THE UNIVERSITY *of* EDINBURGH

Mechanistic Models and Machine Learning for Metabolism

Charlotte Merzbacher

Submitted for the degree of Doctor of Philosophy

The University of Edinburgh

2025

Abstract

Metabolism is the set of all biochemical reactions that sustain life. Recent advances in genetic engineering have enabled the design of metabolic circuits which produce chemicals of interest in microbial hosts, including pharmaceuticals, cosmetics, and biofuels. However, designing new metabolic pathways and understanding complex interactions in metabolism is challenging, especially when considering systems which operate across multiple scales of cellular organization, such as gene regulation, signalling pathways and cellular metabolism. Computational mechanistic models can speed up the development of microbial strains for chemical production and improve our understanding of disease states, and are widely used across the study of metabolism. There are many classes of mechanistic models, including ordinary differential equations and genome-scale linear models, but all can be slow and challenging to construct due to the large amounts of domain-specific knowledge they require. Recent work in machine learning has developed a wide range of tools which can detect patterns in and make highly accurate predictions from large data sets without require many assumptions about biological mechanism. Many of these methods, however, require a large amount of expensive or unavailable data. This PhD thesis presents three approaches which aim to bridge the gap between these two modelling paradigms. Firstly, I present a method for efficient optimization of ordinary differential equation (ODE) models of biological circuits across multiple temporal and spatial scales. The method relies on Bayesian Optimization, a technique commonly used to fine-tune deep neural networks, to learn the shape of a performance landscape and iteratively navigate the design space towards an optimal circuit. This strategy allows the joint optimization of both circuit architecture and parameters, and hence provides a feasible approach to solve a highly non-convex optimization problem in a mixed-integer input space. Second, I integrate ODE models of pathways with genome-scale metabolic models of the production host to create a new joint simulator method, which combines fine-grained concentration trajectory prediction with information about the dynamic global state of native metabolism. I

implement machine learning surrogate models to enable accelerated simulation. Finally, I employ machine learning models to predict gene deletion phenotypes from synthetic data generated from genome-scale mechanistic models. I achieve state-of-the-art accuracy for gene essentiality in various organisms, as well as the first predictive model for small molecule production from deletion screening data. Each of these results demonstrates a way machine learning can be used to improve mechanistic models: to optimize model structure, to replace slow computation, and to improve predictive accuracy.

Lay summary

All living organisms rely on a complex network of chemical reactions to survive and adapt to their environment. Understanding and manipulating these reactions can help us better understand disease and produce pharmaceuticals more sustainably. Computer-based mechanistic models of biological systems are often used to more quickly and cheaply predict how an experimental system will behave using mathematical equations designed based on knowledge of the underlying biological system. There are many different types of mechanistic models, including ordinary differential equations and genome-scale linear models. Mechanistic models are widely used but require a lot of expert knowledge about the reactions under study to construct. In contrast, recent developments in machine learning have led to an explosion of algorithms which can extract patterns from data without assuming much about the underlying biological mechanism. The drawback to these machine learning methods is that they require large amounts of laboratory data, which can often be expensive or impossible to produce. This thesis aims to combine mechanistic models with machine learning to get the best of both approaches. I present three new methods for bridging this gap. First, I use Bayesian Optimization, a machine learning technique, to select mechanistic models of engineered pathways which balance producing lots of a desired product while allowing the bacterial host to grow. Machine learning methods can rapidly navigate the large number of possible design choices and reduce the number of lab experiments needed. Second, I develop a new approach which combines models of engineered pathways with larger models of overall metabolism. I use machine learning in this project to replace the slow and costly genome-scale model simulation. Finally, I use machine learning models to predict how turning off a specific gene will affect a cell, including whether it lives or dies and how much of a small molecule it produces. Machine learning promises to improve mechanistic models substantially by helping engineers select which experiments to do, speeding up difficult computations, or making better predictions about metabolism.

Declaration

I declare that this thesis has been composed solely by myself, and that it contains only my work except where otherwise specified, or where the work is explicitly indicated below to have formed part of a jointly-authored publication. This work has not been submitted for any other degree or professional qualification.

The contents of Chapter 2 are adapted from a paper published in *ACS Synthetic Biology* authored jointly by myself and my advisors, Diego Oyarzún and Oisin Mac Aodha (Merzbacher et al., 2023). The contents of Chapters 3 and 4 are adapted from papers in review authored jointly by the same (Merzbacher et al., 2025). The contents of Chapter 5 contain adapted text from a review in *Biochemical Society Transactions* authored by myself and Diego Oyarzún (Merzbacher and Oyarzún, 2023).

In addition, some of the computational experiments described in Chapter 2 were conducted during the MSc stage of my programme. When applicable, these experiments are noted as their contents were described previously in my MSc thesis (Merzbacher, 2022) and new work conducted in the PhD stage is marked.

Charlotte Merzbacher

June 2025

Acknowledgements

I would like to begin by thanking my advisors, Diego Oyarzún and Oisin Mac Aodha, for their support and guidance throughout the past four years. I could not have begun to accomplish what I did without your knowledge and advice. You have been two of the best advisors a PhD student could hope for, and I will miss working with both of you immensely.

Thanks to UKRI for funding this work and the CDT in Biomedical Artificial Intelligence. Without your funding, I would not have been able to move to the United Kingdom and explore a new field for four years. I only hope that future governments continue to expand support for scientific research and the Home Office keeps the development of future scientific talent at the front of mind in its immigration policies. Basic and applied science research is one of the great inheritances of humanity; we have a responsibility to future generations to continue to steward this collective knowledge with devotion and foresight.

I would like to thank my committee member, Nadanai Laohakunakorn, for his constructive feedback during my yearly reviews. The members of the Biomolecular Control Group provided helpful comments and ideas at our lab meetings throughout the years: special thanks to Evangelos Nikolados, Arin Wongprommoon, Ricardo Albornoz, and Vanessa Smer-Barreto for welcoming me into the lab and providing role models for my development as a scientist. Thanks to Ian Simpson for his three years of CDT leadership and Ekaterina Churkina and Isabelle Hanlon for their administrative support with my many travel bookings. Thanks to my students, Sam Cain, Nicholas Goguen-Compagnoni, and Nicola Hallmann. Supervising you was a pleasure and a privilege.

Completing a PhD follows the common adage about raising children: the days are long, but the years are short. These four years of my PhD have been some of the happiest and most challenging of my life. I cannot detail all the support I received during this degree, for fear the acknowledgements would become as long as the thesis. I will instead list an incomplete subset of names. In Edinburgh: Claire

Gould, Alexandra Lehmann, Zhi Kang Chua, Adelheid Bjornlie, Alys Woodhead, Theo Golden, Tim MacDonald, Hannah Rohde, Ronen Barzel, and Adam Budd. The members of Women Who Write Edinburgh, the Artist's Way Circle, and the Edinburgh Tool Library. Harriet Harris and the University Chaplaincy. The members of the 2021 CDT in Biomedical AI cohort, in particular Barry Ryan, Thibaut Goldsborough, Fiona Smith, and Sebastyen Kamp. Elsewhere: Andy Pelos, Liam Carpenter-Urquhart, Hannah Lam, Cirstyn Michel, Peter Brown, Dominique Martin, and Aviva Davis.

My parents were understandably sceptical when I told them I was moving to Scotland at the tail end of a global pandemic. I am immensely grateful to them for supporting me anyway and finding a second home in Edinburgh. I am excited to see where we explore next. I am proud to join the ranks of the other living Dr. Merzbachers: Matthew and Celia. Thanks to my cousins both in the United States and Europe who have welcomed me into their homes. Special thanks to Alison and Des McKenna for hosting me upon my arrival, Anna Dunthorne for letting me rent her flat, and the Branders for hosting me in Copenhagen, Amsterdam, and Stroanfreggan.

I will end by thanking my ancestors, both those I know and those I did not get the chance to meet. May my efforts make you proud.

Contents

List of Figures

List of Tables

Glossary

ANN Artificial neural network.

CCM Central carbon metabolism.

dFBA Dynamic FBA.

E. coli Escherichia coli.
EFM Elementary flux modes.
EI Expected improvement.
ExP Extreme pathways.

FBA Flux balance analysis.
FFA Free fatty acids.
FPP farnesyl pyrophosphate.
FVA Flux variability analysis.

G6P Glucose-6-phosphate.
GA Glucaric acid.
GGPP geranylgeranyl pyrophosphate.

Ino1 inositol-3-phosphate synthetase.
IPP isopentyl pyrophosphate.

LNML Layered negative metabolic loop.

MCMC Markov chain Monte Carlo.
MI myo-inositol.
MINLP Mixed-integer nonlinear programming.
MIOX myoinositol oxidase.
ML Machine learning.
MOMA Minimization of metabolic adjustment.
MRTF Metabolite-responsive transcription factor.

NGL Negative gene loop.
NML Negative metabolic loop.

ODE Ordinary differential equation.

QSSA Quasi-steady-state assumption.

S. cerevisiae Saccharomyces cerevisiae.

SuhB inositol-1-monophosphatase.

tesA Thioesterase.
TF Transcription factor.
TPE Tree of Parzen estimators.

Udh uronate dehydrogenase.

Chapter 1

Introduction

Inside every cell, a highly coordinated set of biological networks regulates essential cellular functions. Gene regulatory networks determine the temporal and spatial patterns of gene expression, controlling which genes are transcribed under specific conditions. Signalling networks mediate the transmission of intracellular and extracellular cues, allowing cells to respond dynamically to environmental and internal signals. Post-translational modification networks modulate protein activity, stability, and localization, adding an additional regulatory layer beyond gene expression. Protein-protein interaction networks organize proteins into functional complexes, facilitating processes such as signal transduction, enzymatic cascades, and structural assembly. Metabolic networks orchestrate the conversion of substrates to products through enzyme-catalysed reactions, sustaining energy balance and meeting biosynthetic demands. These networks are tightly interconnected, forming a complex systems-level architecture that underlies cellular physiology. Among these, metabolic networks occupy a central position, as they supply the energy and precursors required for gene expression, signalling, and other cellular processes. This thesis uses computational methods to simulate these complex metabolic processes for a variety of applications in biotechnology and biomedicine.

Computational methods can help us understand metabolism better, for example by predicting the effects of environmental or genetic changes. Laboratory experiments are expensive and can take a long time to produce results; simulating

the metabolism accurately can reduce the number of experiments needed to answer many scientific questions. In addition, recent advances in genetic engineering have opened the way to the modification of metabolic pathways, and models of engineered organisms can aid in the design process. As the tools for pathway engineering have improved, the number of options engineers can choose has expanded, and computational models are even more necessary to narrow down this design space.

This thesis uses two families of computational methods to study metabolism: mechanistic modelling and machine learning. Mechanistic models are mathematical descriptions built from expert knowledge of a specific biological mechanism. They can be slow to simulate and improving their predictive accuracy requires manual curation. In contrast, machine learning models find patterns in large amounts of raw data. These models have been applied to a wide range of biological problems in recent years; however, they generally require a large amount of training data which can be unavailable or expensive to generate. Both approaches have their benefits and drawbacks. In this thesis, I present three example problems which are solved best by a combination of these two approaches:

- An algorithm which applies methods from machine learning to optimize mechanistic models of engineered pathways
- A new simulation approach to understanding interactions between a pathway and the rest of metabolism, which uses machine learning to replace slow mechanistic simulations
- A pipeline to predict the effects of genetic changes using machine learning algorithms trained on data generated by mechanistic models.

I begin this thesis introducing the requisite biological context. The final sections of this chapter summarize the research outputs, including papers and presentations, and detail the ethical impacts of this work.

1.1 Biological context

I begin this section by introducing the fundamentals of metabolism and the molecular biological processes underlying it. I introduce enzymes, which speed up metabolic reactions, and discuss the many regulatory mechanisms which keep metabolism in balance. In recent years, bioengineers have introduced methods to modify metabolism by creating new metabolic pathways to produce chemicals of interest. I introduce these genetic engineering methods and then discuss recent advances which have sought to replicate natural metabolic regulatory feedback loops onto engineered metabolic pathways to improve their robustness.

1.1.1 What is metabolism?

Metabolism refers to the complete set of enzyme-catalyzed biochemical reactions within a cell that transform chemical substrates into energy and essential biomolecules. Metabolism enables the conversion of nutrients into usable energy and cellular components, supporting processes such as cell growth, repair, reproduction, and maintenance (Alberts et al., 2022). These reactions are typically divided into two broad categories: catabolism and anabolism (Stephanopoulos et al., 1998). Catabolic reactions break down molecules like glucose or fatty acids into smaller components, releasing energy in the form of adenosine triphosphate (ATP), which then fuels the cell (Voet et al., 2016). Anabolism builds larger and more complex molecules, such as proteins, nucleic acids, and lipid membrane molecules, from simpler precursors through biosynthesis reactions (Palsson, 2011). These macromolecules form the building blocks of every cell, from the outer membrane to the DNA code.

Rather than acting in isolation, various metabolic reactions chain together to form intricate pathways and networks which change dynamically over time. These pathways are highly regulated so they can adapt to the organism's internal state and external environment. The flow of energy and metabolites can shift depending on nutrient availability, cellular stress, or hormonal signals (Berg et al., 2023; Fisher, 2001). Reactions are described in terms of metabolic fluxes, which

are the rate of flow of metabolites through a pathway in units of concentration per time (Matsuda et al., 2017). Systems biology views metabolism as an integrated and dynamic network of genes, proteins, and metabolites (Stephanopoulos et al., 1998). Understanding how these different cellular components interact across multiple scales of time and space can help identify potential drug targets, give insight into disease mechanisms, and aid in strain design for biotechnology applications (McConville, 2014; Spichak et al., 2021).

Enzymes and catalysis

Most metabolic reactions are thermodynamically favorable, meaning they have the potential to occur spontaneously. However, under normal physiological conditions—such as body temperature and pH—these reactions would proceed extremely slowly because they require a significant amount of energy to get started, known as the activation energy barrier (Fisher, 2001). As a result, biomolecules known as enzymes, usually proteins, speed up reactions in a process called catalysis. Enzymes work to lower the activation energy of reactions so they can proceed at rates fast enough to support life. They do this by their unique three-dimensional structure, which brings the reactant (or substrate) molecules together in a small pocket of space called the active site. The active site stabilizes the transition state—the intermediate, high-energy state of the reaction—making it easier for the reaction to proceed (Kraut, 1988). In addition to speeding up reactions, enzymes provide specificity, ensuring that metabolic reactions produce the correct products and minimize unwanted side reactions. Without enzymes, metabolic processes would occur too slowly or non-specifically to sustain life effectively.

Metabolic regulation

While enzymes enable the rapid and specific execution of individual biochemical reactions, they do not act in isolation. In living systems, thousands of enzymatic reactions occur simultaneously and are organized into highly coordinated metabolic pathways. The flux through these pathways must be tightly controlled

to match the cell's changing needs, conserve resources, and prevent harmful imbalances (Palsson, 2011). This coordination is achieved through metabolic regulation—a set of mechanisms that modulate enzyme activity, gene expression, and metabolite availability in response to both internal and external cues. Metabolic regulation is vital as it enables organisms to respond dynamically to internal signals and external environmental factors, such as oxygen and carbon sources in the environment, temperature fluctuations, and signals from surrounding cells.

The metabolic requirements of a specific cell are dependent on its function and environment. Some cells, like the bacterium *Escherichia coli*, grow and divide exponentially when provided with enough sugar. In contrast, human immune cells must remain alive but not proliferating for long periods of time before rapidly multiplying in response to a stimulus (Metallo and Vander Heiden, 2013). Life can also thrive in a wide variety of environments. The parasite *Giardia lamblia* survives in the oxygen-poor environment of the human intestine by relying on specialized organelles which extract energy through anaerobic pathways rather than standard oxidative phosphorylation (Tovar et al., 2003). Thermophilic archaea can grow in temperatures over 70° Celsius and have evolved metabolic pathways that remain stable and efficient in extreme heat, enabling them to survive conditions that would denature most enzymes in other organisms (Stetter, 1999; Straub et al., 2018).

Metabolic needs can fluctuate within a cell across seconds or persist for prolonged periods. As a result, no single biomolecular subsystem can effectively control metabolism alone. Several mechanisms are used to regulate metabolic pathways across multiple time horizons and organizational levels. On short timescales, enzyme activity can be modulated through allosteric interactions, reversible covalent modifications such as phosphorylation, and changes in substrate or cofactor availability (Metallo and Vander Heiden, 2013). For example, phosphorylation of key enzymes in glycolysis and gluconeogenesis rapidly shifts metabolic flux depending on hormonal cues like insulin and glucagon (Fisher, 2001). When slower, more persistent changes are needed, transcriptional regulators and epigenetic modifications can alter the expression of metabolic enzymes in response

to nutrient status, stress, or developmental signals (Nielsen, 2017). Cells may also coordinate metabolism through compartmentalization, where enzymes and pathways are spatially separated into different cellular areas or organelles to allow localized control of flux (Bar-Peled and Kory, 2022; Zecchin et al., 2015). Many regulation mechanisms function as negative feedback loops; that is, the accumulation of a product downstream in a metabolic pathway inhibits the activity of an upstream enzyme, thereby preventing overproduction and maintaining metabolic balance (Ni et al., 2021).

1.1.2 Advances and challenges in metabolic engineering

Metabolic engineering applies DNA recombination techniques to restructure and alter metabolic networks (Bailey, 1991). Metabolic engineers use various techniques, from plasmid transduction to CRISPR gene splicing, to edit a host organism's DNA and insert the genetic code to produce an enzyme (Stephanopoulos et al., 1998). These enzymes are typically encoded by heterologous genes—genes sourced from foreign organisms—which are introduced into the host genome and expressed using the host's native transcriptional and translational machinery. This enables the host cell to produce novel enzymatic functions not present in its natural metabolic repertoire. For example, human insulin was the first licensed drug produced with recombinant technology in 1982 (Baeshen et al., 2014). Scientists insert the genes to produce insulin precursors into the bacterium *Escherichia coli* (*E. coli*) and the budding yeast *Saccharomyces cerevisiae* (*S. cerevisiae*). The antimalarial artemisinin was originally isolated from sweet wormwood, a source which limited its production. Metabolic engineers inserted a three step pathway to create artemisinic acid, the direct precursor, into yeast (Ro et al., 2006). Commercial production of semi-synthetic artemisinin began in 2013 (Paddon et al., 2013; Turconi et al., 2014).

The choice of host organism in metabolic engineering depends on the complexity of the desired biosynthetic pathway, the nature of the product, and the post-translational modifications required for functional activity. Commonly used microbial platforms include *E coli*, favoured for its rapid growth, genetic tractabil-

ity, and well-characterized metabolism. *S. cerevisiae* offers eukaryotic expression capabilities and tolerance to industrial fermentation conditions (Nielsen, 2013). More complex biologics often require mammalian expression systems, such as Chinese hamster ovary (CHO) cells, which are capable of producing glycosylated proteins and other post-translational modifications essential for therapeutic efficacy (Kim et al., 2012).

Examples of products synthesized via metabolic engineering span a broad range of applications. In *E. coli*, researchers have engineered pathways to produce lycopene, a carotenoid with antioxidant properties, and icariside D2, a flavonoid derivative with potential anticancer activity (Liu et al., 2019; Wang et al., 2020b). Recombinant human insulin, a critical therapeutic for diabetes management, is produced in *S. cerevisiae* using optimized expression systems to ensure proper folding and activity (Nielsen, 2013). Monoclonal antibodies used in diagnostics, cancer therapy, and infectious disease treatment are produced in CHO cells due to their ability to perform human-compatible glycosylation and ensure protein functionality (Singh et al., 2018). Additionally, metabolic engineering has enabled the production of chemical precursors that serve as feedstocks for traditional synthetic chemistry, bridging the gap between biology and industrial chemistry (Chartrain et al., 2000).

Despite the success of metabolic engineering, significant challenges to achieving cost-efficient production remain. Increasing product titer (concentration), production rate, and overall yield is challenging because engineered pathways often impose a metabolic burden on the host by competing with native pathways for energy resources, enzyme cofactors, reducing agents, and native metabolite precursors (Keasling, 2010). Production pathways share the host cell machinery with native metabolism, for example ribosomes, which means that to achieve high product titers may require most of the cell's resources to be redirected to produce heterologous enzymes. Furthermore, issues such as pathway bottlenecks, enzyme and genetic instability (Son et al., 2021), and intermediate toxicity (Montaño López et al., 2022) can reduce overall efficiency and necessitate iterative optimization at both the genetic and systems levels (Nielsen, 2017; Van Dien, 2013).

For example, introducing large numbers of heterologous genes into *Pseudomonas putida* caused it to lose the ability to synthesize siderophores, which are necessary for iron metabolism and cell survival (Hong et al., 1991). In another example, attempts to increase the steroid precursor pathway flux led to the accumulation of a toxic intermediate, which stopped cell growth (Kizer et al., 2008). Engineers must consider the costs and benefits of increased metabolic burden when evaluating a particular strategy or pathway design to maximize overall productivity (Wu et al., 2016).

In addition to the practical challenges of implementing engineered systems, identifying promising intervention strategies is itself a formidable theoretical challenge. The discovery of intervention targets typically relies on solving large-scale, nonlinear, and often combinatorial optimization problems over complex metabolic networks (Merzbacher and Oyarzún, 2023). These problems are inherently ill-posed: for instance, many alternative flux configurations can produce equivalent phenotypes, while incomplete kinetic and regulatory data limit predictive power (Hu et al., 2023b; Orth et al., 2010). Moreover, the underlying optimization landscapes are highly nonconvex, making global optima difficult to guarantee. As a result, the theoretical limitations of model inference and optimization strongly constrain what can be achieved experimentally, linking algorithmic tractability to biological feasibility.

1.1.3 Engineering control of metabolism

Traditional metabolic engineering approaches use constitutive promoters that transcribe enzymes at a constant rate. Once the pathway is present in the host cell, engineers use the “push-pull-block" strategy to systematically modify metabolic pathways and improve product yield (Liu et al., 2018). First, they enhance the supply of precursor molecules by increasing the concentration of enzymes that produce them, which “pushes" flux towards the target biosynthetic pathway. The “pull" step introduces or upregulates downstream pathway components to efficiently convert intermediates into the desired product. Finally, engineers inhibit or delete competing pathways that could divert flux from the

target product. However, static enzyme expression is not responsive to changes in the internal or external environment. Additionally, significant experimental fine-tuning is usually required to achieve high product titers. For example, introducing alternative carbon sources in the form of palmitic or oleic acids to the growth medium reduced steroid toxicity in Kizer et al., 2008, but this discovery required multiple costly experimental iterations.

Additionally, scaling up from controlled laboratory conditions like small-scale flasks or bioreactors to industrial-sized facilities is challenging because larger reactors have more complex and spatially heterogeneous environments (Sun and Alper, 2015). For example, oxygen, pH, and nutrient levels can all vary across different zones of the reactor due to gradients in mixing, which can impact microbial metabolism (Heins and Weuster-Botz, 2018). These differences in cellular micro-environment can lead to unpredictable variations in productivity, yield, and overall bioprocess performance (Pigou and Morchain, 2015; Wehrs et al., 2019). Static control systems cannot respond to these changes and are thus often not very robust to scale-up.

Dynamic metabolic engineering (see Figure 1.1) proposes to address the limitations of static control using techniques drawn from control theory (Ni et al., 2021). In natural systems, cells regulate metabolism through many different negative feedback mechanisms which operate at the molecular, genetic, and whole-cell scale. If engineers could introduce feedback loops like these to heterologous pathways, cells would be able to respond autonomously to changes in bioreactor conditions (Hartline et al., 2021).

Previous experimental work has implemented dynamic control feedback loops in multiple systems. Many circuits rely on inducers such as IPTG, which not only add significant cost to the medium (Cardoso et al., 2020) but also limit control to the batch level. Instead, closed-loop control mechanisms automatically maintain enzyme concentrations at a desired set point. These mechanisms use biosensors which respond to a given stimulus, including nutrient levels, surrounding populations (quorum sensing), temperature, light, or stress. Control mechanisms can act at the genetic, transcriptional, translational, or post-translational stages. DNA

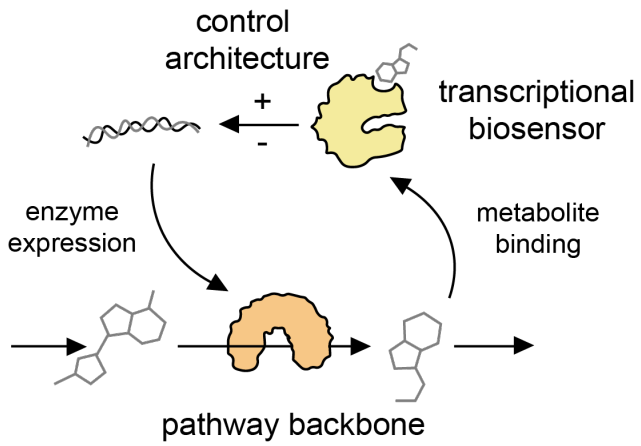


Figure 1.1: Schematic of engineered pathway under transcriptional feedback control. Adapted from Merzbacher and Oyarzún, 2023.

recombination for genetic-level control can take several generations and therefore multiple hours to take effect; in contrast, accelerating posttranslational or translational degradation has a faster response time but can lead to high molecular turnover which impacts cell growth. In this thesis, I model pathways with metabolite-responsive transcription factors (MRTFs), which use metabolites already produced in the pathway to modify enzyme expression (Oyarzún and Stan, 2013). These systems regulate enzyme expression at the transcriptional level, which is the most well-studied and commonly used control point. Dynamic control promises to improve robustness by allowing engineered cells to automatically regulate their new pathways in response to the environment, but these systems can be even more complicated to engineer and have many components. As a result, computational models are even more useful to understand how pathway changes interact with metabolism.

1.2 Research outputs and engagement

This section summarizes the key research outputs of this thesis and give an overview of the ways I have engaged with the scientific community in the past three years. During my PhD, I published several peer-reviewed papers and attended multiple conferences, workshops, and working groups.

Publications

The work presented in this thesis comprises three major research outputs. The contributions of each paper are detailed further in Section 1.4. I also published a literature review in *Biochemical Society Transactions* which is adapted in part into Chapter 6. In addition, I published one of my rotation projects from the first year of the PhD and a paper based on one of my supervised students' projects. Below I list all publications completed during this thesis. I also note the contributions I made to each work.

- Merzbacher, C., Mac Aodha, O. M., and D. A. Oyarzún. "Bayesian optimization for design of multiscale biological circuits". In: *ACS Synthetic Biology*. September 2023. I conceived of the optimization pipeline as part of the MSc phase, performed all simulations myself, created all figures, and wrote the draft of the paper. OMA and DAO advised the project and edited the paper text.
- Merzbacher, C. and D. A. Oyarzún. "Applications of artificial intelligence and machine learning in dynamic pathway engineering". In: *Biochemical Society Transactions*. October 2023. I performed the systematic literature review and wrote the draft of the paper. DAO edited the text.
- Merzbacher, C., Ryan, B., Goldsborough, T. et al. "Integration of datasets for individual prediction of DNA methylation-based biomarkers". In: *Genome Biology*. December 2023. I ran the methylation pipeline modelling experiments in collaboration with BR and TG and produced a report that was adapted into the paper.
- Cain, S., Merzbacher, C., and D. A. Oyarzún. "Low-dimensional representations of genome-scale metabolism". In: *Proceedings of Foundations of Systems Biology in Engineering*. September 2024. I proposed the project, supervised SC, and edited the paper draft.
- Merzbacher, C., Mac Aodha, O. M., and D. A. Oyarzún. "Modelling dynamic host-pathway interactions at the genome scale". In: *Metabolic Engineering*. June 2025. I conceived of the simulator design, ran all experiments, created all figures and wrote the paper. DAO and OMA edited the paper

draft.

- Merzbacher, C., Mac Aodha, O. M., and D. A. Oyarzún. "Accurate prediction of gene deletion phenotypes with Flux Cone Learning". In: *Nature Communications* September 2025. I conceived of the Flux Cone Learning pipeline, ran all experiments, created all figures and wrote the paper draft. DAO and OMA edited the paper draft.

Presentations and Posters

I presented my work at a variety of conferences, both as scientific posters and as talks. I detail these conferences in Table 1.1.

Conference and Workshop Planning

In addition to attending conferences, I organized and delivered two workshops during my PhD: a one-day satellite workshop for the International Conference in Systems Biology in Berlin in October 2022 and a two-day working group at the Santa Fe Institute (SFI) in New Mexico in September 2024. The ICSB workshop was titled "Synthetic Biology in the Age of Machine Learning" and had 40-50 attendees and 10 speakers. The SFI workshop was planned in collaboration with 8 other PhD students over the course of a year and consisted of 2 days of talks and discussion groups on "Assessing Representation in Minds and Artificial Systems".

Student supervision

I proposed three projects to be undertaken by Bachelors' and Masters' students. I supervised two University of Edinburgh BSc students in the 2023-2024 school year, Samuel Cain and Nicholas Goguen-Compagnoni. Samuel's work was written up into a short paper for submission to the Foundations of Systems Biology in Engineering conference in September 2024. I presented this work at the conference and received the Best Talk prize for the work. In Fall 2024, I supervised Nicola Hallmann, a visiting MSc student from ETH Zürich, on a project titled "Optimal media composition using Bayesian Optimization". All three students passed their degrees.

Conference Name	**Dates**	**Location**	**Type**
International Conference in Systems Biology	October 6-10, 2022	Berlin, Germany	Flash Talk
ICSB Satellite Workshop: Synthetic Biology in the Age of Machine Learning	October 5, 2022	Berlin, Germany	Workshop organizer and 30-minute talk
SynBioUK	November 3-5, 2022	Newcastle, UK	Poster
Turing Workshop on AI and Engineering Biology	March 13-14, 2023	Edinburgh, UK	10-minute talk
AI4bio Spring Symposium	April 18-19, 2023	Delft, Netherlands	Conference attendee
AI for Healthcare CDT Symposium	May 3-5, 2023	York, UK	Poster and 15-minute talk
Complexity-GAINS Summer School	August 13-23	Cambridge, UK	Summer school and 10-minute talk
SynBioUK	November 4-8, 2023	Bristol, UK	15-minute talk and poster
Springer Nature Co-Creation Workshop	December 1-3, 2023	Berlin, Germany	Workshop participant
CDT Industry Day	April 24, 2024	Edinburgh, UK	15-minute talk
AI for Healthcare CDT Symposium	May 20-21, 2024	Edinburgh, UK	Poster
Cambridge ELLIS Machine Learning Summer School	July 2024	Cambridge, UK	Summer school and poster
FOSBE Conference	September 2024	Corfu, Greece	20-minute talk

Continued on next page

Conference Name	Dates	Location	Type
Santa Fe Institute Working Group on Representation	September 2024	Santa Fe, USA	Workshop organizer and 2 30-minute talks
SynBioBeta	May 2025	San Jose, USA	Conference attendee
Metabolic Network Analysis	July 2025	Vienna, Austria	Talk and poster

Table 1.1: List of conferences and workshops attended during PhD.

1.3 Responsible research and innovation

My research highlights several ethical considerations associated with the application of machine learning methods to computational models of metabolism. Throughout the PhD, I aligned my work with the principles of Responsible Research and Innovation (RRI) (Owen et al., 2020). While the exact definition of RRI is still debated, it emphasizes the anticipation of potential consequences, reflexivity on purposes and motivations, inclusion of a wide range of stakeholders, and responsiveness to public concerns throughout the research process (Burget et al., 2017; Owen et al., 2013).

A primary ethical concern raised by this thesis is the dual-use potential of the computational methodologies developed. While these methods can accelerate the beneficial production of pharmaceuticals, biofuels, and other valuable chemicals through engineered microbial strains, they may also be exploited for harmful purposes, such as creating pathogenic organisms, synthesizing toxins, or developing biological weapons (Cirigliano et al., 2017). Miller and Selgelid, 2007 proposed a framework for “experiments of concern" to ethically evaluate biological research based on its potential for misuse. However, their framework does not consider recent computational advances. A recent report by the National Academy of Sciences examined how to mitigate the risks of synthetic biology being used for

bioweapons (Sciences et al., 2018). They suggest that the government should maintain registries of known biological threats and use machine learning models to scan ordered DNA sequences, for example, to detect possible bioweapons. In my work, I maintained a private code repository, accessible only to project collaborators, until the work was published in a peer-reviewed journal. If any ethical concerns with the release of the data were raised by reviewers, I proposed the use of a permanently private code repository accessible only to verified and reputable researchers upon request. While this approach has some trade-offs in open research and intellectual freedom, it significantly reduces the potential misuse of computational tools (Bhowmik, 2017). Other guardrails already exist on the procurement of pathogenic DNA sequences and toxic precursor compounds. It is also acknowledged that the high level of domain-specific expertise required to operationalize these technologies serves as a practical barrier against widespread misuse (Council et al., 2006).

Another ethical consideration concerns the accuracy and reliability of the computational models used in this research. Machine learning models can be biased based on the distribution of data included in the training set; these biases can be difficult to detect especially in large biological datasets where the ground truth distribution is not well-characterized. In addition, mechanistic approaches like ordinary differential equations can have inaccuracies in their formulation which can lead to predictions inconsistent with experimental observations. This discrepancy underscores the importance of rigorous validation procedures and responsible application of these predictive models. In some biological tasks such as protein structure prediction, advances in model performance have been assessed using consistent benchmarks (Kryshtafovych et al., 2021; Villaverde et al., 2015); however, such benchmarks remain to be developed for the vast majority of biological tasks.

A final consideration of this thesis is the trade-offs between accuracy and interpretability in machine learning models. Deep learning has many advantages such as being able to flexibly fit large, high-dimensional training sets; however, these models are often considered “black boxes" due to their complex and opaque

internal structures, making it challenging to understand how specific inputs are transformed into outputs. This lack of transparency can hinder scientific innovation because biases in the model can be difficult to detect and correct. Ideally, a model should maximize both accuracy and be fully interpretable; in practice, there are often tradeoffs between flexible deep learning models and their interpretability. In this thesis, I preferentially use interpretable models such as random forest, linear regression, and logistic regression if they achieve similar accuracies to deep learning models. In addition, in some imbalanced classification problems where the majority of the training set comes from one class, I prioritize balancing the class accuracies to be more even over the overall model performance (see Section 5.5). In my applications, I was able to primarily use shallow learning methods; however, in the case that a black-box method achieves substantially better performance and is selected over more interpretable alternatives, there are some techniques which can provide insights into model predictions. These methods work post-hoc on trained models by assigning each feature an importance score (Chen et al., 2024). For instance, Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) both evaluate the importance of input features to final predictions (Xu et al., 2019). The ideal tradeoff between explainability and accuracy depends on the application. For decision-making in experimental design, partial interpretability with quantifiable uncertainty may be preferable to higher black-box accuracy (Sidak et al., 2022.)

To conclude, this thesis advocates the responsible and transparent use of machine learning for biomedical research. I recommend the active engagement of stakeholders, particularly experimental biologists, throughout the research process. Clear and accessible communication regarding the limitations, uncertainties, and assumptions inherent in these computational methods is also crucial to facilitate informed interpretation and utilization by the broader scientific community.

1.4 Contributions of this thesis

This thesis presents three different paradigms for the integration of machine learning and mechanistic models for applications in metabolism and metabolic engineering. I begin with a chapter detailing the computational methods used in the thesis and move on to discuss each approach in three results chapters. Each chapter corresponds to a major research output in the form of a first author primary research paper.

- First, machine learning toolkits can be used to rapidly search high-dimensional spaces and optimize biological problems described by existing mechanistic models. Selecting both genetic control loop architectures and biosensor dose-response parameters is challenging, particularly when taking into account the multiple timescales present in heterologous pathways. In Chapter 3, I introduce BayesOpt, a novel method which applies a toolkit from neural network hyperparameter tuning to the design of multiscale biological circuits.
- Second, machine learning can act as a surrogate, replacing costly-to-simulate mechanistic models based on training on data generated on-demand from the mechanistic model. In Chapter 4, I address some of the limitations of ordinary differential equation models, such as their assumed constant growth rate, by creating a novel simulator which incorporates a genome-scale model (GEM) of native metabolism with an ODE model of an engineered pathway. This integration requires the training of a surrogate ML model to replace the time-consuming FBA optimization step.
- Finally, machine learning can improve the predictive accuracy of mechanistic models for various second-order tasks. In Chapter 5, I introduce Flux Cone Learning, a pipeline which trains a machine learning model on large-scale flux sampling data generated from genome-scale models to predict gene deletion fitness, either for essentiality prediction or product production prediction. Flux Cone Learning beats the current gold standard in metabolic gene essentiality prediction.

In the final chapter, I discuss these three use cases and review the related literature on recent developments in hybrid modelling for problems not covered in the thesis. I present perspectives on the state of the field and where I believe it should go in the near future.

Chapter 2

Computational modelling of metabolism

The complexity of cellular metabolism—spanning multiple biological scales, thousands of interacting components, and nonlinear feedback mechanisms—poses significant challenges for understanding and engineering solely through experimental approaches. To address this, computational models have become central to metabolic research, providing a structured framework to analyse, predict, and manipulate biochemical networks systematically (Orth et al., 2010; Palsson, 2015). A wide variety of computational methods have been developed to facilitate the rational design of engineered cells or to explore the impacts of various changes to metabolism (Gombert and Nielsen, 2000). Before committing to costly and time-consuming experiments, researchers can use these tools to simulate metabolic behaviour *in silico* and predict how genetic or environmental changes will affect the system under study. For example models have been used to identify metabolic bottlenecks, simulate gene knockouts, optimize flux distributions, and couple product formation with cellular growth (Burgard et al., 2003; Patil et al., 2005). Model predictions can help prioritize experiments by selecting a subset of strain variants or culture conditions to test (Liao et al., 2022). Models can incorporate data from many different sources, from genomics data to enzyme activity assays, and express this data as hypotheses about how metabolic variables are expected to change over time.

Until recently, most computational models used in metabolic engineering were mechanistic models, which use established physical and biochemical principles to simulate system behaviour. Mechanistic models are termed as such because they explicitly describe the biological mechanism under study using mathematical equations. Two widely used classes are ordinary differential equation (ODE) models, which describe dynamic changes in metabolite concentrations over time, and genome-scale metabolic models (GEMs), which represent the complete metabolic capabilities of an organism based on its annotated genome. In this section, I will introduce both modelling frameworks, exploring their advantages and limitations in the context of metabolic engineering.

2.1 Mechanistic models of metabolism

Mechanistic models describe biological systems based on the known physical, chemical, and biological principles that govern their behaviour. These models aim to formulate the underlying relationships between components such as molecules or genes into a set of mathematical equations which describe a higher-level cellular process (Curtis, 1991). Mechanistic models have been widely used for the study of metabolism for several reasons. First, because they are based on well-established biological mechanisms (e.g. enzyme kinetics, gene regulatory networks), they provide insights into why and how a system behaves a certain way. Furthermore, hypotheses about a particular biological mechanism or experimental condition can be tested *in silico* to see if they match the results of experiments. Some of the first mechanistic models were developed by Hodgkin and Huxley in 1952 to describe how neurons transmit electrical signals (Hodgkin and Huxley, 1952); since then many thousands of models for various biological phenomena have been built (Malik-Sheriff et al., 2020). The core benefit of mechanistic models is that, because they are constructed from first principles, they can extrapolate to predictions about behaviours not present in the original data (Baker et al., 2018). Two modelling paradigms which are commonly used to model metabolism are nonlinear ordinary differential equations (ODEs) and genome-scale metabolic models. I

introduce each method in Table 2.1; while they stem from the same initial mass-balance and stoichiometric principles, they diverge in their assumptions. I then introduce each method in more detail.

Aspect	**Kinetic ODE models**	**Genome-scale metabolic models (GEMs)**
Common foundation	Both stem from mass balance and stoichiometry.	Both stem from mass balance and stoichiometry.
Governing principle	Dynamic mass balance on metabolite/enzyme pools: $\frac{dx}{dt} = f(x, e, u; \theta)$ $\frac{de}{dt} = g(x, e, u; \theta)$ with kinetic laws (e.g., Michaelis–Menten).	Steady-state mass balance on fluxes: $S\,v = 0, \quad v^{\mathrm{lb}} \le v \le v^{\mathrm{ub}}$ often with an optimality assumption (e.g., maximize biomass).
Key assumption	Nonlinear, non–steady state dynamics	Steady state for intracellular metabolites (quasi-steady-state)
Primary variables	Concentrations $x(t)$, enzyme levels $e(t)$.	Reaction fluxes v
Typical inputs	Kinetic parameters θ, initial conditions, kinetic forms; perturbations $u(t)$.	Stoichiometric matrix S, bounds $v^{\mathrm{lb}}, v^{\mathrm{ub}}$; objective; media/exchange constraints.
Granularity / scale	Pathway scale; selected reactions with detailed kinetics.	Genome scale (thousands of reactions); no explicit kinetics.

Table 2.1: ODE vs GEMs: shared mass-balance foundations, diverging assumptions (dynamic vs steady state), and complementary strengths.

2.1.1 Ordinary differential equations

Ordinary differential equation (ODE) models are widely used to describe the dynamic behaviour of metabolic pathways. These models express how key variables, such as metabolite concentrations or enzyme activities, change over time based on

the underlying biochemical reactions. In many cases, the governing processes and reaction mechanisms are better understood than the exact instantaneous state of the system, making ODEs a natural mathematical framework for modelling (Braun and Golubitsky, 1983). An ODE consists of a function and its derivative with respect to a single independent variable—typically time—capturing the rate of change of system components. ODE models are particularly well suited for representing the nonlinear, time-dependent interactions characteristic of metabolic flux regulation and enzyme-mediated reactions (see Figure 2.1), allowing researchers to explore system dynamics, steady states, and responses to perturbations (Alon, 2019).

Most biological systems do not follow simple linear relationships. As a result, accurate models of these systems are often highly nonlinear and involve multiple interacting components. These dynamics present a computational challenge to integrating these equations. Few differential equations used in biology have closed-form analytical solutions that can be expressed explicitly in terms of elementary functions. Instead, researchers must use numerical methods and often costly computational simulation to simulate these systems and explore their behaviour under different initial conditions and parameter values (Darvishi et al., 2007; Érdi and Tóth, 1989). Many toolkits have been developed to efficiently solve differential equations (Gardner et al., 2022). I will begin by introducing a simple model of an enzyme-mediated chemical reaction which, given simplifying assumptions, can be reduced to the widely used Michaelis-Menten rate law.

2.1.2 Michaelis-Menten kinetics

In 1913, biochemists Leonor Michaelis and Maud Menten developed a model of enzyme catalysis which to this day forms the basis of many differential equation models (Johnson and Goody, 2011; Srinivasan, 2022). They began with a reaction scheme which assumes that enzymatic reactions proceed through the reversible formation of an enzyme-substrate complex ES before conversion of the substrate S to a product P:

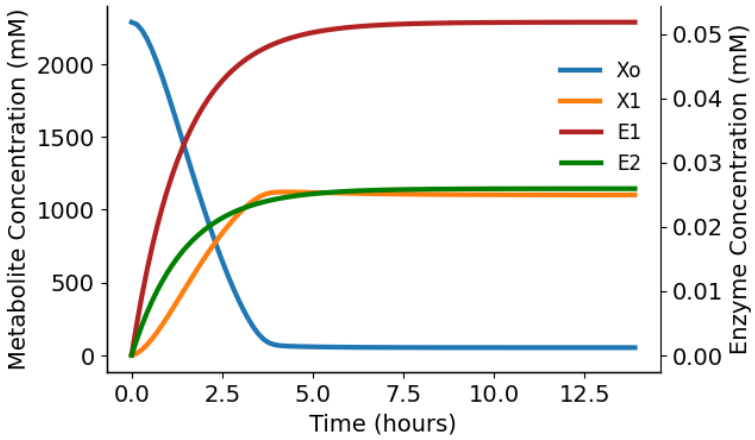


Figure 2.1: Sample ODE trajectories for toy model of metabolic pathway. Generated using toy model from Chapter 3.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P, \tag{2.1}$$

where E is the free enzyme and k_1, k_2, and k_{-1} are the kinetic rate constants which describe the speed of each step. To describe the dynamics of the system, Michaelis and Menten constructed differential equations for each concentration by using the law of mass action, which states that the rate of a reaction is proportional to the product of the concentrations of the reactants. The system of equations is as follows:

$$\frac{d[E]}{dt} = -k_1[E][S] + k_{-1}[ES] + k_2[ES], \tag{2.2}$$
$$\frac{d[ES]}{dt} = k_1[E][S] - k_{-1}[ES] - k_2[ES], \tag{2.3}$$
$$\frac{d[S]}{dt} = -k_1[E][S] + k_{-1}[ES], \tag{2.4}$$
$$\frac{d[P]}{dt} = k_2[ES]. \tag{2.5}$$

Each equation describes how a species (the substrate, product, or substrate-enzyme complex) changes over time as a result of the two different reactions. The key simplification Michaelis and Menten made was the quasi-steady-state assump-

tion (QSSA), which assumes that the binding dynamics of the substrate-enzyme interaction are much faster than the depletion of the substrate or accumulation of the product. As a result, the concentration of the enzyme-substrate complex, $[ES]$, rapidly reaches a steady state. This implies that its rate of change can be approximated as zero:

$$\frac{d[ES]}{dt} \approx 0 \tag{2.6}$$

The substrate-enzyme complex thus reaches a steady-state concentration where no net change is occurring while the rest of the reaction is still in progress and its rate of change can be assumed to be zero.

Additionally, the enzyme is either in its free form $[E]$ or bound in the complex $[ES]$, so given the total enzyme concentration we can find an equation for the free enzyme $[E]$:

$$[E] = [E]_0 - [ES], \tag{2.7}$$

where $[E]_0$ is the total enzyme concentration, or the sum of free enzyme and enzyme bound in the enzyme–substrate complex. Under the QSSA and enzyme conservation, we can solve algebraically for $[ES]$. Substituting (2.7) gives

$$\begin{aligned} 0 &= k_1[E][S] - (k_{-1} + k_2)[ES] \\ &= k_1\big([E]_0 - [ES]\big)[S] - (k_{-1} + k_2)[ES] \\ &= k_1[E]_0[S] - \big(k_1[S] + k_{-1} + k_2\big)[ES]. \end{aligned} \tag{2.8}$$

Rearranging,

$$[ES] = \frac{k_1[E]_0[S]}{k_1[S] + (k_{-1} + k_2)} = \frac{[E]_0[S]}{\dfrac{k_{-1} + k_2}{k_1} + [S]} = \frac{[E]_0[S]}{K_M + [S]}, \tag{2.9}$$

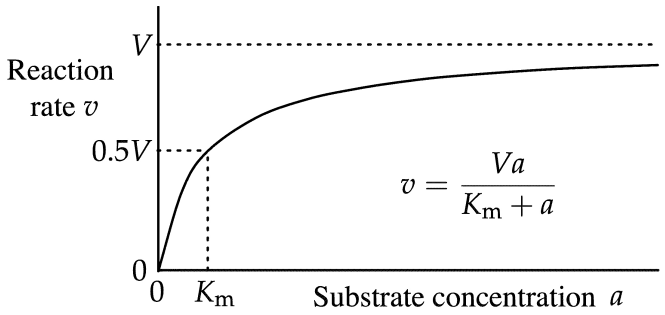


Figure 2.2: Curve of the Michaelis-Menten equation labelled with equation components. The substrate concentration [S] is labelled here as a and $V_{\max}$ is V_a. Sourced from Wikipedia.

where we have defined the Michaelis constant

$$K_M = \frac{k_{-1} + k_2}{k_1}. \tag{2.10}$$

We can then express the product formation rate $\frac{d[P]}{dt}$ in a more concise form, known as the Michaelis-Menten equation:

$$v = \frac{d[P]}{dt} = k_2[ES] = \frac{k_2[E]_0[S]}{K_M + [S]} = \frac{V_{\max}[S]}{K_M + [S]}, \tag{2.11}$$

where v is the initial reaction rate of the product $[P]$, $V_{\max}$ is the maximum rate of the reaction when the enzyme is saturated with substrate, and K_m is the Michaelis constant, representing the substrate concentration at which the reaction rate is half of $V_{\max}$. Figure 2.2 shows the reaction rate v is plotted as a function of the substrate concentration $[S]$. As the substrate concentration increases, the reaction rate initially does too before saturating at $V_{\max}$ when all the enzymes are bound to substrate and the reaction is proceeding at maximum speed. There are many adaptations of the Michaelis-Menten rate law which describe reversible reactions, competitive substrate binding, and allosteric inhibition.

2.1.3 Multiscale models of metabolic pathways

In a network of multiple reactions, the rate of change of a particular metabolite x can be described as a sum of the reaction fluxes producing and consuming it:

$$\frac{dx}{dt} = v_{in,1} - v_{out,1} + v_{in,2}, \tag{2.12}$$

where in this toy example two reactions produce x and one consumes it. Equations for many reactions forming an interlocking network can be written, but if these reactions follow Michaelis-Menten kinetics their rates are dependent on enzyme concentrations. Enzyme concentrations change over time via transcriptional feedback regulation, albeit much more slowly than metabolite concentrations. Unlike in the Michaelis-Menten case, these differences in timescale cannot be ignored. I focus this section on metabolic pathways under feedback control that synthesize high-value products (Verma et al., 2021). These systems have two important timescales that an accurate ODE must capture: the relatively rapid conversions of substrate to product and the slower timescale of enzyme transcription and translation, and thus genetic feedback control. In the general case, a model includes a metabolic branch point through a heterologous pathway with enzymatic steps and contains two sets of equations, one to cover the substrate dynamics and one to cover the enzyme dynamics:

$$\begin{aligned} \frac{ds}{dt} &= f(s,e) - \lambda s, \\ \frac{de}{dt} &= u(s,\gamma) - \lambda e, \end{aligned} \tag{2.13}$$

where s and e are vectors of metabolite and enzyme concentrations, respectively. The term $f(s,e)$ describes the biochemical reactions between pathway intermediates, while the parameter λ models the dilution effect by cell growth. Some entries in $f(s,e)$ will be Michaelis-Menten rate law terms (see Equation 2.11). The vector $u(s,\gamma)$ describes the enzyme expression rates controlled by some pathway intermediates, and typically take the form of sigmoidal dose-response

curves that lump together processes such as metabolite-TF or metabolite-riboregulator interactions (Mannan et al., 2017).The parameters γ are tunable to control the shape of the dose-response curve.

This system of equations represents an exemplar pathway with both a fast, kinetic timescale and slow, genetic timescale. Metabolic reactions operate in the millisecond range or faster (Bar-Even et al., 2011), whilst enzyme expression changes in the scale of minutes or longer. Moreover, metabolites and enzymes are also present in different ranges of concentrations, from nM for enzymes to mM and higher for metabolites (Tonn et al., 2019). Particularly in pathways under dynamic feedback control (see Section 1.1.3), both sets of species change at vastly different rates, which result in stiff systems which rapidly become too computationally intensive to integrate as the number of chemical species increases. Furthermore, the construction of these models relies on detailed mechanistic knowledge and includes many kinetic and regulatory parameters. These parameters can be fit from data or drawn from existing knowledge databases such as BRENDA (Schomburg et al., 2004; Villaverde et al., 2019). The largest existing ODE models of yeast glycolysis can cover large sections of metabolism; however, these models must be parametrized by strain-specific data which is expensive and difficult to obtain (Hu et al., 2023a). Additionally, this model makes several simplifying assumptions. In reality, the cellular growth rate is not constant and changes based on metabolic conditions. Limited parameter availability, **large variation in parameter values across different environmental or cellular cond** and changing external conditions which affect parameter values and uncertainty are all significant challenges to the construction and accuracy of ODE models.

2.2 Genome-scale metabolic models

An alternative, widely used modelling method to ODEs are genome-scale metabolic models (GEMs). Both approaches originate from the same underlying principle

of mass balance, which can be expressed generally as:

$$\mathbf{S}\mathbf{v} = \frac{d\mathbf{x}}{dt}, \tag{2.14}$$

where $\mathbf{S}$ is the stoichiometric matrix, $\mathbf{v}$ is the vector of reaction fluxes, and $\mathbf{x}$ is the vector of metabolite concentrations. In an ordinary differential equation (ODE) model, the right-hand side $\frac{d\mathbf{x}}{dt}$ is explicitly integrated over time, allowing metabolite concentrations and fluxes to vary dynamically.

In contrast, GEMs make the steady-state assumption that intracellular metabolite concentrations remain constant over the timescale of interest, such that

$$\mathbf{S}\mathbf{v} = 0. \tag{2.15}$$

GEMs therefore describe metabolism as a high-dimensional space of feasible steady-state flux distributions, often referred to as the flux cone. Vectors of metabolic fluxes from this flux cone can be generated in various ways, including flux balance analysis and flux sampling.

GEMs are large computational models that describe the metabolic network of an organism as a set of reactions between metabolites defined by gene–protein–reaction associations (Borodina and Nielsen, 2005). The stoichiometric coefficients of every reaction in the GEM form an integer matrix $\mathbf{S}$ which imposes constraints on the flow of metabolites through the network (Figure 2.3). This stoichiometric matrix has dimension $m \times n$, where m is the number of metabolites and n is the number of reactions. An n-dimensional vector of fluxes $\mathbf{v}$ represents the rate of each reaction, and the steady-state assumption (2.15) ensures that all internal metabolite balances are satisfied. The feasible fluxes are further constrained by physiochemical bounds:

$$v_i^{\min} \leq v_i \leq v_i^{\max}, \tag{2.16}$$

where $(v_i^{\min}, v_i^{\max})$ are lower and upper bounds on each reaction flux (Kauffman et al., 2003). Each reaction flux v_i is analogous to the reaction rate in ODE

models (Equation 2.1.3), but whereas ODEs capture transient, time-dependent behaviour, GEMs represent the set of all flux distributions consistent with steady-state mass balance. This conceptual shift from integrating $\frac{d\mathbf{x}}{dt}$ to enforcing $\mathbf{Sv} = 0$ distinguishes dynamic kinetic models from constraint-based steady-state models.

In genome-scale metabolic models, the gene–protein–reaction (GPR) mappings describe how genes encode enzymes that catalyse specific metabolic reactions. These mappings are derived from genome annotation and biochemical databases (Devoid et al., 2013), linking genes to their corresponding enzyme subunits and isoenzymes using logical rules (e.g. AND for protein complexes, OR for isozymes). Through these GPR associations, the annotated genome is systematically converted into a reaction network that captures the enzyme composition and genetic basis of each metabolic function.

Since the 1990s GEMs have been reconstructed for over 6000 organisms, including over 5000 bacteria (Edwards and Palsson, 1999; Gu et al., 2019). Complete GEM models exist for several common host bacteria, including *Escherichia coli*, *Bacillus subtilis*, and the pathogens *Mycobacterium tuberculosis* and *Yersinia pestis*. In particular, the iML1515 model of *E. coli* contains information on 1,515 open reading frames, 2,719 metabolic reactions, and 1,182 unique metabolites and is considered the best-curated and most complete GEM (Monk et al., 2017). In eukarya, the brewer's yeast *Saccharomyces cerevisiae* has two iteratively updated consensus GEM models, Yeast 9 (Zhang et al., 2024) and iMM904 (Herrgård et al., 2008). Several repositories for models, including BiGG models (King et al., 2016) standardize model formats for widespread use by experimental practitioners.

2.2.1 The flux cone

The set of all reaction fluxes in genome-scale models describe a high-dimensional space where each dimension corresponds to the flux of a specific metabolic reaction (Klapper et al., 2021). When the linear constraints are applied to the reaction fluxes (see Equation 2.16), they delineate a subset of this high-dimensional space called the flux cone. This convex polyhedral cone represents the set of all feasible

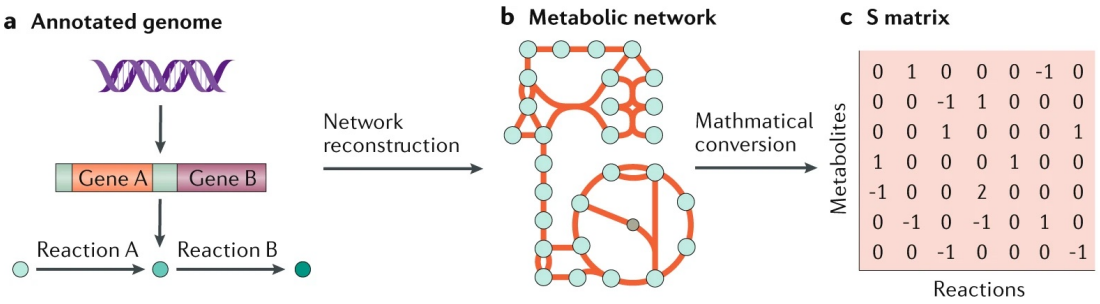


Figure 2.3: Schematic of genome-scale metabolic model construction. Adapted from Fang et al., 2020. This figure is a simplified representation of GEM construction and does not depict isozyme or protein complex mapping that is not 1:1 from gene to reaction.

steady-state flux distributions that satisfy the stoichiometric and thermodynamic constraints of the system. The flux cone encompasses all possible combinations of reaction fluxes that maintain a steady state, ensuring that the production and consumption rates of metabolites are balance (Wagner and Urbanczik, 2005). The structure of the flux cone is influenced by the network stoichiometry and the reversibility of its reactions (Palsson, 2011). It can be characterized by a set of vectors, such as elementary flux modes or extreme pathways, which represent minimal sets of reactions capable of operating at steady state independently. Elementary flux modes (EFMs) are the simplest, non-divisible sets of reactions in a metabolic network that can operate independently (Schuster et al., 1999). Each EFM describes a minimal metabolic route through the network in the sense that you cannot remove any reaction from the set without losing its ability to function at steady state (Zanghellini et al., 2013). Similar to EFMs, extreme pathways (ExP) are another type of minimal set of reactions that define the edges or extreme points of the flux cone. Extreme pathways are unique and non-redundant pathways that capture the boundaries of the metabolic capabilities of the network (Price et al., 2002). Any feasible flux distribution inside the flux cone can be represented as a linear combination of these minimal generating vectors (EFMs and ExPs). Understanding the geometry of the flux cone is crucial for metabolic engineering and systems biology, as it provides insights into the metabolic capabilities, flexibility, and robustness of an organism.

2.2.2 Flux balance analysis

Flux balance analysis (FBA) is a constraint-based linear modelling method used to analyse the flow of metabolites through a GEM metabolic network (Orth et al., 2010). FBA assumes all reactions in the network are at steady state, which means the concentration of all metabolites in the cell remain constant (Kauffman et al., 2003). FBA also maximizes an objective, usually growth rate. Given the constraints imposed by metabolic stoichiometry, FBA solves a linear optimization problem to predict the metabolic fluxes of the cell provided the objective is optimal. All metabolic fluxes are assumed to be constant, which reduces the model to a linear optimization problem of the form:

$$\max_{v}(c^T v), \tag{2.17}$$

$$\text{s.t.} \quad \mathbf{S}v = 0, \tag{2.18}$$

$$v_{\min} < v_i < v_{\max}, \tag{2.19}$$

where $c^T v$ forms an objective, usually representative of the growth rate through a manually constructed biomass reaction. There are infinite possible optimal flux vectors because the problem is underdetermined; that is, the number of of reactions exceeds the number of metabolites, leading to a solution space that contains many different flux distributions which all satisfy the same steady-state and optimality conditions. FBA identifies a single point within the flux cone, often the one that happens to be found by the solver, but not necessarily the biologically correct or most representative one (Mahadevan and Schilling, 2003). Alternate techniques such as flux variability analysis (FVA) and Monte Carlo flux sampling are used to assess the range of possible flux values compatible with optimal growth (Gudmundsson and Thiele, 2010).

Flux balance analysis is widely used by experimentalists to quantitatively predict the effects of adding engineered pathways to a cell, knocking out various genes, or changing medium conditions. Many software toolkits, including RAVEN and COBRA, exist for FBA simulation in various programming languages (Ebrahim et al., 2013; Wang et al., 2018).

2.2.3 Flux sampling

Flux sampling is a collection of methods for randomly generating flux distributions from the solution space of a genome-scale metabolic model. Flux sampling algorithms are high-dimensional samplers which randomly select a point from the flux cone and check its feasibility. Due to the high dimensionality and complexity of the flux cone, researchers often utilize random walk strategies, such as Markov Chain Monte Carlo (MCMC). These strategies iteratively select a point within the flux cone and then move in a random direction to generate a new sample, ensuring that each sampled point adheres to the system's steady-state (see Equation 2.15) and thermodynamic constraints (see Equation 2.16).

To achieve a homogeneous sampling of the flux cone, it is crucial that the random walk is sufficiently long, allowing the sampler to converge towards a uniform sampling of feasible fluxes (Wiback et al., 2004). This convergence ensures that the collected samples accurately reflect the diversity of possible metabolic states within the constraints of the GEM. The resulting flux distributions can then be analysed to identify the most probable flux values, assess metabolic pathway utilization, and understand the organism's metabolic adaptability under various conditions (Gelbach et al., 2024).

Modern flux sampling algorithms are optimized for the non-isotropic geometry of the convex polytope defined by the GEM. For example, OptGPSampler uses uses artificial centering hit-and-run to bias the random walk towards the elongated sections of the flux cone (Megchelenbrink et al., 2014). After an initial random location in the flux space is selected, a warm-up phase iterates until the distribution samples is approximately uniform. Once this warm-up phase is complete, every kth point following is generated by the sampler until N points are generated. These two parameters (k, N) control the number of flux samples generated by the algorithm.

The main trade-off for achieving acceptable computational times for flux sampling algorithms lies between sampling accuracy (how uniformly the feasible flux space is explored) and computational efficiency. Early algorithms such as Hit-and-Run and Gibbs sampling were theoretically accurate but extremely slow

to converge in high-dimensional flux cones. Modern approaches—like Coordinate Hit-and-Run with Rounding (CHRR) or Artificially Centered Hit-and-Run (ACHR)—improve efficiency by relaxing exact uniformity or using preconditioning steps that approximate the flux space geometry. These approximations accelerate convergence and reduce runtime, but at the cost of producing only approximately uniform samples rather than perfectly uniform ones.

One of the significant advantages of flux sampling is its ability to explore the solution space without imposing a specific cellular objective, such as biomass maximization (see Equation 2.2.2). This feature allows for an unbiased exploration of the metabolic network's capabilities (Quek and Turner, 2019). However, flux sampling is computationally costly because it requires performing extensive random walks through a high-dimensional flux space. To accurately approximate the distribution of fluxes within the flux cone, the random walk must continue until it reaches its mixing time—the point at which sampled points are effectively independent and uniformly distributed throughout the feasible space. Achieving mixing in high-dimensional spaces, such as those encountered in typical GEMs, is particularly challenging because the complexity and dimensionality of these spaces significantly increase the number of sampling iterations required (Liphardt, 2018). Consequently, flux sampling methods like Markov Chain Monte Carlo (MCMC) and Hit-and-Run algorithms can be quite resource-intensive, especially for models comprising hundreds or thousands of reactions. For comprehensive GEMs, a single sampling run can take on the order of several minutes to hours, depending on the desired convergence quality and computational resources available (Hermann et al., 2019). This high computational cost has led to the development of advanced, more efficient sampling algorithms and software implementations, such as ACHR and optGP, designed specifically to reduce the computational time required for flux sampling analyses (De Martino et al., 2015; Kaufman and Smith, 1998; Megchelenbrink et al., 2014).

2.2.4 The curse of dimensionality

High-dimensional sampling methods, such as those required for exploring the flux cone, are significantly impacted by the curse of dimensionality. This phenomenon describes how the behaviour of data changes as dimensionality increases, presenting fundamental challenges for statistical and computational methods. Specifically, in high-dimensional spaces, data points tend to become nearly equidistant from one another, which undermines the discriminative power of distance-based learning algorithms. As the dimensionality of a space grows, points cluster disproportionately near the boundaries of the feasible space rather than distributing evenly throughout its volume (Wainwright, 2019). Consequently, uniformly sampling high-dimensional spaces is inherently difficult, leading to computational inefficiencies and inaccuracies in statistical learning and optimization (Hastie et al., 2009). These challenges directly affect flux sampling techniques, where ensuring uniform and representative coverage of the flux cone demands sophisticated algorithms and extensive computational resources.

2.2.5 Dimensionality of flux cones across species

Genome-scale models have been built for a large range of species across the kingdom of life, ranging from bacteria (Monk et al., 2017) to multicellular organisms like *Caenorhabditis elegans* (Gebauer et al., 2016) to human cells (Swainston et al., 2016). The dimensionality of the flux cone varies significantly across different species, reflecting differences in the size, complexity, and metabolic diversity of their respective genome-scale metabolic models. There are several metrics which can be used to describe the dimensionality of a flux cone including the number of genes and the number of reactions present in a GEM. Additionally, the nullity of the stoichiometric matrix in Equation 2.15 refers to the number of linearly independent flux vectors (solutions) that satisfy the steady-state condition, where metabolite production and consumption rates balance out. Mathematically, this nullity corresponds to the dimension of the null space (kernel) of the stoichiometric matrix. Consequently, the nullity directly indicates the dimensionality of

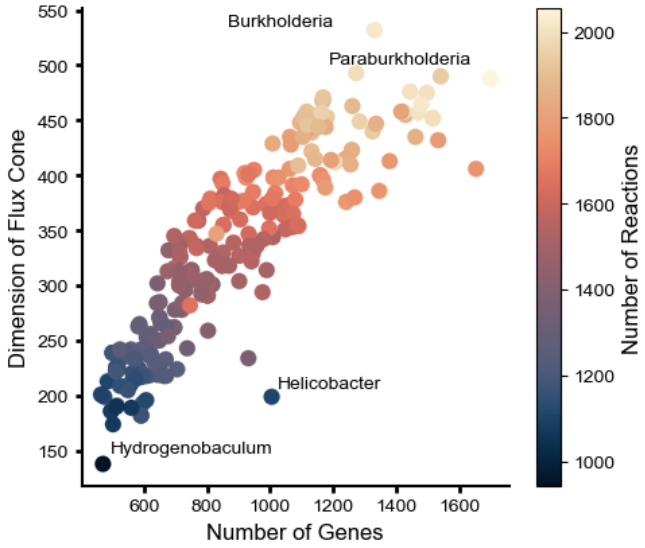


Figure 2.4: Genome-scale metabolic models from across the bacterial kingdom. The dimension (y-axis) is the nullity of the S matrix.

the flux cone: a higher nullity means a higher-dimensional solution space. Figure 2.4 shows the range of models from across the bacterial kingdom. Models for $N = 244$ bacterial species were obtained from Plata et al., 2015 and species names were obtained from Ramon and Stelling, 2023. These GEMs were generated programmatically using the SEED toolkit (Henry et al., 2010) and the stoichiometric matrices of each model were extracted, along with the number of genes and reactions in each model. The largest models have over 1500 genes and 2000 reactions, while the smallest models are below 600 genes and 1000 reactions.

2.3 Hybrid modelling approaches

Machine learning and mechanistic simulation approaches both try to predict the behaviour of a system; however, they have traditionally been applied to different types of problems. ML models thrive in applications where causal relationships are unknown or cannot be assumed but large amounts of data are available. However, since ML techniques are solely dependent on data, they can struggle with sparse or biased data, leading to predictions that are unreliable or not based on physical predictions (Alber et al., 2019). Alternately, mechanistic modelling derives from known causal relationships in fields where data is sparse, noisy,

or unavailable (Rueden et al., 2020). Most modelling in systems biology and metabolism prior to the introduction of machine learning was mechanistic. In recent years, hybrid models which combine aspects of both paradigms have been developed. These hybrid approaches can prove useful when classical mechanistic models cannot handle new high-dimensional data or only partial mechanistic models exist. Several reviews give different pathways hybrid models take to integrate mechanistic modelling and machine learning:

- Synthetic data generated from mechanistic models provides the input to a machine learning algorithm
- Reinforcement learning or optimization techniques guide the mechanistic model towards the right representation of reality
- Pipelines which improve mechanistic model predictions based on machine learning

These methodologies can be applied to a wide range of tasks, from biosensor engineering to metabolic flux prediction. One of the most common hybrid modelling methods is where a machine learning model such as an artificial neural network (ANN) is used as a surrogate to fully replace a complex, computationally costly mechanistic model. One work applied machine learning models such as Gaussian processes, gradient-boosted trees, and ANNs to replace agent-based modelling methods (Angione et al., 2022). Gherman et al., 2023 reviews other methods which aim to combine constraint-based methods with machine learning to identify pharmacological effects, cell fate, improve GEM curation. Procopio et al., 2023 found that when genome-wide models were available, mechanistic models for genomic, proteomic, or metabolomic networks could be exploited to generate large amounts of synthetic data to tune ML-based algorithms. In contrast, when concerned more with cellular dynamics, agent-based models tuned by reinforcement learning techniques were more common, as they are inspired by biological evolution. Alternatively, a genome-scale metabolic model can be used as the graph structure for a graph-based machine learning method such as a graph neural network (GNN) or mass flow graph (Occhipinti et al., 2024). Multi-modal graph neural networks have also been applied to include not just information from

a mechanistic model but data from other sources such as -omics sequence data or imaging data (Holzinger et al., 2021). Hybrid modelling frameworks provide a powerful foundation for tackling complex biological design challenges, especially in contexts where neither purely data-driven nor fully mechanistic approaches are sufficient. In particular, metabolic circuits, which operate across multiple time scales and regulatory layers, present a compelling use case for such methods.

Chapter 3

Design of multiscale engineered gene circuits with Bayesian optimization

Existing computational methods struggle to design biological circuits effectively, particularly when these circuits involve dynamic feedback loops. Such loops often operate across multiple temporal or concentration scales, further complicating accurate modelling and control. This chapter presents a machine learning method for the efficient optimization of biological circuits across scales. The method relies on Bayesian Optimization (BayesOpt), a technique commonly used to fine-tune deep neural networks, to learn the shape of a performance landscape and iteratively navigate the design space towards an optimal circuit. I first enumerate the optimization problem under study and give context for the size and character of the design space. My method allows the joint optimization of both circuit architecture and parameters, and hence provides a feasible approach to solve a highly non-convex optimization problem in a mixed-integer input space. I next provide a demonstration of the method in a toy model system and benchmark BayesOpt's results against several standard methods. I then illustrate the applicability of the method on several gene circuits for controlling biosynthetic pathways with strong nonlinearities, multiple interacting scales, and using various performance objectives. The method efficiently handles large multiscale problems and enables

parametric sweeps to assess circuit robustness to perturbations, serving as an efficient *in silico* screening method prior to experimental implementation. The contents of this chapter are adapted from a paper published in *ACS Synthetic Biology* entitled "Bayesian optimization for design of multiscale biological circuits". The work underlying this paper was partially conducted in the MSc stage of my programme and was previously written up in my MSc thesis (Merzbacher, 2022). Work conducted or presented as part of my MSc will be clearly signposted throughout the chapter to make it clear to the reader the extent of novel work conducted during the PhD stage.

3.1 Background and motivation

Engineering molecular circuits with prescribed functions is a core task in synthetic biology (Brophy and Voigt, 2014; Lazebnik, 2002). Bioengineers aim to construct networks of interacting genes, proteins, or small molecules to execute specific tasks in a cell, inspired by how electrical engineers construct circuits with precise behaviours (Cameron et al., 2014). These molecular circuits typically comprise genetic components that interact through regulatory mechanisms such as transcription, translation, and protein-protein interactions to achieve desired functionalities including oscillation, memory storage, logic operations, and feedback control (Elowitz and Leibler, 2000; Gardner et al., 2000; Nielsen et al., 2016). Computational design tools are widely employed by engineers to discover circuits with specific dynamics and accelerate the experimental design process (Gurdo et al., 2023; Li et al., 2017; Ma et al., 2009; MacDonald et al., 2011; Qiao et al., 2019). In particular, optimization-based strategies can be employed to search over design space and single out circuits predicted to fulfil a desired function or result in specific system dynamics (Dasika and Maranas, 2008; Hiscock, 2019; Otero-Muras and Banga, 2017; Verma et al., 2021). However, significant challenges to rapid computational optimization of circuits remain. First, the large number of possible components and circuit architectures leads to a combinatorial explosion in the size of the design space, which then requires search strategies

which can navigate the large space efficiently. Second, circuit design requires both the specification of circuit architecture, i.e. the circuit "wiring diagram" and the strength of interactions among molecular components. Since circuit architectures are discrete choices and molecular interactions depend on continuous parameters such as binding rate constants, the design problem is a mixed-integer optimization, which can be notoriously difficult to solve (Banga, 2008). Finally, when circuits operate across multiple scales, their computational models become numerically stiff (Hairer and Wanner, 1996), resulting in extremely slow simulations that make the large number of parameter values which must be tried for optimization infeasible. I will address the size of the design space in dynamic systems, the discrete-continuous nature of the circuit optimization problem, and the multiscale nature of biological systems as three features of the problem that make it particularly challenging and that my approach aims to address.

3.1.1 Navigating the gene circuit design space

Previous work on computational circuit design has largely focused on genetic circuits that operate in isolation from other layers of the cellular machinery (Figure 3.1). Networks of transcriptional regulators can influence each other to produce behaviours such as oscillations, ultrasensitivity, bistability, or differentiation cascades (Elowitz and Leibler, 2000; Hiscock, 2019; Oyarzún and Chaves, 2015). However, even with a small number of regulators, the possible combinations of activation and repression loops and the strength ranges of interactions create a large design space that cannot be navigated through experimental design alone. Exhaustive computational search which tests all possible circuit topologies guarantees the optimal solution will be found but is impractically slow for larger circuits (Blanchini et al., 2014; Li et al., 2017; Ma et al., 2009; Qiao et al., 2019). For scale, if one considers all possible circuit with three genes and assume that each gene can activate, repress, or have no effect on itself and the other genes, there are over 19,000 possible circuit topologies. Each of these topologies may have entirely different and non-intuitive dynamics due to the nonlinear interactions between circuit components. Furthermore, as the number of circuit components grows

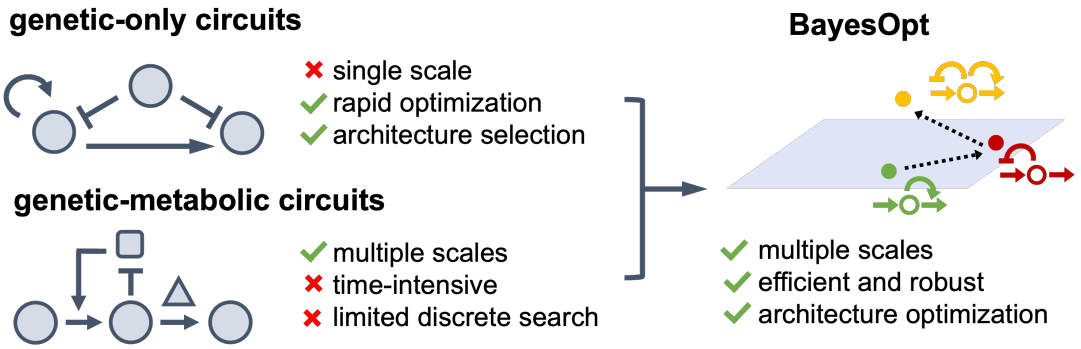


Figure 3.1: Schematic of strategies for the design of circuit architectures and parameters for single and multiscale circuits.

the size of the design space grows exponentially, following a 3^{N^2} scaling for N regulators, making exhaustive exploration increasingly infeasible. Consequently, for anything other than the simplest networks, efficient algorithms to navigate this large design space are necessary.

3.1.2 Mixed-integer optimization of circuit architectures

Designing a biological circuit requires specification of both the discrete interactions between circuit components (e.g. a gene or protein represses or activates a target) and the strength of those interactions. This feature frames the problem as a mixed-integer nonlinear programming problem (MINLP), which already places it in a class of problems which is known to be challenging to solve using traditional optimization methods (Chachuat et al., 2005; Lee and Leyffer, 2011). Many of the approaches for MINPL problems struggle when the objective function landscape is nonconvex; that is that local minima are not the global minima. Exhaustive sampling of biological circuits has revealed that many of their objective function landscapes resemble a series of nonconvex "pockets" (Figure 3.3; Hiscock, 2019). Computational approaches which use traditional optimization methods use decompositions or multiobjective optimization to try to approach the optimal circuit architecture (Dasika and Maranas, 2008; Otero-Muras and Banga, 2017). Bayesian design (Gonzalez et al., 2015; Woods et al., 2016) and gradient-based machine learning methods (Hiscock, 2019; Shen et al., 2021) provide an efficient alternative to exhaustive search but can face some of the same

problems as MINLP methods and get stuck in local minima. Other algorithms use approaches from control theory to analyse network structures and identify circuits which might lead to robust adaptation or other desirable behaviors (Araujo and Liotta, 2018; Bhattacharya et al., 2022; Briat et al., 2016, 2018; Drengstig et al., 2008). While these methods differ on their specific modelling strategies and assumptions, they all require computational simulations at thousands to millions of locations in the design space to optimize circuit architectures. This scale of simulations is feasible for simpler systems; however, as the number of circuit components increases, the computational costs become significant and limit the applicability of current optimization methods. In addition, the circuits under study have previously only included components from a single scale (i.e. protein-protein interactions or gene regulatory networks) and have neglected the dynamics of small-molecule metabolites in the pathways under study. I next consider how incorporating multiple cellular scales into models of biological circuits further complicates the problem.

3.1.3 Multiscale gene circuits and dynamic control systems

Biological circuits include components that operate across various scales of cellular organization, such as gene expression, signalling pathways (Shaw et al., 2019) or metabolic processes (Zhang et al., 2012). One notable application where these challenges appear is in the design of genetic circuits for the dynamic control of metabolic pathways (Doong et al., 2018; Dunlop et al., 2010; Oyarzún and Stan, 2013; Xu et al., 2014; Zhang et al., 2012). These systems have received substantial attention thanks to several successful implementations that improved yields as compared to classic techniques in metabolic engineering (Hartline et al., 2021; Ni et al., 2021). A more detailed overview of these systems is discussed in Section 1.1.3; the key principle is to put enzymatic genes under the control of metabolite-responsive mechanisms that couple heterologous expression to the concentration of a pathway intermediate (Zhang et al., 2012). This creates feedback loops between enzyme expression and pathway intermediates that allow the control of pathway activity in response to upstream changes in growth

conditions or precursor availability. Such dual genetic-metabolic systems are particularly challenging to simulate efficiently because metabolites and enzymes vary in different timescales, from milliseconds (enzyme kinetics) to minutes (enzyme expression), and they also appear in vastly different concentrations; in bacteria enzymes are typically expressed in nanomolar concentrations, whilst metabolites are found typically above the millimolar range (Tonn et al., 2019).

Moreover, the implementation of these systems is costly and requires substantial experimental fine-tuning. As a result, a central task prior to implementation is the choice of a suitable feedback control loops between metabolites and enzymatic genes, and the strength of interactions between metabolites and actuators of gene expression such as transcription factors (Mannan et al., 2017) or riboregulators (Zhou and Zeng, 2015). The design of control architectures is particularly important, because there are many ways of building similar control loops (Chaves and Oyarzún, 2019), for example by employing combinations of transcriptional activators and repressors (Liu and Zhang, 2018; Stevens and Carothers, 2015), that may differ in their performance and cost of implementation.

The computational cost associated with numerically integrating stiff systems make the simulation of thousands of candidate circuits infeasible. Therefore, a more efficient model of candidate circuit selection is needed. When modelling a pathway under gene regulation using ordinary differential equations (see Section 2.1.1), the gap in timescales between metabolite kinetics and transcriptional dynamics leads to numerically stiff systems. Stiff systems change quickly over short timescales while slowly over others. To capture both the rapid dynamics while running simulations for a long enough time to capture slower evolutions to steady state, explicit numerical integration methods require very slow time steps.

3.2 Efficient joint architecture and parameter optimization

In this section, I present a fast and scalable machine learning approach for optimization of multiscale circuit architectures and parameters (Figure 3.1). The

method is based on Bayesian optimization coupled with differential equation models, and I highlight its utility in various models of metabolic pathways under genetic feedback control (Frazier, 2018). I first describe how the method works, using a toy example of a simple pathway. I show that the method converges rapidly and outperforms other optimizers by a substantial margin in benchmarking studies. I also show that the method scales well to large systems of differential equations, provided the number of design parameters remains under approximately 20 dimensions. The method can help speed up the design of synthetic biological circuits and presents a novel approach to explore the design space ahead of implementation. The method was initially developed as part of my MSc thesis work in the first year of the CDT programme; however, all figures were generated during the first year of the PhD. I consider four case studies: a toy model, models of glucaric acid and fatty acid production, and a large model of p-aminostyrene synthesis. All case studies were initially selected during the MSc thesis and the sample results from the toy model and glucaric acid model remain unchanged albeit with new visualizations. The benchmarking experiments were run in the PhD period. The fatty acid model was entirely revamped during the PhD and the work on multiple objective functions is new; as a result, I focus on this work in more detail in Section 3.3.2.

3.2.1 Framing the problem and Bayesian optimization

In general, the circuit design task can be stated as the following mixed-integer optimization problem:

$$\begin{aligned} &\min_{p_d, p_c} J(x, p_c, p_d), \\ &\text{subject to:} \\ &\qquad \mathrm{d}x/\mathrm{d}t = h(x), \\ &\qquad p_c \in \mathcal{C},\ p_d \in \mathcal{D}, \end{aligned} \tag{3.1}$$

where $J(x, p_c, p_d)$ is a performance objective to be optimized over a space of continuous parameters p_c and a discrete set of circuit architectures p_d. The ODE in

Eq. (3.1) describes the temporal dynamics of circuit components and are typically built from mass balance relations comprised in the nonlinear function $h(x)$. Common examples of continuous parameters in applications are binding affinities between DNA and regulatory proteins, or the strength of protein-protein interactions. Conversely, circuit architecture would typically involve various combinations of positive and negative feedback loops among molecular species. I have stated the problem as minimization of J, but similar formulations can be posed as a maximization problem.

In this paper, I propose to solve the design problem in Eq. (3.1) with Bayesian optimization (BayesOpt), a class of algorithms designed for problems with objective functions that are expensive to compute. BayesOpt is a global optimization technique that treats the objective function as a random variable with a prior distribution on it. The algorithm creates a statistical model of the objective through subsequent evaluations, which are employed to build a posterior distribution and determine the next set of inputs to evaluate (Figure 3.2). A typical application of BayesOpt is in design of experiments (Frazier, 2018) where the objective function requires measuring data with costly and/or slow experimental work. In deep learning, BayesOpt is widely employed for model selection, as traditional grid search approaches require large computing resources to train many architectures with combinations of various hyperparameters (Bergstra et al., 2013; Snoek et al., 2012). I implemented BayesOpt using the lightweight hyperparameter tuning package Hyperopt, although alternatives such as BoTorch are available and can provide a wider range of algorithm options (Bagge Carlson, 2018; Balandat et al., 2020).

The performance objective J can be flexibly used to model common design goals such as production flux, yield or titer, as well as cost-benefit tasks that balance production with the deleterious impact of the pathway on the physiology of the host. I discuss multiple possible objectives later in the chapter (see Section 3.3.2; however, in the toy case I consider the minimization of

$$J = \alpha_1 J_{\text{prod}} + \alpha_2 J_{\text{cost}}, \tag{3.2}$$

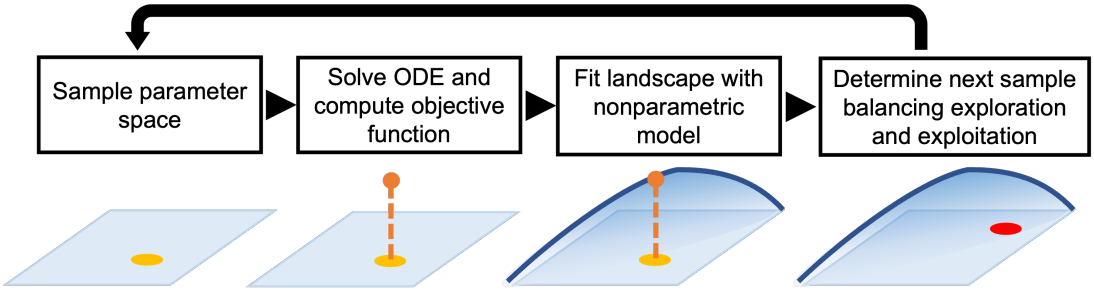


Figure 3.2: Schematic of a mixed-integer Bayesian optimization loop. My algorithm uses a nonparametric statistical model known as Tree of Parzen Estimators (Bergstra et al., 2013) and Expected Improvement (EI) as the acquisition function to balance exploration and exploitation towards the global minimum.

where J_{prod} was designed so that its minimization is equivalent to maximization of the production flux, and J_{cost} penalizes total amount of enzyme expressed during the culture. The parameters α_1 and α_2 are positive weights used to control the balance between the costs and benefits of expressing the heterologous pathway. In general, their values are dependent on the design objectives of the engineer building the system and can be seen as a trade-off hyperparameter between different terms in the scalar objective function. In this chapter, I set them to make the objective terms operate over roughly the same order of magnitude.

The objective function is regarded as a random variable to be optimized over an input space comprised of continuous parameters and a set of discrete circuit architectures. At each iteration, the optimization algorithm selects both a discrete genetic control architecture (p_d) and a set of continuous parameters (p_c) which control the strength of the genetic control circuits. To illustrate the challenge of jointly optimizing circuit architecture and parameters, in Figure 3.3 I show a schematic of the design space. In this example, I consider a toy system with four discrete control architectures (for further details, see Section 3.2.2), which include open loop control as well as three different implementations of negative feedback control using a metabolite-responsive transcription factor. Negative feedback is widely employed in gene circuits as it has substantial benefits in terms of robustness and performance, and their properties have been extensively studied in the literature (Hartline et al., 2020; Venayak et al., 2015; Zhu et al., 2021).

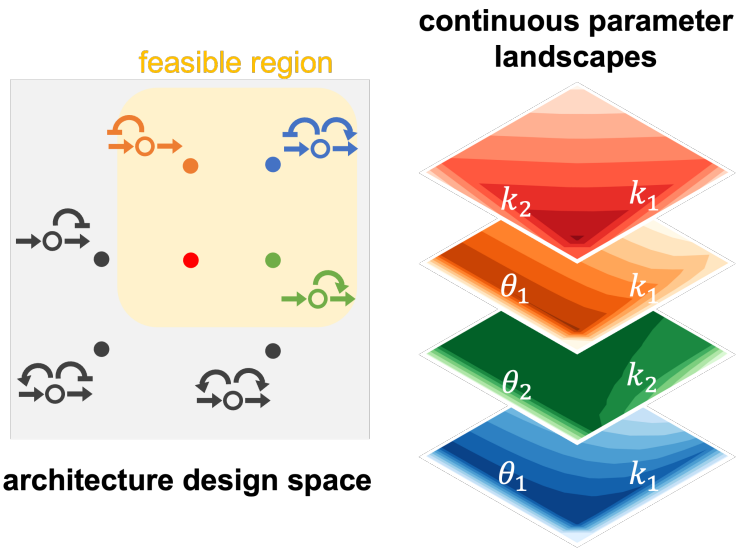


Figure 3.3: Discrete-continuous design space of the toy model. Positive feedback loops are excluded from the architecture space (left) as these are prone to multistability (Oyarzún and Chaves, 2015). Heatmaps (right) show the value of the objective J.

The four control architectures under consideration reside at different discrete points in the architecture space. Within each architecture, I observe substantial variations in the shape of the performance landscape J as a function of the dose-response parameters p_c. There are cases with convex landscapes with a clear optimum (e.g. dual control) and landscapes with flat basins where most optimization algorithms would struggle to find the optimum (e.g. downstream activation). When searching over the space of architectures and parameters simultaneously, the problem becomes a mixed-integer, non-convex optimization that is extremely challenging to solve with traditional approaches.

One key advantage of Bayesian methods is that they are not gradient-based, and therefore are not constrained to navigate the space smoothly in the direction of steepest descent. Gradient-based methods can get trapped in local minima and struggle to find the global optimum, especially in highly nonconvex landscapes like the ones presented here. In contrast, BayesOpt does not converge by chasing minima directly but rather by modelling the entire objective function landscape. The method can perform multiple "jumps" between distant locations in the discrete-continuous search space, where each subsequent sample is selected to maximize the expected improvement on the best sample found so far.

3.2.2 Demonstration of method in toy model

For the circuit design task in Eq. (3.1), if the biological system has multiple scales the computation of objective J requires solving a stiff ODE in many locations of the mixed-integer search space, which can rapidly become infeasible. To first establish a baseline for the performance of my method, I employed a simple toy pathway model that displays common features found in real metabolic pathway (Figure 3.4). The toy pathway under study (see Figure 3.4) has two pathway intermediates and two enzymes under transcriptional control (Verma et al., 2021). The substrate represses a metabolite-responsive transcription factor which can act as an activator or repressor for the transcription of enzymes 1 and 2. (see Figure 3.1). The influx from native precursors and draw from pathway metabolism are modelled as constant rates. While there is no direct biological correlate for this pathway, it exemplifies the most basic engineered pathway possible and thus is useful primarily as a proof-of-concept for the BayesOpt method.

The mass balance equations for the toy branched pathway are

$$\begin{aligned}
\frac{\mathrm{d}x_0}{\mathrm{d}t} &= V_{\mathrm{in}} - e_0\frac{k_{\mathrm{cat}}x_0}{k_{\mathrm{m}}+x_0} - e_1\frac{k_{\mathrm{cat}}x_0}{k_{\mathrm{m}}+x_0} - \lambda x_0,\\
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= e_1\frac{k_{\mathrm{cat}}x_0}{k_{\mathrm{m}}+x_0} - e_2\frac{k_{\mathrm{cat}}x_1}{k_{\mathrm{m}}+x_1} - \lambda x_1,\\
\frac{\mathrm{d}e_1}{\mathrm{d}t} &= u(x_1,k_1,\theta_1,n) - \lambda e_1,\\
\frac{\mathrm{d}e_2}{\mathrm{d}t} &= u(x_1,k_2,\theta_2,n) - \lambda e_2,
\end{aligned} \tag{3.3}$$

where x_0 and x_1 are the metabolite concentrations and e_1 and e_2 are the enzyme concentrations. The input flux V_{in} is the flux from native precursors. The constant λ is the culture growth rate and is fixed at $1.93 \cdot 10^{-4}s^{-1}$, which corresponds to a 26 minute doubling time in *E. coli*. Each enzyme follows standard Michaelis-Menten kinetics has two fixed kinetic parameters, k_{cat} and k_{m}. For simplicity, all enzymes are assumed to have the same kinetic parameter values. The function $u(x, k_i, \theta_i, n)$ describes the expression rates of each enzyme, which can be modelled by one of three functions dependent on genetic control architecture (see Equation 3.4). The dose-response curves are encoded by the optimizable

parameters k_i and θ_i and the fixed Hill coefficient $n = 2$. The form of $u(x, k, \theta)$ is determined by the architecture value and modelled using a sigmoid function which incorporates multiple molecular processes, including metabolite-TF and TF-DNA binding, and is simpler to parameterize using limited kinetic data. Additionally, dynamic pathways often express the TF constitutively, which reduces the effect of TF expression on performance (Xu et al., 2014). I consider three architecture forms: activation, repression, and open loop control:

$$u(x_i, k_i, \theta_i) = \begin{cases} k_i & \text{(no control)}, \\ \dfrac{k_i x^2}{{\theta_i}^2 + x^2} & \text{(activation)}, \\ \dfrac{k_i {\theta_i}^2}{{\theta_i}^2 + x^2} & \text{(repression)}. \end{cases} \tag{3.4}$$

As a result, the toy pathway has four continuous parameters under study: k_1, θ_1, k_2, θ_2, two associated with each enzyme. The four optimizable parameters are constrained on the following ranges:

$$10^{-3} \leq \theta_1, \theta_2 \leq 10, \tag{3.5}$$

$$10^{-7} \leq k_1, k_2 \leq 10^{-3}. \tag{3.6}$$

The possible architectures are limited to three architectures (upstream repression, downstream activation, and dual control) which include only negative feedback loops, in addition to an open-loop control with no dynamic feedback loops (see Figure 3.1). This architecture limitation prevents a trivial equilibrium where $x_1 = e_1 = e_2 = 0$ and removes the possibility of undesirable multistability or oscillatory behaviour. All parameter values are given in Table 3.1.

The α_1 and α_2 values given in Table 3.1 were chosen empirically to scale the two objectives to be similar in magnitude and the overall loss to be between 0

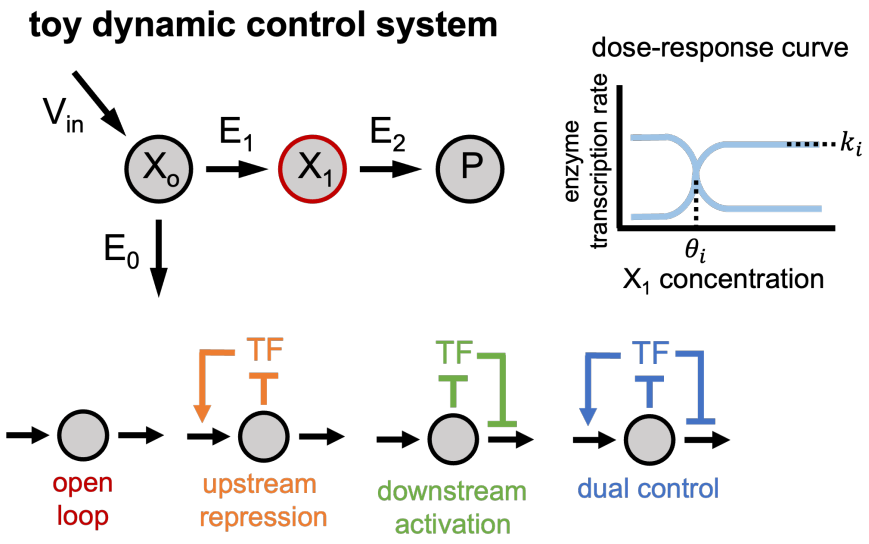


Figure 3.4: Schematic of example metabolic pathway under gene regulation. The dose response curve (top right) for the transcription factor (TF) that binds the intermediate X_1 as either an activator or a repressor. The negative feedback architectures are named based on the net effect of the metabolite on gene expression.

Parameter	Value	Units
k_{cat}	12	μM/s
k_{m}	10	μM
V_{in}	1	μM/s
e_0	0.0467	μM
λ	1.93E-4	1/s
α_1	1E-5	N/A
α_2	1E-2	N/A

Table 3.1: Parameter values of the toy model in Equation 3.3. All model parameters come from Verma et al., 2021.

Pathway Product	Toy Product
Decision Variables	$k_1, k_2, \theta_1, \theta_2$
Architectures	Open Loop, Upstream Repression, Downstream Activation, Dual Control
Pathway Metabolites	X_0, X_1
Pathway Enzymes	E_0 (constant), E_1, E_2
Integration Time	$5 \cdot 10^4$s
Initial Conditions	$x_0(0) = 2290\mu$M $x_1(0) = 0\mu$M $e_1(0) = 0\mu$M $e_2(0) = 0\mu$M

Table 3.2: Toy model summary. The names for the decision variables and pathway metabolites and enzymes are included, as are the initial conditions.

and 1. The loss equation for the objective function is

$$J = \alpha_1 \underbrace{\int_0^T \left| V_{\text{in}} - e_2(t) \frac{k_{\text{cat}} x_1(t)}{k_{\text{m}} + x_1(t)} \right| \mathrm{d}t}_{\text{production loss}} + \alpha_2 \underbrace{\int_0^T \left(u(x_1(t), k_1, \theta_1) + u(x_1(t), k_2, \theta_2) \right) \mathrm{d}t}_{\text{pathway cost}} . \tag{3.7}$$

The first term describes the absolute difference between the influx to the pathway V_{in} and the flux utilized by the pathway to produce product. The second term describes the total heterologous enzyme expression, which exerts a burden on the cell and this is a measure of the pathway cost.

I ran each simulation of the model for a total time of $5 \cdot 10^4$ seconds (13.8 hours) to ensure the final values would be at steady state. All metabolites and enzymes had an initial concentration of 0μM except for x_0, the native enzyme, which had an initial concentration of 2290μM (Verma et al., 2021). A summary of the model details is given in Table 3.2.

3.2.3 Benchmarking of method against state-of-the-art and hyperparameter tuning

I implemented a BayesOpt routine to jointly compute the architecture (p_d) and dose-response parameters (p_c) that minimize the performance objective for the toy model system in Eq. (3.2.2). I benchmarked the algorithm's performance against several other methods, including a random search, an exhaustive grid search, a gradient-based method, and a genetic algorithm (Figure 3.5). Figure 3.5 shows the results for random sampling until a low objective function was reached ($N = 1,000$ samples), grid search ($N = 40,000$), a genetic algorithm (Solgi, 2020) ($N = 100$ individuals, $N = 1000$ generations), and a gradient-based optimizer to find optimal continuous parameter values for each architecture (Virtanen et al., 2020). The algorithm was able to compute optimal solutions rapidly and robustly. I ran the BayesOpt routine for 1000 iterations, which averaged 27 seconds per run across 100 runs. The routine robustly converged to within 2.5% of the mean optimal objective function value (standard deviation). BayesOpt runs significantly faster than the other methods, and provides a 30-fold computational time improvement over a genetic algorithm. The accuracy of the optimum, quantified by the minimal value of the objective function, is on average 11.4% worse than the genetic algorithm, but this falls within the variation of the latter across several runs. I also note that the traditional gradient-based optimizer proved unreliable and failed to converge on 14.5% of runs. The random sampling algorithm converged to the same objective function value (0.065) as the BayesOpt method.

Following benchmarking, I tested the robustness of the BayesOpt method to hyperparameter variations. The tree-structured Parzen estimator method (TPE) implemented in the Hyperopt package has a hyperparameter γ which controls the balance of exploration and exploitation. TPE models the search space using two nonparametric density estimators: $l(x) = p(x \mid y < y^*)$ for the "good" samples (configurations with objective values below a quantile threshold y^*) and $g(x) = p(x \mid y \geq y^*)$ for the "bad" samples. Both $l(x)$ and $g(x)$ are represented as

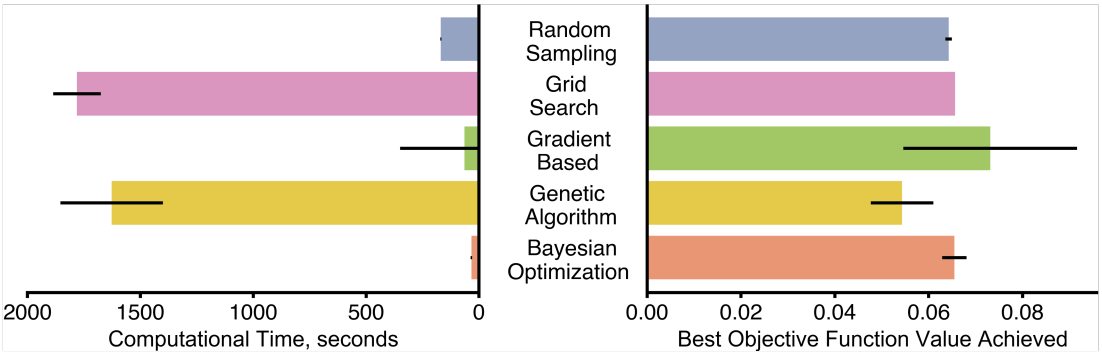


Figure 3.5: Comparison of BayesOpt against other strategies using the toy model as a benchmark.

Tree-structured Parzen Estimators, kernel density estimators that use Gaussian kernels for continuous variables and categorical distributions for discrete ones. The parameter γ defines the quantile threshold y^*, determining the fraction of observations classified as "good." The default value in Hyperopt is $\gamma = 0.15$, meaning that 15% of samples are used to construct $l(x)$ and the remaining 85% form $g(x)$. New candidate points are drawn from $l(x)$ and evaluated according to the expected improvement criterion, which favours configurations where the ratio $l(x)/g(x)$ is high. Larger values of γ promote broader exploration of the loss landscape, while smaller values favour local exploitation around previously successful regions.

I found that regardless of the γ value chosen, the distribution of architectures explored was similar (see Figure 3.6). To give a single scalar metric, I computed the fraction of samples in each quarter taken from the majority architecture, or the architecture most commonly sampled across the optimisation. This fraction was computed as

$$\frac{\text{majority architecture samples}}{\text{optimization iterations}}. \tag{3.8}$$

I split the optimization into four quarters (each 250 iterations) and computed the majority architecture for each fraction for different γ values. In the case of the glucaric acid model, the majority architecture is dual control. The TPE hyperparameter γ was tuned by cloning the Hyperopt package repository and manually changing the value in the source code. Hyperparameter values of $\gamma =$

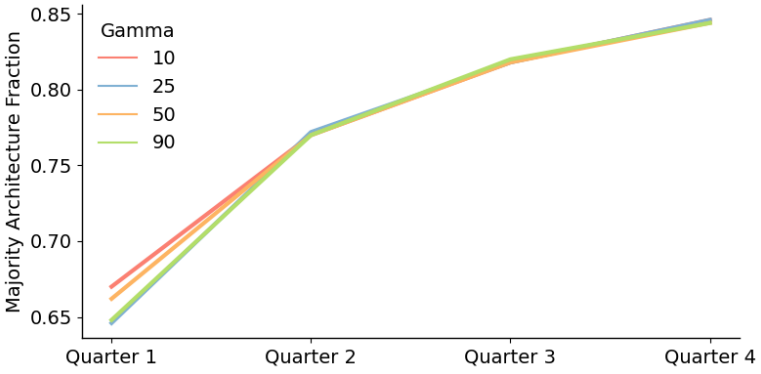


Figure 3.6: Hyperparameter Tuning of TPE. A $N = 1000$ iteration BayesOpt run was split into 4 quarters and the majority architecture fraction computed for each.

10, $\gamma = 25$, $\gamma = 50$, and $\gamma = 90$ were considered and a single optimisation was run for each value. The total optimisations were split into four quarters by iteration and the percentage of sample drawn from the majority architecture (in this case, dual control).

The majority architecture fraction rose with each quarter of iterations as the BayesOpt routine increasingly focused on low-loss dual control parameter value combinations; however, there was no statistical difference between the different γ values chosen. These results show that the default γ value is sufficient as TPE is relatively insensitive to hyperparameter tuning. This feature of TPE makes it quick to implement without significant application-specific tuning but may cause it to fail to adapt to certain specific cases. However, I did not observe TPE fail to converge in any of the systems under study.

The speed of my approach enables the computation of large solution ensembles under model perturbations such as sweeps of key model parameters. In addition, my method can search high-dimensional mixed-integer design spaces. I next illustrate the versatility of the approach in a range of relevant real-world pathways that require solving the optimization problem for large samples of parameter values.

3.3 Applications of method to circuit design

3.3.1 Assessing robustness of control circuits to uncertainty in enzyme kinetic parameters

A challenge in building pathway models is the substantial uncertainty on the enzyme kinetic parameters; this is particularly critical for pathways that include regulatory mechanisms such allostery or product inhibition, which are often poorly characterized. Databases such as BRENDA (Schomburg et al., 2017) often have insufficient data on enzyme kinetics for a particular host strain or substrate of interest. Since pathway dynamics can strongly depend on enzyme kinetics, the parametric uncertainty requires extensive sweeps of kinetic parameters to determine the robustness of a specific control architecture deemed to be optimal.

Glucaric acid model

I focused on a pathway for synthesis of glucaric acid in *E. coli* (Figure 3.7). Glucaric acid is a key precursor for a number of biomedical applications and its pathway has been implemented previously in *E. coli* (Moon et al., 2009). As shown in Figure 3.7, the pathway branches from glucose-6-phosphate (G6P) in upper glycolysis and contains three enzymatic steps (Ino1, SuhB, and MIOX). The first step (inositol-3-phosphate synthetase, Ino1, taken from *S. cerevisiae*) catalyzes glucose-6-phosphate (G6P) from central carbon metabolism into myoinositol-1-phosphate. Myoinositol-1-phosphate is subsequently converted into glucaric acid by inositol-1-monophosphatase (SuhB, native to *E. coli*), myoinositol oxidase (MIOX, from *Mus musculus*) and uronate dehydrogenase (Udh, from *Pseudomonas syringae*). In addition to being exported from the cell and acting allosterically on MIOX, myoinositol can sequester the transcription factor IpsA, a dual transcriptional regulator from *Corynebacterium glutamicum*. IpsA can then act on the transcription of Ino1 or MIOX (Doong et al., 2018). The glucaric acid pathway provides a more complex, real-world application that builds on the toy model and includes reversible and allosteric reactions. The dynamic control im-

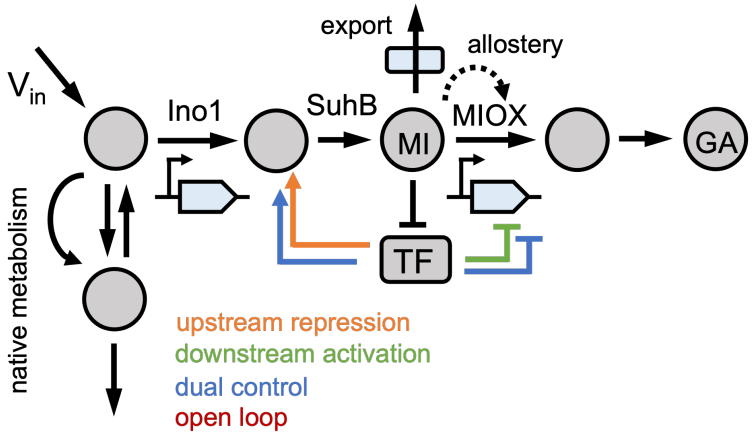


Figure 3.7: Schematic of a dynamic pathway for production of glucaric acid in *Escherichia coli* (Doong et al., 2018). The enzyme SuhB is not rate-limiting and thus is not modeled explicitly. As in Figure 3.4, the architectures are named based on the net effect of the metabolite on gene expression. Figure adapted from (Merzbacher, 2022).

plementation by Doong and colleagues led to a 2.5-fold increase in product titer over static metabolic engineering methods.

The glucaric acid model was chosen for its increased complexity. While maintaining the same number of possible architectures as the toy model, this model incorporates allosteric control and reversible reactions, two more complex biological interactions not present in the toy model. I employed a previously developed ODE model (Verma et al., 2021) that was parameterized using a combination of enzyme kinetic data and omics measurements, and considered the same four control architectures as in the previous toy example, including various alternative implementations of negative feedback control. The mass balance equations for

the glucaric acid production pathway (see Figure 3.7) are

$$\begin{aligned}
\frac{\mathrm{dg6p}}{\mathrm{d}t} &= V_{\mathrm{in}} - \mathrm{zwf}\frac{k_{\mathrm{cat,\ zwf}}\mathrm{g6p}}{k_{\mathrm{m,\ zwf}} + \mathrm{g6p}} - \mathrm{pgi}\frac{k_{\mathrm{cat,\ pgi}}(\mathrm{g6p} - (\mathrm{f6p}/k_{\mathrm{eq}}))}{\mathrm{g6p} + k_{\mathrm{m,\ pgi,\ g6p}}(1 + (\mathrm{f6p}/k_{\mathrm{m,\ pgi,\ f6p}}))} - \lambda \cdot \mathrm{g6p}, \\
\frac{\mathrm{df6p}}{\mathrm{d}t} &= \mathrm{pgi}\frac{k_{\mathrm{cat,\ pgi}}(\mathrm{g6p} - (\mathrm{f6p}/k_{\mathrm{eq}}))}{\mathrm{g6p} + k_{\mathrm{m,\ pgi,\ g6p}}(1 + (\mathrm{f6p}/k_{\mathrm{m,\ pgi,\ f6p}}))} \\
&\quad + 0.5\mathrm{zwf}\frac{k_{\mathrm{cat,\ zwf}}\mathrm{g6p}}{k_{\mathrm{m,\ zwf}}} - \frac{k_{\mathrm{cat,\ pfk}}\mathrm{f6p}^3}{k_{\mathrm{m,\ pfk}}^3 + \mathrm{f6p}^3} - \lambda \cdot \mathrm{f6p}, \\
\frac{\mathrm{dMI}}{\mathrm{d}t} &= \mathrm{ino1}\frac{k_{\mathrm{cat,\ ino1}}\mathrm{g6p}}{k_{\mathrm{m,\ ino1}}} - \frac{V_{\mathrm{m,\ t,\ MI}}\mathrm{MI}}{k_{\mathrm{m,\ t,\ MI}} + \mathrm{MI}} - \mathrm{MIOX}\frac{k_{\mathrm{cat,\ eff}}\mathrm{MI}}{k_{\mathrm{m,\ MIOX}} + \mathrm{MI}} - \lambda \cdot \mathrm{MI}, \\
\frac{\mathrm{dino1}}{\mathrm{d}t} &= u(\mathrm{MI}, k_{\mathrm{ino1}}, \theta_{\mathrm{ino1}}) - \lambda \cdot \mathrm{ino1}, \\
\frac{\mathrm{dMIOX}}{\mathrm{d}t} &= u(\mathrm{MI}, k_{\mathrm{MIOX}}, \theta_{\mathrm{MIOX}}) - \lambda \cdot \mathrm{MIOX},
\end{aligned} \tag{3.9}$$

where Ino1 and MIOX are the enzymes in the pathway and g6p, f6p, and MI are the substrates. The enzyme parameters $k_{\mathrm{cat}}, k_{\mathrm{eq}}, k_{\mathrm{m,\ x}}$ and $k_{\mathrm{m,\ y}}$ are all fixed kinetic parameters specific to each enzyme. The effective substrate activation constant $k_{\mathrm{cat,\ eff}}$ has two additional activation kinetic parameters k_{a} and a which must be specified:

$$k_{\mathrm{cat,\ eff}} = k_{\mathrm{cat,\ MIOX}}\frac{1 + a_{\mathrm{MIOX}}\mathrm{MI}}{k_{\mathrm{a,\ MIOX,\ MI}} + \mathrm{MI}}, \tag{3.10}$$

The function $u(x, k, \theta)$ describes the genetic control topology at the enzyme's promoter. There are three options for this functional form: activation, repression, and no control as in the toy model (see Equation 3.4).

Similarly to the toy pathway, there are four continuous parameters: k_{ino1}, θ_{ino1}, k_{MIOX}, θ_{MIOX}. The four decision variables are constrained on the following ranges:

$$1 \cdot 10^{-7} \leq \theta_{\mathrm{ino1}}, \theta_{\mathrm{MIOX}} \leq 10, \tag{3.11}$$

$$1 \cdot 10^{-7} \leq k_{\mathrm{ino1}}, k_{\mathrm{MIOX}} \leq 5. \tag{3.12}$$

The possible architectures are limited to those only containing negative feedback loops. Similarly to the toy model, I term these three architectures upstream

Parameter	Value	Units	Parameter	Value	Units
λ	2.77E-5	1/s	n_{Pfk}	3	N/A
V_{in}	0.1656	mM/s	$k_{\text{cat, ino1}}$	0.2616	mM/s
$k_{\text{cat, pgi}}$	0.8751	mM/s	$k_{\text{m, ino1, g6p}}$	1.18	mM
$k_{\text{eq, pgi}}$	0.3	mM	$V_{\text{m, t, MI}}$	0.045	mM/s
$k_{\text{m, pgi, g6p}}$	0.28	mM	$k_{\text{m, t, MI}}$	15	mM
$k_{\text{m, pgi, f6p}}$	0.147	mM	$k_{\text{cat, MIOX}}$	0.2201	mM/s
$k_{\text{cat, zwf}}$	0.0853	mM/s	$k_{\text{m, MIOX}}$	24.7	mM
$k_{\text{m, zwf, g6p}}$	0.1	mM	a_{MIOX}	5.422	N/A
$k_{\text{cat, pfk}}$	2.615	mM/s	$k_{\text{a, MIOX, MI}}$	20	mM
$k_{\text{m, pfk, f6p}}$	0.16	mM	α_1	10E-5	N/A
			α_2	10E-3	N/A

Table 3.3: Kinetic parameters of glucaric acid model.

repression, downstream activation, and dual control based on their mode of action (see Figure 3.7). All kinetic parameters are given in Table 3.3.

The α_1 and α_2 values in the objective function were chosen to scale the two objectives to be similar in magnitude and the overall loss to be between 0 and 1 (see Table 3.3). The loss equation for the objective function is

$$J = \alpha_1 \underbrace{\int_0^T \left| V_{\text{in}} - \text{MIOX}(t) \frac{k_{\text{cat, MIOX}} \text{MI}(t)}{k_{\text{m, MIOX}} + \text{MI}(t)} \right| \mathrm{d}t}_{\text{production loss}} \tag{3.13}$$

$$+ \alpha_2 \underbrace{\int_0^T \left(u(\text{MI}(t), k_{\text{ino1}}, \theta_{\text{ino1}}) + u(\text{MI}(t), k_{\text{MIOX}}, \theta_{\text{MIOX}}) \right) \mathrm{d}t}_{\text{pathway cost}} . \tag{3.14}$$

A summary of the model details is given in Table 3.4. I ran each simulation of the model to a final time of $5 \cdot 10^5$ seconds to ensure the final values would be at steady state. All metabolites and enzymes had an initial concentration of 0μM except for g6p and f6p, which had initial concentrations of 0.281mM and 0.0605mM, respectively. These values were obtained by running the model with the engineered pathway removed and taking the steady state values of the unmodified pathway metabolites as a initial condition.

Pathway Product	Glucaric Acid
Decision Variables	$k_{\text{ino1}}, k_{\text{MIOX}}, \theta_{\text{ino1}}, \theta_{\text{MIOX}}$
Architectures	Open Loop, Upstream Repression, Downstream Activation, Dual Control
Pathway Metabolites	g6p, f6p, MI
Pathway Enzymes	Pgi (constant), Zwf (constant), Pfk (constant), Ino1, MIOX
Integration Time	$5 \cdot 10^5$s
Initial Conditions	f6p(0) = 0.281mM g6p(0) = 0.0605mM MI(0) = 0mM Ino1(0) = 0mM MIOX(0) = 0mM

Table 3.4: Glucaric acid model summary. The names for the decision variables and pathway metabolites and enzymes are included, as are the initial condition values. All model values come from Verma et al., 2021.

Representative BayesOpt results

The results in Figure 3.8 show a typical run of the optimizer when using the cost-benefit objective in Eq. (3.2), together with the fraction of samples in which the algorithm explored each control architecture across the successive iterations. The optimal architecture (dual control in this case) was found in under 200 iterations and the algorithm was able to further decrease the value of the objective function by exploring the space of dose-response parameters of IpsA. I observe that as the iterations progress, the algorithm shows a remarkable ability to explore other architectures despite their larger objective function values. The first quarter of the run had the most exploration of architectures other than dual control, with 38.6% of samples coming from non-majority architectures. This percentage steadily decreased over the iterations but did not drop below 20%, illustrating the global nature of the optimization routine.

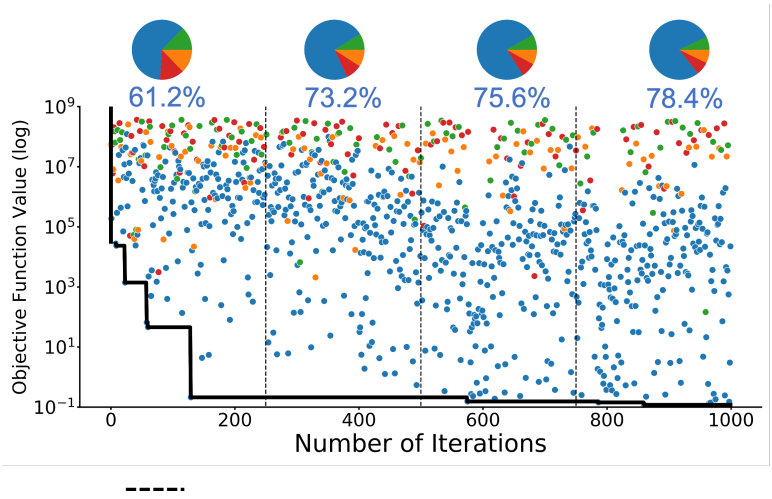


Figure 3.8: Sample run of the BayesOpt algorithm for 1,000 iterations of the loop in Figure 3.2. Black line shows the descent on the value of the objective function. Dots show all samples colored by architecture; pie charts show the fraction of architectures explored by the algorithm, and the fraction of samples taken from the majority architecture (dual control).

Perturbation of kinetic parameters

To explore the impact of uncertain enzyme kinetics, I perturbed the parameters of the rate-limiting MIOX allosteric reaction:

$$\begin{aligned} V_{\text{MIOX}} &= \frac{V_{\text{m, eff}}\text{MI}}{k_{\text{m, MIOX}} + \text{MI}}, \\ \text{given } V_{\text{m, eff}} &= V_{\text{m, MIOX}} \frac{1 + a_{\text{MIOX}}\text{MI}}{k_{\text{a, MIOX}} + \text{MI}}, \end{aligned} \tag{3.15}$$

where $V_{\text{m, MIOX}}$ is the maximum rate of reaction, $k_{\text{m, MIOX}}$ is the Michaelis-Menten constant, and $k_{\text{a, MIOX}}$ and a_{MIOX} are allosteric activation constants. The kinetic parameters were perturbed using Latin Hypercube sampling (Loh, 1996) on the range (-100%, +100%) of the nominal values. I solved the optimization problem for 1000 combinations of these three parameters, which took under 16 hours on a Macbook Air with Apple M1 processor and 8GB of RAM running MacOS Monterey. Parameter value samples which resulted in the ODE solver erroring out due to high model stiffness were removed.

Perturbing the kinetic parameters of the glucaric acid pathway did not significantly affect the minimum objective function value achieved, indicating that the optimum is robust to uncertainty in the kinetic parameters (3.9A). I observed

no statistically significant difference between between background and perturbed objective function values (2-way t-test, p=0.69). Only one of the $N = 100$ runs for perturbed parameters failed to converge the optimum. I found that the dual control architecture was chosen as optimal in more than 85% of samples (Figure 3.9B). I thus sought to examine the optimal dose-response parameters of this architecture in more detail.

The maximal enzyme expression rates (k) and regulatory thresholds (θ) control the shape of the dose-response curves. As shown in Figure 3.10, I found that the upstream repressive loop and downstream activatory loop had different optimal dose-response curves, corresponding to different optimal values of the continuous parameters. Optimal values of the upstream repression threshold θ_1 are low (mean value 0.64) and compressed into a narrow range as compared to the larger standard deviation of the downstream repression threshold θ_2 (mean value 7.24). This is reflected on a larger variation in the shape of the dose response curve for the downstream loop. Experimental fine-tuning of a dual control circuit might target parameters with optimal values with a wide range, such as k_1, as varying these parameters is less likely to impair circuit function. Overall, these results show the robustness of the glucaric acid dual control system to kinetic parameter uncertainty and demonstrate the possibilities enabled by the speed of BayesOpt. I next demonstrate the flexibility of the method with multiple objective functions to optimize diverse circuit goals.

3.3.2 Exploration of alternative objective functions

In the previous case studies I employed a cost-benefit objective designed to account for the tradeoff between heterologous production and the cost of expressing pathway enzymes, as in Eq. 3.2. To demonstrate the flexibility of the method with other objective functions, here I consider the optimization of the temporal trajectories of pathway metabolites. I focused on the joint optimization of the rise time and overshoot in a model of a fatty acid production pathway considered previously in the literature (Liu and Zhang, 2018).

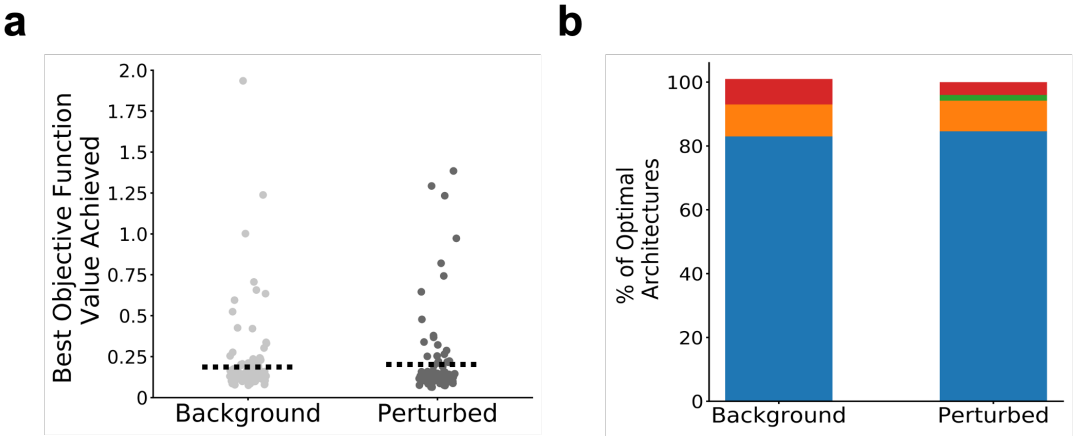


Figure 3.9: Perturbed kinetic parameter simulations. **(a)** Strip plot of best objective function values achieved for perturbed and nonperturbed simulations (N=100 per condition). Dashed line denotes the mean value of the objective function. **(b)** Bar chart of optimal architectures selected for perturbed and nonperturbed conditions.

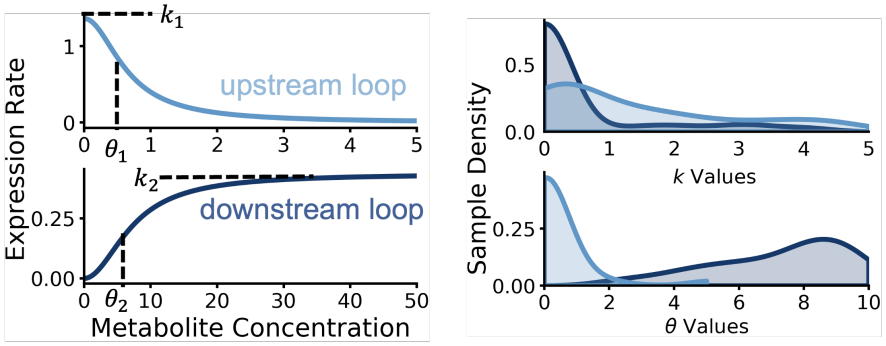


Figure 3.10: Average dose-response curves and distribution of optimal parameters for the dual control architecture with perturbed allosteric parameters. The parameter k_i and θ_i determine the maximal enzyme expression rate and regulatory threshold, respectively.

Fatty acid model

Fatty acids are one of the four macromolecule types necessary for life. In addition to being used to form cell membranes, they are important sources of energy. Furthermore, hydrocarbons derived from fatty acids have attracted attention as a potential biofuel source (Xu et al., 2014; Zhang et al., 2018).

Previous work engineering metabolic and genetic control loops showed that negative feedback control could speed up the rise to steady state conditions (Liu and Zhang, 2018). The engineered fatty acid biosynthetic pathway shown in Figure 3.11 was created by expressing a thioesterase under transcriptional control, shown as the negative metabolic loop (NML) architecture in Figure 3.11. In addition to transcription-factor mediated negative feedback loops, this model also includes individually implemented direct genetic loops where a repressor is expressed on the same promoter as the enzyme. There are thus two different types of feedback loops which interface with different levels of cellular organization. I explore several control architectures previously proposed in the literature (Figure 3.11).

The fatty acid model has multiple different architectures which model different molecular components of the pathway. Due to this model complexity, I consider each architecture in detail.

Open loop architecture. The mass balance equations for the open loop architecture are

$$\begin{aligned} \frac{\mathrm{dFFA}}{\mathrm{d}t} &= k_{\mathrm{tesA}} \cdot \mathrm{tesA} - \lambda \cdot \mathrm{FFA}, \\ \frac{\mathrm{dtesA}}{\mathrm{d}t} &= r_{\mathrm{lac}} - \lambda \cdot \mathrm{tesA}, \end{aligned} \tag{3.16}$$

where FFA is the concentration of free fatty acids and tesA is the concentration of thioesterase enzyme (tesA). The cellular growth rate λ is set for all architectures at $3.85 \cdot 10^{-4}$ mM/s, which is calculated from an *E. coli* doubling time of 26 minutes. The parameter k_{tesA} is the kinetic parameter of the thioesterase reaction. There is one free optimisable parameter in this model, the tesA promoter constant r_{lac}.

Negative gene loop architecture. The negative gene loop architecture expresses a transcriptional repressor on the same promoter as tesA, creating a negative feedback loop on enzyme production. The mass balance equations for this architecture are

$$\begin{aligned}\frac{\mathrm{dFFA}}{\mathrm{d}t} &= k_{\mathrm{tesA}} \cdot \mathrm{tesA} - \lambda \cdot \mathrm{FFA},\\ \frac{\mathrm{dtesA}}{\mathrm{d}t} &= \frac{r_{\mathrm{tl}}\mathrm{tetR}^2}{k_{\mathrm{d,\ tetR}}^2 + \mathrm{tetR}^2} - \lambda \cdot \mathrm{tesA},\\ \frac{\mathrm{dtetR}}{\mathrm{d}t} &= \frac{r_{\mathrm{tl,\ tetR}}\mathrm{tetR}^2}{k_{\mathrm{d,\ tetR}}^2 + \mathrm{tetR}^2} - \lambda \cdot \mathrm{tetR}.\end{aligned} \tag{3.17}$$

The variable tetR is the concentration of the repressor expressed on the same gene as tesA. While the other models allow multiple modes of transcriptional control (activation, repression, no control), this model only allows for activation due to the type of transcription factor used (Liu and Zhang, 2018). The relative expression strengths r_{tl} and $r_{\mathrm{tl,\ tetR}}$ are free parameters.

Negative metabolic loop architecture. The negative metabolic loop architecture includes a transcription factor which acts on the tesA promoter. The mass balance equations for this architecture are

$$\begin{aligned}\frac{\mathrm{dFFA}}{\mathrm{d}t} &= k_{\mathrm{tesA}} \cdot \mathrm{tesA} - \lambda \cdot \mathrm{FFA},\\ \frac{\mathrm{dtesA}}{\mathrm{d}t} &= \frac{r_{\mathrm{fl}'}\mathrm{tetR}^2}{k_{\mathrm{i}}^2} - \lambda \cdot \mathrm{tesA},\end{aligned} \tag{3.18}$$

where the free parameters of this model are k_i and $r_{\mathrm{fl}'}$, which control the shape of the transcription factor dose-response curve. Unlike the other free model parameters, k_i varies on the range 0 to 0.12.

Layered negative feedback loop architecture. Finally, the layered negative feedback loop uses tetR as an intermediate repressor. The transcription factor is repressed by the product and in turn represses the repressor tetR, which can repress the expression of the enzyme tesA. This negative feedback loop is

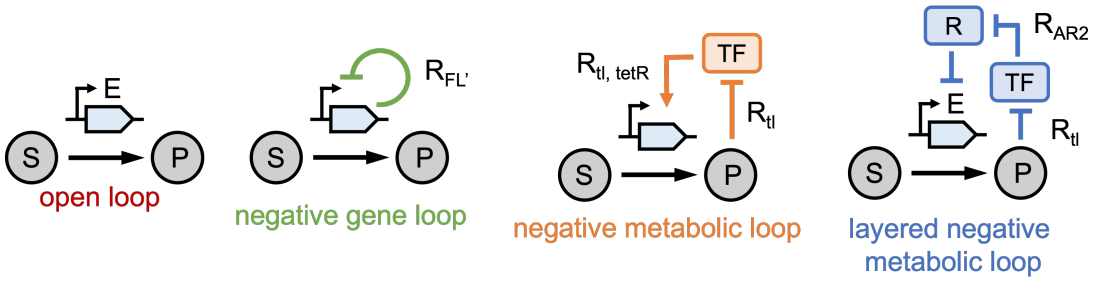


Figure 3.11: Fatty acid pathway diagram with various control architectures implemented in *Escherichia coli* (Liu and Zhang, 2018). Figure adapted from (Merzbacher, 2022).

described by the following equations:

$$\begin{aligned}
\frac{\mathrm{dFFA}}{\mathrm{d}t} &= k_{\mathrm{tesA}} \cdot \mathrm{tesA} - \lambda \cdot \mathrm{FFA}, \\
\frac{\mathrm{dtesA}}{\mathrm{d}t} &= \frac{r_{\mathrm{tl}} \mathrm{tetR}^2}{k^2_{\mathrm{d,\ tetR}}} - \lambda \cdot \mathrm{tesA}, \\
\frac{\mathrm{dtetR}}{\mathrm{d}t} &= \frac{r_{\mathrm{ar2}} k^2_{\mathrm{ar2}}}{(1 + \frac{\mathrm{FFA}}{k_{\mathrm{d,\ FadR,\ FFA}}})^2} - \lambda \cdot \mathrm{tetR}.
\end{aligned} \tag{3.19}$$

The repressor tetR is produced on the AR2 promoter, which has its own strength parameter r_{ar2}. The free parameters on this model are r_{ar2} and r_{tl}.

Unless otherwise stated, all continuous model parameters are restricted to the range from $10 \cdot 10^{-11}$ to $10 \cdot 10^{-8}$. I ran each simulation of the model to $5 \cdot 10^4$s to ensure steady state values had been reached. All pathway components started with a concentration of 0mM. The kintetic parameters are given in 3.5). Table 3.6 summarizes the model details.

Alternative objective functions

Two objective functions were implemented for this pathway: a production-burden function and a speed-accuracy function. I first considered a similar objective function as in Eq. (3.2) so as to compare convergence against the previous case studies. The loss equation for the production-burden objective function is the sum of the production loss and pathway cost:

$$J = \alpha_1 J_{\mathrm{prod}} + \alpha_2 J_{\mathrm{cost}}, \tag{3.20}$$

Architecture	Parameter	Definition	Value	Units
All	λ	growth rate	3.85 $\cdot 10^{-4}$	mM/s
OL	k_{tesA}	tesA catalysis constant	100	mM/s
NGL	k_{tesA}	tesA catalysis constant	105.25	mM/s
NGL	$k_{\text{d, tetR}}$	tetR dissociation constant	$3.0 \cdot 10^{-8}$	mM/s
NML	k_{tesA}	tesA catalysis constant	77.75	mM/s
LNML	k_{tesA}	tesA catalysis constant	230.9	mM/s
LNML	$k_{\text{d, tetR}}$	tetR dissociation constant	$3.85 \cdot 10^{-8}$	mM/s
LNML	$k_{\text{d, FadR, FFA}}$	tetR and FFA dissociation constant	0.001	mM/s
LNML	k_{ar2}	tetR promoter constant	138.50	mM/s
all	α_1	production loss scaling constant	10E-4	mM/s
all	α_2	pathway cost scaling constant	10E-3	mM/s

Table 3.5: Kinetic parameters of fatty acid model.

Pathway Product	Fatty Acid
Decision Variables	$r_{\text{lac}}, r_{\text{bad}}, r_{\text{tl}}, r_{\text{tl, tetR}}, r_{\text{fl'}}, r_{\text{ar2}}$
Architectures	Open Loop, Negative Gene Loop, Negative Metabolic Loop, Layered Negative Metabolic Loop
Pathway Metabolites	FFA, tetR (repressor)
Pathway Enzymes	tesA
Integration Time	$5 \cdot 10^4$s
Initial Conditions	0mM for all metabolites and enzymes

Table 3.6: Fatty acid model summary. Some metabolites and enzymes (tetR) are only modeled for relevant architectures like the negative metabolic loop.

where J_{prod} is the production loss. The α_1 and α_2 values in the objective function were chosen to scale the two objectives to be similar in magnitude and the overall loss to be between 0 and 1 (see Table 3.3). The production loss is defined as

$$J_{\text{prod}} = \frac{1}{\int_0^T |\text{tesA}(t) k_{\text{tesA}}| \, \mathrm{d}t}. \tag{3.21}$$

The pathway cost J_{cost} varies for each architecture:

$$J_{\text{cost}} = \begin{cases} \int_0^T (r_{\text{lac}}) \, \mathrm{d}t & \text{(open loop)}, \\ \int_0^T \left(\frac{r_{\text{tl}} \text{tetR}^2}{k_{\text{d, tetR}}^2 + \text{tetR}^2} + \frac{r_{\text{tl, tetR}} \text{tetR}^2}{k_{\text{d, tetR}}^2 + \text{tetR}^2}\right) \mathrm{d}t & \text{(negative gene loop)}, \\ \int_0^T \left(\frac{r_{\text{fl}}, \text{tetR}^2}{k_{\text{i}}^2}\right) \mathrm{d}t & \text{(negative metabolic loop)}, \\ \int_0^T \left(\frac{r_{\text{tl}} \text{tetR}^2}{k_{\text{d, tetR}}^2} + \frac{r_{\text{ar2}} k_{\text{ar2}}^2}{(1 + \frac{\text{FFA}}{k_{\text{d, FadR, FFA}}})^2}\right) \mathrm{d}t & \text{(negative gene loop)}. \end{cases} \tag{3.22}$$

While the exact equations vary, all of these objectives penalize expression of heterologous enzymes to capture burdens on native metabolism. A representative optimization run for this objective (Figure 3.12) shows that the negative gene loop (NGL, green) and negative metabolic loop (NML, orange) architectures perform, on average, better than the other two architectures. BayesOpt samples taken from the open-loop architecture were, on average, two orders of magnitude worse than samples taken from NML and NGL architectures. Despite such hierarchy of loss values across the four architectures, the method effectively explores all architectures throughout the optimization run.

I next considered the optimization of percent overshoot and rise time presented in the literature (Liu and Zhang, 2018). The second speed-accuracy objective function has two terms, percent overshoot and normalized rise time:

$$J = \alpha J_{\text{rt}} + J_{\text{os}}. \tag{3.23}$$

The percent overshoot objective, J_{os}, measures the maximal deviation of product from its steady state concentration and is defined as the percent difference

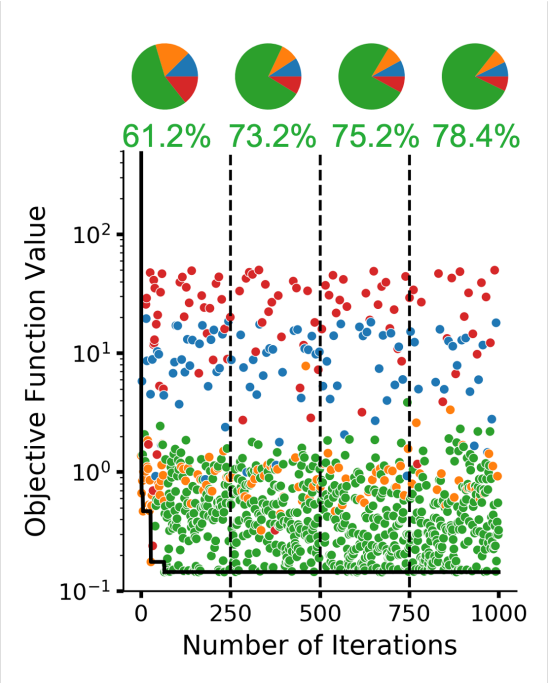


Figure 3.12: Representative run of BayesOpt on fatty acid model with cost-benefit objective showing the best objective function value (black line).

between the maximum fatty acid concentration and the steady state fatty acid concentration. The rise time, J_{rt}, is a measure of how fast fatty acid production rises to steady state and is defined as the first time point where the fatty acid concentration reaches 50% of the steady state value, normalized by the total integration time. I minimized the sum of the overshoot and rise time with a scaling weight α:

$$J = \alpha J_{\text{rt}} + J_{\text{os}}. \tag{3.24}$$

The percent overshoot J_{os} is the percent difference between the maximum FFA concentration and the steady state FFA concentration at the end of the integration time. For example, a 10% overshoot would correspond to a maximum FFA concentration 10% over the steady state concentration. The rise time is the first time point where the FFA concentration reaches 50% of the steady state value, normalized by the total integration time. As a result, a 20% rise time corresponds to the FFA concentration reaching 50% of the steady state value one-fifth of the way through the total integration time.

The objective weight α was swept from $\alpha = 0.01$ to $\alpha = 10,000$ and BayesOpt was run for 100 iterations at each α value. The optimal parameter values were used to compute the rise time and overshoot for visualization. Higher values of α correspond to optimal circuits with low rise times, while lower values of α prioritize circuits with low overshoot. Rise time is a measure of circuit speed, while overshoot is a measure of circuit accuracy. I found that when α is varied across several orders of magnitude, the optimal circuits form a optimal tradeoff curve (Figure 3.13). Different architectures occupy different parts of the optimal tradeoff curve and display markedly different dynamics. The NML optima occupies a single point in the loss space, indicating that multiple continuous parameter values give the same loss function value for multiple values of α. The NML also has the lowest absolute loss function value of all the architectures considered. The NGL and layered negative metabolic loop (LNML) architectures occupy larger ranges on the curve, with NGL giving a low overshoot and LNML a low rise time. After further analysis, I discovered that the open loop architecture does not have the free parameter flexibility to produce more than a single overshoot and rise time value, and thus is never found to be optimal regardless of α value. The product production trajectories on the Figure 3.13 inset show the different curves achieved with various α values. The optimal NML circuit has no overshoot but the slowest rise time, while the LNML has a very rapid rise time but overshoots the steady state value by more than double. These opposing tradeoffs demonstrate the importance of balancing multiple circuit design objectives.

3.3.3 Scalability of models to large and complex pathways

My previous case studies have been limited to circuits with a single metabolite controlling gene expression and a relatively small number of control architectures. In addition, the models contain between 3 and 5 differential equations modelling a select few pathway metabolites and enzymes. I now study a large model for the synthesis of p-aminostyrene (p-AS), an industrially relevant vinyl aromatic monomer with applications in photonics and biomedicine, in *E. coli* (Figure 3.14) (Goikhman et al., 2011) using a cost-benefit objective similar to Eq. (3.2) tailored

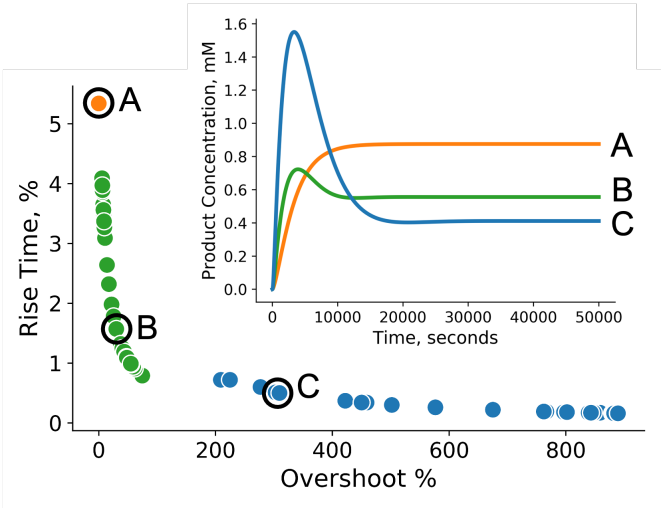


Figure 3.13: Optimal tradeoff curve between overshoot and rise time. The inset shows three sample trajectories illustrating how different optimal architectures navigate the tradeoff between overshoot and rise time.

to the specific pathway.

The cytotoxic intermediate p-amminocinnamic acid (p-ACA) makes p-AS challenging to produce in microbes. Furthermore, another intermediate, p-aminophenylalanine (p-AF), leaks from cells (Carothers et al., 2011). L-amino acid oxidase, one of the enzymes in the pathway (see 3.14) depletes key aromatic amino acid metabolites and creates toxic hydrogen peroxide as a byproduct. Finally, overexpression of the efflux pump used to remove p-ACA from the cell also causes cytotoxicity. In addition to these challenges, the pathway has two possible loci of genetic control. The first three enzymes in the pathway, papA, papB, and papC, are all expressed on the papABC operon, which produces a single mRNA transcript. Operons are common in bacteria, so including one in my pathway models demonstrates an important application of this method.

This model has two possible metabolites that can regulate gene expression, namely p-amminocinnamic acid (p-ACA) and p-aminophenylalanine (p-AF), both of which can act as ligands for aptazyme-regulated expression device (aRED) transcription factors (Ellington and Szostak, 1990), and three genes to be controlled. The aRED transcription factors can also act as dual regulators (activators or repressors) on any of the three promoters involved in the pathway. For simplicity, I limit the design space to control architectures without positive feed-

back loops, as these are prone to bistability (Oyarzún and Chaves, 2015). This results in 27 possible control architectures and 19 continuous parameters to be optimized. The model also has a number of additional complexities. It contains operon-based gene expression commonly found in bacterial systems (genes papA, papB, and papC are expressed on the papABC operon), it includes a detailed description of mRNA dynamics and protein folding, which results in a large model with 23 differential equations, and it can also display oscillatory dynamics. Previous models abstracted translation as a direct function of transcription and did not explicitly separate mRNA and promoter concentrations. The p-AS model has separate equations for each promoter, mRNA, and folded and unfolded proteins, enabling an even higher level of scale specificity.

In addition to expression of heterologous enzymes, the accumulation of toxic intermediates is another major source of genetic burden to host organisms. The p-AS model has several sources of toxicity present in the pathway (Goikhman et al., 2011; Stevens and Carothers, 2015). The intermediate p-ACA and the efflux pump used to remove p-ACA from cells are both cytotoxic, while another intermediate, p-AF, leaks from cells. The pathway enzyme L-amino acid oxidase (LAAO) depletes key aromatic amino acid metabolites and creates toxic hydrogen peroxide as a byproduct. The model incorporates these various types of toxicity in the form of a toxicity factor tau. This toxicity factor is of the form

$$\tau = \frac{k_i}{k_i + \frac{\mathrm{pACA}}{t_a} + \frac{\mathrm{P_{efflux}}}{t_p} + \frac{\mathrm{LAAO}}{t_l}}, \tag{3.25}$$

where t_l, t_a, and t_p are chemical-specific toxicity factors. Enzyme-induced toxicity t_l scales the key metabolite depletion rate driven by the enzyme LAAO. Metabolite-induced toxicity t_a scales the impact of toxic intermediate p-ACA concentration. Finally, protein-induced toxicity t_p reflects the toxicity caused by efflux pump expression. The toxicity factor acts as a scaling coefficient on the pathway synthesis, degradation, and folding reaction rates. An alternate approach would be to incorporate a toxicity term as a cost in the objective function to penalize systems with high intermediate toxicity. Due to the large number of

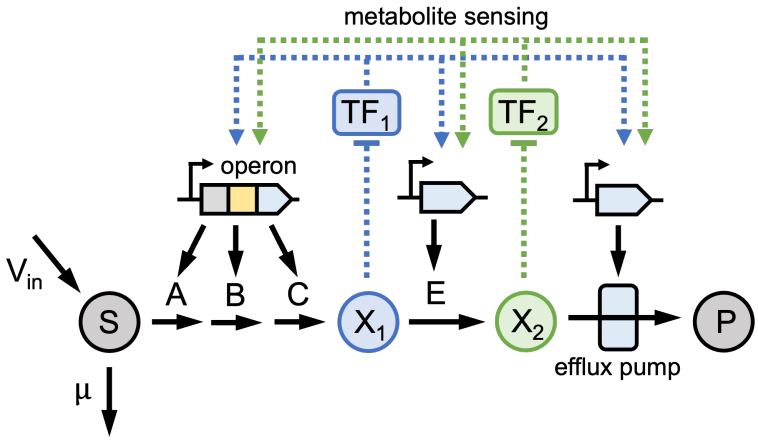


Figure 3.14: Schematic of pathway for production of p-aminostyrene (Stevens and Carothers, 2015). Figure adapted from (Merzbacher, 2022).

equations, I include the full description of the ODE model in Appendix A.

Despite the complexity and size of the p-AS model, I observe that BayesOpt explores many of the 27 possible architectures and converges to a low value of the objective function (Figure 3.15); this was also achieved at a reasonable computational cost (mean run time under two minutes). The best architecture selected in the sample run was a double upstream repression, single downstream activation loop controlled by p-AF (Figure 3.15, inset), but there is no clear best architecture when the optimization is run many times. No architecture is optimal for more than 15% of test runs, demonstrating that there are combinations of architectures and parameter values that achieve a similar optimal loss. I also found that several architectures can display oscillatory solutions, which I chose to exclude from the search by applying a peak detection algorithm (Virtanen et al., 2020) and adding a large regularization term to the loss.

To investigate the robustness to chemical toxicity, I perturbed the metabolite-induced toxicity t_a and protein-induced toxicity t_p in Eq. (3.25). Metabolite-induced toxicity was perturbed on the nominal range $(1{\cdot}10^{-3}, 1{\cdot}10^{-4})$ and protein-induced toxicity on the range $(1 \cdot 10^{-4}, 1 \cdot 10^{1})$ respectively. Both ranges were selected to match the ranges provided in the literature (Stevens and Carothers, 2015). Latin Hypercube sampling was used to generate $N = 100$ perturbed parameter values, and the optimal solutions were compared to an equal number of

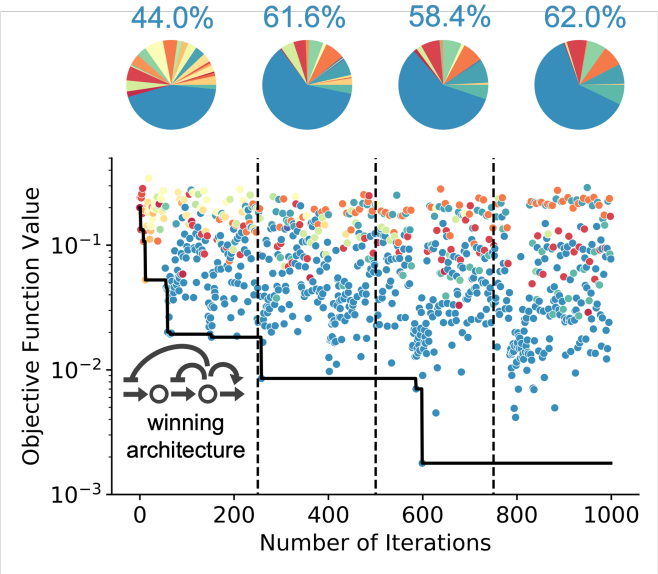


Figure 3.15: Representative run of the BayesOpt algorithm for the p-aminostyrene pathway.

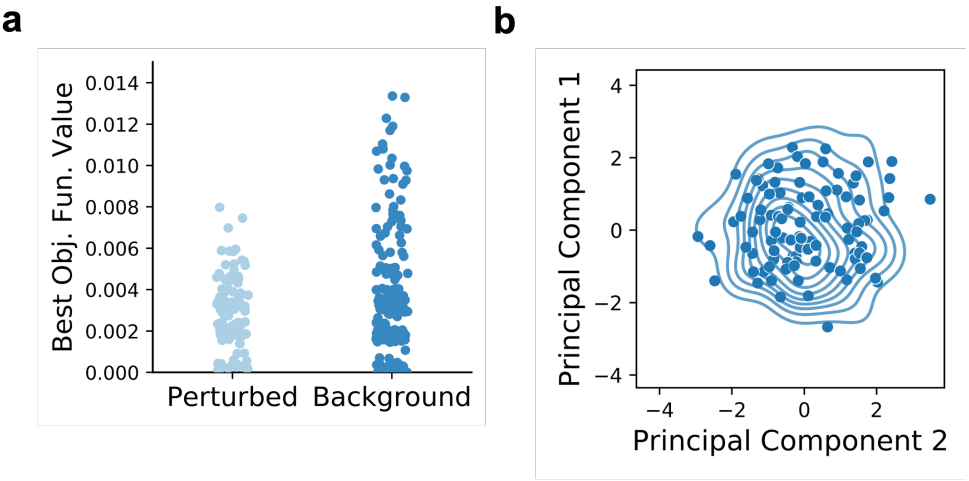


Figure 3.16: P-AS pathway under toxicity factor perturbation. **(a)** Optimal objective function values for perturbed and nonperturbed conditions (N=100). **(b)** Scatter plot of principal components of the optimal parameter values for the model with perturbed toxicity parameters. The first PC explains 9.5% of total variance, the second PC explains 8.9% of total variance.

background solutions using the nominal parameter values. The optimal loss values were found to be comparable between perturbed and background systems (Figure 3.16). Additionally, when projected onto a 2-dimensional space using principal component analysis, the distribution of background parameter values was similar to the distribution of perturbed solutions, indicating that the perturbation did not significantly affect the optimal parameters selected (3.16) (Abdi and Williams, 2010). The p-AS pathway lies at the maximum size of currently experimentally tractable systems and thus illustrates the broad applicability of BayesOpt to realistic design tasks in metabolic engineering.

3.4 Discussion

Progress in synthetic biology has allowed for the construction of circuits on increased complexity which act across various levels of biological organization. However, large design spaces and multiple scales of biological organization can become substantial challenges for the rapid design of functional systems. Machine learning has already proved useful in a range of metabolic engineering tasks (Radivojević et al., 2020) and is gaining substantial interest in other areas of synthetic biology (Carbonell et al., 2019; Nikolados et al., 2022). In this chapter I presented the use of Bayesian optimization for the design of biological circuits. The method can rapidly find combinations of circuit architectures and parameters to optimize a performance objective that captures the target circuit functionality.

The method is particularly well suited for cases in which the multiple scales prevent efficient simulation of ODE models. Gene circuits designed to control metabolic pathways are an excellent example of such multiscale systems, as they combine fast metabolic timescales with the much slower dynamics of gene expression. Moreover, the choice of regulators, control points, and control architectures adds multiple degrees of freedom that are infeasible to explore experimentally. Previously implemented metabolic control systems have been built primarily based on application-specific knowledge of pathway features (Liu et al., 2018; Ni et al., 2021). I have shown that Bayesian optimization can aid the design

of such systems prior to implementation and serve as tools for *in silico* screening of competing designs that may have similar performance but entail different cost of wet lab implementation. I showed the efficiency and scalability of the method in several real-world case studies from metabolic engineering. In particular, the p-aminostyrene pathway is more complex than systems typically implemented in literature so far, which suggests that the method is applicable across a range of real-world design tasks.

The inherent nonconvexity of dual optimization presents a challenge to traditional gradient-based methods. Bayesian optimization has a substantial advantage here in that it can jump around the design space freely, with no implied spatial similarity between subsequent samples. Instead, the acquisition function scores samples across the design space and can select one from any area which maximizes its value, regardless of where the previous sample was located. I chose to use Expected Improvement (EI) as the acquisition function, as it is the most popular choice and automatically captures the balance between exploration of undersampled areas of the design space and exploitation towards the global minima. However, there are several alternative options that are also commonly used, including entropy search and the upper confidence bound (Frazier, 2018).

Another advantage of my method is its flexibility in implementation. Architectural representations can be encoded using a mixed-integer approach or structured as separate systems of equations. In the fatty acid model, each architecture is individually implemented as a separate Python function, enabling the precise modelling of various pathway components. In contrast, the other three models utilize a matrix-based approach, where the architecture is encoded as a one-hot feature vector at each potential control point. This approach facilitates the automatic generation of all possible architectures, making it particularly advantageous for complex systems with numerous potential configurations, such as the p-AS pathway.

One consistent feature of the performance landscape I observed across systems was the presence of flat regions in the objective function space near the global optimum. Achieving similarly low loss values often corresponded to optimal param-

eter values spanning broad ranges for some parameters, while remaining tightly constrained for others. This phenomenon reflects the concept of sloppiness, a well-documented property in complex biological models, where certain parameter combinations exhibit wide tolerances without significantly affecting system behavior, while others are highly sensitive (Gutenkunst et al., 2007; Transtrum et al., 2015).

Sloppiness manifests as an elongated, shallow trough in the objective function landscape, where the loss minimum is broad in some dimensions and sharply curved in others (Waterfall et al., 2006). This characteristic suggests that many parameter values are not uniquely identifiable but instead form an ensemble of near-optimal solutions, a property observed in models of gene regulation, metabolic networks, and signalling pathways (Daniels et al., 2008). Such behaviour can arise due to inherent redundancies in biological systems, where certain parameter variations are buffered by compensatory mechanisms, maintaining functional robustness despite variations in underlying kinetics (Machta et al., 2013).

Gradient-based optimization methods are powerful when the objective function is smooth, differentiable, and inexpensive to evaluate — conditions that are rarely met in the design of biological systems. In contrast, the design landscapes encountered in synthetic and systems biology are typically nonconvex, noisy, and discontinuous, with gradients that are either unavailable or unreliable due to stochastic simulations and measurement noise. Under these circumstances, gradient-based methods can easily become trapped in local optima or fail to converge. Bayesian optimization provides a more suitable alternative: it models uncertainty in the objective explicitly, balances exploration and exploitation through its acquisition function, and can operate effectively with sparse, expensive evaluations. This probabilistic treatment of uncertainty makes it especially appropriate for guiding experiments or simulations where each evaluation represents a costly biological design iteration.

3.4.1 Follow up projects

This work was completed in the first two years of my programme and, following its publication, I began supervising student projects which extended the work to answer new scientific questions. First, I designed a project which was undertaken by a BSc student in Biotechnology, Nicholas Goguen-Compagnoni. Nicholas' project focused on the p-aminostyrene model, which he reimplemented in Julia to improve its performance. The scale and complexity of this model present a challenge for the design of an appropriate objective function that differentiates the many possible optimal architectures. One approach Nicholas implemented was to add a complexity regularization term to the objective function to push the optimization to preferentially select architectures with fewer regulatory loops or transcription factor ligands. The second approach was to add terms to the objective function which penalized samples which had very high total proteomic mass or protein production flux. These objective function constraints represent a first attempt at incorporating constraints on ribosomal machinery and cellular energy resources into the optimization. The modularity of different objective function components is a clear advantage of this approach and can easily be applied to design circuits with more specific temporal dynamics or target goals.

An additional follow-on project was undertaken by Nicola Hallmann, a visiting MSc student from ETH Zurich. The goal of this project was to apply Bayesian optimization to the selection of culture medium components. The medium composition in a bioreactor can significantly affect growth and contains many components, some of which can be expensive. The salts, sugars, serums, and buffers present can range in optimal values and medium selection often requires multiple costly experimental iterations. Nicola used genome-scale models and implemented a Bayesian optimization loop using BoTorch (Balandat et al., 2020), a more developed package which allows for parallelization and alternative acquisition functions (further discussion on improvements to BayesOpt in Section 3.5). She implemented an optimization balancing growth rate cost, and heterologous product production and successfully produced Pareto curves visualizing tradeoffs between the three objective components.

Both of these projects demonstrate the broad applicability of this work to similar biological design problems and as a toolkit for exploring high-dimensional nonconvex design spaces. I also anticipate several other novel applications of this work to other problem areas where discovery or tuning of multiscale circuits has been previously infeasible. For instance, this method could be employed to fit temporal circuit dynamics to data or discern which of several discrete circuit mechanisms most closely matches observed behaviour. Hiscock, 2019 fit the temporal trajectories if single-scale gene circuits to desired behaviour (e.g. bistability, ultrasensitivity, or oscillation). Furthermore, the ODE models could be augmented with Thompson sampling as in Kobiela et al., 2024 to better capture the parameter uncertainty and stochasticity inherent in biological systems.

3.5 Limitations and future work

While the BayesOpt method is flexible, fast, and scalable it still has several limitations and room for future improvement. Unlike exhaustive search methods, Bayesian optimization does not guarantee convergence to the global minimum in practical settings with finite evaluations or noisy objectives. However, formal convergence results exist under certain regularity assumptions—for example, when the objective is continuous, noise-free, and drawn from a Gaussian process prior, acquisition functions such as Expected Improvement and GP-UCB have been shown to converge asymptotically to the global optimum (Garnett, 2023). In practice, however, biological design landscapes often violate these assumptions, and convergence is therefore heuristic rather than guaranteed.

Furthermore, depending on the internal model BayesOpt fits (in this case a Tree of Parzen Estimators model), the number of dimensions BayesOpt can optimize is limited to around 20 (Bagge Carlson, 2018).

While the Hyperopt package worked well for this project without significant fine-tuning, there are a large number of ways the optimization algorithm could be improved. Future optimization strategies could incorporate secondary local optimization methods that exploit these broad optima. One promising approach is a

two-step optimization framework: following Bayesian optimization, a secondary fine-tuning step, such as gradient-based refinement via PyTorch autograd, could further explore the local objective function landscape (Hiscock, 2019). By refining solutions within the flat trough of the loss surface, this approach could identify smaller, high-precision regions of low objective function values, improving parameter specificity and model interpretability.

Other avenues for improving global optimization methods include acquisition function selection, batch optimization, and the integration of more flexible objective function models. The TPE nonparametric function chosen for BayesOpt was shown to be relatively insensitive to hyperparameter tuning (see Figure 3.6). While this makes the method easy to use out of the box, it limits the tunability compared to alternative models. The Hyperopt package implements a random sampling modelling method (Bergstra et al., 2013). Another widely used approach is Gaussian processes, which generate uncertainty distributions over the objective function but can be computationally expensive (Snoek et al., 2012). More recent work has explored alternative acquisition strategies, such as those based on the moment-generating function (MGF) (Wang et al., 2017). Additionally, researchers have proposed multiobjective optimization strategies, such as multiobjective TPE, which allows users to optimize multiple objectives simultaneously without scalarization (Ozaki et al., 2020).

Recent advancements in Bayesian optimization have been driven by frameworks such as BoTorch (Balandat et al., 2020), which provides a flexible and modular approach to designing and implementing Bayesian optimization algorithms. Built on PyTorch, BoTorch supports scalable Gaussian process models, Monte Carlo acquisition functions, and efficient batch optimization strategies, making it well-suited for high-dimensional and multi-objective optimization problems. BoTorch also integrates with Ax (Chang, 2019), a higher-level optimization platform that automates the selection of acquisition functions and experiment design, further enhancing the practicality of Bayesian optimization in real-world applications.

Batch optimization improves Bayesian optimization efficiency by selecting

multiple candidate points per iteration instead of evaluating a single new parameter at a time. Traditional acquisition functions typically select one new sample per iteration and sequentially query the objective function (Frazier, 2018). In contrast, batch optimization methods, such as q-Expected Improvement (qEI) and q-Noisy Expected Improvement (qNEI), accelerate convergence by choosing multiple promising points simultaneously across the search space. This approach is particularly valuable in parallel computing environments where multiple function evaluations can be performed concurrently.

For instance, BoTorch provides built-in support for batch acquisition functions, leveraging Monte Carlo methods to efficiently estimate acquisition values for multiple candidates at once (Balandat et al., 2020). Another practical approach is local penalization, which modifies the acquisition function to discourage redundant evaluations in regions where other batch points are selected (González et al., 2016). While batch optimization is often infeasible for computationally expensive tasks such as training deep neural networks—where each function evaluation requires significant resources—our objective function evaluations are relatively fast. This allows the algorithm to execute multiple evaluations per iteration without substantial overhead. I anticipate that batch optimization would enhance convergence speed by enabling more rapid exploration of the performance landscape, particularly for systems with multiple control points, such as the p-AS pathway.

As with other design strategies based on ODE models, a challenge of my approach is the significant domain knowledge required to construct models for a target pathway, both in terms of the enzyme kinetics and the downstream metabolic processes that affect pathway activity. ODE-based models rely on detailed kinetic parameters and significant assumptions about native metabolic state that can limit their predictive power in complex cellular environments. I elaborate further on the limitations of these models in Section 4.1.2. To address these challenges, the next chapter introduces a novel simulation framework that integrates kinetic models with genome-scale metabolic modelling. This approach captures the nonlinear behaviour of pathway enzymes within the broader

metabolic network of the host, improving predictive accuracy and enabling large-scale computational screening of dynamic control strategies. I apply Bayesian optimization to the host-pathway simulator in Section 4.3.3.

3.6 Conclusion

This chapter describes the application of Bayesian optimization to the design of multiscale biological circuits. I applied the Hyperopt package, originally developed for the hyperparameter optimization of machine learning models, to the simultaneous selection of genetic control architectures and continuous dose-response parameter values. I first review the problem, elaborating on three reasons it is particularly challenging: the size of the design space, its mixed-integer nonconvex nature, and the multiple temporal and spatial scales present in models of engineered pathways under dynamic regulatory control. These production pathways have been implemented to improve yield in experimental applications but are extremely challenging to design without significant laboratory iterations. I present four case studies of production pathways: a toy example and models for the production of glucaric acid, fatty acids, and p-aminostyrene. In all cases, the Bayesian optimization reliably converges to a low objective function value while continuing to explore all architectures across the duration of its run. The fatty acid model is explored in more detail as I implemented two alternative objective functions: a cost-benefit objective that trades off production and genetic burden and a rise time-overshoot objective which aims to find optimal circuits with desired dynamics. I observed a trade-off curve between rise-time and overshoot, with some architectures being optimal for low overshoot circuits and others achieving a low rise time (see Figure 3.13). The p-aminostyrene model represents an upper bound on the currently implementable experimental complexity of dynamic control systems. I demonstrate that both it and the glucaric acid model are robust to perturbations to constants in the ODE, which indicates that the optimization results are likely consistent even under parameter uncertainty. I end by presenting several possible improvements to the method and describing two

spin-off projects conducted by students under my supervision.

Chapter 4

Simulation of dynamic host-pathway interactions at the genome scale

This chapter introduces a novel simulation approach to integrate kinetic heterologous pathway models with genome-scale metabolic models of a host. The simulator enables the prediction of local nonlinear dynamics of pathway enzymes and metabolites, informed by the global metabolic state of the host as predicted by Flux Balance Analysis (FBA). First, the background section will provide context for the types of host-pathway interactions the simulator can predict and existing computational methods used to do similar modelling tasks, such as dynamic FBA and genome-scale kinetic models. I will then describe the simulator methodology in detail, including how I achieved substantial performance improvements by replacing FBA calculations with machine learning surrogate models. There were several challenges to implementation which required novel solutions, including a warm-up routine to determine initial concentrations and a method to balance fluxes at the model boundary. Next, I demonstrate the applicability of the simulator through two case studies of production pathways in *Escherichia coli*. The simulator can predict metabolite dynamics under genetic perturbations and in various carbon sources. I showcase the utility of my method for screening dynamic control circuits through large-scale parameter sampling and Bayesian

optimization. The contents of this chapter are adapted from a paper to appear in *Metabolic Engineering* (Merzbacher et al., 2025).

4.1 Background and motivation

Heterologous production pathways can interact with native cellular processes in complex and nonlinear ways, all of which impact the metabolic capacity and viability of the production host. These impacts can occur across multiple cellular systems and scales. For example, nonnative enzyme production consumes energy resources like ATP, cofactors like NAD(P)H, or carbon sources normally used by native metabolism (Wu et al., 2016). In addition, imbalanced fluxes can lead to the accumulation of toxic intermediates or pathway enzymes (Cachera et al., 2023b). Host-pathway interactions that depend on the temporal dynamics of a production pathway are particularly challenging to predict. For example, pathway bottlenecks can form during fermentation or some toxicity effects can be caused by the transient accumulation of pathway intermediates. Other examples include the use of intracellular control systems to dynamically adjust pathway expression levels (Ni et al., 2021), or the use of external inducers to switch across pathways during fermentation (Hartline et al., 2021). These host-pathway interactions are rarely taken into account or oversimplified when modelling metabolism. In particular, many models do not incorporate the multiple timescales present in metabolism, from fast biochemical reactions to slower enzyme expression relevant for pathway regulatory loops. In addition, predicting local pathway behavior as well as global metabolic state remains a challenge for existing modelling paradigms.

4.1.1 Limitations of flux balance analysis

Pathway designers often use genome-scale metabolic models (GEMs) to predict pathway fluxes and understand interactions with native cellular processes. A key toolkit, Flux Balance Analysis (FBA) (see Section 2.2.2) uses a steady-state optimality assumption, which is typically taken to be growth rate, production

yield, or a combination of both (Orth et al., 2010). FBA assumes all reactions in the network are at steady state and thus that the concentration of all chemicals in the cell remain constant over time (Kauffman et al., 2003). Given this constraint, FBA solves a linear optimization problem to predict the metabolic fluxes of the cell provided the maximum growth rate is equal. Thanks to its linear problem structure, FBA can be solved repeatedly and rapidly, allowing experimentation in a wide range of simulated conditions (Ebrahim et al., 2013; Wang et al., 2018).

However, standard FBA cannot predict effects that depend on the temporal dynamics of pathway enzymes and metabolites during the course of fermentation. Prediction of such dynamic effects is challenging because it requires multi-scale models that simultaneously describe pathway dynamics and the host metabolism. Traditional FBA cannot account for metabolite dynamics because it inherently assumes that metabolites are at steady state and, moreover, it works on the space of fluxes without information on the enzyme kinetics that support them. There are some early proof-of-concept studies which attempted to predict native metabolite dynamics using small stoichiometric models of specific pathways such as central carbon metabolism; however, the construction of such models remains limited to a few strains and pathways (Covert et al., 2008; Min Lee et al., 2008; Oyarzún, 2011).

A number of works have developed extensions of FBA to dynamic settings, and this is subject of active research in the community. For example, several studies have employed techniques from dynamic optimization to recast FBA with time-dependent extracellular fluxes (Jeanne et al., 2018; Mahadevan et al., 2002; Reimers et al., 2017). These approaches include dynamics for extracellular metabolites and assume that the dynamics for these metabolites, usually in a bioreactor system, are slower and therefore intracellular dynamics can be approximated using a steady-state assumption (Oliveira et al., 2023). As a result, dynamic FBA (dFBA) cannot predict trajectories for intracellular metabolites or enzymes such as those in an engineered heterologous metabolic pathway. In addition, dFBA techniques do not incorporate transcriptional or translational control of enzyme expression and genetic feedback circuits. This drawback limits

their applicability to dynamic control applications. Some work has attempted to add gene expression and protein synthesis to dFBA models in order to better predict regulatory control but these models cannot predict individual pathway metabolite dynamics (Waldherr et al., 2015; Yang et al., 2019b).

4.1.2 Limitations of ordinary differential equation models

Kinetic modelling based on ordinary differential equations (ODEs; see Section 2.1.1) is an alternative approach which can capture pathway metabolite and enzyme dynamics. Finding reasonable values for enzyme kinetic parameters or fitting equation parameters to data remains a challenge, especially for engineered systems where pathway components can be drawn from multiple source organisms and inserted into an entirely different host. As a result, ODEs tend to focus on a limited set of metabolites and enzymes in a pathway without interaction with the production host, assuming a fixed growth rate and influxes to the pathway from native metabolism. A recent hybrid modelling approach aimed to inform an ODE model of a bioreactor batch process using a GEM to predict key exchange fluxes (Espinel-Ríos and Avalos, 2024b). This approach predicts intracellular fluxes from FBA to give a dynamic picture of cellular metabolism to an extracellular model but does not predict intracellular metabolites.

Several works have expanded ODEs to the genome scale (Borzou et al., 2023; Fröhlich et al., 2017; Gilbert et al., 2019; Khodayari and Maranas, 2016). These works have had some success in accurately capturing the dynamics of native metabolism in model microbes such as *E. coli* and *S. cerevisiae*. In addition, some work has had success detecting genetic interventions that improve yield (Narayanan et al., 2024). Despite these successes, genome-scale kinetic models are still limited by the parameter estimation problem. With hundreds of parameters that are challenging to fit to limited experimental data and often vary by multiple orders of magnitude across conditions and species, their applicability remains restricted to a few well-documented model organisms (Hu et al., 2023b). In practice, kinetic models require significant strain and condition-specific fine tuning, which can limit their scope and applicability (Hu et al., 2023a; Srinivasan

et al., 2015).

4.1.3 Integrating two model paradigms

This chapter introduces a novel strategy for predicting pathway dynamics coupled with the metabolism of a production host combining FBA and ODE models. The approach produces local simulations of enzyme and metabolite concentrations informed by the global metabolic state of the host. Both the kinetic ODE model and genome-scale FBA optimization iteratively pass information to each other along the temporal trajectory. This requires computing FBA solutions at many points in time and thus leads to a prohibitively long simulation runtimes. This challenge was resolved by training surrogate machine learning models to replace most FBA calculations. By pre-training the surrogate models with a limited number of offline FBA solutions, computational speed improved by at least 100-fold. Two production pathways from the literature (Borkowski et al., 2018; Doong et al., 2018) and the latest genome-scale model of *Escherichia coli* (Monk et al., 2017) demonstrate the validity and effectiveness of the approach. This chapter showcases the application of the method to predict metabolite dynamics in response to genome-wide knockouts, understand product production in various carbon sources, and design dynamic control circuits via large-scale sampling and global optimization. The results provide a novel approach for computational strain design that includes temporal effects and the cross-talk between a pathway and the host where it resides.

4.2 Methodology of simulator

4.2.1 Integration of genome-scale and kinetic models

I focus on the simulation of heterologous pathways that interact with the host metabolism through sharing of metabolites and/or co-factors. As illustrated in Figure 4.1, my strategy relies on successive iterations between two models: a dynamic Ordinary Differential Equation (ODE) model for a target heterologous

pathway, and a static, genome-scale, model for the host metabolism. The approach is centered on the ODE model to obtain local predictions of the temporal dynamics of pathway metabolites and enzymes, using the global metabolic state of the host as predicted by Flux Balance Analysis (FBA). The simulator loops between the two models, passing information back and forth from one to the other.

Specifically, I first consider a kinetic model for the heterologous pathway:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= f(x,e) - \lambda x, \\ \frac{\mathrm{d}e}{\mathrm{d}t} &= g(x) - \lambda e, \end{aligned} \tag{4.1}$$

where x and e are vectors of metabolite and enzyme concentrations, respectively, and the rate constant λ models the dilution effect caused by cell growth. The vector function $f(x,e)$ models the mass balance relationships and enzyme kinetics, and $g(x)$ is a lumped model for the transcriptional and translational processes that control expression of pathway enzymes. This function can be used to flexibly accommodate for either constitutive enzyme expression, in which case $g(x)$ is a constant, or metabolite-dependent enzyme expression as commonly encountered in dynamic pathway engineering (Liu et al., 2018), in which case $g(x)$ can be modelled as a Hill function of a pathway metabolite (Mannan et al., 2017).

The second model corresponds to a standard FBA formulation applied to a genome-scale model of the host:

$$\begin{aligned} &\lambda = \max_{v \in \mathbb{R}^m} c^T v \\ &\text{subject to: } \begin{cases} \mathbf{S}v = 0, \\ v_{\mathrm{lb}} \leq v \leq v_{\mathrm{ub}}. \end{cases} \end{aligned} \tag{4.2}$$

In this FBA problem, λ is the solution of a linear optimization problem with growth rate set as the objective function to be maximized (Orth et al., 2010). In Eq. (4.2), $\mathbf{S}$ is the stoichiometric matrix of the host metabolism, v is a vector of m metabolic fluxes, c is a weight vector that describes the contribution of

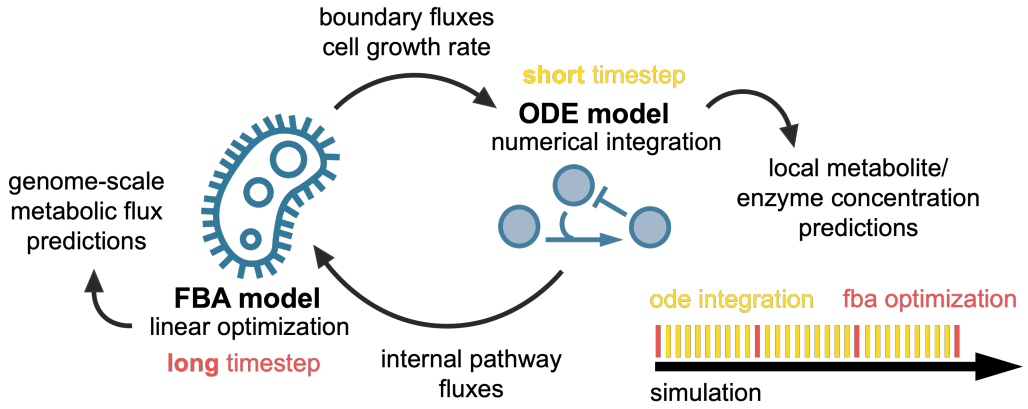


Figure 4.1: Schematic of my strategy to model dynamic host-pathway interactions.

key reactions to biomass formation, and (v_{lb}, v_{ub}) are vectors of lower and upper bounds for each metabolic flux. The stoichiometric relation $\mathbf{S}v = 0$ imposes a steady-state constraint for all metabolites included in the genome-scale model.

The iterations between both models (Figure 4.2) follow these principles: First, the FBA model in Eq. (4.2) produces a vector of fluxes V_{b} (which I call the "boundary fluxes") and a predicted growth rate λ, all of which are passed as parameters to the ODE model in Eq. (4.1). Second, I integrate the ODE model on a short time interval (yellow blocks in Figure 4.1) to produce temporal predictions of metabolite and enzyme concentrations. Third, at the end of the integration interval, I compute a flux V_{p} consumed by the pathway from the host (which I call the "pathway flux") that gets passed back to the FBA model as an equality constrain. By iterating these steps, the algorithm produces temporal trajectories for fluxes, metabolites, and enzymes of the heterologous pathway, alongside the dynamics of growth rate of the host. This approach to jointly simulate FBA-ODE models operates on two timescales: a fast metabolic timescale comprised in the FBA problem and kinetics of the production pathway, and a slow timescale of heterologous enzyme expression and cell growth.

I now present a more mathematically rigorous description of the simulation. I can express the simulation loop in pseudocode as:

The first iteration of the loop begins by initializing the boundary flux vector V_{b} and the growth rate λ using FBA with the pathway flux V_{p} set to zero. I consider both models operating in distinct alternating timesteps and divide the

Algorithm 1 FBA-ODE simulation loop

$V_\mathrm{b}, \lambda \leftarrow \mathrm{fba}(V_\mathrm{p} = 0)$
$u0 \leftarrow \text{warm-up}()$
while $t < T$ **do**
    $t_\mathrm{span} \leftarrow [t, t + \Delta t]$
    $V_\mathrm{p} \leftarrow \mathrm{ode}(V_\mathrm{b}, \lambda, u0, t_\mathrm{span})$
    $V_\mathrm{b}, \lambda \leftarrow \mathrm{fba}(V_\mathrm{p})$
    $t = t + \Delta t$
end while

simulation interval $[0, T]$ into N equispaced subintervals of length Δt. Within each subinterval $[t, t + \Delta t]$ I solve the ODE model for the heterologous pathway:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= f(x, e, V_\mathrm{b}) - \lambda x, \\ \frac{\mathrm{d}e}{\mathrm{d}t} &= g(x) - \lambda e, \end{aligned} \tag{4.3}$$

where the function $f(x, e, V_\mathrm{b})$ depends explicitly on the boundary fluxes (V_b) feeding into the pathway from the host. The enzyme and substrate concentration values at the end of the integration time $t + \Delta t$ are used to compute the pathway flux V_p. The pathway flux V_p is passed to the FBA solver as a constraint on the added sink reaction $P \rightarrow \emptyset$:

$$\begin{aligned} &\lambda = \max_{v \in \mathbb{R}^m} c'v \\ &\text{subject to:} \begin{cases} \mathbf{S}v = 0, \\ v_\mathrm{lb} \leq v \leq v_\mathrm{ub} \\ v_\mathrm{p} = V_\mathrm{p}, \end{cases} \end{aligned} \tag{4.4}$$

The updated FBA problem re-computes the optimal flux vector and growth rate λ to start a new integration cycle of the ODE model. The time t is updated every loop iteration to move it forward one interval with length Δt. In all my simulations I employed $\Delta t = 1\mathrm{s}$.

In my formulation, the heterologous pathway is not included in the genome-scale model but instead simulated separately. I implemented this by adding a sink reaction with pathway flux V_p to the genome-scale model with stoichiometry

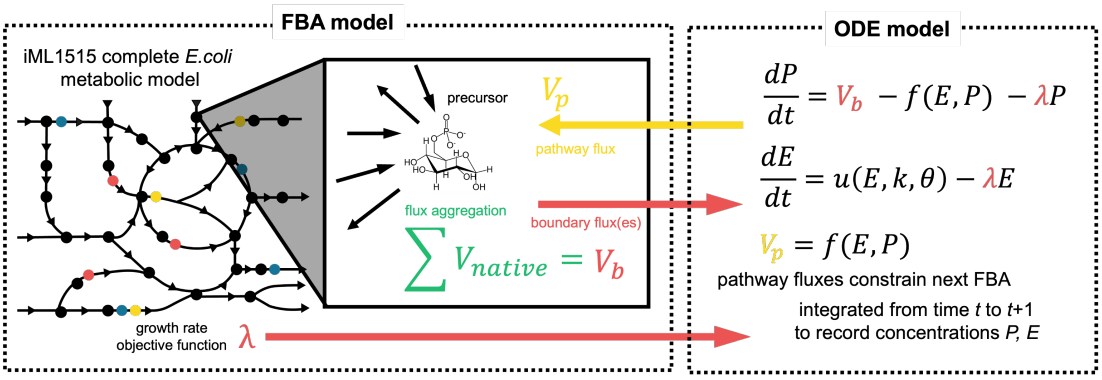


Figure 4.2: Detailed schematic description of simulator. Coloured dots in the metabolic network represent redox cofactors and high-energy phosphates like ATP which participate in many reactions.

$P \rightarrow \emptyset$, where P is the host precursor that feeds into the heterologous pathway. I note that boundary fluxes present in both models must satisfy the steady-state constraint imposed by FBA at each iteration of the simulation loop, so the ODEs are modified to ensure this relationship holds. Another challenge is the selection of initial metabolite and enzyme concentrations for the ODE simulation, as these can produce a pathway flux V_p that violates the GEM constraints. I resolved this through a warm-up routine which iteratively finds initial conditions using a Bayesian optimization method (Merzbacher et al., 2023). Details on the balancing equations and the warm-up routine can be found in Section 4.2.5.

4.2.2 Dynamic host-pathway case studies in *Escherichia coli*

I implemented the simulator in two different product production pathways already experimentally implemented in literature (Borkowski et al., 2018; Doong et al., 2018), one with an existing ODE model in literature (Verma et al., 2021). I begin by describing each pathway qualitatively before including details of the ODE models, and modifications made to the genome-scale models for integration. In both case studies I employed the iML1515 model for *E. coli* (Monk et al., 2017) as the GEM for native metabolism. An additional reaction was added to the model to represent the dynamic pathway with stoichiometry of -1.0 for all precursors and

no product. Case-study specific details of machine learning surrogate models and pathway balancing equations are given in Sections 4.2.4 and 4.2.5, respectively. I include representative sample simulator results in each section.

4.2.3 Surrogate models using machine learning

While the initial algorithm design produced promising results and converged to realistic phenotypes, I found that simulation run times were prohibitively long for practical use (e.g. over 24 hours of runtime). I performed a runtime analysis on a preliminary model and found that the FBA optimization step was responsible for more than 95% of runtime (see Table 4.1). To produce simulations in feasible computational time, I substituted all FBA-related steps with pre-trained machine learning surrogates. I split the surrogates into three supervised tasks (Figure 4.3). To generate the training data, I scanned the pathway flux V_{p} on a range and ran FBA N=500 times constrained by the pathway flux values (see top of Figure 4.3). The FBA then outputs three values as labels for the ML models to predict: a binary feasibility flag (1 if FBA finds a solution, 0 if FBA is infeasible), the growth rate, and the boundary flux(es). The boundary flux or fluxes vary per pathway; in the glucaric acid pathway, there is only one, V_{in} and in the beta carotene pathway, there are two, V_{fpp} and V_{ipp}. Once trained, the ML model then predicts from the pathway flux value (given from the ODE) to each of these three values.

I first trained a binary classification algorithm to determine whether a given pathway flux (V_{p}) resulted in a feasible FBA problem. An FBA problem is infeasible when there is no flux vector that satisfies the GEM constraints and the steady state assumption. I trained a logistic regression classifier to predict a binary outcome (1 if feasible; 0 if infeasible) from V_{p} values, so as to perform the growth rate and pathway flux predictions only for feasible samples.

All results in the following parts are based on a model trained on the wild type iML1515 model grown on glucose at default uptake rates of 10mM/h with the boundary flux computed as the V_{in} from the glucaric acid model. The dynamic pathway flux V_{p} was swept from 0 to 10mM/h and FBA was run to determine

feasibility. A logistic regression model was trained on a binary feasibility class (1 = feasible, 0 = infeasible) to predict FBA feasibility. FBA is infeasible when no positive growth rate exists within the constrained flux cone. I generated a confusion matrix for a 20% held test set ($N = 100$). The predicted labels were identical to the ground truth labels with zero false positives or false negatives. Feasible training samples were then selected for further model training. If a sample was classified as feasible, I also computed the growth rate and boundary flux. I split the data generated from FBA ($N = 500$ samples) into a training (80%) and test (20%) set and reserved the test set for model evaluation.

No. iterations	**Simulation time (hours)**	**FBA optimization time (s)**	**ODE integration time (s)**	**FBA time (%)**
10	0.0027	2.73	0.139	95.1
100	0.027	27.433	0.6012	97.8
1000	0.27	270.970	4.606	98.3
3600	1	1005.366	12.430	98.8

Table 4.1: Runtime analysis of integrated FBA and ODE simulations. Study was done on the glucaric acid pathway. The simulation was run following Algorithm 1.

I then trained two other models on samples classified as feasible: a linear regression between pathway flux (V_p) and optimal growth rate (λ), and a nonlinear regression with a neural network to predict the boundary fluxes (V_b) from the pathway flux (V_p). For the boundary fluxes, the shape of the ground truth curve varies based on the medium conditions and thus cannot be assumed to be linear like the growth rate relationship. A feedforward neural network with 3 hidden layers, 500 units per layer, and ReLu activation function was trained for 1000 epochs. The loss converged after approximately 250 epochs (see Figure 4.5).

All regressors were trained on data pairs $(V_\mathrm{p}, V_\mathrm{b})$ and (V_p, λ), respectively, generated through a large collection of offline FBA simulations; these data need to be generated only once per genome-scale metabolic model, thus leading to substantial gains in performance as compared to online FBA simulations run

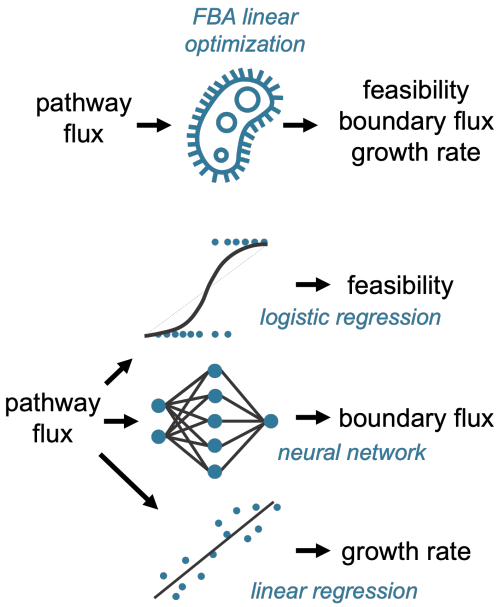


Figure 4.3: Schematic of machine learning surrogates to replace the FBA calculations and increase simulation speed.

at every iteration. The surrogate models accelerated my simulations by more than 100-fold (Figure 4.6), which enabled prediction of long time courses and exploration of the impact of various perturbations.

Once trained, I can implement the machine learning surrogates in the existing algorithm loop as follows:

Algorithm 2 Surrogate-ODE Optimization Loop

$V_b \leftarrow \text{model}_{V_b}(V_p = 0)$
$\lambda \leftarrow \text{model}_{\lambda}(V_p = 0)$
$u0 \leftarrow$ warm-up()
while $\tau < T$ **do**
    $t_{span} \leftarrow [\tau, \tau + \Delta t]$
    $V_p \leftarrow \text{ode}(V_b, \lambda, u0, t_{span})$
    if $\text{model}_{feasible}$ **then**
        $V_b \leftarrow \text{model}_{V_b}(V_p)$
        $\lambda \leftarrow \text{model}_{\lambda}(V_p)$
    end if
    $\tau = \tau + \Delta t$
end while

Instead of computing λ and the boundary flux directly using FBA, I first check if FBA finds a particular V_p feasible and then generate the V_b and λ values from their respective models. Neural networks were trained using the Flux.jl package

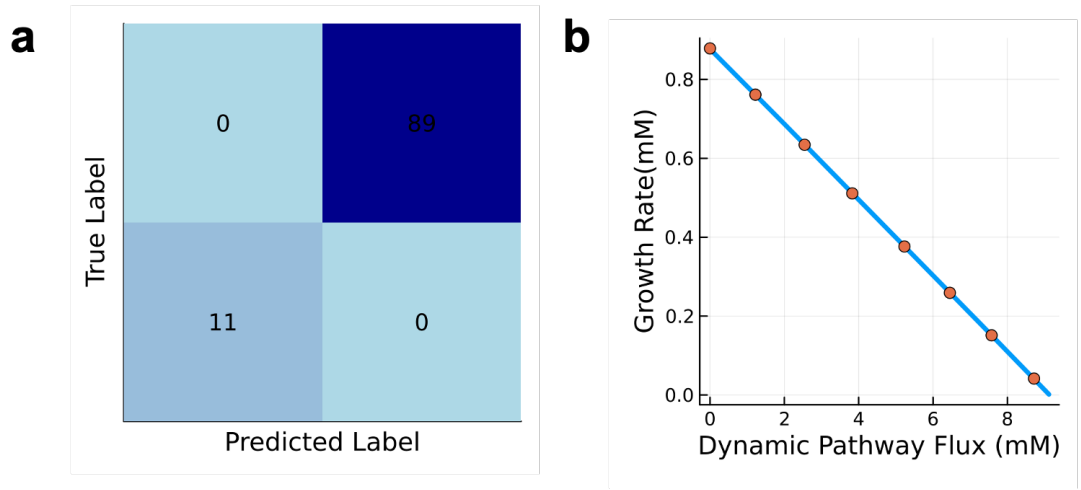


Figure 4.4: ML surrogate model training. **(a)** Confusion matrix of feasibility classification model. **(B)** A linear regression model was trained to predict the growth rate from the dynamic pathway flux V_p. The blue line is the ground truth and the orange circles are selected trained model predictions from the held-out test set.

and linear models using the GLM package. The exact training data generation routines and model specifications for each case study are given in the following sections.

Model	**Test Accuracy**	**Model R^2**	**Training MSE**	**Test MSE**
Feasibility	99.8 ± 0.005	N/A	N/A	N/A
Growth Rate	N/A	0.999 ± 1.62E-7	3.45E-33 ± 3.86E-33	3.56E-9 ± 4.88E-9
Pathway Influx	N/A	N/A	0.0124 ± 0.0198	0.0167 ± 0.0226

Table 4.2: Performance of machine learning models. The machine learning models for the glucaric acid model were run N =5 times with different random initializations and the accuracy metrics for each model reported with standard deviation across repeats.

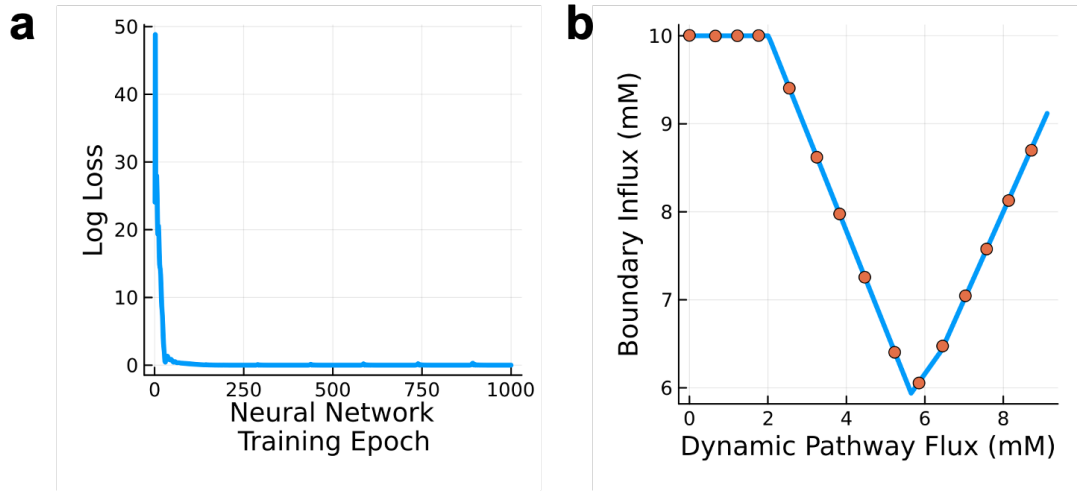


Figure 4.5: ML surrogate model training. **(a)** Training loss for boundary flux neural network. **(b)** Boundary flux predictions. The blue line is the ground truth and the orange circles are a subset of test set predictions.

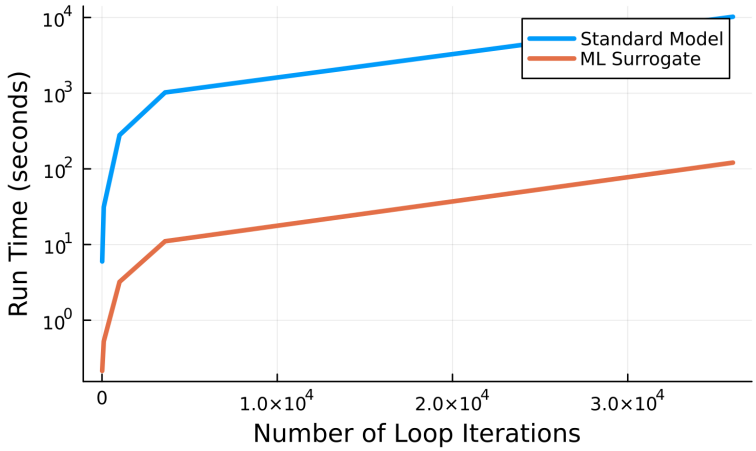


Figure 4.6: Timing comparison between FBA and ML surrogate

4.2.4 Methodological details of case studies

Synthesis of glucaric acid

I first focused on a pathway that branches off of central carbon metabolism, and hence is expected to interact strongly with the host metabolism. Glucose-6-phosphate is converted to glucaric acid via myoinositol (MI). The enzyme MIOX is activated allosterically by its substrate MI and MI is also exported from the cell. The heterologous enzymes Ino1 and MIOX can be regulated via a metabolite-responsive transcription factor, IpsA, that can be repressed upon binding to MI.

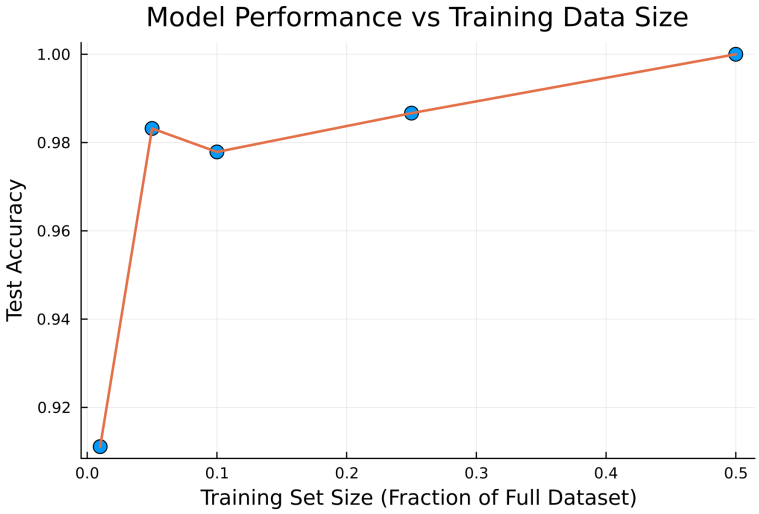


Figure 4.7: Degradation of feasibility model performance.

This feature can be exploited to build dynamic control circuits that respond to the intracellular levels of myoinositol, for example by using IpsA to control expression of Ino1 or MIOX. For the representative simulation shown in Figure 4.9, I use a dual control genetic feedback architecture where IpsA represses the upstream enzyme Ino1 and activates the downstream enzyme MIOX, creating two negative feedback loops (Doong et al., 2018). The sole boundary flux V_{in} is a sum of all reaction fluxes that produce and consume G6P in the genome-scale model, and V_{p} is the flux of the reaction catalyzed by Ino1.

Glucose-6-phosphate is the product of the first step of glycolysis and catalyzed directly from glucose, the standard carbon source for *E. coli*. Genome-scale models for *E. coli* include G6P in the FBA objective function because of its direct impact on growth rate. I integrated a previously published kinetic model of the GA pathway (Verma et al., 2021) with the iML1515 genome-scale model. My host-pathway simulations indeed show a sustained drop in growth rate (Figure 4.9), in line with the centrality of the precursor G6P. While the drop in growth rate depends on the expression levels of the heterologous enzymes, I generally observed realistic growth defects of 5–50% from the wild type (Labhsetwar et al., 2013).

I also observed a reasonable timescale of growth rate dynamics: a steady-state growth rate was achieved within 10 hours and 90% of the drop in growth

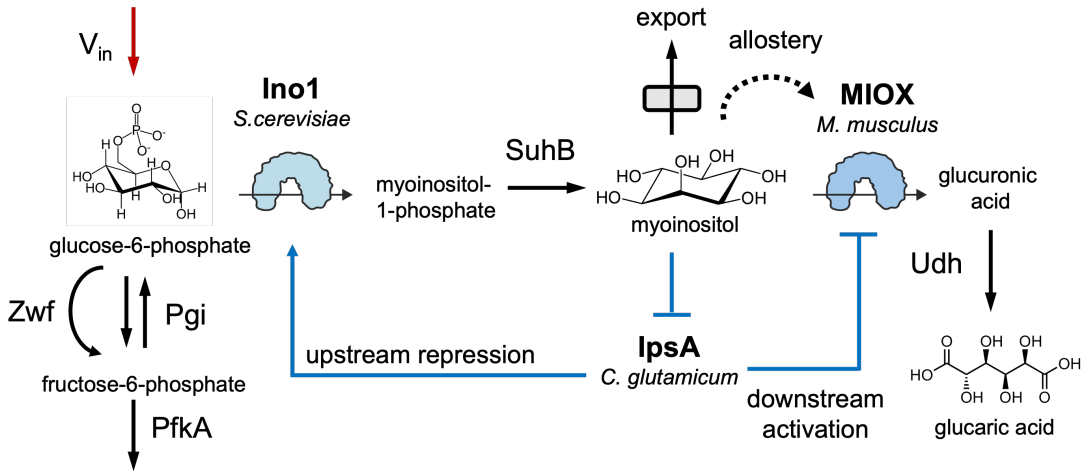


Figure 4.8: Schematic of glucaric acid production pathway. Regulatory architectures mediated by IpsA can repress Ino1 transcription upstream or activate MIOX transcription downstream, or both in a dual control architecture (Verma et al., 2021) (shown in blue). The boundary flux V_{in} computed from FBA is labeled in red.

rate occurs in the first 6 hours (Taymaz-Nikerel et al., 2013). The steady state concentrations for glucaric acid and both heterologous enzymes are within physiologically realistic ranges. The metabolite dynamics are more rapid than the enzyme dynamics, in line with the fast kinetic and slower genetic timescale separation present in the ODE model.

I employed the default growth media conditions in the iML1515 model (glucose as carbon source with a maximal uptake rate of 10mM/h). The ODE model for the glucaric acid pathway in Figure 4.8 was modified from literature (Verma et al., 2021) and the equations given in Chapter 3 to allow a variable influx and growth rate passed from FBA. The model includes three metabolites (glucose-6-phosphate, fructose-6-phosphate, and myoinositol) and two enzymes (Ino1 and MIOX). The other steps in the conversion of myoinositol to glucaric acid were assumed to be non-rate-limiting and were not included in the model. All kinetic parameters were taken from the BRENDA database (Schomburg et al., 2017). The parameters and model equations are given in 3.3.1.

Connecting the two models requires defining the boundary fluxes which enter the ODE model from the FBA model, and the pathway fluxes computed by the ODE model as constraints for the FBA model. All fluxes into and out of

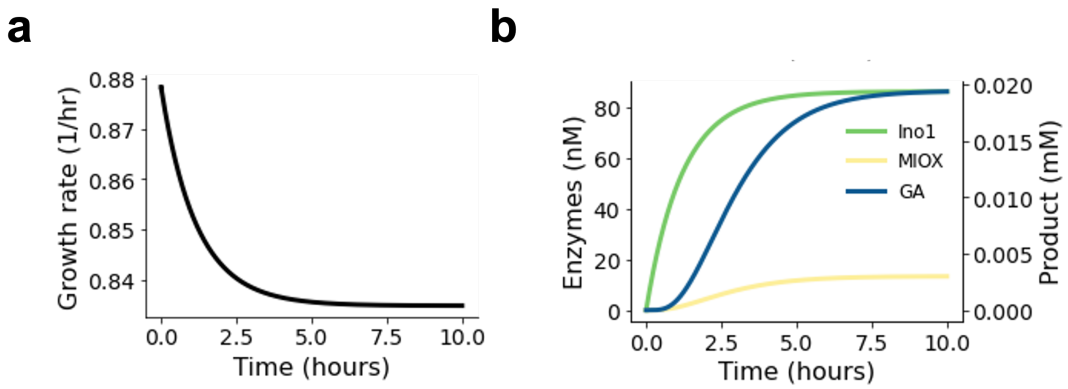


Figure 4.9: Sample simulation of host-pathway dynamics for glucaric acid production. The host-pathway model was simulated for 10 hours of simulation time (approx. 2.5 minutes of runtime) under a dual control architecture with parameters given in Appendix B.

a particular boundary metabolite in the pathway can be summed into a single boundary flux. The boundary flux (V_{in}) in this case is defined as the sum of 11 reactions that produce or consume glucose-6-phosphate: TRE6PH, PGMT, HEX1, AB6PGH, TRE6PS, FRULYSDG, GLCptspp, G6PP, G6Pt62pp, and BGLA1. To model the consumption of G6P by the glucaric acid pathway, a new forward reaction was added to the iML1515 model (V_{p}, which consumes 1 mol of G6P, has no products, and is constrained to run forward to prevent flux being drawn backwards through the production pathway.

I generated training data by sweeping the pathway flux (V_{p}) from 0 to 10 mM/h and sampled 500 linearly spaced values to constrain the FBA problem. Generation of the training data and model training took around 5 minutes on a standard laptop. I randomly split the data into training and testing using a 80:20 split, and trained a logistic regressor to predict model feasibility. I consistently achieved test accuracy above 95% (Figure 4.4a). To regress the optimal growth rate and boundary fluxes, I first I filtered the training set to include only feasible data points, and then trained a linear regressor that achieved R^2 of 0.99 or above (Figure 4.4b). To regress the boundary fluxes, I trained a feedforward neural network with 3 layers, 500 units per layer, and ReLu activation function for 1500 epochs (Figure 4.5A). When comparing the predicted results to ground truth, I find a good fit that avoids overfitting to numerical noise (Figure 4.5B).

I trained $N = 5$ training repeats and achieved consistently high performance (see Table 4.2). I also confirmed that model performance degraded with smaller training set sizes by reducing the training set size for the feasibility assessment by 50-99% and plotted the test accuracy of the feasibility binary classification (see Figure 4.7). The performance of the regression models also degrades when assessing test MSE but less substantially, likely due to the simple functional form of the prediction task. I incorporated the machine learning models into the loop as a replacement for FBA, which improved the model run time by several orders of magnitude. For example, model runs that took >1000 seconds in the standard loop were accelerated to 12 seconds with the surrogate models. Figure 4.6) shows the loop runtime for standard FBA and ML surrogate loop, in seconds (log scaled). Both increase linearly with the number of loop iterations; however, standard FBA becomes rapidly computationally infeasible due to the cost of linear FBA optimization. For comparison, most simulations run for the results require 8.6×10^4 loop iterations and would take $>$ 6.6hrs to run with a standard FBA model, compared to around 5 minutes with the ML surrogate. Models and training data objects were serialized to .jls files to minimize retraining time. For the variable growth conditions in Section 4.3.1, I modified the uptake reactions for all carbon sources to limit influx to a single source and regenerated training data for each carbon source. All models were retrained for a total of 15 conditions ($N = 45$ new models).

Synthesis of beta-carotene

As a second example that branches from a downstream region of metabolism, I selected the beta-carotene production pathway. I chose this pathway because it branches from native metabolites not included in the biomass equation and thus is not directly coupled to growth in the FBA formulation. Beta-carotene is a high-value metabolite with applications in medicine, cosmetics, animal feed, and nutritional supplements (Hernández-Almanza et al., 2016; Khan et al., 2008). The biosynthetic beta-carotene production pathway has been implemented experimentally using codon-optimized enzymes from *Erwinia uredovora* engineered

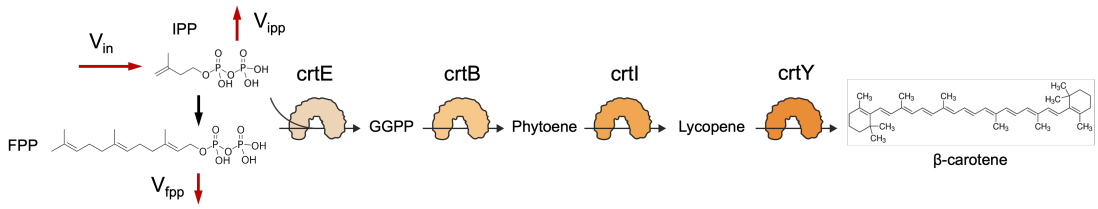


Figure 4.10: Beta-carotene production pathway schematic. The three boundary fluxes computed from FBA are labelled in red.

into *E. coli* (Kim et al., 2006; Misawa et al., 1990; Yang and Guo, 2014). The pathway has four enzymes: crtE, crtB, crtI, and crtY (see Figure 4.10). The two precursors farnesyl pyrophosphate (FPP) and isopentyl pyrophosphate (IPP) are catalyzed by crtE to form geranylgeranyl pyrophosphate (GGPP). The enzyme crtB then catalyzes the conversion to phytoene, followed by crtI converting phytoene to lycopene. The enzyme crtY catalyzes the final step in the pathway to produce beta-carotene.

Unlike the glucaric acid pathway, which has only one boundary flux (V_{in} in Figure 4.8), the beta-carotene model combines two precursors, FPP and IPP, both of which have their own effluxes and influxes. As shown in Figure 4.10, the boundary flux vector V_{b} is thus composed of three elements: V_{in} is the influx to IPP, V_{ipp} is the efflux from IPP, and V_{fpp} is the efflux from FPP. The pathway flux V_{p} is the crtE-catalyzed conversion from the precursors to GGPP.

I built a kinetic model for the beta-carotene pathway from scratch and integrated it with the iML1515 genome-scale model. The standard iML1515 GEM was modified by adjusting its medium conditions to fit an experimentally determined wild type growth rate of 0.65mM/h (Borkowski et al., 2018). I adjusted the medium conditions in the FBA to a limited fructose source by setting a constraint on the fructose influx (-7.5 mM/h) that resulted in an equivalent growth rate when the heterologous pathway was not induced. Since the pathway has two precursors (farnesyl pyrophposhate, FPP, and isopentyl pyrophosphate, IPP), the boundary flux vector V_{b} has several components. I define a boundary influx into IPP (V_{in}) as the sum of the reactions IPDDI and IPDPS. I define an efflux to IPP (V_{ipp}) as the sum of the reactions UDCPDPS, OCTDPS, DMATT, IPDPS

with appropriate stoichiometric weighting. FPP is entirely produced by IPP and its efflux (V_{fpp}) is defined by the sum of the reactions UDCPDPS, HEMEOS, and OCTDPS. A new forward reaction (V_{p}) was added to the iML1515 model to represent consumption of FPP and IPP by the beta-carotene pathway. This reaction consumes 1 mol of FPP and 1 mol of IPP, has no products, and is constrained to run forward to prevent flux being drawn backwards through the production pathway.

For the kinetic model, I included the four enzymes in the pathway (crtE, crtB, crtI, crtY) along with six metabolites: FFP, IPP, GGPP, phytoene, lycopene, and beta-carotene. A constant influx to the pathway V_{in} and a growth rate λ are set as parameters in the ODE. The first reaction in the pathway follows Michaelis-Menten kinetics with two substrates:

$$V_{\text{crtE}} = \frac{k_{\text{cat}}\text{crtE} \cdot \left(\frac{\text{FPP}\cdot\text{GGPP}}{k_{\text{M,FPP}}k_{\text{M,GGPP}}}\right)}{1 + \left(\frac{\text{FPP}}{k_{\text{M,FPP}}} + \frac{\text{GGPP}}{k_{\text{M, GGPP}}}\right)}, \tag{4.5}$$

where $k_{\text{M,FPP}}$ and $k_{\text{M,GGPP}}$ are substrate-specific enzyme affinity parameters. All other enzymes are assumed to follow standard Michaelis-Menten kinetics:

$$V = E\frac{k_{\text{cat}}\text{S}}{k_{\text{m}} + \text{S}}, \tag{4.6}$$

where E is the enzyme concentration, S is the substrate concentration, and k_{cat} is the turnover number.

I sourced k_{m} values and averaged available k_{cat} numbers from BRENDA (Schomburg et al., 2017); however, the turnover numbers were not available for some enzymes and often available only for unrelated organisms. I instead used a deep learning model trained on all available enzyme kinetic information to predict k_{cat} values from enzyme amino acid sequence (Li et al., 2022). For those k_{cat} values available in BRENDA, the deep learning model delivered predictions in the same order of magnitude.

The substrate equations can thus be written as:

$$\begin{aligned}
\frac{\mathrm{dFPP}}{\mathrm{d}t} &= V_{\mathrm{in}} - 2 \cdot V_{\mathrm{crtE}} - V_{\mathrm{FPP}} - \lambda \cdot \mathrm{FPP} \\
\frac{\mathrm{dIPP}}{\mathrm{d}t} &= V_{\mathrm{in}} - 2 \cdot V_{\mathrm{crtE}} - V_{\mathrm{IPP}} - V_{\mathrm{FPP}} - \lambda \cdot \mathrm{FPP}, \\
\frac{\mathrm{dGGPP}}{\mathrm{d}t} &= -V_{\mathrm{crtE}} - V_{\mathrm{crtB}} - \lambda \cdot \mathrm{GGPP}, \\
\frac{\mathrm{dPhy}}{\mathrm{d}t} &= V_{\mathrm{crtB}} - V_{\mathrm{crtI}} - \lambda \cdot \mathrm{Phy}, \\
\frac{\mathrm{dLyc}}{\mathrm{d}t} &= V_{\mathrm{crtI}} - V_{\mathrm{crtY}} - \lambda \cdot \mathrm{Lyc}, \\
\frac{\mathrm{dBcar}}{\mathrm{d}t} &= V_{\mathrm{crtY}} - \lambda \cdot \mathrm{Bcar}.
\end{aligned} \tag{4.7}$$

The equations for all four enzymes follow standard open-loop production with a promoter strength parameter k_E. For example, for the first enzyme in the pathway, the equation is

$$\frac{\mathrm{dcrtE}}{\mathrm{d}t} = k_{\mathrm{crtE}} - \lambda \cdot \mathrm{crtE}. \tag{4.8}$$

Table 4.3 contains all kinetic parameter values for the ODE model.

My host-pathway simulations (Figure 4.11) show that in this case pathway activity leads to a transient growth defect, in line with the fact that precursors FPP and IPP are not weighed in the FBA objective function. Production of beta-carotene results in a defect to growth rate which recovers upon consumption of the precursors and reduction of pathway flux. The metabolite dynamics include consumption of the precursors followed by peaks of each intermediate in the pathway. The timing and size of these peaks depends on the enzyme promoter strengths, and I generally observed beta-carotene concentrations in a feasible mM range. The enzyme expression dynamics are also realistic, taking over 10 hours to reach steady state, and operate in a feasible nM range. Altogether, both case studies (glucaric acid and beta-carotene) demonstrate the reasonable results produced by msimulation approach and its applicability to a wide range of metabolic pathways.

Parameter	Value	Units	Source
$k_{\text{cat},crtE}$	0.2456	1/s	deep learning
$k_{\text{cat},crtB}$	0.066	1/s	deep learning
$k_{\text{cat},crtI}$	4.2255	1/s	deep learning
$k_{\text{cat},crtY}$	42.9099	1/s	deep learning
$k_{\text{m},crtE,\text{FPP}}$	0.0321	μM	BRENDA
$k_{\text{m},crtE,\text{IPP}}$	0.0234	μM	BRENDA
$k_{\text{m},crtB}$	0.01682	μM	BRENDA
$k_{\text{m},crtI}$	9.179	μM	BRENDA
$k_{\text{m},crtY}$	0.035	μM	BRENDA

Table 4.3: Kinetic parameters of the beta-carotene pathway model. For those parameters absent from the BRENDA database, I employed the deep learning predictor from Li et al., 2022.

For the surrogate models, training data was generated by varying the pathway flux (V_{p}) from 0 to 1mM/h and sampling 500 linearly spaced values to constrain the FBA problem. Generation of the training data set and model training took around 5 minutes on a standard laptop. I held out 20% of the training set for model testing. I trained a logistic regression model to predict model feasibility. I trained a linear regression model to predict growth rate. I found a linear relationship between V_{p} and all boundary fluxes, so to speed up training time I trained three linear regression models to predict the boundary flux components V_{in}, V_{ipp}, and V_{fpp}. All linear models obtained an R^2 of 0.99 or better and the logistic regression obtained a test accuracy of 0.99. Models and training data objects were serialized to .jls files to minimize retraining time. For the genome-wide knockout screen in Section 4.3.2 the FBA model was modified by knocking out each gene and generating new training data in each case, and then retraining all models ($N = 1,515$ knockouts, $N = 7,575$ models).

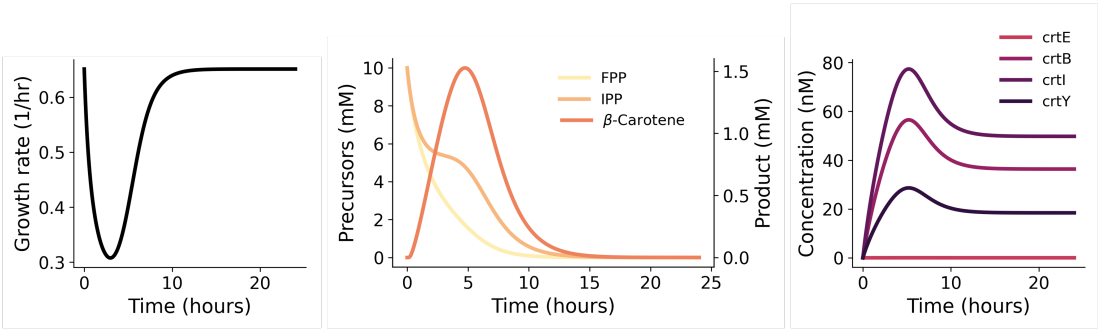


Figure 4.11: Simulation of host-pathway dynamics for beta-carotene production. The host-pathway model was simulated for 24 hours of simulation time (approx. 5 minutes of runtime) with parameters given in Appendix B.

4.2.5 Challenges to algorithm implementation

I encountered several challenges in implementing the simulator loop, even once the basic structure of the modelling paradigm was established. I introduce these challenges in more detail here: first, that the fluxes included in both models must balance so the FBA optimization problem remains feasible, and second, that the initial concentration values for the ODE native metabolites must be determined before the simulation loop can start. Both solutions are pathway-specific, but I provide guidance for how to implement them in future case studies in Appendix C.

Balancing fluxes at the model boundary

Since the precursor and dynamic pathway flux V_{p} are included in both models, the ODE must satisfy the steady-state constraint imposed by FBA at each iteration of the simulation loop; that is, the reaction fluxes that produce and consume the precursor(s) must equal each other and balance out. To force this constraint on the pathway model, I modified the precursor equation(s) in the ODEs to express all fluxes as a function of the boundary flux V_{in} and the pathway flux V_{p}. The exact balancing equations are pathway-dependent and described for each of my case studies next.

The glucaric acid pathway required no modifications to the ODEs as it is the simplest case in which one boundary flux feeds into the precursor G6P and one

pathway flux consumes it. The flux balance is therefore simply:

$$V_{\text{in}} = V_{\text{p}}. \tag{4.9}$$

The precursor equation remains unchanged from the literature model:

$$\frac{d\text{G6P}}{dt} = V_{\text{in}} - V_{\text{p}} - \lambda \cdot \text{P}. \tag{4.10}$$

For the beta-carotene pathway, both precursors (FPP and IPP) combine in an equimolar reaction to produce the pathway flux V_{p}. I define three components of the boundary flux vector: V_{in} and the two precursor effluxes V_{fpp} and V_{ipp}. The unbalanced precursor equations are as follows:

$$\begin{aligned} \frac{\text{dIPP}}{\text{d}t} &= V_{\text{in}} - V_{\text{ipp}} - V_{\text{p}} - V_{\text{conv}} - \lambda \cdot \text{IPP} \\ \frac{\text{dFPP}}{\text{d}t} &= V_{\text{conv}} - V_{\text{fpp}} - V_{\text{p}} - \lambda \cdot \text{FPP}, \end{aligned} \tag{4.11}$$

where V_{conv} is the flux converting IPP to FPP, which is included in the FBA but is not a boundary flux and therefore not passed to the ODE. There is no inherent constraint on the ODE which ensures that the boundary fluxes and V_{conv} must satisfy the steady-state assumption, so I must impose one by expressing V_{conv} in terms of the boundary fluxes. Taking a flux balance of each precursor, I obtain the following equations:

$$\begin{aligned} 0 &= V_{\text{in}} - V_{\text{ipp}} - V_{\text{p}} - V_{\text{conv}} \\ 0 &= V_{\text{conv}} - V_{\text{fpp}} - V_{\text{p}}. \end{aligned} \tag{4.12}$$

I rearrange the second equation to find an expression for V_{conv} in terms of only the boundary and pathway fluxes:

$$V_{\text{conv}} = V_{\text{fpp}} + V_{\text{p}}. \tag{4.13}$$

I substitute this expression into the the precursor equations in Eq. (4.11) and

obtain:

$$\begin{aligned} \frac{d\text{IPP}}{dt} &= V_{\text{in}} - V_{\text{ipp}} - 2V_{\text{p}} - V_{\text{fpp}} - \lambda \cdot \text{IPP} \\ \frac{d\text{FPP}}{dt} &= -V_{\text{fpp}} - \lambda \cdot \text{FPP}. \end{aligned} \tag{4.14}$$

This modification to the ODE model ensures that the boundary fluxes (V_{in}, V_{fpp}, and V_{ipp}) satisfy the steady-state constraint at each iteration of the simulator loop. These ODE modifications are pathway-specific and in Appendix C I show the balancing equations for several common topologies encountered in applications.

Warm-up routine for initial conditions

The initial conditions of the ODE must be chosen prior to the first iteration of the simulation loop. In some cases, the native concentrations of metabolites might be known from literature or lab experiments, however this was not the case for my pathways. The initial conditions selected must result in a FBA-feasible pathway flux V_{p}. To satisfy this constraint, I created a warm-up routine which iteratively tries various initial conditions using a Bayesian optimization method (Bagge Carlson, 2018). These initial conditions are then run for 500 iterations of the simulator (500 seconds) and each iteration is checked for FBA feasibility. If initial conditions are not valid for all 500 iterations, the objective function of the Bayesian optimizer is heavily penalized. The objective is:

$$J = \frac{1}{\sum_{i=1}^{N} C_i} + f \cdot 10^7, \tag{4.15}$$

where f is the binary feasibility flag (1 if infeasible) and C_i are the concentrations of native metabolites, i.e. G6P and F6P for the glucaric acid pathway, and FPP and IPP for the beta-carotene pathway. I assume that the initial concentrations of heterologous enzymes and pathway metabolites are zero, as the host does not produce them natively before the pathway is induced. The warm-up routine runs for 1000 initial condition values and the initial conditions which achieve the lowest value of J are selected for further simulation. If no feasible conditions are found, i.e. when the minimum objective function value is greater than 10^7, no further

simulation runs. For the Bayesian optimization loop, I employed a Tree of Parzen Estimator (TPE) algorithm implemented in the TreeParzen.jl Julia package. Expected Improvement was used as the acquisition function and initial conditions for the native metabolites (G6P, F6P, FPP, and IPP) were drawn from uniform priors with a maximum of 0.75mM (glucaric acid) and 10mM (beta-carotene), set by integration the ODE to steady state without the pathway induced, i.e. without heterologous enzymes. This warm-up routine reliably converges to qualitatively reasonable values and can easily be adapted to integrate additional information about the native metabolite concentrations as priors on the Bayesian Optimization.

4.3 Applications and results

4.3.1 Growth and production in different carbon sources

Since *E. coli* can grow in a variety of carbon sources, I sought to predict the impact of growth media on the temporal dynamics of the glucaric acid pathway in Figure 4.8. I deliberately chose a panel of 15 carbon sources that enter central metabolism at different points, both upstream and downstream of glucose-6-phosphate, which feeds into glucaric acid synthesis. As shown in Figure 4.8, the glucaric acid (GA) production pathway branches from CCM at glucose-6-phosphate (G6P). Besides galactose and glucose, all considered carbon sources enter CCM below G6P. (Figure 4.12A). This allows checking the consistency of my simulation approach: the host should be able to grow in all carbon sources and glucaric acid should be produced only in conditions where carbon enters metabolism at or upstream from glucose-6-phosphate. To perform these simulations, the machine learning FBA surrogates had to be re-trained for each of the 15 growth conditions, using data produced with offline FBA simulations (see Section 4.2.3). Simulations were run for 24 hours of simulation time with parameters given in Appendix B.

Pathway simulations (Figure 4.12B) indeed predict *E. coli* growth in all tested carbon sources, but glucaric acid production was observed only in galactose and

glucose, both of which enter glycolysis upstream of glucose-6-phosphate. For all carbon sources except galactose and glucose, the predicted growth rates match those reported previously in the literature (Monk et al., 2017), ranging from 0.25 mM/h up to approximately 0.8 mM/hr. Growth on galactose and glucose resulted in a 10.3% and 9.3% drop in growth rate, respectively, due to diversion of central carbon flux towards glucaric acid production. These simulations serve as a computational validation of the consistency of my approach and demonstrate its ability to predict dynamics of heterologous pathways in a media-specific fashion.

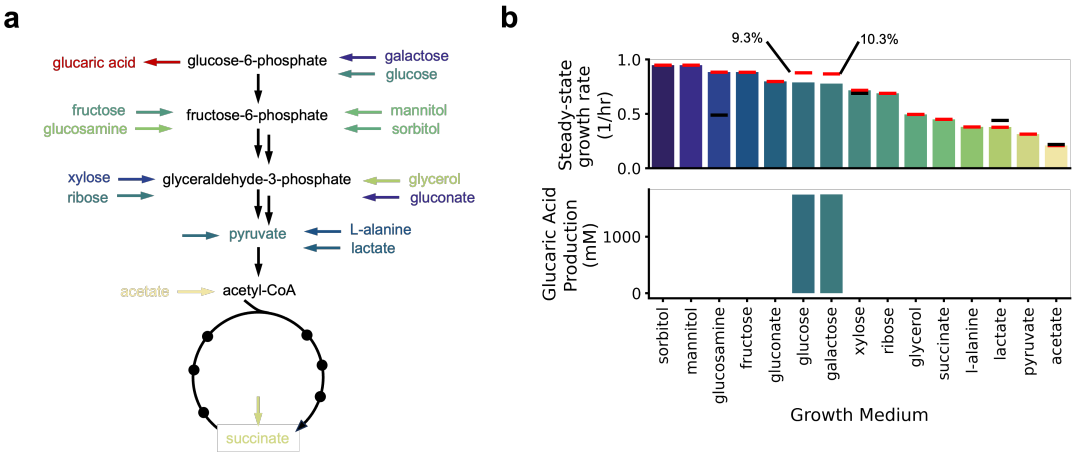


Figure 4.12: Simulator performance across carbon sources. **(A)** Schematic of central carbon metabolism (CCM) with various carbon sources entering at different points. **(B)** Predicted steady-state growth rates and cumulative GA production, computed as the temporal integral of the GA production flux. Wild type growth rates predicted by FBA are shown as red lines.

4.3.2 Impact of gene deletions on pathway dynamics

To test the ability of my approach to predict metabolite dynamics under genetic perturbations, I ran a genome-scale knockout screen on the *E. coli* model, and predicted the dynamics of the beta-carotene pathway (Figure 4.13) in each condition. To this end, I zeroed out the flux bounds for the reaction associated to each knockout, and then ran my algorithm with the modified genome-scale model. As in the previous section, this requires re-training the machine learning FBA surrogates for each of the 1,515 knockouts, which in this case involves the generation of large scale data for training, totalling over 757,500 samples computed through

off-line FBA simulations.

A genome-scale knockout screen of strains engineered with the beta-carotene pathway in Figure 4.10 was performed. Figure 4.13 shows a histogram of all knockouts and their resulting growth defect. The majority of knockouts either have no effect on growth, i.e. 0% growth defect from wild type (0.65mM/h), or are essential, i.e. >50% defect shown by red dotted line. Knockouts with a defect in the range 5–20% are defined as intermediate knockouts and highlighted in red. Simulations were run for 24 hours of simulation time with parameters given in Appendix B.

I found that my knockout simulations predicted the same set of growth essential genes as the wild type iML1515 model, with 12.3% of enzymatic genes being essential with a growth defect above 50% from the wild type (Figure 4.13). In line with expectations, beta-carotene production was observed only in the nonessential knockouts. Examination of beta-carotene dynamics across all nonessential knockouts ($N = 1,310$) suggests that the large majority do not affect the temporal trajectory of product concentration (Figure 4.15). However, the simulations also identified a small set of 25 nonessential knockouts associated with intermediate growth defects, ranging from 6% to 18% from the wild type (Figure 4.13), that had a pronounced effect on beta-carotene dynamics 4.14). Each row of the heatmap is a timecourse of beta-carotene production. Darker colors indicate a peak in the production. Knockouts with significant impact on product dynamics can be seen as streaks on the plot. Among these nonessential knockouts, I found a larger growth defects led to a more pronounced peak in beta-carotene concentration (Figure 4.15). The knockouts selected were: b1779, b2277, b2779, b0432, b0721, and b3919.

Since my approach also allows simulation of pathway dynamics in response to genetic perturbations in the heterologous pathway itself, I selected 6 representative intermediate knockouts from Figure 4.13 and varied the expression level of the rate limiting enzyme (crtE, shown in Figure 4.10) through changes to the strength of its promoter. I swept the promoter strength of the rate-limiting enzyme (crtE) across four orders of magnitude. The other enzymes (crtB, crtI,

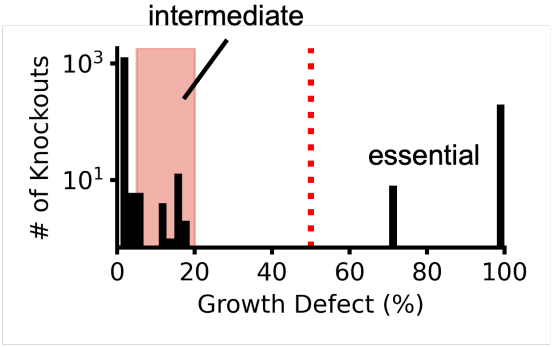


Figure 4.13: Histogram of knockout screen growth defects.

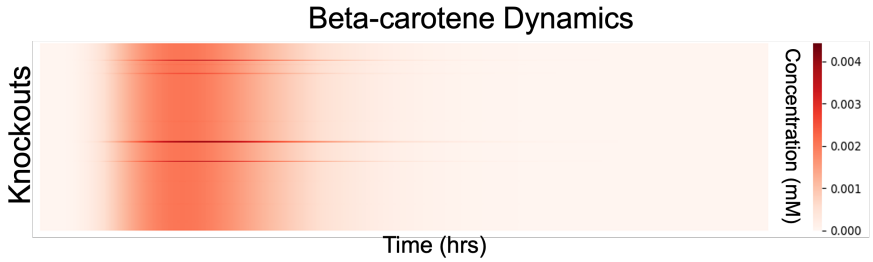


Figure 4.14: Beta-carotene concentration dynamics for all nonessential knockouts.

crtY) had fixed promoter strengths equal to those used in the knockout screen. The results in Figure 4.16 suggest that peak beta-carotene concentration is primarily sensitive to the promoter strength, while its rise time is primarily affected by the size of the growth defect. Altogether these results underline the utility of my approach to predict metabolite dynamics in response to local and global genetic perturbations.

Despite the computational overhead required to re-train the surrogate models for each of the 1,515 knockouts, I found that my approach provides substantial gains in efficiency as compared to the use of FBA optimization. Even for a relatively high-density training set of $N = 500$ data points, generation of training data took less than 2min/knockout with a negligible time for model training. I performed an additional timing study varying the training set size (Table 4.4). The timing study was conducted on the glucaric acid pathway with glucose as the carbon source. Fresh training data for growth in glucose was generated N =5 times and the feasibility, growth rate, and pathway flux machine learning models were re-trained from scratch $N = 5$ times to compute average times and their

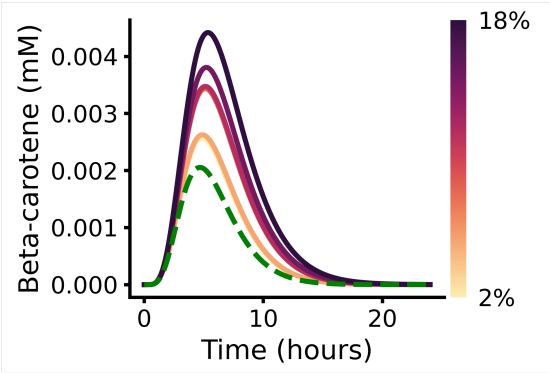


Figure 4.15: Beta-carotene dynamics for representative intermediate knockouts, coloured by the size of growth defect. The no knockout (NK) curve is shown for reference (green dashed line).

standard deviation. When extrapolated to all 1,515 knockouts, these results suggest that data generation and model training take approximately 49h and 6h, respectively. In contrast, a 24h simulation with the full FBA calculation takes almost 7h/knockout, totalling over 10,000h of computation time for all knockouts.

Training set size	**Training data generation time (s)**	**ML model training time (s)**
500	117.2 ± 8.60	14.27 ± 3.87
100	27.9 ± 3.24	4.12 ± 0.46
50	21.1 ± 2.85	3.17 ± 0.29
25	8.57 ± 0.45	2.36 ± 0.095

Table 4.4: Computational cost for generation of training data.

4.3.3 Host-aware screening of metabolic control circuits

Given that my simulation approach can produce temporal predictions in reasonable computational time, I sought to explore its use as platform for computational screening of metabolic control circuits. Dynamic control circuits (see Section 1.1.3) have emerged as a powerful tool to build responsive pathways that self-adapt to changes in growth conditions (Liu et al., 2018; Ni et al., 2021). The large design space and high implementation costs of these systems have lead to

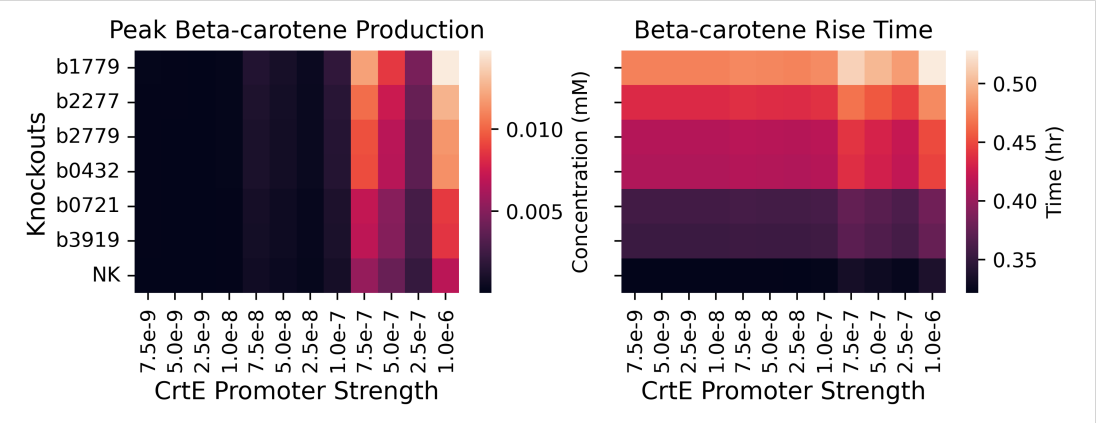


Figure 4.16: Heatmaps of peak beta-carotene concentration and rise time.

a number of computational approaches being developed for exploring the space of circuit architectures such as the BayesOpt method developed in Chapter 3 (Stevens and Carothers, 2015; Verma et al., 2021). However, these methods generally overlook the interactions with the host. To illustrate the utility of my approach for host-aware screening of control circuits, I employed two complementary approaches based on parameter sampling for a fixed architecture, and apply BayesOpt to optimize both parameters and control architectures.

Large-scale parameter sampling

I first assessed the performance of negative feedback loops from beta-carotene to the promoters producing the pathway enzymes, and compared it to the open-loop case in which all pathway enzymes are constitutively expressed (Figure 4.17). A negative feedback loop is considered between beta-carotene and the expression of all pathway enzymes via a metabolite-responsive transcription factor. Following the design from Borkowski et al., 2018, I considered the four enzymes expressed in an operon with variable ribosomal binding site (RBS) strengths, modelled through four parameters k_i and a single regulatory threshold parameter (θ), as shown in Eq. 4.16 of the main text. The negative feedback acts to downregulate expression of the *crt* enzyme genes in response to beta-carotene, and its has been shown to improve robustness to perturbations (Oyarzún and Stan, 2013).

Following the notation in my general model in Eq. (4.1), I assumed that all

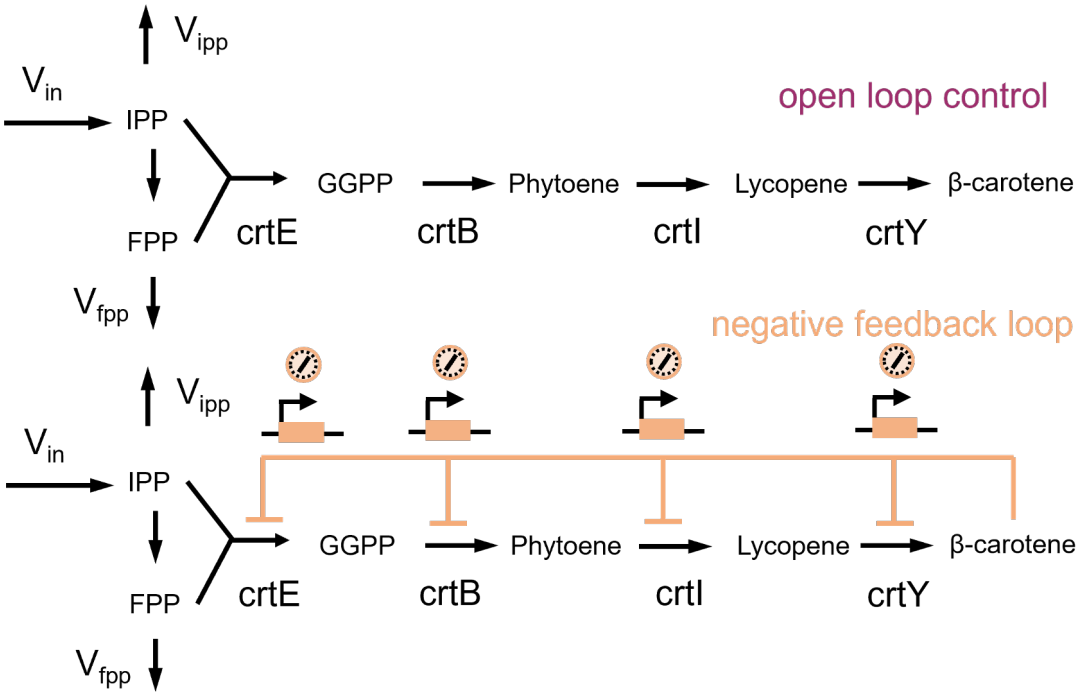


Figure 4.17: Diagram of circuit architectures for beta-carotene pathway in Figure 4.10.

enzymes are expressed at a rate:

$$g_i(\text{Bcar}) = \frac{k_i}{1 + \left(\frac{\text{Bcar}}{\theta_i}\right)^n}, \tag{4.16}$$

where Bcar is the concentration of beta-carotene, and (k_i, θ_i) are the promoter strength and regulatory threshold, respectively. The Hill coefficient n is set to 2. As noted in Figure 4.13, the beta-carotene model results in a transient production and growth defect that returns to the wild type conditions as the precursors are depleted. I thus performed large-scale random sampling of the (k_i, θ_i) space and scored each design according to the cumulative beta-carotene production and drop in growth rate, both computed as temporal integrals of the simulated dynamics. Latin hypercube sampling was used to produce $N = 1,000$ parameter sets for both open loop control (4-dimensional; no θ sampling) and negative feedback (5-dimensional). Bounds for the parameter search space are given in Appendix B. Host-pathway simulations were run for 24h and then scored with the cumulative

beta-carotene production and cumulative growth defect, defined as $\int_0^T \text{Bcar}(t)\,\mathrm{d}t$ and $\int_0^T (\lambda_{\text{WT}} - \lambda(t))\,\mathrm{d}t$, respectively; λ_{WT} is the wild type growth rate. The results in Figure 4.18 show the emergence of a trade-off front (dashed black line) across designs, with high producers resulting in larger drop in growth rate. This trade off is qualitatively similar to the relationship between circuit capacity and growth rate reported by Borkowski et al., 2018. I also found that the negative feedback loop increased the median beta-carotene production timecourse (Figure 4.19).

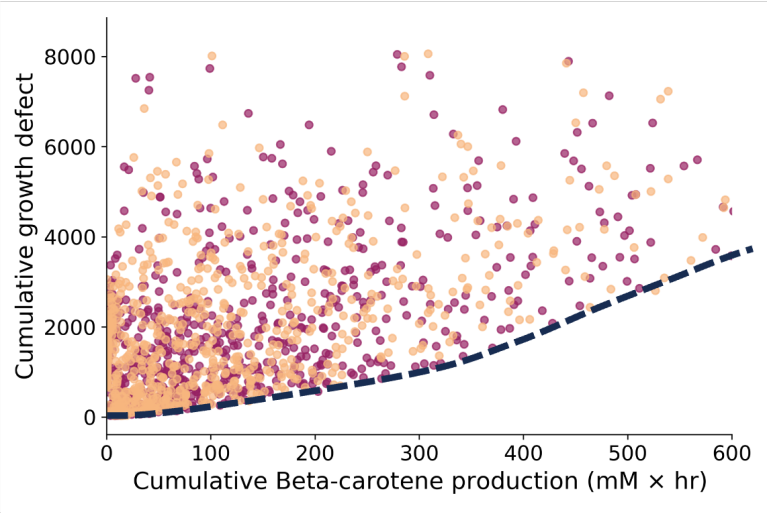


Figure 4.18: Random sampling of genetic parameter space. Latin hypercube sampling was used to produce $N = 1,000$ parameter sets for both open loop control (4-dimensional) and negative feedback (5-dimensional). Bounds for the parameter search space are given in Appendix B. Host-pathway simulations were run for 24h and then scored with the cumulative beta-carotene production and cumulative growth defect, defined as $\int_0^T \text{Bcar}(t)\,\mathrm{d}t$ and $\int_0^T (\lambda_{\text{WT}} - \lambda(t))\,\mathrm{d}t$, respectively; λ_{WT} is the wild type growth rate. An trade-off front emerges between production and growth defect (dashed black line).

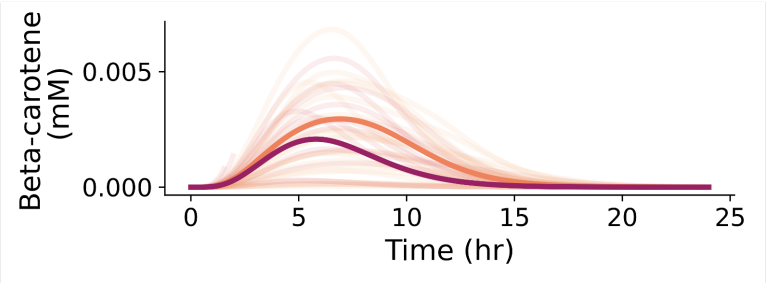


Figure 4.19: Beta-carotene production dynamics. The median curves are darker; lighter curves are sampled randomly from ±1 standard deviation around the median beta-carotene timecourse. The negative feedback architecture (orange) tends to increase peak beta-carotene concentration when compared to the open loop (red).

Global optimization of control circuits

As a final test of my approach, I used it to automatically search for optimal control parameters and architectures using the BayesOpt framework introduced in Chapter 3. I chose the glucaric acid pathway as a test case and restricted the search to three genetic feedback architectures in addition to open loop control (Figure 4.20). This is a challenging optimization problem that requires simulating the pathway dynamics at many points in the input space, and thus a ideal test to assess if my simulation approach could be deployed within such computationally intensive tasks.

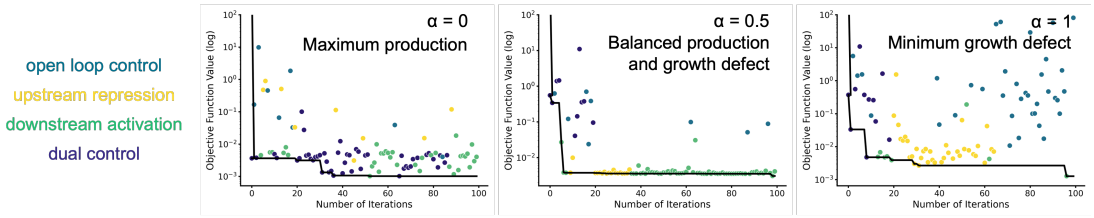


Figure 4.20: Control circuit screening for glucaric acid production using mixed-integer optimization of both regulatory architectures and parameters. Points are coloured by their respective architectures (shown in Figure 4.8) and the black line traces the best architecture found throughout the optimization run. Bounds for the parameter search space are given in Appendix B.

Since the glucaric acid pathway leads to a permanent growth defect (Figure 4.9), I constructed an objective function J that weighs the drop in steady state growth rate and the cumulative production of glucaric acid

$$J = \alpha \frac{\lambda_{\mathrm{WT}} - \lambda_{\mathrm{SS}}}{\lambda_{\mathrm{WT}}} + \sigma(1-\alpha)\frac{1}{\int_0^T \mathrm{GA}(t)\,\mathrm{d}t}, \tag{4.17}$$

where λ_{SS} and λ_{WT} are the steady state growth rate of the production and wild type strain, respectively, and GA is the concentration of glucaric acid. The parameter σ is scaling factor to ensure both terms are in the same magnitude, and α is a weight to control the balance between growth defect and production. The scaling factor was set to $\sigma = 1,256$. The optimization results in Figure 4.20 show that the optimizer can effectively single out optimal design; for a fixed value of α, each optimization run ($N = 100$ iterations, 10h of simulation time per iteration) took 4.1h of computations. In all considered cases I observed the

optimizer converging to an optimal control circuit.

4.4 Discussion

The interplay between pathways and the cellular growth rate, consumption of native metabolite precursors, and the accumulation of toxic pathway intermediates all affect production phenotypes. My work contributes to the emerging area of hybrid metabolic modelling that attempts to expand the predictive power of traditional approaches and incorporate a broader range of cellular processes relevant for pathway engineering. Recent examples include the use of machine learning to augment the predictive power of GEMs (Faure et al., 2023; Hasibi et al., 2024) and improve genome-scale kinetic modelling (Choudhury et al., 2022).

Our approach includes several key innovations which give it advantages over previous modelling approaches in the literature. Several works have already proposed extensions to genome-scale metabolic modelling (GEM) to include temporal dynamics, including various extensions of Flux Balance Analysis (FBA) (Jeanne et al., 2018; Reimers et al., 2017; Waldherr et al., 2015; Yang et al., 2019b), but these methods are not able to account for metabolite or enzyme dynamics. In the synthetic biology literature, dynamic models for host-circuit interactions have been subject of much attention in the past decade, including several approaches to model competition for shared cellular resources (Gorochowski et al., 2016; Liao et al., 2017; Nikolados et al., 2019) or the impact of growth feedback on circuit function (Melendez-Alvarez et al., 2021). Yet so far these approaches do not incorporate metabolic dynamics and are limited to predictions on gene expression alone. My approach can predict intracellular metabolite trajectories, unlike dFBA or other hybrid modelling approaches considered in literature which can only include extracellular ones (Espinel-Ríos and Avalos, 2024a; Gilbert et al., 2019). In addition, my approach predicts enzyme concentrations and includes transcriptional control of enzyme expression as a multiscale feedback circuit in the ODE model. To my knowledge, no other computational approaches in the literature can produce such predictions.

The simulator can be employed to explore the impact of pathway dynamics on production phenotypes, for example by studying pathway bottlenecks due to metabolite accumulation, or the onset of toxicity or stress responses upon enzyme overexpression. This may also provide insights on dynamic control strategies that can potentially mitigate the impact host-pathway interactions and enable the identification of control architectures with improved performance (Liu et al., 2018). Such control strategies have been shown to help manage the cross-talk between gene circuits and their host (Ceroni et al., 2018; Stone et al., 2024), and they may also provide benefits in pathway engineering.

Another promising application of this work is the study of metabolic burden (Snoeck et al., 2024). The synthetic biology literature has so far focused mostly on expression burden and its impact on cell physiology (Darlington et al., 2018; Grob et al., 2021; Nikolados et al., 2019; Weiße et al., 2015). In a seminal work, Borkowski and colleagues (Borkowski et al., 2018) employed a cell free platform to show that metabolic burden, as quantified by a decreased pathway titer, could not be explained solely by the increased competition for gene expression resources. This hints to a metabolic source of burden that is challenging to define and quantify, which is an area of research where my approach can help to draw new hypotheses on the sources and impact of metabolic burden.

Moreover, my method can serve as a tool for genome-scale knockout screens to identify genomes with improved production phenotypes. While host-pathway interactions are normally seen as deleterious for production, it is well known that genetic knockouts can cause unexpected effects on cell physiology; for example, a large-scale study showed that a small fraction of double yeast knockouts can improve fitness (Costanzo et al., 2016), while multiplexed CRISPR screens have found combinations of knockouts with improved metabolic phenotypes (Cachera et al., 2023b; Lian et al., 2017). Such findings pave the way for the use of such synergistic genetic interactions as a novel strategy to improve production. My simulation approach could be thus employed in combination with well-adopted strain optimization tools such as OptKnock or OptForce (Burgard et al., 2003; Ranganathan et al., 2010) to identify genetic edits with improved pathway dy-

namics.

4.5 Limitations and future work

A key enabler in my approach was the use of machine learning models to replace costly FBA calculations. This led to substantial gains in simulation speed that would have otherwise made the FBA-ODE integration too slow for practical use. However, there is additional room for further improvements to computational efficiency. The main bottleneck is the generation of data for training the surrogate machine learning models, which must be instantiated separately for every modification of the GEM, including knockouts and changes to media conditions. A transfer learning approach that selects minimal training data computed on modified GEMs could potentially speed up this process (Gherman et al., 2023). Previous work has used iterative active learning approaches to train surrogate models with fewer expensive computational simulations (Lye et al., 2021) or used iterative parallel computing across multiple cores to speed up the training data generation (Balaprakash et al., 2013).

Another drawback is that model integration must be done in a pathway-specific fashion. I have illustrated how to define the boundary fluxes and modify the kinetic model so that the FBA steady state constraints are respected. In general, however, such model modifications are pathway-dependent and thus may require substantial user input. In Appendix C I have enumerated some representative pathway topologies to ease the application of my strategy across other pathways with more complex branch point topologies than the considered case studies. In cases where engineered strains or deletion mutants cannot maximize their growth rate as in wild-type (Patil et al., 2005), alternate approaches are used to determine the growth rate, including MOMA (Segre et al., 2002) and Energy Balance Analysis (EBA) (Beard et al., 2002). Future work could incorporate these methods, which are commonly used for gene deletion screens, into the simulator.

The simulator is currently implemented only in systems with engineered path-

ways branching from native metabolism. The steady-state constraint imposed by FBA at the model boundary requires that all pathway fluxes can be summed as a single outward flux from metabolism. Loops or cycles which direct flux from the ODE model back into native metabolism (modelled by the GEM) would require the ODE model to conform to the steady-state constraint, which it cannot while still including nonlinear metabolite dynamics. As a result, the simulator as currently designed must describe pathways which branch off metabolism at a single point and are not included in the GEM. Future work could focus on possible solutions to this caveat, including removing reactions from the GEM or modifying reaction connectivity.

4.6 Conclusion

This chapter introduces a novel modelling approach which combines dynamic ODE models with genome-scale FBA models to simulate nonlinear interactions between host metabolism and an engineered pathway. By iterating between short ODE integrations and FBA optimizations, I pass information about native metabolism from the genome-scale model to the differential equation, and constrain metabolic states based on dynamic pathway fluxes. I demonstrate the effectiveness of this approach through two case studies of glucaric acid and beta-carotene production in *E. coli*. The glucaric acid pathway branches directly off of central carbon metabolism and thus production is directly linked to a sustained drop in growth rate, whereas the beta-carotene pathway is farther downstream and affects growth via the consumption of key precursors during product production. I faced several challenges when implementing my new simulator method. First, the large number of computationally expensive FBA iterations required the training of a surrogate model to reduce simulation runtime. These surrogate models present one path forward for the integration of machine learning with mechanistic models. Second, I had to perform pathway-specific balancing of fluxes present in both the ODE and GEM to avoid violating FBA's steady-state constraint. Finally, I had to select appropriate initial concentrations for the

ODE, which required the construction of a Bayesian warm-up routine. I confirmed the validity of my simulation results by examining results across various carbon sources. I also performed a genome-wide screen of all metabolic gene deletions, identifying which ones resulted in changes to product production dynamics. Finally, I explored the applicability of my approach to host-aware metabolic circuit design via large-scale random parameter sampling and Bayesian optimization of control circuit parameters. This novel method presents the opportunity to explore metabolic burden and the complex interactions between a heterologous pathway and native metabolism. Future work could improve the machine learning surrogates using transfer or active learning and develop a less pathway-specific method for model integration.

Chapter 5

Prediction of gene deletion phenotypes from high-dimensional metabolic spaces

The prediction of mutant phenotypes is a key problem in industrial biotechnology, drug discovery, precision medicine, and numerous other fields. Gene deletions, in particular, can fundamentally impair cell physiology and lead to cell death. For example, identifying lethal deletions is key for new cancer therapies (Chang et al., 2021) or antimicrobial treatments that bypass drug resistance (Rosconi et al., 2022). In biotechnology, non-lethal deletions are a powerful strategy to redirect chemical flux toward production of high-value compounds for the food, energy, and pharmaceutical sectors, using genetically engineered cells as an alternative to petrochemicals (Rancati et al., 2018). In this chapter, I show that deletions can alter the shape of the metabolic flux space, and these changes can be learned using supervised learning algorithms. Flux Cone Learning uses Monte Carlo sampling of genome-scale metabolic models in tandem with supervised learning of fitness scores from deletion screens. I demonstrate unparalleled predictive accuracy for metabolic gene essentiality in organisms of varied complexity (*Escherichia coli*, *Saccharomyces cerevisiae*, Chinese Hamster Ovary cells). To demonstrate the method's robustness, I reduce the training set size and apply it to less well-curated models of *E. coli*. I also investigate which reactions given to the machine learning

model as training features are most predictive of essentiality. Finally, I present the first predictive model for small molecule production from deletion screening data. This approach lays the groundwork for the development of predictive algorithms for a range of cellular phenotypes.

5.1 Background and motivation

5.1.1 Mutational phenotype prediction

Mutations are changes to the genetic sequence which encodes proteins. Changes to this code, whether via substitution, deletion, or insertion, can cause changes to overall cellular phenotypes, ranging from adaptive improvements to fitness to lethal mutations that result in cell death. Multiple mutations can result in similar phenotypes, but a particular mutation's effect on overall fitness can be difficult to intuit from the gene's function, even if it is known (Dowell et al., 2010). This is particularly true for metabolic gene mutations, where a loss-of-function to one particular enzyme, for example, can have significant downstream effects on other interconnected reactions in metabolism. Understanding how genetic deletions affect cellular phenotypes is a core problem in fields across biotechnology and medicine. In biotechnology, the ability to predict and select deletions that yield industrially relevant phenotypes is essential for optimizing microbial cell factories, enabling more efficient bioprocesses and increasing product titers (Fernández-Cabezón et al., 2019). However, identifying deletions without extensive trial-and-error experimentation remains a significant challenge.

In precision medicine, decoding the phenotypic effects of rare mutations is crucial for advancing targeted therapies. Such insights can provide valuable information on the potential efficacy and toxicity of targeting that gene product for treating more common human diseases (Dugger et al., 2018). For example, in oncology, discovering tumour-specific deletions which in combination with a cancer therapeutic cause lethality is key to drug discovery (O'Neil et al., 2017). Despite its importance, our ability to systematically map mutations to phenotypic outcomes remains limited, underscoring the need for innovative computational and

experimental strategies to bridge this gap.

5.1.2 High-throughput screening of gene knockouts

Recent developments in high-throughput screening data have opened new avenues to predicting phenotype from genotype, especially when combined with advances in machine learning (Borkenhagen et al., 2021; Danilevicz et al., 2022; Nikolados et al., 2022). Since the 1990s, advances in automation, genetic engineering tools such as CRISPR/Cas9 (Hart et al., 2017) and detection technologies such as fluorescence activated cell sorting (FACS) (Lanier, 2014) have allowed for screens of large numbers of mutations, including full-genome screens (Ipsen et al., 2022; Yeung et al., 2019). For instance, CRISPR/Cas9 has accelerated functional genomics by facilitating precise gene editing, allowing researchers to conduct genome-wide knockout screens to identify gene functions and interactions. Similarly, FACS technology has improved the sorting and analysis of cells based on specific fluorescent markers, streamlining the identification of phenotypic changes resulting from genetic modifications. The case of single-gene metabolic deletions, where an entire gene is removed or its product rendered non-functional by a deletion mutation, is a common screening task with multiple studies producing genome-wide results for several tasks. I next discuss two of these tasks addressed by my method: gene essentiality and metabolite production prediction.

Gene essentiality prediction

Gene essentiality prediction involves identifying genes that are critical for an organism's survival and reproduction. A gene is deemed essential if its deletion leads to cell death. Essential genes are required for core biological processes, including DNA replication, transcription, translation, and cellular metabolism (Dubois-Mignon and Monget, 2022). It's important to note that gene essentiality is not an absolute attribute; rather, it is highly context-dependent. The essential nature of a gene can vary based on environmental conditions, genetic background, and developmental stages. For instance, a gene that is essential under specific environmental conditions may be non-essential under others (Larrimore and Ran-

cati, 2019). Even defining essentiality in multicellular organisms can be difficult, as certain genes may be required for the organism's overall growth and fitness while others may be lethal at the individual cellular level (Cacheiro and Smedley, 2023; Rancati et al., 2018).

Identifying essential genes is pivotal in functional genomics, as it provides insights into the minimal genetic requirements for life and informs the development of therapeutic strategies targeting essential genes in pathogens or cancer cells (Patel et al., 2017). Additionally, studying essential genes across different species enhances our comprehension of evolutionary conservation and divergence in gene function. As a result, substantial previous work has been done to develop computational methods of predicting gene essentiality. There are several different approaches in this field, ranging from sequence-based models (Aromolaran et al., 2021; Guo et al., 2021) to those analysing biological networks (Freischem et al., 2022). However, the gold standard for metabolic genes is FBA, which is particularly effective at predicting gene essentiality in microbes but less so in higher-order organisms (Bordbar et al., 2014; Lin et al., 2025; Segre et al., 2002).

Alternate methods for essentiality prediction include gene Minimal Cut Sets, non-metabolic network-based methods, and sequence models. gMCS methods identify combinations of gene deletions that eliminate specific cellular functions (Apaolaza et al., 2017, 2019). This approach has been particularly valuable for predicting synthetic lethality in cancer, where the simultaneous targeting of multiple pathways can selectively kill cancer cells while sparing normal cells (Olaverri-Mendizabal et al., 2024). Non-metabolic network-based methods leverage protein-protein interaction (PPI) networks under the assumption that essential genes often encode highly connected proteins or occupy critical positions in cellular networks (Hahn and Kern, 2005; Joy et al., 2005). Centrality measures such as degree, betweenness, and eigenvector centrality have been widely used to rank protein importance. However, these approaches are inherently limited to proteins present in the PPI network and suffer from the incomplete and biased nature of current interactome data (Li et al., 2016). Recent efforts have attempted to overcome these limitations by integrating multiple network types, including

metabolic, regulatory, and signalling networks, to create more comprehensive cellular network models (Kim et al., 2016). More recently, sequence-based models have emerged alternatives that can predict gene essentiality across all genes without requiring prior network knowledge (Hasan and Lonardi, 2020; Zhang et al., 2020b). These deep learning approaches typically employ convolutional neural networks or transformer architectures to extract features directly from DNA or protein sequences (Kang et al., 2025). Despite these advances, sequence-based methods are generally are tested on all genes, including significantly easier-to-predict structural proteins, which means their performance on metabolic genes may be significantly lower.

Prediction of metabolite production

Predicting the production of target compounds in metabolic engineering can accelerate the design of optimized microbial strains (Long et al., 2015). Traditional approaches often rely on GEMs to identify potential genetic modifications which could improve product titer. However, there have been increased attempts to apply machine learning (ML) strategies to predict beneficial gene deletions (Kim et al., 2020). ML techniques can analyse large-scale -omics datasets to uncover complex, non-linear relationships between genetic modifications and metabolic outputs. For instance, supervised learning algorithms have been employed to predict the efficacy of specific gene knockouts or overexpressions on product formation, thereby guiding strain design decisions (Cheng et al., 2023; Lawson et al., 2021). ML models can integrate diverse data types, such as transcriptomic, proteomic, and metabolomic profiles. However, the primary challenge of ML methods is how data-hungry they are; they typically require large, high-quality datasets to accurately learn predictive patterns, and acquiring such comprehensive experimental data in metabolic engineering contexts can be resource-intensive and time-consuming. Some recent work has attempted to address this problem by creating hybrid models that combine mechanistic insights from GEMs with data-driven ML predictions (Zhang et al., 2020a). These models leverage the strengths of both approaches, using ML to refine and inform the constraints and

parameters within GEMs, leading to more reliable predictions of metabolic behaviour under genetic perturbations. However, currently there are no algorithms capable of predicting the impact of metabolic gene deletions on non-metabolic phenotypes from GEMs.

5.1.3 Flux balance analysis, optimality, and alternative methods

In Section 2.2.2, I introduced Flux Balance Analysis (FBA), a computational method that predicts metabolic phenotypes by combining genome-scale metabolic models (GEM) with an optimality principle (Orth et al., 2010). This technique can model many metabolic tasks such as growth capabilities in various substrates (Monk et al., 2017), cell-specific auxotrophies (Wang et al., 2018), or responses to drug interventions (Kim et al., 2011). FBA operates under an optimality assumption; that is, it uses linear optimization to find a set of fluxes that maximizes an objective, usually a reaction in the GEM that represents the biomass accumulation rate or growth rate. For organisms like wild-type *E. coli*, the assumption that the organism does in fact attempt to maximize its growth rate and that that growth rate can be specified in terms of other known metabolic reactions present in the GEM is a relatively accurate one, and FBA achieves high predictive accuracy (Bordbar et al., 2014). However, when FBA is applied to higher-order organisms where the optimality objective is unknown, its performance drops (Lin et al., 2025; Segre et al., 2002). For models of mammalian cells, including human cell lines (Agren et al., 2014; Swainston et al., 2016), what the optimality objective would even be is quite unclear, and different cell types, for example, are likely maximizing very different fitness metrics in the context of multicellular tissues.

Furthermore, even a successful FBA run only generates a single point in the flux cone. Additionally, FBA can only predict metabolic phenotypes encoded in the genome-scale model. There are several alternatives to FBA including Flux Variability Analysis (FVA) and Minimization of Metabolic Adjustment (MOMA).

FVA extends FBA by calculating the possible range for each metabolic flux within the constraints of the model, thereby identifying which fluxes are constrained and which can vary over large ranges (Gudmundsson and Thiele, 2010). MOMA is another alternative that predicts the metabolic state of a mutant organism by assuming that, following a genetic perturbation, the organism's metabolism will adjust minimally from its wild-type state (Segre et al., 2002). MOMA employs quadratic programming to find a flux distribution in the mutant that is closest to the wild-type flux distribution while satisfying the new constraints imposed by the mutation. This method has been shown to more accurately predict the behaviour of perturbed metabolic networks, as demonstrated in studies where MOMA's predictions correlated well with experimental data from *E. coli* mutants. However, both MOMA and FVA have limitations. FVA only determines the maximum and minimum possible flux values for each reaction rather than their distribution. MOMA assumes that genetic mutants minimize their deviation from a wild-type state, which may not always be an accurate assumption. These limitations make it challenging to fully capture the range of possible metabolic states and how they are distributed within the solution space. To address this, flux sampling generates a distribution of feasible fluxes without relying on strict optimization criteria or assumptions about mutants' deviation from wild-type optimality.

5.1.4 Flux sampling and genome scale models across the kingdom of life

Flux sampling, an alternative to optimization-based techniques, was introduced in Section 2.2.3. In this chapter, I sample several different GEMs from various organisms of increasing complexity (see Figure 5.1). The number of genes and reactions in a GEM varies based on both the complexity of the organism and the quality of model curation. The programmatically generated SEED models shown in Figure 2.4, for example, are smaller than the best manually curated bacterial models (iML1515 of *E. coli*), even for species with a larger number of

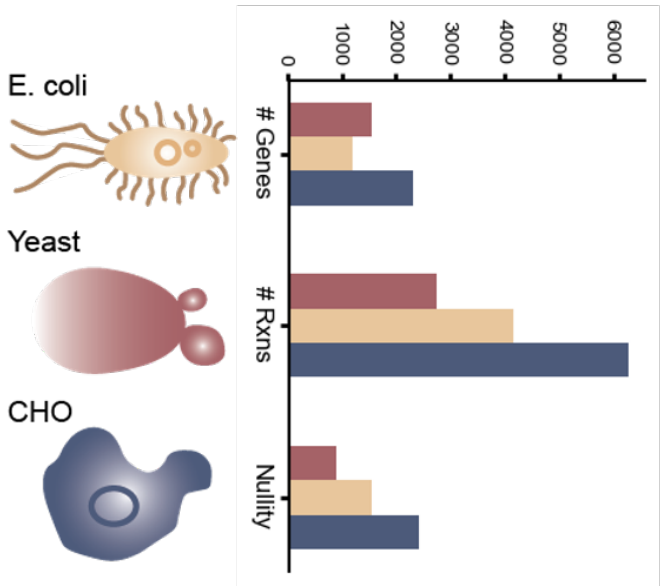


Figure 5.1: Dimensionality of genome-scale models across species.

genes. More complex organisms like the yeast *Saccharomyces cerevisiae* (Yeast9) and Chinese Hamster Ovary cells (iCHO2291) have much higher stoichiometric nullities of over 1,000 dimensions (see Figure 2.4). These models are much slower to sample due to their higher dimensionality. This computational cost becomes increasingly substantial when many modified versions of a GEM must be sampled, such as in the case of various mutant versions of an organism. I list the GEMs sampled in this chapter in Table 5.1: Four *E. coli* models and the state-of-the-art yeast and CHO models.

GEM	Number of genes	Number of reactions	Stoichiometric nullity
iML1515	1516	2712	867
iJO1366	1367	2583	817
iAF1260	1261	2382	752
ijr904	904	1075	332
Yeast9	1162	4130	1539
iCHO2291	2291	6236	2390

Table 5.1: Summary of genome-scale metabolic models employed in this chapter.

5.2 Predicting fitness from the shape of the flux cone

In this section, I describe Flux Cone Learning, a versatile machine learning strategy for predicting deletion phenotypes from the shape of the metabolic space. Flux Cone Learning utilizes mechanistic information encoded in a genome-scale metabolic model to produce a large corpus of training data for each deletion. These data can then be paired with an experimental fitness readout for any phenotype of interest which correlates with metabolic changes and then employed for training predictive models with supervised learning.

5.2.1 Learning the shape of the flux cone

Flux Cone Learning relies on the observation that the high-dimensional cone formed by the set of all reactions in a genome-scale model (see Section 2.2.1) has a shape that can be learned by machine learning algorithms. I hypothesized that the shape differences between cones could be captured from random samples taken by a flux sampler. To test this hypothesis and check if such learned representations could preserve biologically relevant structure in the data, I decided to apply an unsupervised learning technique we developed in Cain et al., 2024. I first sampled five metabolically diverse pathogens (*Bordetella pertussis*, *Pseudomonas aeruginosa*, *Helicobacter pylori*, *Mycobacterium tuberculosis*, and *Streptococcus pneumoniae*) from programatically generated GEMs to avoid confounders introduced by variations in model quality (Devoid et al., 2013). I then trained a variational autoencoder (VAE) (Kingma, Welling, et al., 2019) based on two fully-connected neural networks to compute low-dimensional representations of each species cone, using a large set of Monte Carlo samples of the $N = 494$ metabolic reactions shared across the five species and removing species-specific reactions. The final embedding dimension $D = 8$ was reached via 5 linearly decreasing ($D = 494$, $D = 397$, $D = 300$, $D = 203$, $D = 107$, $D = 8$) fully connected layers and the VAE was trained with PyTorch for 100 epochs with default hyperparameters using the Adam optimizer; t-SNE embeddings were employed for

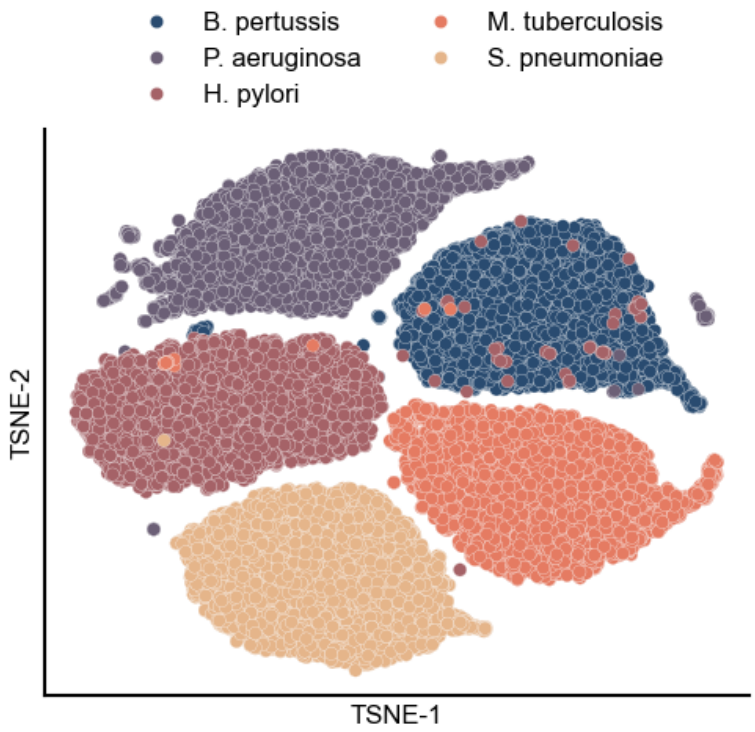


Figure 5.2: Variational autoencoder compression of five bacterial pathogens. A variational autoencoder (VAE) was trained on the common $N = 494$ reactions shared across all species. The final embedding dimension D =8 was reached via 5 linearly decreasing ($D = 494$, $D = 397$, $D = 300$, $D = 203$, $D = 107$, $D = 8$) fully connected layers and the VAE was trained with PyTorch for 100 epochs with default hyperparameters using the Adam optimizer; t-SNE embeddings were employed for 2-dimensional visualization.

2-dimensional visualization (see Figure 5.2). The embeddings capture the separation of metabolic spaces across species. While the learned representation displays a small number of outliers, overall I found that all 5 pathogens retain distinct clustering despite the VAE only being trained on non species-specific reactions. This suggests that the cone geometry can be learned from Monte Carlo samples, and offers a path toward the construction of metabolic foundation models across many species and genomic perturbations.

However, these results are across five very different species distributed across the tree of life. To better understand if lower-dimensional manifolds captured biologically relevant information in a single species, I sampled the iML1515 genome-scale model of *E. coli* $N = 5000$ times. The resulting data was normalized using zero mean and unit variance. I performed PCA, a dimensionality reduction technique, to visualize the data (Maćkiewicz and Ratajczak, 1993). The first two

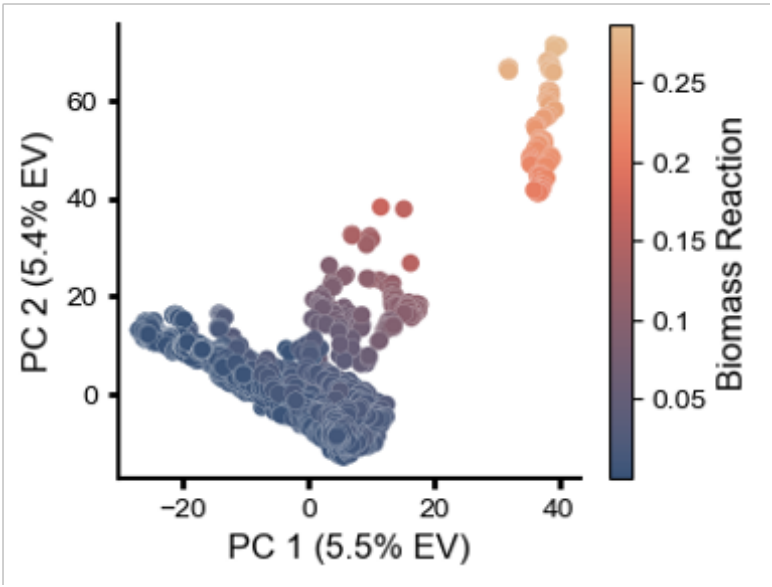


Figure 5.3: PCA representation of flux cone of wild type *E. coli* metabolism.

principal components, plotted on Figure 5.3, are the two directions where the data varies the most.

In this analysis, the first two principal components (PCs) explained 5.5% and 5.4% of the variance, respectively. To capture 95% of the total variance, 621 PCs were needed. Given the original feature set of $R = 2712$ reactions, this represents a 77% reduction in dimensionality while preserving most of the variance. Biomass reactions were excluded from the PCA (see Table 5.3), but in Figure 5.3, I coloured the first two PCs by the value of the forward biomass reaction. A clear trend emerges: high-growth samples cluster in the top right of the plot, while low-growth samples (the majority) cluster in the bottom left. This suggests that biologically relevant features like growth rate can be captured by a relatively low-dimensional subset of the full reaction space, even when sampled randomly from the flux cone.

These results support the core hypothesis of Flux Cone Learning: that the structure of high-dimensional metabolic spaces can be effectively learned and compressed into lower-dimensional representations while preserving biologically meaningful signals. Across both inter-species and intra-species analyses, unsupervised learning methods (VAEs and PCA) revealed consistent organization in the sampled flux spaces, reflecting traits such as species and growth rate. The

ability to extract this structure—even from random samples and in the absence of species-specific features—demonstrates the robustness of cone geometry as a foundation for data-driven exploration of metabolism. Having established that biologically relevant information is embedded within these learned representations, I next explore how this structure can be leveraged for predictive tasks, including supervised classification and regression across diverse metabolic phenotypes.

5.2.2 Framework for Flux Cone Learning

Flux Cone Learning utilizes sampling data generated with a random walk on a deletion-specific genome-scale metabolic model (GEM). In a GEM, a gene deletion is modelled by zeroing out the bounds on a reaction v_i:

$$V_i^{\min} \leq v_i \leq V_i^{\max}, \tag{5.1}$$

where $(V_i^{\min}, V_i^{\max})$ are flux bounds that can be used to model gene deletions through a Gene-Protein-Reaction (GPR) map. Upon deletion of gene g_j, the GPR determines which flux bounds need to be zeroed out in the GEM, i.e. by setting $V_i^{\min} = V_i^{max} = 0$. Changing these bounds also changes the shape of the flux cone. A single gene deletion can affect multiple reaction fluxes in the GEM, and conversely, a single reaction may be influenced by the deletion of multiple genes. Despite this complexity, gene deletions represent a relatively minor perturbation to the flux cone compared to the substantial differences observed across species, cell types (Cain et al., 2024), or growth media. Therefore, I anticipated that detecting such subtle changes would be more challenging.

The steps of the pipeline are shown in Figure 5.4. First, a wild type GEM is modified with a gene deletion by setting the corresponding reaction bounds to zero. The high-dimensional flux cone of the deletion GEM is sampled using a random walk sampler; in my implementations I opted for OptGPSampler (Megchelenbrink et al., 2014), a fast Monte Carlo method that aims to uniformly sample the flux cone. The sampler first transforms the problem into a convex optimization problem in logarithmic space using geometric programming, then

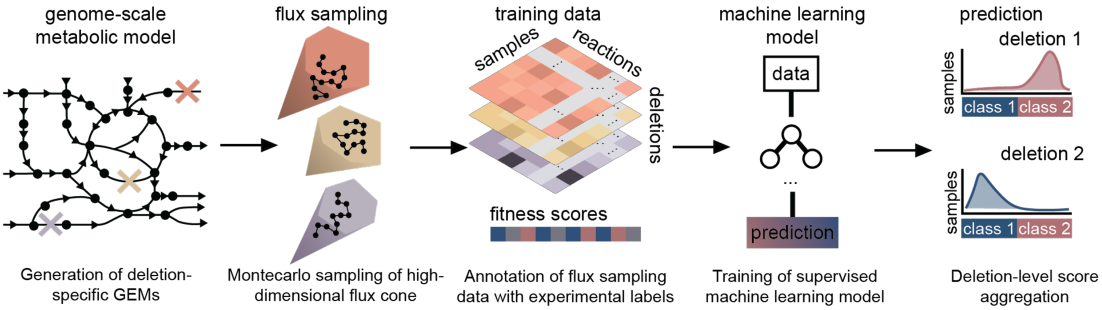


Figure 5.4: Flux Cone Learning of metabolic deletion phenotypes.

employs a hit-and-run algorithm to sample the interior of the cone. The feature matrix for model training has $k \times q$ rows and n columns, where k is the number of gene deletions, q is the number of flux samples per deletion cone, and n is the number of reactions in the GEM. This approach leads to large datasets; for example, in the case of the iML1515 model of *Escherichia coli* (Monk et al., 2017), acquiring 100 Monte Carlo samples for the 2,712 reactions and every gene deletion leads to a dataset over 3Gb in single-precision floating-point format.

A supervised machine learning model is then trained on the flux samples alongside measured phenotypic fitness labels for each deletion; all samples in a deletion cone get assigned the same label. FCL does not prescribe the choice of machine learning model and can be applied to both regression and classification tasks. The fitness score can be either discrete or continuous depending on the fitness readout under study. The model predictions are made at level of flux samples, i.e. the model is trained on single-sample flux vectors. Therefore, every sample from each deletion GEM is assigned an individual predicted score, and the distribution of these scores is finally averaged to obtain a gene-level prediction. Flux Cone Learning can deliver high predictive accuracy because it is trained to learn correlations between the geometry of the flux cone and the resulting phenotype. This method improves the classification performance, in some cases significantly, because it allows Flux Cone Learning to weigh samples across the cone, including from regions especially perturbed by the deletion of interest. I next provide a more precise methodological overview of the steps of Flux Cone Learning, including the details necessary to reproduce all experiments in Sections 5.3, 5.4, and 5.5.

A. Generation of deletion-specific GEMs and flux sampling

I chose to use OptGPSampler for all sampling runs in this chapter; I ran the sample function on all single-gene deletions in four *Escherichia coli* models, the Yeast9 model for *Saccharomyces cerevisiae*, and the iCHO2291 model for Chinese Hamster Ovary cells (see Table 5.1). For training supervised machine learning models, sampling data were normalized to zero mean and unit variance. There were a small number of deletions in each GEM where the sampling failed to converge; these were not included in training or testing. A summary of the GEMs sampled is in Table 5.1 and details of sampling runs are found in Table 5.2. For example, in the Yeast9 model I sampled 1,159 single-gene deletions with a step size of k =5,000 for a sampling density of N =124 samples/cone, leading to a total of 143,716 samples with D =4,130 fluxes each (total data size 4.43Gb). The resulting NPZ file with Float32 precision is 4.43Gb; compression is of limited use here as the flux sampling data matrix is highly non-sparse.

GSM	Sampled KOs	Total S	Density per cone	Step size	Subsampling	Size of NPZ file
iML1515	1502	150313	101	100	50	3.06Gb
iJO1366	1318	163680	124	5000	1	3.17Gb
iAF1260	1214	150784	124	5000	1	2.69Gb
ijr904	866	107508	124	5000	1	0.89Gb
Yeast9	1121	143716	124	1	1	4.43Gb
iCHO2291	2290	291028	127	1	1	13.53Gb

Table 5.2: Summary of model sampling details. All models were sampled with the same step size, except iML1515 which employed a smaller step size to generate more samples and the subsampled.

For all models except *Escherichia coli*, I sampled with a high step size of k =5,000. To ensure robust performance evaluations in the *Escherichia coli* iML1515 model (Figure 5.6), I retrained models many times using different training sets. For computational efficiency and due to large data sizes, after computing an initial large set of samples, using a fine step size of k = 100, I subsampled the data 10 times to have the same number of samples per deletion (N =100 samples/cone). Three smaller *E. coli* models (iAF1260, iJO1366, iJR904) were

employed for the comparison in Figure 5.11. In these models, deletion GEMs were sampled with N =100 samples/cone and k =5,000. To equalize the amount of training features between models, only the deletions present in all models (D =864 reactions) were included in the training and test sets for the models in Figure 5.11. The biomass reactions were removed to ensure the models were learning from the true reaction fluxes, not the biomass reaction used to compute FBA predictions (see Table 5.3).

GSM	Biomass Reactions Removed [IDs]
iML1515	BIOMASS_Ec_iML1515_core_75p37M [2669], BIOMASS_Ec_iML1515_core_75p37M_reverse_35685 [2670]
iJO1366	BIOMASS_Ec_iJO1366_core_53p95M [19], BIOMASS_Ec_iJO1366_core_53p95M_reverse_5c8b1 [14]
iAF1260	BIOMASS_Ec_iAF1260_core_59p81M [926]
ijr904	BIOMASS_Ecoli [269], BIOMASS_Ecoli_reverse_bf7a1 [270]
Total Common Reactions	864

Table 5.3: Biomass reactions removed for *E. coli* models and their common reactions.

B. Annotation with experimental fitness labels

Flux sampling generates a large data set with the number of features equal to the number of reactions and the number of data points equal to the number of samples generated. However, training a supervised machine learning model to predict phenotype requires phenotypic labels. I extracted these labels from the literature; in particular experimental results from high-throughput deletion screens. In Sections 5.1.2 and 5.1.2, I introduced the two prediction tasks for FCL: gene essentiality classification and product production prediction. I will discuss how I obtained the labels for each task in turn.

Gene essentiality is usually measured by the cell growth rate, with the threshold for classification as essential or nonessential being set empirically (Monk et al., 2017). In some cases, the raw growth rate is reported; in others the growth rate relative to wild type. The gene essentiality labels for *E. coli*, *S. cerevisiae*

and CHO cells were obtained from the literature (Monk et al., 2017; Xiong et al., 2021; Zhang et al., 2024). The yeast essentiality labels included non-metabolic genes and were labelled with both gene and ORF labels. Gene names were standardized to their systematic names from the Saccharomyces Genome Database, resulting in $N = 1121$ metabolic gene deletions labelled with essentiality data, sampled, and included in the final dataset for model training. ORFs and gene names were linked using a tool from Yeastract+.

For the production task, I chose to focus on a large deletion screen of *S. cerevisiae* mutants engineered to synthesize betaxanthin (Cachera et al., 2023a), a tyrosine-derived pigment widely employed in the food sector. The screen measured betaxanthin autofluorescence readouts for $N = 811$ yeast deletions averaged across four cultures (see Figure 5.5. While one gene (YBR011C) was also identified as essential in the essentiality dataset, I included it in my analysis as I hypothesized this could be a conditionally essential deletion which can grow in alternative strain and media conditions. The raw mean betaxanthin production fluorescence values varied from 0.28 to 0.61. I normalized the average autofluorescence for each deletion to have zero mean and unit variance. To simplify the production prediction task, I binned the data into three classes. This class split was determined qualitatively from the distribution of autofluorescence. In all case studies, labels were highly class imbalanced, as shown in Table 5.4.

Task	Class imbalance	Class breakdown
E. coli essentiality	83/17	N=1252 non-essential, N=251 essential
Yeast essentiality	86/14	N=964 non-essential, N=157 essential
CHO essentiality	83/17	N=1898 non-essential, N=392 essential
Yeast production	17/67/16	N=138 low, N=545 medium, N=128 high

Table 5.4: Class imbalances for each task.

C. Supervised machine learning on fluxomic data

Next, a supervised machine learning model is trained on the flux samples alongside measured phenotypic fitness labels for each deletion. Flux Cone Learning does not prescribe the choice of machine learning model and can be applied to

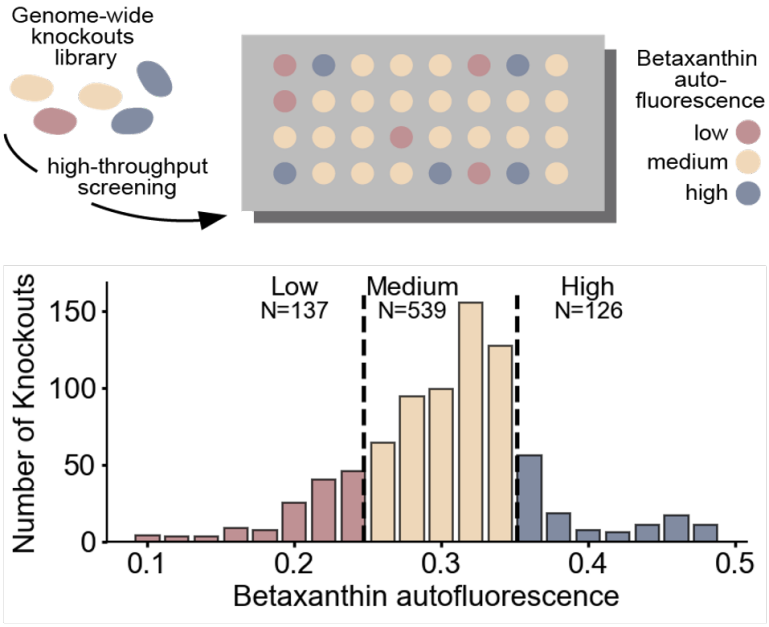


Figure 5.5: High-throughput deletion screening data of *S. cerevisiae* strains engineered to produce betaxanthin (Cachera et al., 2023a).

both regression and classification tasks. I will discuss the exact details of the model(s) trained for each task in the corresponding results sections. All models were trained using the scikit-learn package in Python.

D. Knockout-level score aggregation

The final step of the Flux Cone Learning pipeline is the aggregation of scores across samples into a single deletion-level label. The training procedure is done sample-wise, with each training data point being a single sample vector of fluxes taken at a single point in the cone. Many samples from the same flux cone are passed into the machine learning model. However, at test time the same call must be made for all samples taken from the same cone. I term this the knockout-level call and compute it by averaging the prediction scores across all samples from the same cone. If this average is above the predetermined threshold, the knockout is classified as the positive class and vice versa. This knockout-level score aggregation procedure improves the performance of Flux Cone Learning in some tasks by making the knockout level call robust to outliers from nonperturbed areas of the cone which could be misclassified.

5.3 Accurate prediction of *E. coli* essentiality

5.3.1 Benchmarking against FBA

I first tested FCL as a predictor of gene essentiality in *Escherichia coli*. I chose *E. coli* because it has the best curated GEM in the literature (iML1515, Monk et al., 2017) which mitigates the impact of poor GEM quality on predictive performance. In addition, FBA has a strong performance in wild-type *E. coli* because its assumption of optimized growth is accurate to the bacterium. When tested across different carbon sources, FBA delivers a maximal accuracy of 93.5% correctly predicted genes for *E. coli* growing aerobically in glucose with biomass synthesis as the optimization objective (Monk et al., 2017).

Details of machine learning model

A random forest model classifier was trained using N =1,202 gene deletions (80% training set) with q = 100 samples/cone to be a binary classifier of gene essentiality. The training set was stratified to maintain the class imbalance and the biomass reaction was removed from training features to prevent the model from learning the correlation between biomass and essentiality that supports FBA predictions (see Table 5.3, Figure 5.3). This led to a training dataset with N =120,285 samples with n =2,712 features (see Table 5.2). I chose a random forest classifier as a suitable compromise between model flexibility, which allows it to fit complex functions learned across the high-dimensional training set and interpretability, as every decision tree can be extracted from the model and its structure attributed to specific training features. I fixed the model hyperparameters to standard values: `max_depth` None and `min_samples_split` 2. The random forest was retrained N =5 times with different test sets to confirm that performance was not significantly affected by the composition of the training set. The FBA baseline was obtained using the `single_gene_deletion` function in the CobraPy package (Ebrahim et al., 2013) applied to all genes in the iML1515 model with default biomass objective function, aerobic conditions, and glucose as carbon source. I chose 0.4 1/hr as the cutoff for FBA essentiality predictions.

This was chosen to match the experimental growth rate cutoff employed by the original iML1515 source (Monk et al., 2017), which is 50% of the wild-type growth rate (predicted to be ∼0.8 1/hr by FBA). The naive baseline was compared by predicting all genes as non-essential (majority class). Once trained on the sample level, the prediction score of all samples from a single deletion was averaged; if this score was less than 0.5, the deletion was classified as essential.

Results of benchmarking

Test results in a random set of N =300 genes (20%) outperformed the FBA predictions in accuracy, precision and recall, achieving an average 95% accuracy for all test genes across training repeats (Figure 5.6). I used several different metrics to evaluate the performance of FCL. In a class-imbalanced task such as essentiality prediction, only using standard accuracy (the fraction of correct predictions out of all test set samples) can hide significant discrepancies in class-to-class prediction (Powers, 2020). I also computed the precision, recall, and F1 score. I refer to nonessential genes as the positive class since they are the majority. Thus, precision is the proportion of true nonessential predictions among all deletions predicted as nonessential, while recall is the proportion of actual nonessential genes that were correctly predicted. The F1 score is the harmonic mean of precision and recall, which balances false negatives and false positives (Sokolova and Lapalme, 2009). The difference between the three metrics can be seen most clearly in the naive baseline case, which has an accuracy of 82% despite incorrectly classifying all essential deletions as nonessential. I also plotted the class-level accuracy scores in Figure 5.7. FCL achieved a 1% and 6% improvement in classification of non-essential and essential genes, respectively, as compared to FBA. As FBA is the gold standard for metabolic gene essentiality prediction, the improvements FCL achieves result in the best known predictive performance for gene essentiality in the literature.

I randomly selected a model to create the prediction score distributions in Figure 5.8. The prediction scores for each sample in the test set cone ($N = 100$ samples per deletion) were plotted as a density histogram in one dimension. In-

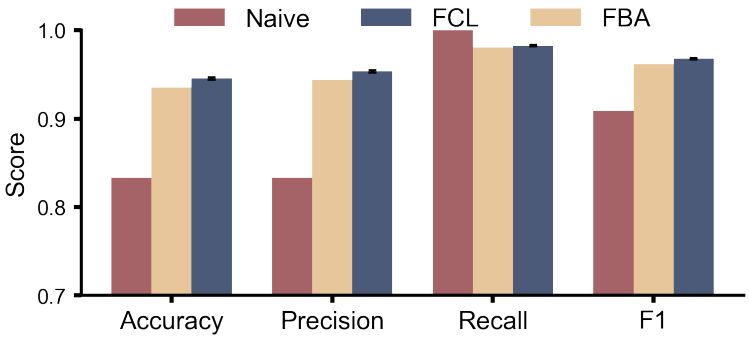


Figure 5.6: Flux Cone Learning (FCL) delivers best results for metabolic gene essentiality prediction in *E. coli*, outperforming the current gold standard predictions from Flux Balance Analysis (FBA). Error bars denote standard error computed across $N = 5$ training repeats using $N = 10$ training subsamples with $q = 100$ samples/cone.

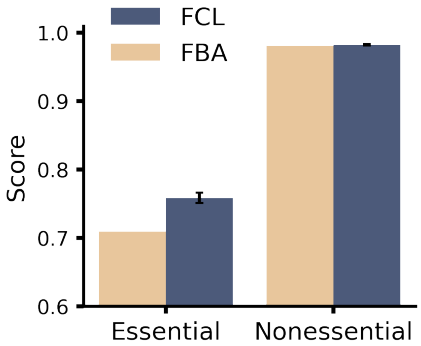


Figure 5.7: Prediction accuracy for essential and non-essential genes in *E. coli*.

spection of the distributions show that a small number of deletions get incorrectly classified as their prediction score distributions have a mean on the wrong side of the thresholds. I examined several of these distributions in more detail (bottom of Figure 5.8) and saw that some (b1812) had much higher variance in their prediction scores than others (b2435), even if the deletion-level prediction was correct. Incorrectly predicted knockouts fell into two categories: first were knockouts (b1252) which had very low variance in prediction scores and a mean score close to zero or one. These misclassifications are cases where the model is "confidently" incorrect, likely due to GEM misspecifications. The other case are knockouts (b2416, b2103) which had a much higher variance in the distribution and have a mean close to 0.5. These deletions are more likely due to the knockout having a smaller or larger than average perturbation to the flux cone, or due to a flux sampling run that did not explore the perturbed areas of the cone sufficiently.

5.3.2 Feature importance and model explainability

I next trained $N = 50$ random forest classifiers on a consistent single subsample, with random held-out test sets. The feature importance scores were extracted from the Random Forest object. A feature importance score quantifies how much each input feature contributes to the model's predictions. The metric used is the average decrease in Gini impurity when each feature is used to split data across all trees in the forest. The input features to the random forest are the set of all non-biomass reactions in iML1515; as a result, more important features are reactions which have a higher impact on essentiality prediction. Interpretability analysis of the 50 repeats revealed that as few as $\sim$100 reactions can explain model predictions (see top of Figure 5.9). There are 40 categories of reaction, but the top 20 most important reactions contains only 5 categories, which are highly enriched for transport and exchange reactions. The variance in feature importance is significantly different between different reactions; this could mean that some reactions are consistently very important (copper exchange, for example) while others are not required for good predictive accuracy.

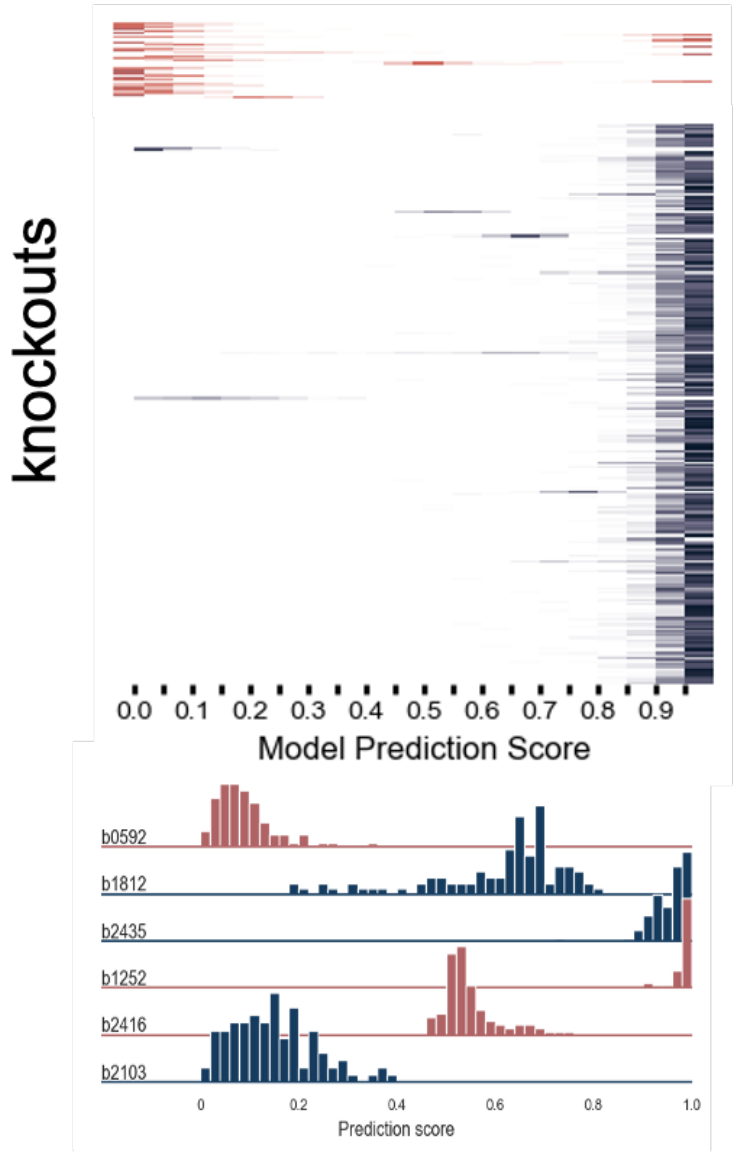


Figure 5.8: Top: Distribution of sample-level FCL prediction scores across 300 test genes for one representative random forest model. The intensity of the colour correlates with the number of samples in that bin.The red samples are ground truth essential; the blue are nonessential. Bottom: Representative prediction score distributions for correctly and incorrectly predicted genes.

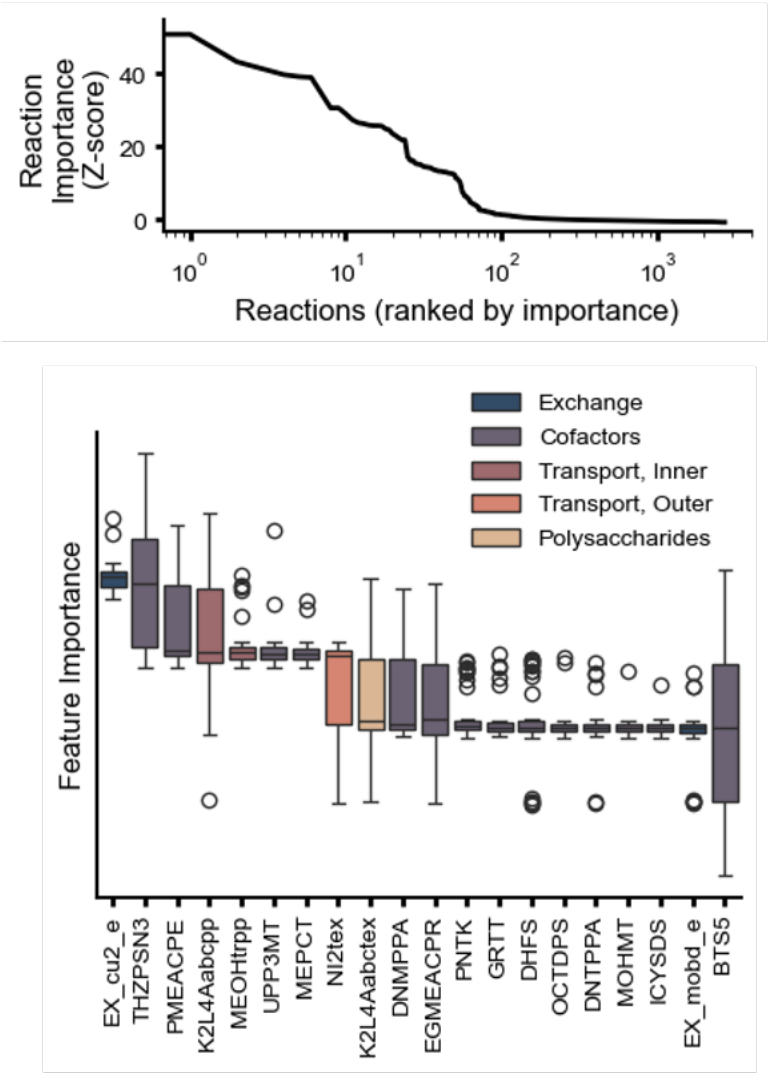


Figure 5.9: Top: reaction feature importance across all genes employed for training, using the random forest feature importance scores. Bottom: Importance of top 20 features across 50 repeats; box plots show mean, IQR, and whiskers are all samples not determined to be outliers.

I removed the biomass reaction from the training features to prevent FCL from learning the exact objective function required for FBA. The feature importance data shows that are other reactions strongly associated with essentiality such as exchange reactions and the random forest model picks up these correlations. However, the point of FCL is that it can predict the correlations between reaction changes and a fitness label like essentiality without a manually constructed objective function. The random forest model can learn that similar reactions to those in the objective function are important. However, the feature importance results in Figure 5.9 do not suggest this as there are several reactions with consistently high feature importance to the RF which have substrates not present in the biomass equation (e.g. THZPSN3).

5.3.3 Degrading model performance

To investigate which factors determine FCL performance, I first retrained the model with less dense cone sampling and fewer gene deletions, both of which reduce the size of the training set. The training set size was reduced in two ways: first, by varying the train/test split, and second by decreasing the number of samples per deletion cone in the training set. I performed experiments for three deletion densities: 100 samples per deletion, 10 samples per deletion, and 1 sample per deletion. I used four test splits: 20% (the baseline), 80%, 95%, and 99%. These splits reduce the number of discrete deletions in the training set from N=1212 to N=15 in the extreme case. The class stratification is preserved across all splits. Sample-level accuracy was used as the comparison metric for all models. I found that the models degraded to the naive baseline accuracy as the number of discrete deletions in the training set was reduced, and degraded more quickly for 1 sample/deletion models (see Figure 5.10). Models trained on as few as 10 samples/cone already matched the current state-of-the-art FBA accuracy. These results indicate that denser or longer sampling runs can improve performance but only to a point; even fairly limited training set sizes are sufficient to achieve better than baseline accuracy.

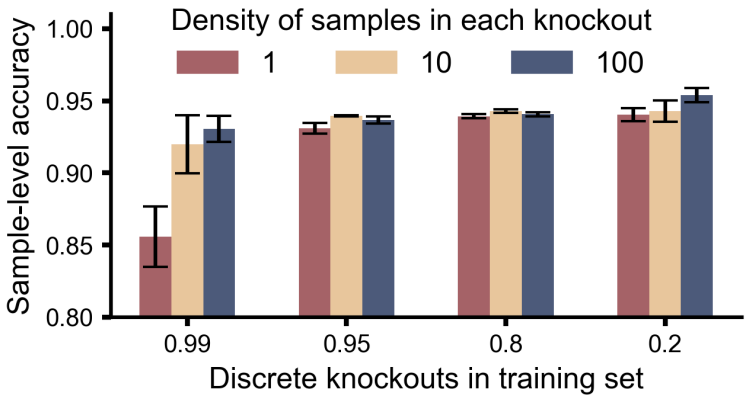


Figure 5.10: Performance of FCL with smaller and less dense training data; error bars are standard error across 5 training repeats with different initializations.

5.3.4 Smaller GSMs of *E. coli*

I additionally retrained FCL with smaller and less complete GEMs for *E. coli*: iAF1260, iJO1366, and iJR904. I trained N=5 random forests on all four *E. coli* GEMs with different 20% class-stratified test sets (Figure 5.11). Two-sided pairwise t-tests were conducted between iML1515 and the other models with a significance level of p=0.05. I found that only the smallest GEM (iJR904) displayed a statistically significant drop in performance when compared to the iML1515 accuracy (p=0.006). These results indicate that improving GEMs through additional reactions and correcting structure via manual curation can improve predictive accuracy. However, there is a limit to the accuracy benefits of adding more genes. The two larger models iAF1260 and iJO1366 approach a similar number of reactions and the stoichiometric nullity of iJO1366 is only 5.7% lower than iML1515 despite having over 10% fewer genes. Model curation improves FBA-only predictions in a similar fashion to FCL; this is also likely due to further manual curation of the GEM. iJR904 (the smallest GEM) has 4% lower essentiality accuracy than iAF1260 for genes in common between the two GEMS (Feist et al., 2007). However, FBA-only performance in smaller genome-scale models is lower than FCL - in iAF1260, for example, FBA achieves 92% accuracy compared to 94% accuracy with FCL.

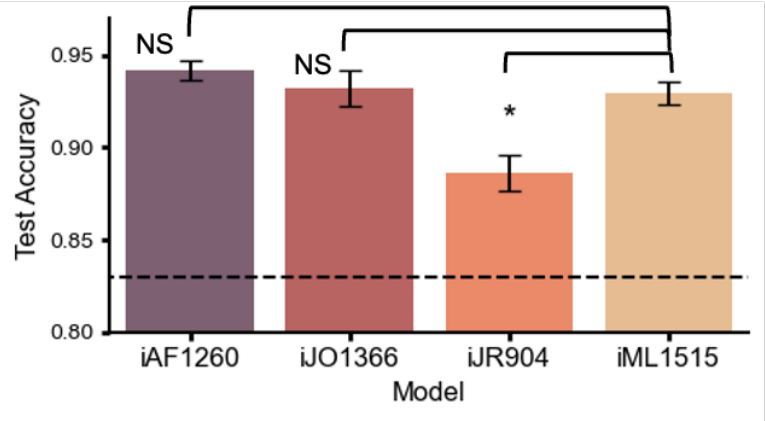


Figure 5.11: Performance of FCL with earlier versions of the *E. coli* genome-scale metabolic model. Test results were computed across N =848 genes shared by the four models (King et al., 2016). Significance was determined at $p < 0.05$ with a one-sided t-test; error bars are standard error across 5 training repeats.

5.3.5 Defining a metabolic distance metric

I next sought to exploit the excellent predictive performance of FCL to define a distance metric between deletions and the wild type strain. To this end, I retrained FCL on all *E. coli* gene deletions (N =1515 genes), and computed the distribution of prediction scores for all flux samples in the wild type and each deletion strain. I then queried the model with flux samples of the wild-type GEM and each deletion cone (100 samples/cone), to produce distributions of prediction scores for the WT and each deletion strain. I scored each strain with the Jensen-Shannon divergence and Hellinger distance between the score distributions of each deletion and the wild type. The Jensen-Shannon Divergence is a smoothed and symmetrical version of the Kullback-Leibler divergence $D(P\|Q)$ defined between two distributions P and Q:

$$JSD(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M), \tag{5.2}$$

where $M = \frac{1}{2}(P+Q)$ is a mixture distribution between P and Q. The Kullback-Leibler divergence is defined as:

$$D(P\|Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{5.3}$$

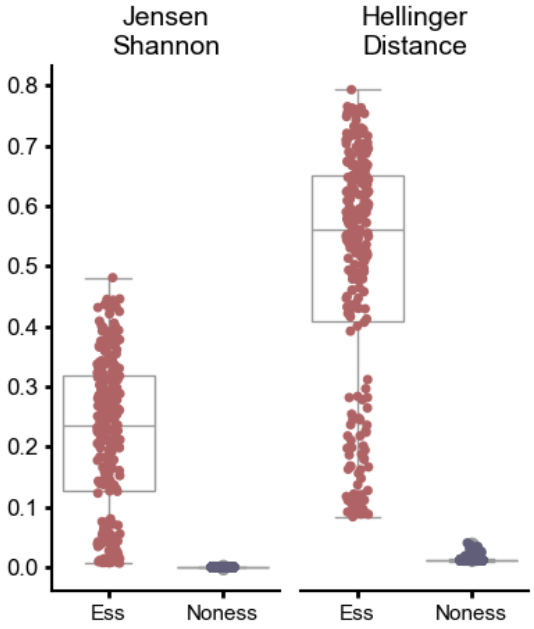


Figure 5.12: Complete classifier sample scoring with Jensen-Shannon divergence and Hellinger distance.

The Hellinger distance between two distributions P and Q is defined as:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2}, \tag{5.4}$$

and is bounded from 0 to 1, with higher values denoting larger differences between the distributions. Based on a one-sided t-test with p-value < 0.05, I found statistically significant differences in score distribution of non-essential and essential genes, which reinforces the conclusion that perturbations to the flux cone are indicative of gene essentiality (see Figure 5.12).

5.4 High performance across more complex organisms

I tested FCL for essentiality prediction in *Saccharomyces cerevisiae* and Chinese Hamster Ovary (CHO) cells, two more complex organisms with well-curated GEMs and essentiality screens (Yeo et al., 2020; Zhang et al., 2024). These models have 52% and 130% more reactions than *E. coli*, respectively, leading to a

higher dimensionality of the flux cone and more features for training (see Figure 5.1 and Table 5.1).

5.4.1 Yeast model training

The Yeast9.0.1 model was downloaded from https://github.com/SysBioChalmers/yeast-GEM as a SBML XML file. The OptGPSampler from CobraPy was used to sample all genes present in the model genes list. 3 genes were not possible to sample due to repeated failures; these were excluded from further analysis. 1159 genes were sampled 100 times with a step size of 5000. For essentiality prediction, a class-stratified 20% of deletions (192 non-essential; 31 essential) was held out as a consistent test set. The remaining 80% of deletions (772 non-essential; 126 essential) were split into 5-fold cross validation sets and a random forest model was trained on each fold. The `max_depth`, `n_estimators`, and `min_samples_split` hyperparameters were tuned using a grid search and the model with the highest average cross validation accuracy was selected and the confusion matrix and ROC curve computed for Figure 5.13. The best `max_depth` value was 30, the best `n_estimators` value (the number of trees) was 300, and the best `min_samples_split` (the minimum number of data points to split a leaf on the random forest) value was 2. The minimum deletion-level accuracy was 87.5%, the maximum was 90.3%. The test set results was computed by running the held-out test set through all 5 fold models and averaging the deletion-level scores across all models. The FBA baseline was computed using the `single_gene_deletion` function in Cobrapy for all genes with glucose as the carbon source and the standard biomass reaction. AUROC metrics for FCL models were computed as an average across folds $\pm$ one standard deviation. The solid black line in Figure 5.13 is the FBA baseline predictions computed for all genes in the GEM.

5.4.2 Chinese Hamster Ovary model training

For the CHO case, models were trained on N =1,832 class-stratified gene deletions with q =127 samples/cone computed from a well-adopted genome-scale model

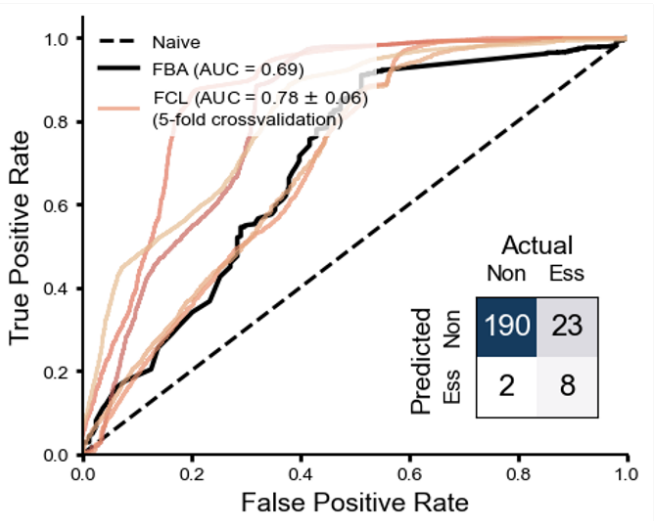


Figure 5.13: Reciever operating characteristic (ROC) curves of FCL model for *Saccharomyces cerevisiae* in 5-fold cross validation.

(Yeo et al., 2020). The large training set data size required training models across 4 CPU nodes of to load all training data into memory. Instead of a random forest, I trained the more memory-efficient HistGradientBoosting classifier on a 5-fold cross validation of the training set. A 20% test set was held out and not included in the cross validation. The hyperparameters `learning_rate`, `max_iter`, and `max_depth` parameters were tuned via grid search and the model with the highest average cross validation accuracy was selected and the confusion matrix and ROC curve computed for Figure 5.14. The `learning_rate` was varied between 0.01 and 0.2, the `max_iter` between 100 and 500, and the `max_depth` set to 5, 10, or None. The best model had a `learning_rate` of 0.05, a `max_iter` of 100, and a `max_depth` of None. The test set results were computed by running the held-out test set through all 5 fold models and averaging the deletion-level scores across all models. The confusion matrix was computed for a class threshold value of 0.5 for each fold and counts were averaged across all 5 folds. The FBA results were computed using the `single_gene_deletion` function in Cobrapy for all deletions and the default carbon source and objective function in the iCHO2291 model.

FCL achieved better receiver operating characteristic (ROC) performance than FBA in both models (Figures 5.13 and 5.14), with improved sensitivity and specificity (11.4% and 14.3% increase in AUROC, respectively). I found

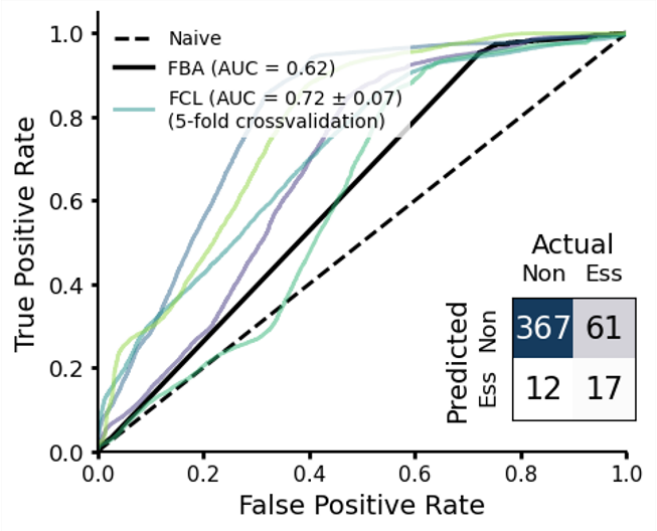


Figure 5.14: Receiver operating characteristic (ROC) curves of FCL model for Chinese Hamster Ovary cells in 5-fold cross validation.

that FCL showed similar prediction errors as FBA, with a tendency to misclassify some essential genes as non-essential, likely due to the class imbalance in the training data (most genes are non-essential). Taken together, the improved performance of FCL on *E. coli*, *S. cerevisiae* and CHO cells strongly suggests that the optimality assumption of FBA is not required to predict metabolic gene essentiality.

5.5 Expanding deletion prediction to non-metabolic phenotypes

To explore the power of FCL for predicting other phenotypes, I focused on small molecule biosynthesis in microbial strains engineered with heterologous pathways (Han et al., 2023). Non-essential deletions can both suppress or boost metabolite production; for example, deletions that disrupt enzymatic co-factor homeostasis are deleterious for product synthesis, while other non-essential deletions can re-direct metabolic flux away from non-essential pathways toward increased production (Rancati et al., 2018). I selected a large deletion screen *S. cerevisiae* mutants engineered to synthesize betaxanthin from the literature (Cachera et al., 2023a). The average autofluorescence was (0,1) normalized (raw range 0.28, 0.61).

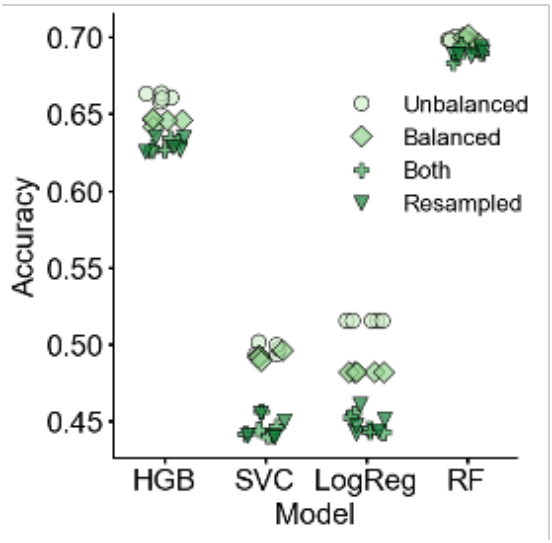


Figure 5.15: Accuracy results for several FCL models with different algorithms for multiclass classification of deletion strains.

I first framed the problem as a regression task, but this proved challenging with the limited number of knockouts at the high and low ends of the autofluorescence distribution. Recognizing that predicting high or low producers is a core task in several applications, I chose to train a three-class classifier by binning the data into three classes of high, medium, and low producers (see Figure 5.5). I set the thresholds qualitatively to label 67% of samples as medium producers (within ∼1 standard deviation from the mean).

I employed FCL to build a 3-class classifier that predicts betaxanthin synthesis using Monte Carlo sampling of the deletion GEMs. Due to the imbalanced data size across classes (17.1%, 67.2% and 15.7%, resp., see Table 5.4), I trialled various model architectures in combination with re-balancing strategies (Figure 5.15). The following model types were trained: HistGradientBoostingClassifier, Linear Support Vector Classifier, Logistic Regression Classifier, Random Forest Classifier (Figure 5.16). I implemented two class balancing techniques to improve the minority class performance: balancing, which weights the class labels to account for the class imbalance, and resampling, which subsamples the majority class to be the same size as the minority classes. The best performing model (random forest) delivered promising accuracy (69.8%). I observed a tendency to underpredict the high-producing deletions due to these being underrepresented

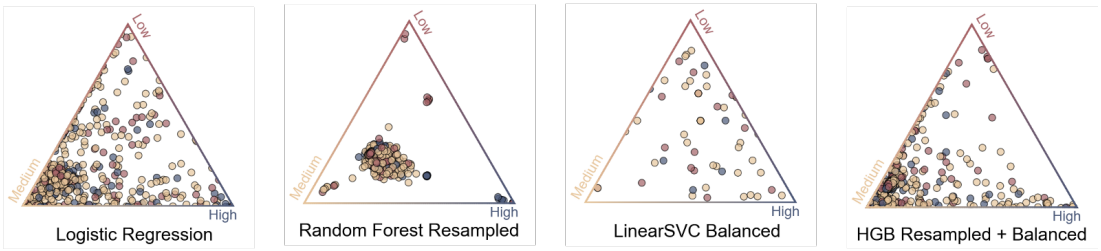


Figure 5.16: Ternary plots of model predictions on the test set for representative models with varying predictive accuracy across the three classes. Vertices represents class prediction with probability one (full confidence), whereas central points deletions predicted to be equally likely to be any of the three classes. Each sample has been colour coded according to their ground truth class labels.

in the training data, though high producer accuracy improvements between 5.5% and 28.3% could be obtained via various class balancing techniques (see Table 5.5).

Model Name	Baseline	Resampled	Balanced	Both	Max % improvement
HistGradient Boosting Classifier	11.4	13.6	14.1	14.6	28.3
Linear SVC	23.8	24.0	27.2	26.8	27.2
Logistic Regression	23.3	26.2	29.5	28.9	14.2
Random Forest Classifier	18.1	11.7	19.1	18.4	5.5

Table 5.5: Accuracy for high producer deletion yeast deletion strains for each class balancing methods in Figure 5.15. Models were assessed across all genes in a held-out, class-stratified 20% test set (N =649 deletions).

To the best of my knowledge, this is the first demonstration that small molecule synthesis can be computationally predicted from deletion screening data, and adds to the growing number of tools to predict metabolite production using various data modalities and computational approaches Djoumbou-Feunang et al., 2019; Schneider et al., 2020. Since FCL relies purely on the wild-type GEM and experimental fitness readouts, it does not require extending the GEM with a heterologous pathway of interest, which can be beneficial in use cases where pathway stoichiometry is not well characterized.

5.6 Discussion

Advances in high-throughput genetic engineering and automated screening have made it possible to conduct studies that were once infeasible, generating large amounts of data. These quantities of data open an opportunity to utilize such data for building predictors of the phenotypic response to genetic mutations, including single-gene deletions. Recent work integrating learning algorithms with genome-scale metabolic models has shown substantial promise for improved predictivity across various tasks (Faure et al., 2023; Gopalakrishnan et al., 2024; Lin et al., 2025; Yang et al., 2019a). In this chapter, I presented Flux Cone Learning, a general strategy to detect correlations between metabolic genotypes and phenotypic readouts. The method combines experimental fitness data with mechanistic knowledge into a machine learning system able to draw phenotypic predictions for a specific gene deletion. I applied Flux Cone Learning to two tasks: gene essentiality prediction and betaxanthin production prediction, though the algorithm can be applied to phenotypes created by many different fitness assays. Although FCL is agnostic to the fitness score employed for training, its effectiveness is limited by the strength of correlations between metabolic activity and the phenotype of interest. In the case of gene essentiality, for example, FCL works well because deletions in pathways that supply key metabolites for growth can strongly impact cell viability. Other phenotypes with weaker or no associations to metabolic activity may require additional modalities of data for accurate prediction.

For the essentiality task, I demonstrated that FCL outperforms the state-of-the-art FBA predictions of metabolic gene essentiality. One limitation of my method is that it requires a training and test set split. In contrast, FBA is a zero-shot predictor because it does not need to be trained on fitness data. Instead, FBA uses a biological optimality assumption (see Section 5.1.3) to draw predictions. For non-engineered microbes, maximal growth rate or biomass synthesis rate are well validated metabolic objectives. However, for most organisms beyond the microbial world as well as engineered organisms, such optimality assumptions are not warranted and there is no consensus on how to define suitable metabolic

objectives for higher-order organisms (Bordbar et al., 2014; Lin et al., 2025).

Various studies have developed computational and theoretical strategies to account for the inherently multiobjective nature of metabolic optimality in biological systems. This complexity arises because organisms often need to balance competing objectives—such as maximizing growth rate, minimizing energy expenditure, and maintaining metabolic robustness—within the constraints of a finite resource pool. Evolutionary pressures can shape phenotypes, even higher-order morphological traits, to lie near low-dimensional trade-off surfaces, thereby enabling the inference of underlying selective pressures from phenotypic data (Shoval et al., 2012). Schuetz et al., 2012 applied this approach in microbes using flux balance analysis (FBA) and C-13 flux data to demonstrate that no single objective function universally describes cellular metabolism. Instead, cells may operate at different points along a Pareto front, balancing tradeoffs between multiple competing objectives. Other work has attempted to reverse-engineer metabolic objectives from empirical -omics data. These methods seek to infer what objective functions organisms are optimizing, rather than presupposing them (Richelle et al., 2021; Zhao et al., 2016). Tradeoff analysis has been used to understand how cells prioritize between conflicting metabolic goals, particularly in pathological contexts (Hausser and Alon, 2020). Lin et al., 2025 introduced novel inference techniques for extracting tradeoff strategies directly from single-cell data, revealing the adaptive strategies cells employ to navigate metabolic constraints under varying environmental pressures.

Yet even in cases where an objective can be identified, however, it is often only one of multiple competing optimization principles. Furthermore, there is little evidence that optimal objectives of the wild-type organism would be preserved upon a gene deletion (Dekel and Alon, 2005; Leiby and Marx, 2014). Mutants are likely to be subject to different evolutionary pressures that shift their genetic programs away from the physiological objectives of the wild-type. By basing its performance on flux sampling data rather than optimization, Flux Cone Learning allows essentiality predictions in a much wider range of cell types than current methods, including those with unknown optimality principles such as human cell

types (Brunk et al., 2018) or the gut microbiome (Heinken et al., 2023).

The insight behind Flux Cone Learning is that it is possible to learn the shape of the metabolic space through random sampling of genome-scale metabolic models. High-dimensional sampling remains a key challenge in statistical learning (Hastie et al., 2009) because in high dimensions samples tend to be equidistant and concentrate on the boundaries of the space (Wainwright, 2019). In traditional probabilistic inference problems, this boundary concentration property renders many sampling-based techniques less useful, since reaching the interior of the space can be difficult without extensive sampling. While expectation would suggest that dense sampling is needed to accurately capture the cone geometry, in my tests I consistently found that accurate FCL models could be trained from shallow sampling with as few as 100 samples per deletion. I hypothesize that this could be a case of the curse of dimensionality operating as a "blessing of dimensionality". Deletions change the boundaries of the cone by definition rather than the interior. To capture these changes with a machine learning algorithm only requires samples at the boundary, and therefore a relatively small number of samples are sufficient for accurate prediction.

This observation is supported by preliminary work from Alperen Dalkiran, a postdoctoral research associate in the Oyarzún lab, who trained deep learning models—including feedforward and convolutional neural networks. These models did not yield improved performance, even when trained on larger datasets containing more than q =5,000 samples/cone. This is likely due to the fact that such models are overparameterized to capture complex nonlinear relationships, whereas flux samples in this context exhibit primarily linear correlations. Dimensionality reduction using Principal Component Analysis (PCA) prior to model training was also explored, but consistently resulted in reduced accuracy. As with the neural networks, this outcome may stem from the removal of features encoding subtle linear correlations between essentiality and cone shape, or from the need for a high-dimensional representation to effectively capture these relationships.

Recent advances in large language and vision models have demonstrated that

training on extremely large datasets enables broad task generalization (Bommasani et al., 2021). This approach has been extended to molecular biology, where foundation models trained on DNA sequence data have been applied to a variety of genomic tasks (Si et al., 2024). These models typically operate by learning lower-dimensional embeddings of sequence data, which are then used to generate novel sequences or evaluate the functional consequences of specific mutations (Dalla-Torre et al., 2024; Nguyen et al., 2024). The performance of FCL indicates that similarly informative representations of metabolic function can be derived from Monte Carlo sampling of genome-scale metabolic models. With sufficiently large-scale sampling across diverse species, growth conditions, and gene deletion genotypes, future work could enable the development of metabolic foundation models capable of generalizing across taxonomic domains and a range of fitness scores with minimal fine-tuning. Such efforts would expand the scope of biological foundation models to encompass a broader range of data modalities than is currently achievable (Brixi et al., 2025; Cui et al., 2024).

Sequence-based prediction represent an alternate approach to predicting gene essentiality, which until recently had achieved limited success and low accuracy rates compared to FBA (Aromolaran et al., 2021). However, Kang et al., 2024 recently tuned a pretrained protein language model to predict human protein essentiality. While they achieve very high ROC scores in human and mouse proteins, they do not restrict their analysis to metabolic proteins only, which are often harder to predict than core structural components of the cell such as membrane proteins. The performance of these models may well continue to improve as protein language models such as evolutionary-scale predictions improve (Lin et al., 2023); however, their combination with multimodal forms of data and mechanistic models is likely to provide further information for the machine learning algorithm. Combining various data modalities and incorporating non-sequence data remains a challenge, but recent work has combined protein interaction networks and single-cell -omics data to improve prediction of gene networks (Afshar et al., 2023). Overall, Flux Cone Learning represents an important first step towards predicting not just essentiality but other metabolic and non-metabolic

phenotypes. I expect that Flux Cone Learning will open new routes for computational prediction of many cellular fitness metrics, with applications in basic discovery, biotechnology and future therapies.

5.7 Limitations and future work

While Flux Cone Learning achieves state-of-the-art prediction in essentiality tasks across species, it has known limitations and areas for future improvement. First, the method can only predict metabolic gene phenotypes; the deleted genes must be present in the genome-scale metabolic model. Furthermore, deletions that do not perturb the flux cone enough for the sampler to detect can be difficult to predict. The method does not achieve 100% accuracy, which could be due to a variety of factors including inaccurate genome-scale model curation or insufficient sampling. Another limitation of the model is its need for training data. Improvements in the future could reduce the size of the training set needed or even train low-shot predictors based on generalization across multiple species.

An area of the pipeline where substantial improvements could be implemented is the sampling step. There are several alternatives to OptGPSampler and improvements that could reduce the total time needed to sample all deletion flux cones. For context, the time needed to sample the CHO flux cone exceeded 3 minutes per deletion, which was not possible without significant parallelization on an external server. One option would be to sample the wild-type flux cone deeply ($N = 5000$ samples per cone or more) and then use these samples to warm-start each deletion. It is relatively quick to check if a specific flux vector satisfies the constraints of a deletion GEM, after which feasible samples can be included in the training dataset as present in the cones of that deletion. An alternative would be to use an improved flux sampler of which there are several (Fallahi et al., 2020).

A different family of samplers are based on geometric programming, which use the Billiard Walk algorithm. Unlike traditional Hit-and-Run algorithms, which select random directions and step sizes to generate sample points, Billiard Walk simulates the trajectory of a particle reflecting off the polytope's boundaries, akin

to a billiard ball's motion (Polyak and Gryazina, 2014). Billiard walk can in some cases converge much faster than Hit-and-Run algorithms, which can shorten the computational time required to sample each deletion (Chalkis and Fisikopoulos, 2020). Recent work has developed a Multiphase Monte Carlo Sampler (MMCS) that combines flux cone rounding and sampling into a single pass based on the Billiard Walk (Chalkis et al., 2020). This algorithm brings into reach uniform sampling of human genome-scale models such as Recon3D, which expands the applications possible for Flux Cone Learning.

Another option to improve the performance of FCL is to oversample minority class deletions to balance the distribution of the training set. Because the training set for FCL is generated on-demand from a mechanistic model, more samples can be generated from knockouts that are more challenging to predict, either because they are in the minority class (essential knockouts and high producers of betaxanthin, for example) or because they are less perturbed and therefore differences in the cone are less identifiable. This oversampling could be done either based on the ground truth label (oversampling minority classes) or as part of an active learning loop where knockouts with consistent incorrect calls during training receive further sampling to improve the machine learning model iteratively (Liu et al., 2022).

In addition to essentiality, Flux Cone Learning can be applied to the prediction of other deletion phenotypes. These phenotypes do not need to be based on elements present in the genome-scale model, or indeed even be metabolic in scope. I introduced a first known model for prediction small molecule product prediction from a genome-scale model in this chapter. Betaxanthin, the chemical I chose to predict, is produced by an engineered pathway not present in the genome-scale Yeast9 model; however, I still achieved above-baseline results for high producer accuracy and other metrics (see Section 5.5). Future work could employ FCL to predict single-cell metabolic capabilities, providing insight into metabolic heterogeneity across individual cells and enabling more precise identification of cell-specific metabolic pathways and vulnerabilities (Gustafsson et al., 2023). FCL could also accelerate the discovery of synthetic lethal genes for

cancer therapies (Srivatsa et al., 2022). Synthetic lethal genes are deletions that individually have minimal impact but become lethal in combination with various chemotherapeutics (Kaelin Jr, 2005). Furthermore, FCL might facilitate the identification of gain-of-function deletions (mutations that confer advantageous metabolic traits), thus enabling targeted metabolic engineering strategies aimed at optimizing microbial production systems or adapting cells to specific environmental stresses (Ye et al., 2022). Additional high-throughput screens published during the preparation of this thesis could provide further case studies to demonstrate FCL's efficacy in metabolic engineering applications (Fang et al., 2025). Even outside of gene deletions, there are perturbations to the flux cone caused by changes in environmental conditions, exchange fluxes, or larger mutations that could be captured by my pipeline of flux sampling followed by machine learning. Future work could expand FCL to predict the effects of various medium conditions based on large metabolomic screens (Albornoz et al., 2024).

5.8 Conclusion

In this chapter, I introduced Flux Cone Learning (FCL), a novel strategy for the prediction of gene deletion phenotype from the shape of a high-dimensional metabolic flux cone generated by sampling from a genome-scale metabolic model. I first introduced the challenges to sampling high-dimensional spaces and the scale of the problem faced when sampling even relatively simple organisms. Following this introduction, I introduced recent advances in high-throughput genome-wide knockout deletion studies, which have enabled significant advances in machine learning. I next give background on the problem of predicting mutational effects, specifically essentiality and product production for metabolic engineering applications. I then walk through the Flux Cone Learning pipeline, giving further methodological detail on each step. I present state-of-the-art results in *E. coli* essentiality prediction in the best-curated GEM, iML1515. I explore this result further by reducing the training set size in two ways: the number of samples per cone and the number of discrete knockouts in the training set. I show

that model performance degrades to the naive baseline as the training set gets smaller and less dense. The prediction accuracy of FCL also drops in smaller, less curated models of *E. coli*. I finally show that by two metrics of divergence, essential and nonessential knockouts are significantly different. I next apply FCL to essentiality prediction in yeast and CHO cells, achieving state-of-the-art performance in these applications as well. Finally, I use FCL to classify high producer knockouts for betaxanthin production, presenting the first-of-its kind model for production prediction. I expect that Flux Cone Learning will open new routes for computational prediction of many cellular phenotypes, with applications in basic discovery, biotechnology and future therapies.

Chapter 6

Outlook and future perspectives

Biological systems are inherently complex, characterized by non-linear dynamics and multi-scale interactions. This complexity presents challenges for both traditional simulation approaches and purely data-driven machine learning (ML) models. In this thesis, I presented three different approaches to combining machine learning and mechanistic modelling: the use of machine learning tools to optimize mechanistic models (as in Chapter 3), the replacement of mechanistic models with machine learning surrogates (Chapter 4), and the improvement of model predictions with machine learning (Chapter 5). In this chapter, I summarize each of the results chapters in turn and extrapolate to the broader field, including examples of previous methods from literature that apply machine learning in a similar manner. I discuss some general challenges to the widespread adoption of hybrid modelling methods and end with a discussion of future directions for the field. For conciseness, I do not discuss details of specific machine learning or mechanistic models presented in previous literature, as this is an extensive subject beyond the scope of this chapter. Parts of this chapter are adapted from my paper published in *Biochemical Society Transactions* entitled "Applications of artificial intelligence and machine learning in dynamic pathway engineering". For a primer on AI and machine learning for biological applications, I refer the reader to the excellent review by Greener and colleagues (Greener et al., 2022).

6.1 Machine learning to optimize mechanistic models

The design and optimization of mechanistic models which match observed macro-level behaviour is a core problem in systems biology. Selecting and designing mechanistic models often requires significant expertise and fine-tuning. Machine learning methods offer one alternative. Chapter 3 describes a Bayesian optimization method for the optimization of genetic control circuits in engineered metabolic pathways. I applied Hyperopt, a package previously used to select hyperparameters in deep neural networks, to rapidly and robustly find ordinary differential equation (ODE) models which maximize product production while minimizing cellular costs. Engineered pathways exert metabolic and genetic burdens on the host cell when producing a desired product, which can lead to slow growth and other deleterious effects. As a result, dynamic control circuits have been built which respond to intracellular conditions by regulating an engineered pathway via transcription factor-mediated feedback loops. However, designing these circuits without computational optimization is challenging due to their large design space, multiple timescales, non-linearities, and the need to select both a discrete control architecture and a set of continuous dose-response parameters which determine the strength and response curve of the feedback loops. Previous work used random and exhaustive search as well as genetic algorithms. Bayesian optimization iteratively selects a candidate circuit, computes the value of a user-defined objective function, updates a nonparametric model of the objective function space based on the samples seen so far, and selects the next sample balancing exploration and exploitation.

My work explored various objective functions in a model of free fatty acid (FFA) production, specifically including objectives which incorporate the nonlinear dynamics present in a multiscale model which includes both fast metabolic reactions and slower gene expression dynamics. I discovered a trade-off between product rise time and overshoot where different areas of the optimality curve are occupied by different control architectures (Figure 3.13), indicating that cir-

cuit designers must balance several competing objectives when selecting which architecture is globally optimal for a certain task. A core benefit of Bayesian optimization is that it can easily be applied to relatively large systems (such as the p-aminostyrene model, which has 27 architectures, see Figure 3.14) and still converge in minutes. This flexibility makes it applicable to a wide variety of circuit design problems in metabolic engineering.

In parallel with Bayesian approaches, machine learning methods have increasingly been applied to accelerate and improve the design of gene circuits. Gradient descent algorithms, for instance, have enabled efficient parameter optimization in ODE-based models of gene circuits, supporting both rapid exploration and real-time tuning (Hiscock, 2019). Beyond parameter fitting, machine learning can assist in model discrimination, selection, and generation. Techniques such as Bayesian inference and Gaussian processes have been successfully used to infer model parameters and identify suitable dynamical system models (Devoid et al., 2013; Järvenpää et al., 2018; Toni et al., 2009).

Alternatively, model reduction approaches that generate smaller dynamic models which accurately represent overall system behaviour in fewer equations often use projection approaches similar to principal component analysis (PCA) (Benner et al., 2015). For example, ML models can also be trained on a set of candidate ODE models to reduce mechanistic model dimensionality and complexity (Brunton et al., 2016; Regazzoni et al., 2019). In many cases, the computational costs of simulating a mechanistic model many times to optimize it remains a bottleneck. In these cases, surrogate machine learning models which replace the mechanistic model entirely offer a practical and efficient alternative for navigating high-dimensional design spaces. In (Pfrommer et al., 2018), circuit optimization was sped up by training a neural network surrogate model on results from a finite element mechanistic model, then querying that model during the optimization process. The machine learning model was able to interpolate the high-dimensional design space and navigate towards the optimal bioprocess.

6.2 Replacing a mechanistic model with a machine learning surrogate

As our understanding of biological systems has grown in scope and detail, so have the mathematical models constructed to emulate their dynamics. While the predictive performance of these larger models can encompass a wider range of biological phenomena, their size and non-linearities make them intractable analytically and require slow numerical simulation methods (Wang et al., 2019). Problems that require many thousands of model simulations, such as parameter selection, sensitivity analysis, or iterative model design, are particularly challenging. In Chapter 4, running FBA thousands of times in the simulator loop quickly became computationally infeasible. The goal of the novel simulator was to combine ODE models of heterologous pathways with genome-scale models of native metabolism. The complex and nonlinear interactions between host and pathway are often neglected in ODE-only or GEM-only modelling frameworks. For example, growth rates are assumed to be constant in many ODE models including those considered in Chapter 3, whereas GEMs assume all reactions in metabolism to be at steady state. I built a simulator loop which iteratively passes information between a GEM and an ODE. Once initial concentrations for native metabolites contained in the ODE are established via a Bayesian warmup routine, the loop begins by integrating the ODE for 1 second of simulation time. After each timepoint, the pathway fluxes generated by the ODE constrain the next FBA optimization, which produces a growth rate and boundary influxes to the pathway from native metabolism.

However, after discovering that the FBA optimization comprised 90% of the simulation runtime and rendered long simulations intractable, I decided to replace it with a machine learning surrogate model. Surrogate models aim to completely replace computationally costly mechanistic model simulations with a machine learning model which regresses the desired inputs and outputs (Gherman et al., 2023). Mechanistic models can be simulated enough times to generate a large set of training data. While model training can take significant resources, the abil-

ity to parallelize many models across GPUs and the constant test-time inference costs make up for the upfront time sink (Cozad et al., 2014). My surrogate models included logistic regression to classify the feasibility of the FBA task, linear regression to predict growth rate, and a deep neural network to predict boundary fluxes. Even when accounting for total training time, the ML surrogate approach achieved a several orders of magnitude speed-up on the simulation process and enabled large-scale parameter sampling experiments.

The simulator was applied to two case study pathways already implemented experimentally in the literature. One pathway produces glucaric acid and branches from central carbon metabolism, which leads to a direct dependence on growth and a sustained drop in the predicted growth rate. The other produces beta-carotene and impacts growth not directly through a precursor reaction present in the FBA objective function but rather through depletion of native metabolites which indirectly slows biomass-linked metabolism. I validated the results of the simulator by comparing to previous results across different carbon sources and then applied Bayesian optimization and random sampling to explore the dynamic control circuit design space, finding trade-offs between production and burden in both pathways. The scale of the experimentation in Section 4.3.2 would not have been possible without the surrogate machine learning models.

Surrogate models have been built previously for ODE, PDE, and constraint-based systems, although many of the recently developed methods have not yet been applied to metabolic engineering (Gilpin et al., 2020). For example, a work trained machine learning models on proteomic and metabolomic timeseries data (Costello and García Martín, 2018). However, traditional supervised ML models are data-hungry and not optimized to learn physical dynamics. Newer methods seek to enforce mechanistic model structure which reflects domain knowledge and constrain the model search space using ODEs to represent underlying system dynamics. These methods can reduce the amount of data needed to train learning models. Physics-informed neural networks (PINNs) and their biologically-informed extensions (BINNs), use general forms of known ODEs where individual terms are replaced by an ANN trained on sparse experimental data (Lagergren et

al., 2020; Raissi et al., 2019). Universal Differential Equations similarly use neural networks to represent terms in an ODE (Rackauckas et al., 2020). The structure and domain knowledge encoded by the ODE allows for more data-efficient reconstruction of system dynamics. Alternate approaches to solve similar problems include ODENet, a deep neural network which uses numerical ODE integration as a forward pass and traditional neural network backpropagation, and a Gaussian process approach which directly incorporates ODE dynamics into the method kernel (Hu et al., 2022; Raissi and Karniadakis, 2018). As an example of an application to biology, an LSTM neural network surrogate was trained to predict spatial patterns in *E. coli* programmed with synthetic gene circuits. This surrogate achieved a 30,000-fold speed-up, enabling the screening of parameter sets that would have taken thousands of years to simulate using the original PDE model (Wang et al., 2019).

Unlike ODEs, traditional fully-connected neural networks do not incorporate an inherent sense of time. There are several ways to work around this limitation. Models can be trained on only equilibrium or steady-state conditions, though this prevents learning of time-dependent system dynamics. Other machine learning techniques take entire time trajectories as learning inputs, such as recent work learning to predict proteomics trajectories from previous timesteps (Costello and García Martín, 2018). Alternate approaches include recurrent neural networks, which keep track of a current system state and update it iteratively with each time step. Finally, the layers of a deep neural network can be conceptualized as different time points. Once trained, propagating a signal forward through the network gives a time trajectory of system dynamics (Gilpin et al., 2020).

6.3 Improving predictive accuracy using ML models trained on mechanistic and experimental data

For many problems, replacement of an entire mechanistic model is not necessary and additional labels from experiments are available. In these case, supervised machine learning models can be applied to both improve a mechanistic model and to improve the predictions generated from said model. Flux Cone Learning, the strategy I introduced in Chapter 5, applies machine learning to fluxomic data generated from large-scale flux sampling of genome-scale models to several biologically relevant prediction tasks, including gene essentiality prediction and classifying high producers of relevant chemicals. Flux Cone Learning relies on the insight that gene deletions affect the shape of the high-dimensional flux cone defined by a genome-scale model. These changes in shape can be detected by machine learning models, which are extremely good at detecting patterns across thousands of features and hundreds of thousands of data points. Genome-scale models can be modified to knock out genes by adjusting the constraints on reactions catalyzed by a gene's product enzyme. Monte Carlo flux sampling algorithms then perform random walks across the flux cone, converging to a uniform distribution and generating hundreds to thousands of samples per gene deletion. These samples are then paired with high-throughput experimental data from genome-wide screens of deletions paired with a fitness phenotype. The two phenotypes I considered were gene essentiality, which describes whether a cell lives or dies following a gene deletion, and betaxanthin production for metabolic engineering applications. A supervised machine learning algorithm takes the fluxomic data and fitness phenotypes as training data pairs. The models are trained on single samples from the flux cone; to improve the deletion-level accuracy, all samples from a single knockout are averaged via their model prediction score, which helps downweight samples from noninformative parts of the cone. Flux Cone Learning achieves state-of-the-art accuracy on gene essentiality prediction in *E. coli*, *S. cerevisiae*,

and Chinese Hamster Ovary cells. While FBA works well in *E. coli* as a zero-shot predictor of gene essentiality based on the biomass objective function, it assumes cells optimize a defined objective, which in higher organisms is often unknown or untrue. Flux Cone Learning does not rely on such an objective and does not require the predicted variable to be present in the metabolic model; as a result, it can be applied to a wider range of contexts and phenotypes. I presented a first demonstration of small molecule synthesis prediction for betaxanthin, a pigment, in *S. cerevisiae*. To simplify the machine learning task, I converted the task from regression to three-way classification. Betaxanthin autofluorescence data from a high-throughput screen was binned into high, medium, and low producers (approximately 15/60/15 split) and a suite of models were trained to predict each class. I applied several class balancing techniques to improve high producer accuracy, as in a real-world application this accuracy is most relevant to metabolic engineers. Often, the key to significant improvements in prediction accuracy is framing the problem as one that is most tractable to machine learning algorithms. In Flux Cone Learning, that included performing a second deletion-level aggregation step following the machine learning model to allow it to be trained on sample-level feature vectors and reframing the betaxanthin prediction problem from a regression to a classification task.

Machine learning has also been employed to estimate ordinary differential equation (ODE) parameters and quantify their associated uncertainties (Presnell and Alper, 2019). For example, the iSCHRUNK toolbox uses decision tree models to characterize uncertainty distributions over kinetic parameters (Andreozzi et al., 2016), while Bayesian approaches have been applied to the same task, offering probabilistic interpretations and robustness to limited data (St. John et al., 2019; Vega-Ramon et al., 2021). In cases where experimental data on enzyme kinetics is scarce, machine learning models can be used to infer missing values. Heckmann et al., 2018 used random forests to predict enzyme turnover numbers from structural and biochemical features. Similarly, in my work, I employed a deep learning model trained on the BRENDA database to estimate missing kinetic parameters for a beta-carotene production model, allowing for more accurate simulations and

predictions.

Beyond parameter estimation, machine learning can facilitate iterative model refinement through active and reinforcement learning. These approaches integrate feedback between experimental results, mechanistic modelling, and algorithmic predictions, enabling multiple design cycles with improved efficiency (Faulon and Faure, 2021). Once implemented, such frameworks can reduce the number of required experiments while enhancing model accuracy, even in data-limited contexts (HamediRad et al., 2019; Wang et al., 2020a). While active learning has been successfully demonstrated in protein and pathway design (Lu et al., 2022; Xiao et al., 2015), further proof-of-concept studies are needed to fully realize its potential in metabolic engineering.

6.4 Challenges and drawbacks

While this thesis proposes several approaches to apply machine learning to mechanistic modelling, substantial challenges to the widespread integration of the two paradigms remain. In this section, I will discuss downsides to the approaches presented in this thesis, including data availability, noise, data formatting, and explainability.

One of the consistent challenges to applying machine learning to biological problems is data availability. Unlike language or vision models, which can be trained on a large amount of data already present and easily accessible in the world (internet posts, Google images, YouTube videos, etc.), biological problems require data which is only obtainable through costly laboratory experiments, clinical trials, patient records, or peer-reviewed scientific literature. Aside from the privacy and ethical implications of using human biomedical data, which is outside the scope of this thesis, the curation and management of these large multimodal datasets requires computational resources, time, and often expert labelling.

An inherent drawback of supervised machine learning methods is that most require a training set to fit the model parameters. In contrast, mechanistic meth-

ods such as FBA are zero-shot predictors once the model is built. Commonly, the training set must be a majority of the total data available (often upwards of 75%) to achieve good generalization to novel data. Once trained, the machine learning model can be applied to new data quickly and easily, but many problems do not have the scale of data required to train a ML model in the first place. One of the ways I circumvented this problem in my thesis was the use of mechanistic models as generators of synthetic data. In cases where a limited amount of experimental data is available in addition to a mechanistic model, synthetic data generated from the mechanistic model can prime models to generalize within reasonable biological limits.

Finally, mechanistic models are written in human-readable equations generally composed of terms which represent physical processes. These equations are often complex and highly nonlinear, but their components can be characterized and understood (for example, the Michaelis-Menten equation as a component of substrate dynamics). Experts can make modifications to the equations based on hypotheses, which can then be validated or disproven with experimental data showing the downstream dynamics. Furthermore, the parameters of the equations are often directly measurable, as in the case of the cellular concentrations of metabolites or the kinetic parameters of enzymes. In contrast, machine learning methods are often considered "black boxes"; that is, the causal links between their input data and output predictions are not clear. Neural networks can be thought of as very flexible function approximators which can map data from one dimension to another; however, the functions and their weights learned from the training data are not interpretable to humans. While significant work has been done to improve the black box nature of ML models, it fundamentally remains.

6.5 Expected trajectory of the field

As the current literature shows, machine learning methods have so far been applied to a wide variety of tasks in systems biology, all of which can require different input data modalities, model architectures and strategies for perfor-

mance evaluation. Although this flexibility endows designers with a wide range of powerful algorithms, it comes at the cost of large datasets available for model training. Progress in laboratory automation and high-throughput screening are paving the way to a data-rich approach for biological design. The development of biofoundries across the globe (Hillson et al., 2019) together with progress in self-driving laboratories (Martin et al., 2023) offer exciting opportunities for large-scale data acquisition, which can pave the way for the systematic integration of AI and machine learning into pathway design pipelines. The interface between machine learning and metabolism is a relatively new and evolving field, with much of the recent work is still at a proof-of-concept stage. Future efforts will likely place an increasing focus on more user-friendly software tools that can bring this technology into the hands of wet lab practitioners, much like in other areas that enjoy a growing number of bespoke software packages aimed at end users (Chen et al., 2019; Hérisson et al., 2022; Nielsen et al., 2016). One area of particular interest is the use of active learning for pathway design. Active learning is a machine learning paradigm where the model selects the most informative designs to implement, thereby reducing the number of experiments required to explore the design space effectively. Several software packages such as BioAutomata (HamediRad et al., 2019), ART (Radivojević et al., 2020), ActiveOpt (Kumar et al., 2021), and METIS (Pandi et al., 2022) have implemented active learning pipelines for the design of static production pathways. In the case of dynamic pathways, however there is a pressing lack of comprehensive computational tools that support end-to-end system design. Given the complexity and number of designable components of dynamic pathways, the application of active learning tools could lead to important efficiency gains in implementation and prototyping. With the growing number of applications of machine learning in dynamic pathway engineering and the continued efforts to develop comprehensive software packages, I expect significant advancements in this area in the coming years that will support the wider adoption of AI and machine learning in strain design.

Bibliography

Abdi, H. and L. J. Williams (2010). "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4, pp. 433–459.

Afshar, S., P. R. Braun, S. Han, and Y. Lin (2023). "A multimodal deep learning model to infer cell-type-specific functional gene networks". In: *BMC bioinformatics* 24.1, p. 47.

Agren, R., A. Mardinoglu, A. Asplund, C. Kampf, M. Uhlen, and J. Nielsen (2014). "Identification of anticancer drugs for hepatocellular carcinoma through personalized genome-scale metabolic modeling". In: *Molecular systems biology* 10.3, p. 721.

Alber, M., A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, et al. (2019). "Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences". In: *NPJ digital medicine* 2.1, p. 115.

Alberts, B., R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, et al. (2022). *Molecular biology of the cell: seventh international student edition with registration card*. WW Norton & Company.

Albornoz, R. V., D. Oyarzún, and K. Burgess (2024). "Optimisation of surfactin yield in Bacillus using data-efficient active learning and high-throughput mass spectrometry". In: *Computational and Structural Biotechnology Journal* 23, pp. 1226–1233.

Alon, U. (2019). *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC.

Andreozzi, S., L. Miskovic, and V. Hatzimanikatis (2016). "iSCHRUNK–in silico approach to characterization and reduction of uncertainty in the kinetic models of genome-scale metabolic networks". In: *Metabolic engineering* 33, pp. 158–168.

Angione, C., E. Silverman, and E. Yaneske (2022). "Using machine learning as a surrogate model for agent-based simulations". In: *Plos one* 17.2, e0263150.

Apaolaza, I., E. San José-Eneriz, L. Tobalina, E. Miranda, L. Garate, X. Agirre, et al. (2017). "An in-silico approach to predict and exploit synthetic lethality in cancer metabolism". In: *Nature communications* 8.1, p. 459.

Apaolaza, I., L. V. Valcarcel, and F. J. Planes (2019). "gMCS: fast computation of genetic minimal cut sets in large networks". In: *Bioinformatics* 35.3, pp. 535–537.

Araujo, R. P. and L. A. Liotta (2018). "The topological requirements for robust perfect adaptation in networks of any size". In: *Nature communications* 9.1, p. 1757.

Aromolaran, O., D. Aromolaran, I. Isewon, and J. Oyelade (2021). "Machine learning approach to gene essentiality prediction: a review". In: *Briefings in bioinformatics* 22.5, bbab128.

Baeshen, N. A., M. N. Baeshen, A. Sheikh, R. S. Bora, M. M. M. Ahmed, H. A. Ramadan, et al. (2014). "Cell factories for insulin production". In: *Microbial cell factories* 13, pp. 1–9.

Bagge Carlson, F. (2018). "Hyperopt. jl: Hyperparameter optimization in Julia." In: URL: https://lup.lub.lu.se/search/publication/6ec19989-9b30-448c-be5e-bae4c4257c7b.

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

Baker, R. E., J.-M. Pena, J. Jayamohan, and A. Jérusalem (2018). "Mechanistic models versus machine learning, a fight worth fighting for the biological community?" In: *Biology letters* 14.5, p. 20170660.

Balandat, M., B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, et al. (2020). "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization". In: *Advances in Neural Information Processing Systems 33*. URL: https://proceedings.neurips.cc/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

Balaprakash, P., R. B. Gramacy, and S. M. Wild (2013). "Active-learning-based surrogate models for empirical performance tuning". In: *2013 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, pp. 1–8.

Banga, J. R. (2008). "Optimization in computational systems biology". In: *BMC systems biology* 2, pp. 1–7.

Bar-Even, A., E. Noor, Y. Savir, W. Liebermeister, D. Davidi, D. S. Tawfik, et al. (2011). "The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters". In: *Biochemistry* 50.21, pp. 4402–4410.

Bar-Peled, L. and N. Kory (2022). "Principles and functions of metabolic compartmentalization". In: *Nature metabolism* 4.10, pp. 1232–1244.

Beard, D. A., S.-d. Liang, and H. Qian (2002). "Energy balance for analysis of complex metabolic networks". In: *Biophysical journal* 83.1, pp. 79–86.

Benner, P., S. Gugercin, and K. Willcox (2015). "A survey of projection-based model reduction methods for parametric dynamical systems". In: *SIAM review* 57.4, pp. 483–531.

Berg, J. M., G. J. Gatto Jr, J. Hines, J. L. Tymoczko, and L. Stryer (2023). *Biochemistry*. Macmillan Higher Education.

Bergstra, J., D. Yamins, and D. Cox (2013). "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *International conference on machine learning*. PMLR, pp. 115–123.

Bhattacharya, P., K. Raman, and A. K. Tangirala (2022). "Discovering adaptation-capable biological network structures using control-theoretic approaches". In: *PLOS Computational Biology* 18.1, e1009769.

Bhowmik, A. C. (2017). "Dual-Use in Synthetic Biology: Balancing Intellectual Freedom with Regulations on Research". In: *Pathways: Stanford Journal of Public Health (SJPH)* 6, pp. 7–10.

Blanchini, F., E. Franco, and G. Giordano (2014). "A structural classification of candidate oscillatory and multistationary biochemical systems". In: *Bulletin of mathematical biology* 76.10, pp. 2542–2569.

Bommasani, R., D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, et al. (2021). "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258*.

Bordbar, A., J. M. Monk, Z. A. King, and B. O. Palsson (2014). "Constraint-based models predict metabolic and associated cellular functions". In: *Nature Reviews Genetics* 15.2, pp. 107–120.

Borkenhagen, L. K., M. W. Allen, and J. A. Runstadler (2021). "Influenza virus genotype to phenotype predictions through machine learning: a systematic review: computational prediction of influenza phenotype". In: *Emerging microbes & infections* 10.1, pp. 1896–1907.

Borkowski, O., C. Bricio, M. Murgiano, B. Rothschild-Mancinelli, G.-B. Stan, and T. Ellis (2018). "Cell-free prediction of protein expression costs for growing cells". In: *Nature communications* 9.1, p. 1457.

Borodina, I. and J. Nielsen (2005). "From genomes to in silico cells via metabolic networks". In: *Current opinion in biotechnology* 16.3, pp. 350–355.

Borzou, P., J. Ghaisari, I. Izadi, Y. Eshraghi, and Y. Gheisari (2023). "A novel strategy for dynamic modeling of genome-scale interaction networks". In: *Bioinformatics* 39.2, btad079.

Braun, M. and M. Golubitsky (1983). *Differential equations and their applications*. Vol. 1. Springer.

Briat, C., A. Gupta, and M. Khammash (2016). "Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks". In: *Cell systems* 2.1, pp. 15–26.

Briat, C., A. Gupta, and M. Khammash (2018). "Antithetic proportional-integral feedback for reduced variance and improved control performance of stochastic reaction networks". In: *Journal of The Royal Society Interface* 15.143, p. 20180079.

Brixi, G., M. G. Durrant, J. Ku, M. Poli, G. Brockman, D. Chang, et al. (2025). "Genome modeling and design across all domains of life with Evo 2". In: *bioRxiv*, pp. 2025–02.

Brophy, J. A. and C. A. Voigt (2014). "Principles of genetic circuit design". In: *Nature methods* 11.5, pp. 508–520.

Brunk, E., S. Sahoo, D. C. Zielinski, A. Altunkaya, A. Dräger, N. Mih, et al. (2018). "Recon3D enables a three-dimensional view of gene variation in human metabolism". In: *Nature biotechnology* 36.3, pp. 272–281.

Brunton, S. L., J. L. Proctor, and J. N. Kutz (2016). "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15, pp. 3932–3937.

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

Burget, M., E. Bardone, and M. Pedaste (2017). "Definitions and conceptual dimensions of responsible research and innovation: A literature review". In: *Science and engineering ethics* 23, pp. 1–19.

Cacheiro, P. and D. Smedley (2023). "Essential genes: a cross-species perspective". In: *Mammalian Genome* 34.3, pp. 357–363.

Cachera, P., H. Olsson, H. Coumou, M. L. Jensen, B. J. Sánchez, T. Strucko, et al. (2023a). "CRI-SPA: a high-throughput method for systematic genetic editing of yeast libraries". In: *Nucleic Acids Research* 51.17, e91–e91.

Cachera, P. P.-Y. J., N. C. Kurt, A. Ropke, T. Strucko, U. H. Mortensen, and M. K. Jensen (2023b). "Microbial cell factory optimisation using genome-wide host-pathway interaction screens". In: *bioRxiv*, pp. 2023–08.

Cain, S., C. Merzbacher, and D. A. Oyarzun (2024). "Low-dimensional representations of genome-scale metabolism". In: *Foundations of Systems Biology in Engineering Conference*, pp. 2024–05.

Cameron, D. E., C. J. Bashor, and J. J. Collins (2014). "A brief history of synthetic biology". In: *Nature Reviews Microbiology* 12.5, pp. 381–390.

Carbonell, P., T. Radivojevic, and H. Garcia Martin (2019). *Opportunities at the intersection of synthetic biology, machine learning, and automation.*

Cardoso, V. M., G. Campani, M. P. Santos, G. G. Silva, M. C. Pires, V. M. Gonçalves, et al. (2020). "Cost analysis based on bioreactor cultivation conditions: production of a soluble recombinant protein using Escherichia coli BL21 (DE3)". In: *Biotechnology Reports* 26, e00441.

Carothers, J. M., J. A. Goler, D. Juminaga, and J. D. Keasling (2011). "Model-driven engineering of RNA devices to quantitatively program gene expression". In: *Science* 334.6063, pp. 1716–1719.

Ceroni, F., A. Boo, S. Furini, T. E. Gorochowski, O. Borkowski, Y. N. Ladak, et al. (2018). "Burden-driven feedback control of gene expression". In: *Nature methods* 15.5, pp. 387–393.

Chachuat, B., A. B. Singer, and P. I. Barton (2005). "Global mixed-integer dynamic optimization". In: *AIChE Journal* 51.8, pp. 2235–2253.

Chalkis, A., V. Fisikopoulos, E. Tsigaridas, and H. Zafeiropoulos (2020). "Geometric algorithms for sampling the flux space of metabolic networks". In: *arXiv preprint arXiv:2012.05503*.

Chalkis, A. and V. Fisikopoulos (2020). "volesti: Volume approximation and sampling for convex polytopes in r". In: *arXiv preprint arXiv:2007.01578*.

Chang, D. T. (2019). "Bayesian hyperparameter optimization with BoTorch, GPyTorch and Ax". In: *arXiv preprint arXiv:1912.05686*.

Chang, L., P. Ruiz, T. Ito, and W. R. Sellers (2021). "Targeting pan-essential genes in cancer: challenges and opportunities". In: *Cancer cell* 39.4, pp. 466–479.

Chartrain, M., P. M. Salmon, D. K. Robinson, and B. C. Buckland (2000). "Metabolic engineering and directed evolution for the production of pharmaceuticals". In: *Current Opinion in Biotechnology* 11.2, pp. 209–214.

Chaves, M. and D. A. Oyarzún (2019). "Dynamics of complex feedback architectures in metabolic pathways". In: *Automatica* 99, pp. 323–332.

Chen, K. M., E. M. Cofer, J. Zhou, and O. G. Troyanskaya (2019). "Selene: a PyTorch-based deep learning library for sequence data". In: *Nature methods* 16.4, pp. 315–318.

Chen, V., M. Yang, W. Cui, J. S. Kim, A. Talwalkar, and J. Ma (2024). "Applying interpretable machine learning in computational biology—pitfalls, recommendations and opportunities for new developments". In: *Nature methods* 21.8, pp. 1454–1461.

Cheng, Y., X. Bi, Y. Xu, Y. Liu, J. Li, G. Du, et al. (2023). "Machine learning for metabolic pathway optimization: a review". In: *Computational and Structural Biotechnology Journal* 21, pp. 2381–2393.

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

Cirigliano, A., O. Cenciarelli, A. Malizia, C. Bellecci, P. Gaudio, M. Lioj, et al. (2017). "Biological dual-use research and synthetic biology of yeast". In: *Science and engineering ethics* 23, pp. 365–374.

Costanzo, M., B. VanderSluis, E. N. Koch, A. Baryshnikova, C. Pons, G. Tan, et al. (2016). "A global genetic interaction network maps a wiring diagram of cellular function". In: *Science* 353.6306, aaf1420.

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

Council, N. R., B. on Global Health, G. Affairs, Security, Cooperation, C. on Advances in Technology, et al. (2006). *Globalization, biosecurity, and the future of the life sciences*. National Academies Press.

Covert, M. W., N. Xiao, T. J. Chen, and J. R. Karr (2008). "Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli". In: *Bioinformatics* 24.18, pp. 2044–2050.

Cozad, A., N. V. Sahinidis, and D. C. Miller (2014). "Learning surrogate models for simulation-based optimization". In: *AIChE Journal* 60.6, pp. 2211–2227.

Cui, H., C. Wang, H. Maan, K. Pang, F. Luo, N. Duan, et al. (2024). "scGPT: toward building a foundation model for single-cell multi-omics using generative AI". In: *Nature Methods* 21.8, pp. 1470–1480.

Curtis, S. B. (1991). "Mechanistic models". In: *Physical and Chemical Mechanisms in Molecular Radiation Biology*, pp. 367–386.

Dalla-Torre, H., L. Gonzalez, J. Mendoza-Revilla, N. Lopez Carranza, A. H. Grzywaczewski, F. Oteri, et al. (2024). "Nucleotide Transformer: building and evaluating robust foundation models for human genomics". In: *Nature Methods*, pp. 1–11.

Daniels, B. C., Y.-J. Chen, J. P. Sethna, R. N. Gutenkunst, and C. R. Myers (2008). "Sloppiness, robustness, and evolvability in systems biology". In: *Current opinion in biotechnology* 19.4, pp. 389–395.

Danilevicz, M. F., M. Gill, R. Anderson, J. Batley, M. Bennamoun, P. E. Bayer, et al. (2022). "Plant genotype to phenotype prediction using machine learning". In: *Frontiers in Genetics* 13, p. 822173.

Darlington, A. P., J. Kim, J. I. Jiménez, and D. G. Bates (2018). "Dynamic allocation of orthogonal ribosomes facilitates uncoupling of co-expressed genes". In: *Nature communications* 9.1, p. 695.

Darvishi, M. T., F. Khani, and A. Soliman (2007). "The numerical simulation for stiff systems of ordinary differential equations". In: *Computers & Mathematics with Applications* 54.7-8, pp. 1055–1063.

Dasika, M. S. and C. D. Maranas (2008). "OptCircuit: an optimization based method for computational design of genetic circuits". In: *BMC systems biology* 2, pp. 1–19.

De Martino, D., M. Mori, and V. Parisi (2015). "Uniform sampling of steady states in metabolic networks: heterogeneous scales and rounding". In: *PloS one* 10.4, e0122670.

Dekel, E. and U. Alon (2005). "Optimality and evolutionary tuning of the expression level of a protein". In: *Nature* 436.7050, pp. 588–592.

Devoid, S., R. Overbeek, M. DeJongh, V. Vonstein, A. A. Best, and C. Henry (2013). "Automated genome annotation and metabolic model reconstruction in the SEED and Model SEED". In: *Systems metabolic engineering: methods and protocols*, pp. 17–45.

Djoumbou-Feunang, Y., J. Fiamoncini, A. Gil-de-la-Fuente, R. Greiner, C. Manach, and D. S. Wishart (2019). "BioTransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification". In: *Journal of cheminformatics* 11, pp. 1–25.

Doong, S. J., A. Gupta, and K. L. Prather (2018). "Layered dynamic regulation for improving metabolic pathway productivity in Escherichia coli". In: *Proceedings of the National Academy of Sciences* 115.12, pp. 2964–2969.

Dowell, R. D., O. Ryan, A. Jansen, D. Cheung, S. Agarwala, T. Danford, et al. (2010). "Genotype to phenotype: a complex problem". In: *Science* 328.5977, pp. 469–469.

Drengstig, T., H. R. Ueda, and P. Ruoff (2008). "Predicting perfect adaptation motifs in reaction kinetic networks". In: *The Journal of Physical Chemistry B* 112.51, pp. 16752–16758.

Dubois-Mignon, T. and P. Monget (2022). "Gene essentiality and variability: What is the link? A within-and between-species perspective". In: *Bioessays* 44.11, p. 2200132.

Dugger, S. A., A. Platt, and D. B. Goldstein (2018). "Drug development in the era of precision medicine". In: *Nature reviews Drug discovery* 17.3, pp. 183–196.

Dunlop, M. J., J. D. Keasling, and A. Mukhopadhyay (2010). "A model for improving microbial biofuel production using a synthetic feedback loop". In: *Systems and synthetic biology* 4, pp. 95–104.

Ebrahim, A., J. A. Lerman, B. O. Palsson, and D. R. Hyduke (2013). "COBRApy: constraints-based reconstruction and analysis for python". In: *BMC systems biology* 7.1, pp. 1–6.

Edwards, J. S. and B. O. Palsson (1999). "Systems properties of the Haemophilus influenzaeRd metabolic genotype". In: *Journal of Biological Chemistry* 274.25, pp. 17410–17416.

Ellington, A. D. and J. W. Szostak (1990). "In vitro selection of RNA molecules that bind specific ligands". In: *nature* 346.6287, pp. 818–822.

Elowitz, M. B. and S. Leibler (2000). "A synthetic oscillatory network of transcriptional regulators". In: *Nature* 403.6767, pp. 335–338.

Érdi, P. and J. Tóth (1989). *Mathematical models of chemical reactions: theory and applications of deterministic and stochastic models*. Manchester University Press.

Espinel-Ríos, S. and J. L. Avalos (2024a). "Hybrid physics-informed metabolic cybergenetics: process rates augmented with machine-learning surrogates informed by flux balance analysis". In: *Industrial & Engineering Chemistry Research* 63.15, pp. 6685–6700.

Espinel-Ríos, S. and J. L. Avalos (2024b). "Linking intra-and extra-cellular metabolic domains via neural-network surrogates for dynamic metabolic control". In: *IFAC-PapersOnLine* 58.23, pp. 115–120.

Fallahi, S., H. J. Skaug, and G. Alendal (2020). "A comparison of Monte Carlo sampling methods for metabolic network models". In: *Plos one* 15.7, e0235393.

Fang, L., X. Hao, J. Fan, X. Liu, Y. Chen, L. Wang, et al. (2025). "Genome-scale CRISPRi screen identifies pcnB repression conferring improved physiology for overproduction of free fatty acids in Escherichia coli". In: *Nature Communications* 16.1, p. 3060.

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

Faulon, J.-L. and L. Faure (2021). "In silico, in vitro, and in vivo machine learning in synthetic biology and metabolic engineering". In: *Current opinion in chemical biology* 65, pp. 85–92.

Faure, L., B. Mollet, W. Liebermeister, and J.-L. Faulon (2023). "A neural-mechanistic hybrid approach improving the predictive power of genome-scale metabolic models". In: *Nature Communications* 14.1, p. 4669.

Feist, A. M., C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, et al. (2007). "A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information". In: *Molecular systems biology* 3.1, p. 121.

Fernández-Cabezón, L., A. Cros, and P. I. Nikel (2019). "Evolutionary approaches for engineering industrially relevant phenotypes in bacterial cell factories". In: *Biotechnology journal* 14.9, p. 1800439.

Fisher, M. (2001). "Lehninger principles of biochemistry, ; by David L. Nelson and Michael M. Cox". In: *The Chemical Educator* 6, pp. 69–70.

Frazier, P. I. (2018). "A tutorial on Bayesian optimization". In: *arXiv preprint arXiv:1807.02811*.

Freischem, L. J., M. Barahona, and D. A. Oyarzún (2022). "Prediction of gene essentiality using machine learning and genome-scale metabolic models". In: *IFAC-PapersOnLine* 55.23, pp. 13–18.

Fröhlich, F., B. Kaltenbacher, F. J. Theis, and J. Hasenauer (2017). "Scalable parameter estimation for genome-scale biochemical reaction networks". In: *PLoS computational biology* 13.1, e1005331.

Gardner, D. J., D. R. Reynolds, C. S. Woodward, and C. J. Balos (2022). "Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers". In: *ACM Transactions on Mathematical Software (TOMS)* 48.3, pp. 1–24.

Gardner, T. S., C. R. Cantor, and J. J. Collins (2000). "Construction of a genetic toggle switch in Escherichia coli". In: *Nature* 403.6767, pp. 339–342.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.

Gebauer, J., C. Gentsch, J. Mansfeld, K. Schmeißer, S. Waschina, S. Brandes, et al. (2016). "A genome-scale database and reconstruction of Caenorhabditis elegans metabolism". In: *Cell systems* 2.5, pp. 312–322.

Gelbach, P. E., H. Cetin, and S. D. Finley (2024). "Flux sampling in genome-scale metabolic modeling of microbial communities". In: *BMC bioinformatics* 25.1, p. 45.

Gherman, I. M., Z. S. Abdallah, W. Pang, T. E. Gorochowski, C. S. Grierson, and L. Marucci (2023). "Bridging the gap between mechanistic biological models and machine learning surrogates". In: *PLoS Computational Biology* 19.4, e1010988.

Gilbert, D., M. Heiner, Y. Jayaweera, and C. Rohr (2019). "Towards dynamic genome-scale models". In: *Briefings in bioinformatics* 20.4, pp. 1167–1180.

Gilpin, W., Y. Huang, and D. B. Forger (2020). "Learning dynamics from large biological data sets: machine learning meets systems biology". In: *Current Opinion in Systems Biology* 22, pp. 1–7.

Goikhman, M. Y., N. Yevlampieva, N. Kamanina, I. Podeshvo, I. Gofman, S. Mil'tsov, et al. (2011). "New polyamides with main-chain cyanine chromophores". In: *Polymer Science Series A* 53.6, pp. 457–468.

Gombert, A. K. and J. Nielsen (2000). "Mathematical modelling of metabolism". In: *Current opinion in biotechnology* 11.2, pp. 180–186.

Gonzalez, J., J. Longworth, D. C. James, and N. D. Lawrence (2015). "Bayesian optimization for synthetic gene design". In: *arXiv preprint arXiv:1505.01627*.

González, J., Z. Dai, P. Hennig, and N. Lawrence (2016). "Batch Bayesian optimization via local penalization". In: *Artificial intelligence and statistics*. PMLR, pp. 648–657.

Gopalakrishnan, S., W. Johnson, M. A. Valderrama-Gomez, E. Icten, J. Tat, M. Ingram, et al. (2024). "COSMIC-dFBA: A novel multi-scale hybrid framework for bioprocess modeling". In: *Metabolic Engineering* 82, pp. 183–192.

Gorochowski, T. E., I. Avcilar-Kucukgoze, R. A. Bovenberg, J. A. Roubos, and Z. Ignatova (2016). "A minimal model of ribosome allocation dynamics captures trade-offs in expression between endogenous and synthetic genes". In: *ACS synthetic biology* 5.7, pp. 710–720.

Greener, J. G., S. M. Kandathil, L. Moffat, and D. T. Jones (2022). "A guide to machine learning for biologists". In: *Nature Reviews Molecular Cell Biology* 23.1, pp. 40–55. DOI: 10.1038/s41580-021-00407-0.

Grob, A., R. Di Blasi, and F. Ceroni (2021). "Experimental tools to reduce the burden of bacterial synthetic biology". In: *Current Opinion in Systems Biology* 28, p. 100393. DOI: 10.1016/j.coisb.2021.100393.

Gu, C., G. B. Kim, W. J. Kim, H. U. Kim, and S. Y. Lee (2019). "Current status and applications of genome-scale metabolic models". In: *Genome biology* 20, pp. 1–18.

Gudmundsson, S. and I. Thiele (2010). "Computationally efficient flux variability analysis". In: *BMC bioinformatics* 11, pp. 1–3.

Guo, Y., Y. Ju, D. Chen, and L. Wang (2021). "Research on the computational prediction of essential genes". In: *Frontiers in Cell and Developmental Biology* 9, p. 803608.

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

Gustafsson, J., M. Anton, F. Roshanzamir, R. Jörnsten, E. J. Kerkhoven, J. L. Robinson, et al. (2023). "Generation and analysis of context-specific genome-scale metabolic models derived from single-cell RNA-Seq data". In: *Proceedings of the National Academy of Sciences* 120.6, e2217868120.

Gutenkunst, R. N., J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna (2007). "Universally sloppy parameter sensitivities in systems biology models". In: *PLoS computational biology* 3.10, e189.

Hahn, M. W. and A. D. Kern (2005). "Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks". In: *Molecular biology and evolution* 22.4, pp. 803–806.

Hairer, E. and G. Wanner (1996). *Solving Ordinary Differential Equations II: Stiff and differential-algebraic problems*. Springer-Verlag. ISBN: 3642081584.

HamediRad, M., R. Chao, S. Weisberg, J. Lian, S. Sinha, and H. Zhao (2019). "Towards a fully automated algorithm driven platform for biosystems design". In: *Nature communications* 10.1, pp. 1–10. DOI: 10.1038/s41467-019-13189-z.

Han, T., A. Nazarbekov, X. Zou, and S. Y. Lee (2023). "Recent advances in systems metabolic engineering". In: *Current Opinion in Biotechnology* 84, p. 103004.

Hart, T., A. H. Y. Tong, K. Chan, J. Van Leeuwen, A. Seetharaman, M. Aregger, et al. (2017). "Evaluation and design of genome-wide CRISPR/SpCas9 knockout screens". In: *G3: Genes, Genomes, Genetics* 7.8, pp. 2719–2727.

Hartline, C. J., A. A. Mannan, D. Liu, F. Zhang, and D. A. Oyarzún (2020). "Metabolite sequestration enables rapid recovery from fatty acid depletion in Escherichia coli". In: *MBio* 11.2, pp. 10–1128.

Hartline, C. J., A. C. Schmitz, Y. Han, and F. Zhang (2021). "Dynamic control in metabolic engineering: Theories, tools, and applications". In: *Metabolic engineering* 63, pp. 126–140.

Hasan, M. A. and S. Lonardi (2020). "DeeplyEssential: a deep neural network for predicting essential genes in microbes". In: *BMC bioinformatics* 21, pp. 1–19.

Hasibi, R., T. Michoel, and D. A. Oyarzún (2024). "Integration of graph neural networks and genome-scale metabolic models for predicting gene essentiality". In: *npj Systems Biology and Applications* 10.1, p. 24.

Hastie, T., R. Tibshirani, J. Friedman, et al. (2009). *The elements of statistical learning*.

Hausser, J. and U. Alon (2020). "Tumour heterogeneity and the evolutionary trade-offs of cancer". In: *Nature Reviews Cancer* 20.4, pp. 247–257.

Heckmann, D., C. J. Lloyd, N. Mih, Y. Ha, D. C. Zielinski, Z. B. Haiman, et al. (2018). "Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models". In: *Nature communications* 9.1, pp. 1–10.

Heinken, A., J. Hertel, G. Acharya, D. A. Ravcheev, M. Nyga, O. E. Okpala, et al. (2023). "Genome-scale metabolic reconstruction of 7,302 human microorganisms for personalized medicine". In: *Nature Biotechnology* 41.9, pp. 1320–1331.

Heins, A.-L. and D. Weuster-Botz (2018). "Population heterogeneity in microbial bioprocesses: origin, analysis, mechanisms, and future perspectives". In: *Bioprocess and biosystems engineering* 41, pp. 889–916.

Henry, C. S., M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens (2010). "High-throughput generation, optimization and analysis of genome-scale metabolic models". In: *Nature biotechnology* 28.9, pp. 977–982.

Hérisson, J., T. Duigou, M. Du Lac, K. Bazi-Kabbaj, M. Sabeti Azad, G. Buldum, et al. (2022). "The automated Galaxy-SynBioCAD pipeline for synthetic biology design and engineering". In: *Nature Communications* 13.1, p. 5082.

Hernández-Almanza, A., J. Montanez, G. Martínez, A. Aguilar-Jiménez, J. C. Contreras-Esquivel, and C. N. Aguilar (2016). "Lycopene: Progress in microbial production". In: *Trends in Food Science & Technology* 56, pp. 142–148.

Herrgård, M. J., N. Swainston, P. Dobson, W. B. Dunn, K. Y. Arga, M. Arvas, et al. (2008). "A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology". In: *Nature biotechnology* 26.10, pp. 1155–1160.

Herrmann, H. A., B. C. Dyson, L. Vass, G. N. Johnson, and J.-M. Schwartz (2019). "Flux sampling is a powerful tool to study metabolism under changing environmental conditions". In: *NPJ systems biology and applications* 5.1, p. 32.

Hillson, N., M. Caddick, Y. Cai, J. A. Carrasco, M. W. Chang, N. C. Curach, et al. (2019). "Building a global alliance of biofoundries". In: *Nature communications* 10.1, p. 2040.

Hiscock, T. W. (2019). "Adapting machine-learning algorithms to design gene circuits". In: *BMC bioinformatics* 20.1, pp. 1–13.

Hodgkin, A. L. and A. F. Huxley (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of physiology* 117.4, p. 500.

Holzinger, A., B. Malle, A. Saranti, and B. Pfeifer (2021). "Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI". In: *Information Fusion* 71, pp. 28–37.

Hong, Y., J. Pasternak, and B. R. Glick (1991). "Biological consequences of plasmid transformation of the plant growth promoting rhizobacterium Pseudomonas putida GR12-2". In: *Canadian Journal of Microbiology* 37.10, pp. 796–799.

Hu, M., H. V. Dinh, Y. Shen, P. F. Suthers, C. J. Foster, C. M. Call, et al. (2023a). "Comparative study of two Saccharomyces cerevisiae strains with kinetic models at genome-scale". In: *Metabolic Engineering*.

Hu, M., P. Suthers, and C. D. Maranas (2023b). "Parameterization of Large-Scale Kinetic Models of Metabolism Using Datasets with Different Reference States". In: *2023 AIChE Annual Meeting*. AIChE.

Hu, P., W. Yang, Y. Zhu, and L. Hong (2022). "Revealing hidden dynamics from time-series data by ODENet". In: *Journal of Computational Physics* 461, p. 111203.

Ipsen, M. B., E. M. G. Sørensen, E. A. Thomsen, S. Weiss, J. Haldrup, A. Dalby, et al. (2022). "A genome-wide CRISPR-Cas9 knockout screen identifies novel PARP inhibitor resistance genes in prostate cancer". In: *Oncogene* 41.37, pp. 4271–4281.

Järvenpää, M., M. U. Gutmann, A. Vehtari, and P. Marttinen (2018). "Gaussian process modelling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria". In: *The Annals of Applied Statistics* 12.4, pp. 2228–2251.

Jeanne, G., A. Goelzer, S. Tebbani, D. Dumur, and V. Fromion (2018). "Dynamical resource allocation models for bioreactor optimization". In: *IFAC-PapersOnLine* 51.19, pp. 20–23.

Johnson, K. A. and R. S. Goody (2011). "The original Michaelis constant: translation of the 1913 Michaelis–Menten paper". In: *Biochemistry* 50.39, pp. 8264–8269.

Joy, M. P., A. Brock, D. E. Ingber, and S. Huang (2005). "High-betweenness proteins in the yeast protein interaction network". In: *BioMed research international* 2005.2, pp. 96–103.

Kaelin Jr, W. G. (2005). "The concept of synthetic lethality in the context of anticancer therapy". In: *Nature reviews cancer* 5.9, pp. 689–698.

Kang, B., R. Fan, C. Cui, and Q. Cui (2024). "Comprehensive prediction and analysis of human protein essentiality based on a pretrained large language model". In: *Nature Computational Science*, pp. 1–11.

Kang, B., R. Fan, C. Cui, and Q. Cui (2025). "Comprehensive prediction and analysis of human protein essentiality based on a pretrained large language model". In: *Nature Computational Science* 5.3, pp. 196–206.

Kauffman, K. J., P. Prakash, and J. S. Edwards (2003). "Advances in flux balance analysis". In: *Current opinion in biotechnology* 14.5, pp. 491–496.

Kaufman, D. E. and R. L. Smith (1998). "Direction choice for accelerated convergence in hit-and-run sampling". In: *Operations Research* 46.1, pp. 84–95.

Keasling, J. D. (2010). "Manufacturing molecules through metabolic engineering". In: *Science* 330.6009, pp. 1355–1358.

Khan, N., F. Afaq, and H. Mukhtar (2008). "Cancer chemoprevention through dietary antioxidants: progress and promise". In: *Antioxidants & redox signaling* 10.3, pp. 475–510.

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

Kim, G. B., W. J. Kim, H. U. Kim, and S. Y. Lee (2020). "Machine learning applications in systems metabolic engineering". In: *Current opinion in biotechnology* 64, pp. 1–9.

Kim, H. U., S. Y. Kim, H. Jeong, T. Y. Kim, J. J. Kim, H. E. Choy, et al. (2011). "Integrative genome-scale metabolic analysis of Vibrio vulnificus for drug targeting and discovery". In: *Molecular systems biology* 7.1, p. 460.

Kim, J. Y., Y.-G. Kim, and G. M. Lee (2012). "CHO cells in biotechnology for production of recombinant proteins: current state and further potential". In: *Applied microbiology and biotechnology* 93, pp. 917–930.

Kim, M., N. Rai, V. Zorraquino, and I. Tagkopoulos (2016). "Multi-omics integration accurately predicts cellular state in unexplored conditions for Escherichia coli". In: *Nature communications* 7.1, p. 13090.

Kim, S.-W., J.-B. Kim, W.-H. Jung, J.-H. Kim, and J.-K. Jung (2006). "Over-production of β-carotene from metabolically engineered Escherichia coli". In: *Biotechnology letters* 28, pp. 897–904.

King, Z. A., J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, et al. (2016). "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models". In: *Nucleic acids research* 44.D1, pp. D515–D522.

Kingma, D. P., M. Welling, et al. (2019). "An introduction to variational autoencoders". In: *Foundations and Trends in Machine Learning* 12.4, pp. 307–392.

Kizer, L., D. J. Pitera, B. F. Pfleger, and J. D. Keasling (2008). "Application of functional genomics to pathway optimization for increased isoprenoid production". In: *Applied and environmental microbiology* 74.10, pp. 3229–3241.

Klapper, I., D. B. Szyld, and K. Zhao (2021). *Metabolic Networks, Elementary Flux Modes, and Polyhedral Cones*. SIAM.

Kobiela, M., D. A. Oyarzun, and M. U. Gutmann (2024). "Risk-averse optimization of genetic circuits under uncertainty". In: *bioRxiv*, pp. 2024–11.

Kraut, J. (1988). "How do enzymes work?" In: *Science* 242.4878, pp. 533–540.

Kryshtafovych, A., T. Schwede, M. Topf, K. Fidelis, and J. Moult (2021). "Critical assessment of methods of protein structure prediction (CASP)—Round XIV". In: *Proteins: Structure, Function, and Bioinformatics* 89.12, pp. 1607–1617.

Kumar, P., P. A. Adamczyk, X. Zhang, R. B. Andrade, P. A. Romero, P. Ramanathan, et al. (2021). "Active and machine learning-based approaches to rapidly enhance microbial chemical production". In: *Metabolic Engineering* 67, pp. 216–226.

Labhsetwar, P., J. A. Cole, E. Roberts, N. D. Price, and Z. A. Luthey-Schulten (2013). "Heterogeneity in protein expression induces metabolic variability in a modeled Escherichia coli population". In: *Proceedings of the National Academy of Sciences* 110.34, pp. 14006–14011.

Lagergren, J. H., J. T. Nardini, R. E. Baker, M. J. Simpson, and K. B. Flores (2020). "Biologically-informed neural networks guide mechanistic modeling from sparse experimental data". In: *PLoS computational biology* 16.12, e1008462.

Lanier, L. L. (2014). "Just the FACS". In: *The Journal of Immunology* 193.5, pp. 2043–2044.

Larrimore, K. E. and G. Rancati (2019). "The conditional nature of gene essentiality". In: *Current Opinion in Genetics & Development* 58, pp. 55–61.

Lawson, C. E., J. M. Martí, T. Radivojevic, S. V. R. Jonnalagadda, R. Gentz, N. J. Hillson, et al. (2021). "Machine learning for metabolic engineering: A review". In: *Metabolic Engineering* 63, pp. 34–60.

Lazebnik, Y. (2002). "Can a biologist fix a radio?—Or, what I learned while studying apoptosis". In: *Cancer cell* 2.3, pp. 179–182.

Lee, J. and S. Leyffer (2011). *Mixed integer nonlinear programming*. Vol. 154. Springer Science & Business Media.

Leiby, N. and C. J. Marx (2014). "Metabolic erosion primarily through mutation accumulation, and not tradeoffs, drives limited evolution of substrate specificity in Escherichia coli". In: *PLoS biology* 12.2, e1001789.

Li, F., L. Yuan, H. Lu, G. Li, Y. Chen, M. K. Engqvist, et al. (2022). "Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction". In: *Nature Catalysis* 5.8, pp. 662–672.

Li, G., M. Li, J. Wang, J. Wu, F.-X. Wu, and Y. Pan (2016). "Predicting essential proteins based on subcellular localization, orthology and PPI networks". In: *BMC bioinformatics* 17, pp. 571–581.

Li, Z., S. Liu, and Q. Yang (2017). "Incoherent inputs enhance the robustness of biological oscillators". In: *Cell systems* 5.1, pp. 72–81.

Lian, J., M. HamediRad, S. Hu, and H. Zhao (2017). "Combinatorial metabolic engineering using an orthogonal tri-functional CRISPR system". In: *Nature communications* 8.1, p. 1688.

Liao, C., A. E. Blanchard, and T. Lu (2017). "An integrative circuit–host modelling framework for predicting synthetic gene network behaviours". In: *Nature microbiology* 2.12, pp. 1658–1666.

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

Lin, D.-W., L. Zhang, J. Zhang, and S. Chandrasekaran (2025). "Inferring metabolic objectives and trade-offs in single cells during embryogenesis". In: *Cell Systems* 16.1.

Lin, Z., H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, et al. (2023). "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637, pp. 1123–1130.

Liphardt, T. (2018). "Efficient computational methods for sampling-based metabolic flux analysis". PhD thesis. ETH Zurich.

Liu, D., A. A. Mannan, Y. Han, D. A. Oyarzún, and F. Zhang (2018). "Dynamic metabolic control: towards precision engineering of metabolism". In: *Journal of Industrial Microbiology and Biotechnology* 45.7, pp. 535–543.

Liu, D. and F. Zhang (2018). "Metabolic feedback circuits provide rapid control of metabolite dynamics". In: *ACS synthetic biology* 7.2, pp. 347–356.

Liu, P., L. Wang, R. Ranjan, G. He, and L. Zhao (2022). "A survey on active deep learning: From model driven to data driven". In: *ACM Computing Surveys (CSUR)* 54.10s, pp. 1–34.

Liu, X., L. Li, J. Liu, J. Qiao, and G.-R. Zhao (2019). "Metabolic engineering Escherichia coli for efficient production of icariside D2". In: *Biotechnology for biofuels* 12.1, pp. 1–12.

Loh, W.-L. (1996). "On Latin hypercube sampling". In: *The annals of statistics* 24.5, pp. 2058–2080.

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

Lu, H., D. J. Diaz, N. J. Czarnecki, C. Zhu, W. Kim, R. Shroff, et al. (2022). "Machine learning-aided engineering of hydrolases for PET depolymerization". In: *Nature* 604.7907, pp. 662–667. DOI: 10.1038/s41586-022-04599-z.

Lye, K. O., S. Mishra, D. Ray, and P. Chandrashekar (2021). "Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 374, p. 113575.

Ma, W., A. Trusina, H. El-Samad, W. A. Lim, and C. Tang (2009). "Defining network topologies that can achieve biochemical adaptation". In: *Cell* 138.4, pp. 760–773.

MacDonald, J. T., C. Barnes, R. I. Kitney, P. S. Freemont, and G.-B. V. Stan (2011). "Computational design approaches and tools for synthetic biology". In: *Integrative Biology* 3.2, pp. 97–108.

Machta, B. B., R. Chachra, M. K. Transtrum, and J. P. Sethna (2013). "Parameter space compression underlies emergent theories and predictive models". In: *Science* 342.6158, pp. 604–607.

Maćkiewicz, A. and W. Ratajczak (1993). "Principal components analysis (PCA)". In: *Computers & Geosciences* 19.3, pp. 303–342.

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

Mahadevan, R. and C. H. Schilling (2003). "The effects of alternate optimal solutions in constraint-based genome-scale metabolic models". In: *Metabolic engineering* 5.4, pp. 264–276.

Malik-Sheriff, R. S., M. Glont, T. V. Nguyen, K. Tiwari, M. G. Roberts, A. Xavier, et al. (2020). "BioModels—15 years of sharing computational models in life science". In: *Nucleic acids research* 48.D1, pp. D407–D415.

Mannan, A. A., D. Liu, F. Zhang, and D. A. Oyarzún (2017). "Fundamental design principles for transcription-factor-based metabolite biosensors". In: *ACS synthetic biology* 6.10, pp. 1851–1859.

Martin, H. G., T. Radivojevic, J. Zucker, K. Bouchard, J. Sustarich, S. Peisert, et al. (2023). "Perspectives for self-driving labs in synthetic biology". In: *Current Opinion in Biotechnology* 79, p. 102881.

Matsuda, F., Y. Toya, and H. Shimizu (2017). "Learning from quantitative data to understand central carbon metabolism". In: *Biotechnology Advances* 35.8, pp. 971–980.

McConville, M. (2014). "Open questions: microbes, metabolism and host-pathogen interactions". In: *BMC biology* 12.1, p. 18.

Megchelenbrink, W., M. Huynen, and E. Marchiori (2014). "optGpSampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks". In: *PloS one* 9.2, e86587.

Melendez-Alvarez, J., C. He, R. Zhang, Y. Kuang, and X.-J. Tian (2021). "Emergent damped oscillation induced by nutrient-modulating growth feedback". In: *ACS synthetic biology* 10.5, pp. 1227–1236.

Merzbacher, C. (2022). "A Machine Learning Approach for Optimization of Gene Circuits for Metabolic Engineering". MA thesis. University of Edinburgh.

Merzbacher, C. and D. A. Oyarzún (2023). "Applications of artificial intelligence and machine learning in dynamic pathway engineering". In: *Biochemical Society Transactions* 51.5, pp. 1871–1879.

Merzbacher, C., O. Mac Aodha, and D. A. Oyarzún (2023). "Bayesian optimization for design of multiscale biological circuits". In: *ACS Synthetic Biology* 12.7, pp. 2073–2082.

Merzbacher, C., O. M. Aodha, and D. A. Oyarzún (2025). "Modelling dynamic host-pathway interactions at the genome scale". In: *Metabolic Engineering*.

Metallo, C. M. and M. G. Vander Heiden (2013). "Understanding metabolic regulation and its influence on cell physiology". In: *Molecular cell* 49.3, pp. 388–398.

Miller, S. and M. J. Selgelid (2007). "Ethical and philosophical consideration of the dual-use dilemma in the biological sciences". In: *Science and engineering ethics* 13, pp. 523–580.

Min Lee, J., E. P. Gianchandani, J. A. Eddy, and J. A. Papin (2008). "Dynamic analysis of integrated signaling, metabolic, and regulatory networks". In: *PLoS computational biology* 4.5, e1000086.

Misawa, N., M. Nakagawa, K. Kobayashi, S. Yamano, Y. Izawa, K. Nakamura, et al. (1990). "Elucidation of the Erwinia uredovora carotenoid biosynthetic pathway by functional analysis of gene products expressed in Escherichia coli". In: *Journal of bacteriology* 172.12, pp. 6704–6712.

Monk, J. M., C. J. Lloyd, E. Brunk, N. Mih, A. Sastry, Z. King, et al. (2017). "i ML1515, a knowledgebase that computes Escherichia coli traits". In: *Nature biotechnology* 35.10, pp. 904–908.

Montaño López, J., L. Duran, and J. L. Avalos (2022). "Physiological limitations and opportunities in microbial metabolic engineering". In: *Nature Reviews Microbiology* 20.1, pp. 35–48.

Moon, T. S., S.-H. Yoon, A. M. Lanza, J. D. Roy-Mayhew, and K. L. J. Prather (2009). "Production of glucaric acid from a synthetic pathway in recombinant Escherichia coli". In: *Applied and environmental microbiology* 75.3, pp. 589–595.

Narayanan, B., D. Weilandt, M. Masid, L. Miskovic, and V. Hatzimanikatis (2024). "Rational strain design with minimal phenotype perturbation". In: *Nature Communications* 15.1, p. 723.

Nguyen, E., M. Poli, M. G. Durrant, B. Kang, D. Katrekar, D. B. Li, et al. (2024). "Sequence modeling and design from molecular to genome scale with Evo". In: *Science* 386.6723, eado9336.

Ni, C., C. V. Dinh, and K. L. Prather (2021). "Dynamic control of metabolism". In: *Annual Review of Chemical and Biomolecular Engineering* 12, pp. 519–541.

Nielsen, A. A., B. S. Der, J. Shin, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, et al. (2016). "Genetic circuit design automation". In: *Science* 352.6281, aac7341.

Nielsen, J. (2013). "Production of biopharmaceutical proteins by yeast: advances through metabolic engineering". In: *Bioengineered* 4.4, pp. 207–211.

Nielsen, J. (2017). "Systems biology of metabolism". In: *Annual review of biochemistry* 86.1, pp. 245–275.

Nikolados, E.-M., A. Y. Weiße, F. Ceroni, and D. A. Oyarzún (2019). "Growth defects and loss-of-function in synthetic gene circuits". In: *ACS synthetic biology* 8.6, pp. 1231–1240. DOI: 10.1021/acssynbio.8b00531.

Nikolados, E.-M., A. Wongprommoon, O. M. Aodha, G. Cambray, and D. A. Oyarzún (2022). "Accuracy and data efficiency in deep learning models of protein expression". In: *Nature Communications* 13.1, p. 7755.

O'Neil, N. J., M. L. Bailey, and P. Hieter (2017). "Synthetic lethality and cancer". In: *Nature Reviews Genetics* 18.10, pp. 613–623.

Occhipinti, A., S. Verma, C. Angione, et al. (2024). "Mechanism-aware and multimodal AI: beyond model-agnostic interpretation". In: *Trends in Cell Biology* 34.2, pp. 85–89.

Olaverri-Mendizabal, D., L. V. Valcárcel, N. Barrena, C. J. Rodríguez, and F. J. Planes (2024). "Review and meta-analysis of the genetic Minimal Cut Set approach for gene essentiality prediction in cancer metabolism". In: *Briefings in Bioinformatics* 25.3, bbae115.

Oliveira, R. D. de, G. A. Le Roux, and R. Mahadevan (2023). "Nonlinear programming reformulation of dynamic flux balance analysis models". In: *Computers & Chemical Engineering* 170, p. 108101.

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

Otero-Muras, I. and J. R. Banga (2017). "Automated design framework for synthetic biology exploiting pareto optimality". In: *ACS Synthetic Biology* 6.7, pp. 1180–1193.

Owen, R., J. Stilgoe, P. Macnaghten, M. Gorman, E. Fisher, and D. Guston (2013). "A framework for responsible innovation". In: *Responsible innovation: managing the responsible emergence of science and innovation in society*, pp. 27–50.

Owen, R., P. Macnaghten, and J. Stilgoe (2020). "Responsible research and innovation: From science in society to science for society, with society". In: *Emerging technologies*. Routledge, pp. 117–126.

Oyarzún, D. (2011). "Optimal control of metabolic networks with saturable enzyme kinetics". In: *IET systems biology* 5.2, pp. 110–119.

Oyarzún, D. A. and G.-B. V. Stan (2013). "Synthetic gene circuits for metabolic control: design trade-offs and constraints". In: *Journal of the Royal Society Interface* 10.78, p. 20120671.

Oyarzún, D. A. and M. Chaves (2015). "Design of a bistable switch to control cellular uptake". In: *Journal of The Royal Society Interface* 12.113, p. 20150618.

Ozaki, Y., Y. Tanigaki, S. Watanabe, and M. Onishi (2020). "Multiobjective tree-structured parzen estimator for computationally expensive optimization problems". In: *Proceedings of the 2020 genetic and evolutionary computation conference*, pp. 533–541.

Paddon, C. J., P. J. Westfall, D. J. Pitera, K. Benjamin, K. Fisher, D. McPhee, et al. (2013). "High-level semi-synthetic production of the potent antimalarial artemisinin". In: *Nature* 496.7446, pp. 528–532.

Palsson, B. (2015). *Systems biology*. Cambridge university press.

Palsson, B. Ø. (2011). *Systems biology: simulation of dynamic network states*. Cambridge University Press.

Pandi, A., C. Diehl, A. Yazdizadeh Kharrazi, S. A. Scholz, E. Bobkova, L. Faure, et al. (2022). "A versatile active learning workflow for optimization of genetic and metabolic networks". In: *Nature Communications* 13.1, p. 3876.

Patel, S. J., N. E. Sanjana, R. J. Kishton, A. Eidizadeh, S. K. Vodnala, M. Cam, et al. (2017). "Identification of essential genes for cancer immunotherapy". In: *Nature* 548.7669, pp. 537–542.

Patil, K. R., I. Rocha, J. Förster, and J. Nielsen (2005). "Evolutionary programming as a platform for in silico metabolic engineering". In: *BMC bioinformatics* 6, pp. 1–12.

Pfrommer, J., C. Zimmerling, J. Liu, L. Kärger, F. Henning, and J. Beyerer (2018). "Optimisation of manufacturing process parameters using deep neural networks as surrogate models". In: *Procedia CiRP* 72, pp. 426–431.

Pigou, M. and J. Morchain (2015). "Investigating the interactions between physical and biological heterogeneities in bioreactors using compartment, population balance and metabolic models". In: *Chemical Engineering Science* 126, pp. 267–282.

Plata, G., C. S. Henry, and D. Vitkup (2015). "Long-term phenotypic evolution of bacteria". In: *Nature* 517.7534, pp. 369–372.

Polyak, B. T. and E. Gryazina (2014). "Billiard walk-a new sampling algorithm for control and optimization". In: *IFAC Proceedings Volumes* 47.3, pp. 6123–6128.

Powers, D. M. (2020). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061*.

Presnell, K. V. and H. S. Alper (2019). "Systems metabolic engineering meets machine learning: a new era for data-driven metabolic engineering". In: *Biotechnology journal* 14.9, p. 1800416.

Price, N. D., I. Famili, D. A. Beard, and B. Ø. Palsson (2002). "Extreme pathways and Kirchhoff's second law". In: *Biophysical journal* 83.5, pp. 2879–2882.

Procopio, A., G. Cesarelli, L. Donisi, A. Merola, F. Amato, and C. Cosentino (2023). "Combined mechanistic modeling and machine-learning approaches in systems biology– a systematic literature review". In: *Computer methods and programs in biomedicine* 240, p. 107681.

Qiao, L., W. Zhao, C. Tang, Q. Nie, and L. Zhang (2019). "Network topologies that can achieve dual function of adaptation and noise attenuation". In: *Cell systems* 9.3, pp. 271–285.

Quek, L.-E. and N. Turner (2019). "Using the human genome-scale metabolic model Recon 2 for steady-state flux analysis of cancer cell metabolism". In: *Cancer Metabolism: Methods and Protocols*, pp. 479–489.

Rackauckas, C., Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, et al. (2020). "Universal differential equations for scientific machine learning". In: *arXiv preprint arXiv:2001.04385*.

Radivojević, T., Z. Costello, K. Workman, and H. Garcia Martin (2020). "A machine learning Automated Recommendation Tool for synthetic biology". In: *Nature communications* 11.1, p. 4879.

Raissi, M. and G. E. Karniadakis (2018). "Hidden physics models: Machine learning of nonlinear partial differential equations". In: *Journal of Computational Physics* 357, pp. 125–141.

Raissi, M., P. Perdikaris, and G. E. Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378, pp. 686–707.

Ramon, C. and J. Stelling (2023). "Functional comparison of metabolic networks across species". In: *Nature Communications* 14.1, p. 1699.

Rancati, G., J. Moffat, A. Typas, and N. Pavelka (2018). "Emerging and evolving concepts in gene essentiality". In: *Nature Reviews Genetics* 19.1, pp. 34–49.

Ranganathan, S., P. F. Suthers, and C. D. Maranas (2010). "OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions". In: *PLoS computational biology* 6.4, e1000744.

Regazzoni, F., L. Dede, and A. Quarteroni (2019). "Machine learning for fast and reliable solution of time-dependent differential equations". In: *Journal of Computational physics* 397, p. 108852.

Reimers, A.-M., H. Knoop, A. Bockmayr, and R. Steuer (2017). "Cellular trade-offs and optimal resource allocation during cyanobacterial diurnal growth". In: *Proceedings of the National Academy of Sciences* 114.31, E6457–E6465.

Richelle, A., B. P. Kellman, A. T. Wenzel, A. W. Chiang, T. Reagan, J. M. Gutierrez, et al. (2021). "Model-based assessment of mammalian cell metabolic functionalities using omics data". In: *Cell reports methods* 1.3.

Ro, D.-K., E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, et al. (2006). "Production of the antimalarial drug precursor artemisinic acid in engineered yeast". In: *Nature* 440.7086, pp. 940–943.

Rosconi, F., E. Rudmann, J. Li, D. Surujon, J. Anthony, M. Frank, et al. (2022). "A bacterial pan-genome makes gene essentiality strain-dependent and evolvable". In: *Nature Microbiology* 7.10, pp. 1580–1592.

Rueden, L. von, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke (2020). "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions". In: *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18*. Springer, pp. 548–560.

Schneider, P., A. von Kamp, and S. Klamt (2020). "An extended and generalized framework for the calculation of metabolic intervention strategies based on minimal cut sets". In: *PLoS computational biology* 16.7, e1008110.

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

Schomburg, I., A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, et al. (2004). "BRENDA, the enzyme database: updates and major new developments". In: *Nucleic acids research* 32.suppl_1, pp. D431–D433.

Schuetz, R., N. Zamboni, M. Zampieri, M. Heinemann, and U. Sauer (2012). "Multidimensional optimality of microbial metabolism". In: *Science* 336.6081, pp. 601–604.

Schuster, S., T. Dandekar, D. A. Fell, S. Schuster, T. Dandekar, and D. A. Fell (1999). "Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering". In: *Trends in biotechnology* 17.2, pp. 53–60.

Sciences, N. A. of, Medicine, D. on Earth, L. Studies, B. on Life Sciences, B. on Chemical Sciences, et al. (2018). *Biodefense in the age of synthetic biology*. National Academies Press.

Segre, D., D. Vitkup, and G. M. Church (2002). "Analysis of optimality in natural and perturbed metabolic networks". In: *Proceedings of the national academy of sciences* 99.23, pp. 15112–15117.

Shaw, W. M., H. Yamauchi, J. Mead, G.-O. F. Gowers, D. J. Bell, D. Öling, et al. (2019). "Engineering a model cell for rational tuning of GPCR signaling". In: *Cell* 177.3, pp. 782–796.

Shen, J., F. Liu, Y. Tu, and C. Tang (2021). "Finding gene network topologies for given biological function with recurrent neural network". In: *Nature communications* 12.1, pp. 1–10.

Shoval, O., H. Sheftel, G. Shinar, Y. Hart, O. Ramote, A. Mayo, et al. (2012). "Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space". In: *Science* 336.6085, pp. 1157–1160.

Si, Y., J. Zou, Y. Gao, G. Chuai, Q. Liu, and L. Chen (2024). "Foundation models in molecular biology". In: *Biophysics Reports* 10.3, p. 135.

Sidak, D., J. Schwarzerová, W. Weckwerth, and S. Waldherr (2022). "Interpretable machine learning methods for predictions in systems biology from omics data". In: *Frontiers in molecular biosciences* 9, p. 926623.

Singh, S., N. K. Tank, P. Dwiwedi, J. Charan, R. Kaur, P. Sidhu, et al. (2018). "Monoclonal antibodies: a review". In: *Current clinical pharmacology* 13.2, pp. 85–99.

Snoeck, S., C. Guidi, and M. De Mey (2024). ""Metabolic burden" explained: stress symptoms and its related responses induced by (over) expression of (heterologous) proteins in Escherichia coli". In: *Microbial Cell Factories* 23.1, p. 96.

Snoek, J., H. Larochelle, and R. P. Adams (2012). "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25.

Sokolova, M. and G. Lapalme (2009). "A systematic analysis of performance measures for classification tasks". In: *Information processing & management* 45.4, pp. 427–437.

Solgi, R. M. (2020). *Geneticalgorithm package*. https://pypi.org/project/geneticalgorithm/.

Son, H.-I., A. Weiss, and L. You (2021). "Design patterns for engineering genetic stability". In: *Current opinion in biomedical engineering* 19, p. 100297.

Spichak, S., T. F. Bastiaanssen, K. Berding, K. Vlckova, G. Clarke, T. G. Dinan, et al. (2021). "Mining microbes for mental health: determining the role of microbial

metabolic pathways in human brain health and disease". In: *Neuroscience & Biobehavioral Reviews* 125, pp. 698–761.

Srinivasan, B. (2022). "A guide to the Michaelis–Menten equation: steady state and beyond". In: *The FEBS journal* 289.20, pp. 6086–6098.

Srinivasan, S., W. R. Cluett, and R. Mahadevan (2015). "Constructing kinetic models of metabolism at genome-scales: a review". In: *Biotechnology journal* 10.9, pp. 1345–1359.

Srivatsa, S., H. Montazeri, G. Bianco, M. Coto-Llerena, M. Marinucci, C. K. Ng, et al. (2022). "Discovery of synthetic lethal interactions from large-scale pan-cancer perturbation screens". In: *Nature communications* 13.1, p. 7748.

St. John, P. C., J. Strutz, L. J. Broadbelt, K. E. Tyo, and Y. J. Bomble (2019). "Bayesian inference of metabolic kinetics from genome-scale multiomics data". In: *PLoS computational biology* 15.11, e1007424.

Stephanopoulos, G., A. A. Aristidou, and J. Nielsen (1998). "Metabolic engineering: principles and methodologies". In.

Stetter, K. O. (1999). "Extremophiles and their adaptation to hot environments". In: *FEBS letters* 452.1-2, pp. 22–25.

Stevens, J. T. and J. M. Carothers (2015). "Designing RNA-based genetic control systems for efficient production from engineered metabolic pathways". In: *ACS synthetic biology* 4.2, pp. 107–115.

Stone, A., A. Youssef, S. Rijal, R. Zhang, and X.-J. Tian (2024). "Context-dependent redesign of robust synthetic gene circuits". In: *Trends in Biotechnology* 42.7, pp. 895–909.

Straub, C. T., J. A. Counts, D. M. Nguyen, C.-H. Wu, B. M. Zeldes, J. R. Crosby, et al. (2018). "Biotechnology of extremely thermophilic archaea". In: *FEMS Microbiology Reviews* 42.5, pp. 543–578.

Sun, J. and H. S. Alper (2015). "Metabolic engineering of strains: from industrial-scale to lab-scale chemical production". In: *Journal of Industrial Microbiology and Biotechnology* 42.3, pp. 423–436.

Swainston, N., K. Smallbone, H. Hefzi, P. D. Dobson, J. Brewer, M. Hanscho, et al. (2016). "Recon 2.2: from reconstruction to model of human metabolism". In: *Metabolomics* 12, pp. 1–7.

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

Toni, T., D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf (2009). "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems". In: *Journal of the Royal Society Interface* 6.31, pp. 187–202.

Tonn, M. K., P. Thomas, M. Barahona, and D. A. Oyarzún (2019). "Stochastic modelling reveals mechanisms of metabolic heterogeneity". In: *Communications biology* 2.1, p. 108.

Tovar, J., G. León-Avila, L. B. Sánchez, R. Sutak, J. Tachezy, M. Van Der Giezen, et al. (2003). "Mitochondrial remnant organelles of Giardia function in iron-sulphur protein maturation". In: *Nature* 426.6963, pp. 172–176.

Transtrum, M. K., B. B. Machta, K. S. Brown, B. C. Daniels, C. R. Myers, and J. P. Sethna (2015). "Perspective: Sloppiness and emergent theories in physics, biology, and beyond". In: *The Journal of chemical physics* 143.1.

Turconi, J., F. Griolet, R. Guevel, G. Oddon, R. Villa, A. Geatti, et al. (2014). "Semisynthetic artemisinin, the chemical path to industrial production". In: *Organic Process Research & Development* 18.3, pp. 417–422.

Van Dien, S. (2013). "From the first drop to the first truckload: commercialization of microbial processes for renewable chemicals". In: *Current opinion in biotechnology* 24.6, pp. 1061–1068.

Vega-Ramon, F., X. Zhu, T. R. Savage, P. Petsagkourakis, K. Jing, and D. Zhang (2021). "Kinetic and hybrid modeling for yeast astaxanthin production under uncertainty". In: *Biotechnology and Bioengineering* 118.12, pp. 4854–4866.

Venayak, N., N. Anesiadis, W. R. Cluett, and R. Mahadevan (2015). "Engineering metabolism through dynamic control". In: *Current opinion in biotechnology* 34, pp. 142–152.

Verma, B. K., A. A. Mannan, F. Zhang, and D. A. Oyarzún (2021). "Trade-offs in biosensor optimization for dynamic pathway engineering". In: *ACS Synthetic Biology* 11.1, pp. 228–240.

Villaverde, A. F., D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, et al. (2015). "BioPreDyn-bench: a suite of benchmark problems for dynamic modelling in systems biology". In: *BMC systems biology* 9, pp. 1–15.

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

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al. (2020). "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature methods* 17.3, pp. 261–272.

Voet, D., J. G. Voet, and C. W. Pratt (2016). "Fundamentals of". In: *Biochemistry: Life at the Molecular Level*.

Wagner, C. and R. Urbanczik (2005). "The geometry of the flux cone of a metabolic network". In: *Biophysical journal* 89.6, pp. 3837–3845.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press.

Waldherr, S., D. A. Oyarzún, and A. Bockmayr (2015). "Dynamic optimization of metabolic networks coupled with gene expression". In: *Journal of theoretical biology* 365, pp. 469–485.

Wang, H., B. van Stein, M. Emmerich, and T. Back (2017). "A new acquisition function for Bayesian optimization based on the moment-generating function". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 507–512.

Wang, H., S. Marcišauskas, B. J. Sánchez, I. Domenzain, D. Hermansson, R. Agren, et al. (2018). "RAVEN 2.0: A versatile toolbox for metabolic network reconstruction and a case study on Streptomyces coelicolor". In: *PLoS computational biology* 14.10, e1006541.

Wang, S., K. Fan, N. Luo, Y. Cao, F. Wu, C. Zhang, et al. (2019). "Massive computational acceleration by using neural networks to emulate mechanism-based biological models". In: *Nature communications* 10.1, p. 4354.

Wang, Y., H. Wang, L. Wei, S. Li, L. Liu, and X. Wang (2020a). "Synthetic promoter design in Escherichia coli based on a deep generative network". In: *Nucleic Acids Research* 48.12, pp. 6403–6412.

Wang, Z., J. Sun, Q. Yang, and J. Yang (2020b). "Metabolic engineering Escherichia coli for the production of lycopene". In: *Molecules* 25.14, p. 3136.

Waterfall, J. J., F. P. Casey, R. N. Gutenkunst, K. S. Brown, C. R. Myers, P. W. Brouwer, et al. (2006). "Sloppy-model universality class and the Vandermonde matrix". In: *Physical review letters* 97.15, p. 150601.

Wehrs, M., D. Tanjore, T. Eng, J. Lievense, T. R. Pray, and A. Mukhopadhyay (2019). "Engineering robust production microbes for large-scale cultivation". In: *Trends in microbiology* 27.6, pp. 524–537.

Weiße, A. Y., D. A. Oyarzún, V. Danos, and P. S. Swain (2015). "Mechanistic links between cellular trade-offs, gene expression, and growth". In: *Proceedings of the National Academy of Sciences* 112.9, E1038–E1047. DOI: 10.1073/pnas.1416533112.

Wiback, S. J., I. Famili, H. J. Greenberg, and B. Ø. Palsson (2004). "Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space". In: *Journal of theoretical biology* 228.4, pp. 437–447.

Woods, M. L., M. Leon, R. Perez-Carrasco, and C. P. Barnes (2016). "A statistical approach reveals designs for the most robust stochastic gene oscillators". In: *ACS synthetic biology* 5.6, pp. 459–470.

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

Xiao, H., Z. Bao, and H. Zhao (2015). "High throughput screening and selection methods for directed enzyme evolution". In: *Industrial & engineering chemistry research* 54.16, pp. 4011–4020. DOI: 10.1021/ie503060a.

Xiong, K., K. J. la Cour Karottki, H. Hefzi, S. Li, L. M. Grav, S. Li, et al. (2021). "An optimized genome-wide, virus-free CRISPR screen for mammalian cells". In: *Cell Reports Methods* 1.4.

Xu, F., H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu (2019). "Explainable AI: A brief survey on history, research areas, approaches and challenges". In: *Natural language processing and Chinese computing: 8th cCF international conference, NLPCC 2019, dunhuang, China, October 9–14, 2019, proceedings, part II 8*. Springer, pp. 563–574.

Xu, P., L. Li, F. Zhang, G. Stephanopoulos, and M. Koffas (2014). "Improving fatty acids production by engineering dynamic pathway regulation and metabolic control". In: *Proceedings of the National Academy of Sciences* 111.31, pp. 11299–11304.

Yang, J. H., S. N. Wright, M. Hamblin, D. McCloskey, M. A. Alcantar, L. Schrübbers, et al. (2019a). "A white-box machine learning approach for revealing antibiotic mechanisms of action". In: *Cell* 177.6, pp. 1649–1661.

Yang, J. and L. Guo (2014). "Biosynthesis of β-carotene in engineered E. coli using the MEP and MVA pathways". In: *Microbial cell factories* 13, pp. 1–11.

Yang, L., A. Ebrahim, C. J. Lloyd, M. A. Saunders, and B. O. Palsson (2019b). "DynamicME: dynamic simulation and refinement of integrated models of metabolism and protein expression". In: *BMC systems biology* 13, pp. 1–15.

Ye, L., J. J. Park, L. Peng, Q. Yang, R. D. Chow, M. B. Dong, et al. (2022). "A genome-scale gain-of-function CRISPR screen in CD8 T cells identifies proline metabolism as a means to enhance CAR-T therapy". In: *Cell metabolism* 34.4, pp. 595–614.

Yeo, H. C., J. Hong, M. Lakshmanan, and D.-Y. Lee (2020). "Enzyme capacity-based genome scale modelling of CHO cells". In: *Metabolic Engineering* 60, pp. 138–147.

Yeung, A. T., Y. H. Choi, A. H. Lee, C. Hale, H. Ponstingl, D. Pickard, et al. (2019). "A genome-wide knockout screen in human macrophages identified host factors modulating Salmonella infection". In: *MBio* 10.5, pp. 10–1128.

Zanghellini, J., D. E. Ruckerbauer, M. Hanscho, and C. Jungreuthmayer (2013). "Elementary flux modes in a nutshell: properties, calculation and applications". In: *Biotechnology journal* 8.9, pp. 1009–1016.

Zecchin, A., P. C. Stapor, J. Goveia, and P. Carmeliet (2015). "Metabolic pathway compartmentalization: an underappreciated opportunity?" In: *Current opinion in biotechnology* 34, pp. 73–81.

Zhang, C., B. J. Sánchez, F. Li, C. W. Q. Eiden, W. T. Scott, U. W. Liebal, et al. (2024). "Yeast9: a consensus genome-scale metabolic model for S. cerevisiae curated by the community". In: *Molecular Systems Biology* 20.10, pp. 1134–1150.

Zhang, F., J. M. Carothers, and J. D. Keasling (2012). "Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids". In: *Nature biotechnology* 30.4, pp. 354–359.

Zhang, J., S. D. Petersen, T. Radivojevic, A. Ramirez, A. Pérez-Manríquez, E. Abeliuk, et al. (2020a). "Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism". In: *Nature communications* 11.1, p. 4880.

Zhang, X., W. Xiao, and W. Xiao (2020b). "DeepHE: Accurately predicting human essential genes based on deep learning". In: *PLOS Computational Biology* 16.9, e1008229.

Zhang, Y., J. Nielsen, and Z. Liu (2018). "Metabolic engineering of Saccharomyces cerevisiae for production of fatty acid–derived hydrocarbons". In: *Biotechnology and bioengineering* 115.9, pp. 2139–2147.

Zhao, Q., A. I. Stettner, E. Reznik, I. C. Paschalidis, and D. Segrè (2016). "Mapping the landscape of metabolic goals of a cell". In: *Genome biology* 17, pp. 1–11.

Zhou, L.-B. and A.-P. Zeng (2015). "Exploring lysine riboswitch for metabolic flux control and improvement of L-lysine synthesis in Corynebacterium glutamicum". In: *ACS synthetic biology* 4.6, pp. 729–734.

Zhu, Y., Y. Li, Y. Xu, J. Zhang, L. Ma, Q. Qi, et al. (2021). "Development of bifunctional biosensors for sensing and dynamic control of glycolysis flux in metabolic engineering". In: *Metabolic Engineering* 68, pp. 142–151.

Appendix A

Specification of p-aminostyrene model

The p-aminostryene synthesis model is the largest and most complex pathway under study. It includes two metabolites which can act on transcription factors: p-ACA and p-AF. There are a large number of possible architectures possible for this model (27 excluding positive feedback loops). As a result, we do not explicitly name the architectures. The p-aminostyrene pathway is based on one studied in (Stevens and Carothers, 2015). There are 7 pathway metabolites, 3 DNA promoter elements, 5 mRNA transcripts, 4 unfolded enzymes, 5 folded enzymes, and a folded and unfolded efflux pump protein described in the mass balance equations. We modified the Stevens model to simplify the multiple explicit equations describing aRED aptamer folding into a single sigmoid similar to

those in Equation 3.4. The metabolite mass balance equations are

$$\begin{aligned}
\frac{\mathrm{dchorismate}}{\mathrm{d}t} &= V_{\mathrm{chorismate}}\tau - f(\mathrm{papA},\, \mathrm{chorismate}) - \delta \cdot \mathrm{chorismate},\\
\frac{\mathrm{dpA1}}{\mathrm{d}t} &= f(\mathrm{papA},\, \mathrm{chorismate}) - f(\mathrm{papB},\, \mathrm{pA1}) - \delta \cdot \mathrm{pA1},\\
\frac{\mathrm{dpA2}}{\mathrm{d}t} &= f(\mathrm{papB},\, \mathrm{pA1}) - f(\mathrm{papC},\, \mathrm{pA2}) - \delta \cdot \mathrm{pA2},\\
\frac{\mathrm{dpA3}}{\mathrm{d}t} &= f(\mathrm{papC},\, \mathrm{pA2}) - f(\mathrm{deaminase},\, \mathrm{pA3}) - \delta \cdot \mathrm{pA3},\\
\frac{\mathrm{dpAF}}{\mathrm{d}t} &= f(\mathrm{deaminase},\, \mathrm{pA3}) - f(\mathrm{LAAO},\, \mathrm{pAF}) - \delta \cdot \mathrm{pAF} - \mathrm{L},\\
\frac{\mathrm{dpACA}}{\mathrm{d}t} &= f(\mathrm{LAAO},\, \mathrm{pAF}) - f(\mathrm{P_{efflux}},\, \mathrm{pACA}) - \delta \cdot \mathrm{pACA},\\
\frac{\mathrm{dpAS}}{\mathrm{d}t} &= f(\mathrm{P_{efflux}},\, \mathrm{pACA}).
\end{aligned} \tag{A.1}$$

Here, the constant L is the rate of loss of p-AF through the leaky cellular membrane, and $V_{\mathrm{chorismate}}$ is a constant parameter defined in Table A.3. The constant τ is the toxicity factor which scales concentrations to account for intermediate metabolite toxicity. The constant δ is the dilution rate due to cellular growth. The function f is a variation on the Michaelis-Menten equation:

$$\begin{aligned}
&f(e,x) = k_{\mathrm{cat}} e \frac{\frac{x}{\mathrm{N_A Vol_{cell}}}}{k_{\mathrm{m}} + \frac{x}{\mathrm{N_A + Vol_{cell}}}}\tau,\\
&\text{where:}\\
&\qquad e \in \{\mathrm{papA},\, \mathrm{papB},\, \mathrm{papC},\, \mathrm{deaminase},\, \mathrm{LAAO},\, \mathrm{P_{efflux}}\},\\
&\qquad x \in \{\mathrm{chorismate},\, \mathrm{pA1},\, \mathrm{pA2},\, \mathrm{pA3},\, \mathrm{pAF},\, \mathrm{pACA}\}.
\end{aligned} \tag{A.2}$$

The promoter and mRNA transcript concentrations are modeled separately here,

unlike in the previous models. The mass-balance equations for these elements are

$$
\begin{aligned}
\frac{\mathrm{dPr_1}}{\mathrm{d}t} &= \mathrm{Pr_1}\beta - \delta \cdot \mathrm{Pr_1}, \\
\frac{\mathrm{dPr_2}}{\mathrm{d}t} &= \mathrm{Pr_2}\beta - \delta \cdot \mathrm{Pr_2}, \\
\frac{\mathrm{dPr_3}}{\mathrm{d}t} &= \mathrm{Pr_3}\beta - \delta \cdot \mathrm{Pr_3}, \\
\frac{\mathrm{dmRNA_{papA}}}{\mathrm{d}t} &= \omega_1 u(\mathrm{pAF}, k_{1,\mathrm{pAF}}, \theta_{1,\mathrm{pAF}}) + (1-\omega_1) u(\mathrm{pACA}, k_{1,\mathrm{pACA}}, \theta_{1,\mathrm{pACA}}) \\
&\quad - \delta \cdot \mathrm{mRNA_{papA}} - \tau \cdot \mu \cdot \mathrm{mRNA_{papA}}, \\
\frac{\mathrm{dmRNA_{papB}}}{\mathrm{d}t} &= \omega_1 u(\mathrm{pAF}, k_{2,\mathrm{pAF}}, \theta_{1,\mathrm{pAF}}) + (1-\omega_1) u(\mathrm{pACA}, k_{2,\mathrm{pACA}}, \theta_{1,\mathrm{pACA}}) \\
&\quad - \delta \cdot \mathrm{mRNA_{papB}} - \tau \cdot \mu \cdot \mathrm{mRNA_{papB}}, \\
\frac{\mathrm{dmRNA_{papC}}}{\mathrm{d}t} &= \omega_1 u(\mathrm{pAF}, k_{3,\mathrm{pAF}}, \theta_{1,\mathrm{pAF}}) + (1-\omega_1) u(\mathrm{pACA}, k_{3,\mathrm{pACA}}, \theta_{1,\mathrm{pACA}}) \\
&\quad - \delta \cdot \mathrm{mRNA_{papC}} - \tau \cdot \mu \cdot \mathrm{mRNA_{papC}}, \\
\frac{\mathrm{dmRNA_{LAAO}}}{\mathrm{d}t} &= \omega_2 u(\mathrm{pAF}, k_{4,\mathrm{pAF}}, \theta_{2,\mathrm{pAF}}) + (1-\omega_2) u(\mathrm{pACA}, k_{4,\mathrm{pACA}}, \theta_{2,\mathrm{pACA}}) \\
&\quad - \delta \cdot \mathrm{mRNA_{LAAO}} - \tau \cdot \mu \cdot \mathrm{mRNA_{LAAO}}, \\
\frac{\mathrm{dmRNA_{P_{efflux}}}}{\mathrm{d}t} &= \omega_3 u(\mathrm{pAF}, k_{5,\mathrm{pAF}}, \theta_{3,\mathrm{pAF}}) + (1-\omega_3) u(\mathrm{pACA}, k_{5,\mathrm{pACA}}, \theta_{3,\mathrm{pACA}}) \\
&\quad - \delta \cdot \mathrm{mRNA_{P_{efflux}}} - \tau \cdot \mu \cdot \mathrm{mRNA_{P_{efflux}}},
\end{aligned}
\tag{A.3}
$$

where the function $u(x, k, \theta)$ is the sigmoidal control function defined in Equation 3.4. The binary parameters w_i define which metabolite exerts control on each promoter:

$$
\begin{aligned}
\omega_1 &= \begin{cases} 1 & \text{papABC controlled by p-AF,} \\ 0 & \text{papABC controlled by p-ACA,} \end{cases} \\
\omega_2 &= \begin{cases} 1 & \text{LAAO controlled by p-AF,} \\ 0 & \text{LAAO controlled by p-ACA,} \end{cases} \\
\omega_3 &= \begin{cases} 1 & \mathrm{P_{efflux}}\text{controlled by p-AF,} \\ 0 & \mathrm{P_{efflux}}\text{controlled by p-ACS.} \end{cases}
\end{aligned}
\tag{A.4}
$$

ω_1	ω_2	ω_3
0	0	0
1	0	0
0	1	0
0	0	1
1	1	0
1	0	1
0	1	1
1	1	1

Table A.1: Ligand binding encodings for all possible combinations of ligand control points. The binary variables ω_1, ω_2, and ω_3 turn transcriptional control on and off at each locus (see Equation A.3). There are three possible control types by each ligand at each locus (activation, repression, no control) determined by the form of the function u.

To avoid circuits with multistable dynamics, we restrict the search to architectures without positive feedback loops. This means that we consider a total of $3^3 = 27$, i.e. those where:

- promoter papABC is either unregulated or repressed by one of the two intermediates (p-AF or p-ACA),
- promoter LAAO is unregulated, activated by p-AF, or repressed by p-ACA,
- promoter P_{efflux} is either unregulated or activated by one of the two intermediates (p-AF or p-ACA).

We restrict the architectures to only those with no positive feedback loops, which limits the possible control topologies to activation at the first promoter and repression at the last promoter. p-ACA is downstream of the middle/second promoter and therefore is limited to repression to create a negative feedback loop on that promoter (upstream repression). Conversely, p-AF is upstream of the second promoter and is limited to activation at each promoter. As a result, there are 3 remaining control topologies possible at each locus (including no control at each), resulting $3 \cdot 3 \cdot 3 = 27$ architectures with only negative feedback loops.

The constant β is the DNA duplication rate. The constant μ is the mRNA degradation rate constant, which is assumed to be constant for all mRNAs. PapA, PapB, and PapC are all expressed from the same promoter but their translation rates are variable, so their mRNAs are modeled separately. Finally, the protein

folding process is modeled explicitly, with each of the 4 enzymes and the efflux pump having a folded and unfolded state. Additionally, the enzyme deaminase is expressed constitutively but its concentration rises to steady state and thus its dynamics are also modeled. The enzyme mass balance equations are

$$\begin{aligned}
\frac{\mathrm{dpapA_{uf}}}{\mathrm{d}t} &= v(\mathrm{mRNA_{papA}}) - \eta\cdot\tau\cdot\mathrm{papA_{uf}} - \delta\cdot\mathrm{papA_{uf}} - \rho\cdot\tau\cdot\mathrm{papA_{uf}},\\
\frac{\mathrm{dpapB_{uf}}}{\mathrm{d}t} &= v(\mathrm{mRNA_{papB}}) - \eta\cdot\tau\cdot\mathrm{papB_{uf}} - \delta\cdot\mathrm{papB_{uf}} - \rho\cdot\tau\cdot\mathrm{papB_{uf}},\\
\frac{\mathrm{dpapC_{uf}}}{\mathrm{d}t} &= v(\mathrm{mRNA_{papC}}) - \eta\cdot\tau\cdot\mathrm{papC_{uf}} - \delta\cdot\mathrm{papC_{uf}} - \rho\cdot\tau\cdot\mathrm{papC_{uf}},\\
\frac{\mathrm{dLAAO_{uf}}}{\mathrm{d}t} &= v(\mathrm{mRNA_{LAAO}}) - \eta\cdot\tau\cdot\mathrm{LAAO_{uf}} - \delta\cdot\mathrm{LAAO_{uf}} - \rho\cdot\tau\cdot\mathrm{LAAO_{uf}},\\
\frac{\mathrm{P_{efflux,\ uf}}}{\mathrm{d}t} &= v(\mathrm{mRNA_{P_{efflux,\ uf}}}) - \eta\cdot\tau\cdot\mathrm{P_{efflux,\ uf}} - \delta\cdot\mathrm{P_{efflux,\ uf}} - \rho\cdot\tau\cdot\mathrm{P_{efflux,\ uf}},\\
\frac{\mathrm{dpapA}}{\mathrm{d}t} &= \eta\cdot\tau\cdot\mathrm{papA_{uf}} - \delta\cdot\mathrm{papA} - \rho\cdot\tau\cdot\mathrm{papA},\\
\frac{\mathrm{dpapB}}{\mathrm{d}t} &= \eta\cdot\tau\cdot\mathrm{papB_{uf}} - \delta\cdot\mathrm{papB} - \rho\cdot\tau\cdot\mathrm{papB},\\
\frac{\mathrm{dpapC}}{\mathrm{d}t} &= \eta\cdot\tau\cdot\mathrm{papC_{uf}} - \delta\cdot\mathrm{papC} - \rho\cdot\tau\cdot\mathrm{papC},\\
\frac{\mathrm{dLAAO}}{\mathrm{d}t} &= \eta\cdot\tau\cdot\mathrm{LAAO_{uf}} - \delta\cdot\mathrm{LAAO} - \rho\cdot\tau\cdot\mathrm{LAAO},\\
\frac{\mathrm{dP_{efflux}}}{\mathrm{d}t} &= \eta\cdot\tau\cdot\mathrm{P_{efflux,\ uf}} - \delta\cdot\mathrm{P_{efflux}} - \rho\cdot\tau\cdot\mathrm{P_{efflux}},\\
\frac{\mathrm{ddeaminase}}{\mathrm{d}t} &= V_{\mathrm{deaminase}} - \delta\cdot\mathrm{deaminase}.
\end{aligned} \tag{A.5}$$

Here, the constant ρ is the protein degradation rate and η is the protein folding rate. The function v is the translation rate equation:

$$v(m) = \frac{m}{\mathrm{T_{init}} + \frac{\mathrm{L}_m}{\mathrm{R}}}, \tag{A.6}$$

where $\mathrm{T_{init}}$ is the transcription initiation rate, L_m is the length of the mRNA, and

Pathway Product	P-Aminostyrene
Decision Variables	$\theta_{1,\text{ pAF}}$, $\theta_{1,\text{ pACA}}$, $\theta_{2,\text{ pAF}}$, $\theta_{2,\text{ pACA}}$, $\theta_{3,\text{ pAF}}$, $\theta_{3,\text{ pACA}}$, $k_{1,\text{ pAF}}$, $k_{1,\text{ pACA}}$, $k_{2,\text{ pAF}}$, $k_{2,\text{ pACA}}$, $k_{3,\text{ pAF}}$, $k_{3,\text{ pACA}}$, $k_{4,\text{ pAF}}$, $k_{4,\text{ pACA}}$, $k_{5,\text{ pAF}}$, $k_{5,\text{ pACA}}$
Pathway Metabolites	Chorismate, PA1, PA2, PA3, pAF, pACA, PAS
Pathway Enzymes	papA, papB, papC, LAAO, Efflux pump, Deaminase
Integration Time	$1.73 \cdot 10^5$s
Initial Conditions	$x(0) = 0$mM (for all metabolites) $e(0) = 0$mM (for all enzymes)

Table A.2: P-Aminostryene model summary. The initial conditions for all model components are set to 0mM. Architectures are unnamed and thus not listed here.

R is the ribosome elongation rate. The objective function for the p-AS model is

$$\begin{aligned} J = \alpha_1 \int_0^T & |V_{\text{chorismate}}\tau - f(\text{P}_{\text{efflux}}, \text{pACA})| \, \text{d}t \\ + \alpha_2 \int_0^T & \{(\omega_1 u(\text{pAF}, k_{1,\text{ pAF}}, \theta_{1,\text{ pAF}}) + (1-\omega_1) u(\text{pACA}, k_{1,\text{ pACA}}, \theta_{1,\text{ pACA}}) \\ & + (\omega_1 u(\text{pAF}, k_{2,\text{ pAF}}, \theta_{1,\text{ pAF}}) + (1-\omega_1) u(\text{pACA}, k_{2,\text{ pACA}}, \theta_{1,\text{ pACA}}) \\ & + (\omega_1 u(\text{pAF}, k_{3,\text{ pAF}}, \theta_{1,\text{ pAF}}) + (1-\omega_1) u(\text{pACA}, k_{3,\text{ pACA}}, \theta_{1,\text{ pACA}}) \\ & + (\omega_2 u(\text{pAF}, k_{4,\text{ pAF}}, \theta_{2,pAF}) + (1-\omega_2) u(\text{pACA}, k_{4,\text{ pACA}}, \theta_{2,\text{ pACA}}) \\ & + (\omega_3 u(\text{pAF}, k_{5,\text{ pAF}}, \theta_{3,\text{ pAF}}) + (1-\omega_3) u(\text{pACA}, k_{5,\text{ pACA}}, \theta_{3,\text{ pACA}})\} \, \text{d}t, \end{aligned} \tag{A.7}$$

where pathway cost is the sum of all heterologous enzyme transcription rates across all loci and ligands, and the rate $f(\text{P}_{\text{efflux}}, \text{pACA})$ is defined in Equation A.2. The constant $V_{\text{chorismate}}$ as well as any other fixed parameter values in the model are defined in Table A.3. Table A.2 summarizes the model details.

Parameter	Symbol	Value	Units
Chorismate production rate	$V_{\text{chorismate}}$	1100.	1/s
Deaminase production rate	$V_{\text{deaminase}}$	10.	1/s
p-AF Loss	L	1.4E-5	1/s
mRNA degradation rate	M	3E-3	1/s
Protein degradation rate	P	2E-4	1/s
Protein folding rate	F	20	1/s
Dilution rate	δ	5.79E-4	1/s
DNA duplication rate	β	5.78E-4	1/s
Avogadro's number	N_A	6.02214E23	N/A
Cell volume	Vol_{cell}	2.5E-15	L
Metabolite-induced toxicity	t_a	5E-4	N/A
Protein-induced toxicity	t_p	50	N/A
Enzyme-induced toxicity	t_l	50	N/A
Toxicity constant	k_i	5E-5	M
Pap operon mRNA length	L_m	3400	nucleotides
Efflux pump mRNA length	L_m	2900	nucleotides
LAAO mRNA length	L_m	1600	nucleotides
Ribosome elongation rate	R	20	amino acids/s
Translation initiation rate	T_{init}	2E-1	1/s
Deaminase k_{cat}	k_{cat}	5	M/s
Deaminase k_{m}	k_{m}	1E-6	M
papA k_{cat}	k_{cat}	0.2975	M/s
papA k_{m}	k_{m}	0.056	M
papB k_{cat}	k_{cat}	39	M/s
papB k_{m}	k_{m}	0.38	M
papC k_{cat}	k_{cat}	20.44	M/s
papC k_{m}	k_{m}	0.555	M
LAAO k_{cat}	k_{cat}	1.29	M/s
LAAO k_{m}	k_{m}	10.82	M
Efflux pump rate	k_{m}	275	M

Table A.3: P-Aminostyrene model kinetic parameters

Appendix B

Parameters for Chapter 3 simulations

Figure	Model	Simulation	Parameter Values
2	glucaric acid	sample simulation	dual control architecture $k_1 = 2 \times 10^{-5}$ $k_2 = 2.2 \times 10^{-3}$ $\theta_1 = 3.3$ $\theta_2 = 1.0$
2	beta-carotene	sample simulation	$k_1 = 6.5 \times 10^{-6}$ $k_2 = 2.3 \times 10^{-2}$ $k_3 = 3.2 \times 10^{-2}$ $k_4 = 1.2 \times 10^{-2}$
3	glucaric acid	medium conditions	open loop architecture $k_1 = 4.1 \times 10^{-5}$ $k_2 = 2.27 \times 10^{-4}$
3	beta-carotene	knockouts	$k_1 = 2.41 \times 10^{-7}$ $k_2 = 9.7 \times 10^{-5}$ $k_3 = 9.8 \times 10^{-5}$ $k_4 = 3.67 \times 10^{-4}$

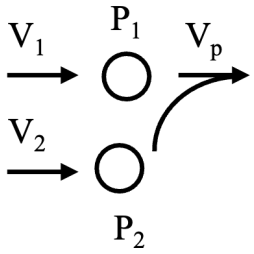
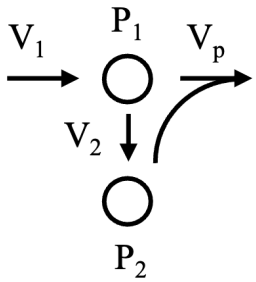
Continued on next page

Figure	Model	Simulation	Parameter Values
4	beta-carotene	large-scale parameter sampling	$10^{-8} \leq k_1 \leq 10^{-6}$ $10^{-7} \leq k_2 \leq 10^{-5}$ $10^{-6} \leq k_3 \leq 10^{-5}$ $10^{-5} \leq k_4 \leq 10^{-4}$ $10^{-5} \leq \theta \leq 10^{1}$
4	glucaric acid	global optimization of control circuits	4 architectures: open loop control upstream repression downstream activation dual control $10^{-7} \leq k_1 \leq 10^{-3}$ $10^{-7} \leq k_2 \leq 10^{-3}$ $10^{-7} \leq \theta_1 \leq 10^{1}$ $10^{-7} \leq \theta_1 \leq 10^{1}$

Table B.1: Parameter values and bounds for simulations in Figures 4.9, 4.11 in the main text.

Appendix C

Pathway balancing equations

Reaction topology	Balance equations	Modified precursor equations
	$V_{\mathrm{in}} = V_{\mathrm{p}}$	$\frac{\mathrm{d}P}{\mathrm{d}t} = V_{\mathrm{in}} - V_{\mathrm{p}} - \lambda P$
	$V_1 = V_2 = V_{\mathrm{p}}$	$\frac{dP_1}{dt} = V_1 - V_{\mathrm{p}} - \lambda P_1$ $\frac{\mathrm{d}P_2}{\mathrm{d}t} = V_2 - V_{\mathrm{p}} - \lambda P_1$
	$V_1 = V_{\mathrm{p}} + V_2$ $V_2 = V_{\mathrm{p}}$	$\frac{\mathrm{d}P_1}{\mathrm{d}t} = 2V_{\mathrm{p}} - \lambda P_1$

Continued on next page

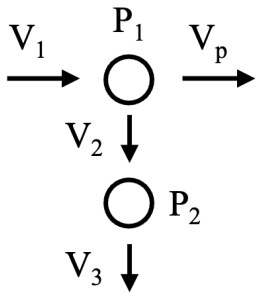
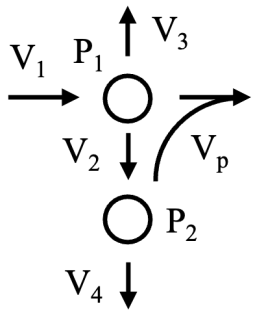
Reaction topology	Balance equations	Modified precursor equations
	$V_1 = V_2 + V_\mathrm{p}$	$\frac{\mathrm{d}P_2}{\mathrm{d}t} = V_1 - V_\mathrm{p} - \lambda P_2$
	$V_1 = V_2 + V_\mathrm{p} + V_3$ $V_2 = V_4 + V_\mathrm{p}$	$\frac{\mathrm{d}P_1}{\mathrm{d}t} = V_1 - V_3 - 2V_\mathrm{p} - \lambda P_1$ $\frac{\mathrm{d}P_2}{\mathrm{d}t} = -V_4 - \lambda P_1$

Table C.1: Reaction topologies and their associated balancing equations. Balancing the pathway fluxes at the boundary for various reaction topologies requires algebraic substitutions to modify the ODE model so that they maintain the steady-state assumption at each timestep. Table shows only the precursor equations that need to be modified. The glucaric acid and beta-carotene pathways in Figures 4.8 and 4.10 correspond to the first and last topology, respectively.