FACULTY OF
BIOSCIENCE ENGINEERING

EVOLUTION AND DESIGN OF METABOLIC NETWORKS USING QUALITY-DIVERSITY OPTIMIZATION ALGORITHMS

word count: 16,038

Shauny Van Hoye
Student ID: 01700330

Supervisors: Prof. dr. Bernard De Baets, dr. ir. Michiel Stock
Tutor: ir. Kirsten Van Huffel

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of master in Bioscience Engineering: Cell and Gene Biotechnology.
Academic year: 2021 - 2022

GHENT
UNIVERSITY

Gent, June 10, 2022

The promotors,

Prof. dr. Bernard De Baets,
dr. ir. Michiel Stock

The author,

Shauny Van Hoye

DANKWOORD

As this master thesis was not a well defined work were all the milestones that should be achieved were known at the start, I quickly discovered that knowing what you want do can be more difficult that the actual task at hand. Most of the time it felt as though I was running through a thick mist unaware that the next thing to come would be a ravine or another mountain to climb. I did however learn a lot, both knowledge and skill. Evolution has always intrigued me and being able to combine that interest with the ability to learn more about bio-inspired computing to build a framework with real world applicability was exhilarating. As many achievements in science, this thesis would not have been possible if it weren't for the support of several other people. First, I would like to thank prof. dr. Bernard De Baets and dr. ir. Michiel Stock, my two promotors. Professor De Beats, thank you for your helpful insights in the search for a thesis subject worth exploring and your library full of interesting books which helped to get a better understanding of optimization algorithms and bio-inspired computing. Michiel, thank you for the weekly update meetings, the interesting ideas, books and papers you suggested and the reoccurring feedback throughout this year. Secondly, I would like to thank ir. Kirsten Van Huffel, my tutor, you were always available for questions and your feedback has helped tremendously both to structure this thesis and make sure the content was well written. I would also like to thank all my professors and teaching assistants who educated me and introduced me to a multitude of interesting scientific field which inspire me to keep on learning every day. Finally, I would like to thank my family and friends for supporting me. I would like to especially thank my parents, for putting up with me these past twenty-two years and providing me with everything I could possibly need to bring this thesis to an end.

CONTENTS

SAMENVATTING

Het fenotype van een micro-organisme wordt bepaald door zowel de genen van het micro-organisme als zijn omgevingsfactoren. De combinatorische explosie die gepaard gaat met alle mogelijke modificaties van het genoom van het micro-organisme en zijn omgevingsfactoren, maakt metabolic engineering van micro-organismen in het laboratorium zeer uitdagend. Het optimaliseren van de groei van micro-organismen of de productie van metabolieten in een laboratorium kan daarom veel tijd en middelen kosten. *In silico* optimalisatie van het organisme voordat het opgegroeid en getest wordt in het laboratorium kan daarom helpen om de zoekruimte, gecreëerd door de genen en de omgevingsfactoren, te exploreren. In deze thesis worden Quality-Diversity evolutionaire algoritmen gebruikt om metabole netwerken van organismen *in silico* te evolueren met als doel de organismen te kunnen groeien op bepaalde substraten, specifieke metabolieten te produceren enz. We focussen ons op het gebruik van één bepaald Quality-Diversity Evolutionaire Algoritme genaamd Multi-dimensional Archive of Phenotypic (MAP)-Elites. Quality diversity Evolutionaire Algoritmen hebben als voordeel dat ze zoeken naar oplossingen die zowel goed presteren als divers zijn. Dit zorgt ervoor dat er een evenwicht bewaard wordt tussen exploratie om locale minima te vermijden en convergentie tijdens het zoeken naar een set van oplossingen die verspreid is over de zoekruimte. Dit contrasteert met traditionele evolutionaire algoritmen, die op zoek gaan naar slechts één optimale oplossing. Doorheen de thesis worden genome-scale metabole modellen gebruikt om de organismen *in silico* voor te stellen. De genome-scale metabole modellen werden gesimuleerd en gemanipuleerd met behulp van Cobrapy, een Python bibliotheek die een eenvoudige interface biedt voor metabolische constraint-based reconstructie en analyse. Het MAP-Elites algoritme maakt het mogelijk om de zoekruimte intelligent te verkennen door het proces van evolutie *in silico* te simuleren. OptMAP, het framework dat verkregen wordt door genome-scale metabole modellen te integreren met het MAP-Elites algoritme, wordt in eerste instantie gebruikt om uit te zoeken welke gen knockouts het metabool netwerk van *Escherichia coli* in staat stellen maximaal succinaat, acetaat, ethanol en flavanonen te produceren. Nadien wordt OptMAP ook uitgebreid om rekening te houden met verschillende medium samenstellingen die de groei van het micro-organisme en productie van metabolieten kan beïnvloeden. De resultaten verkregen voor de productie van succinaat, acetaat, ethanol en flavanonen tonen aan dat OptMAP in staat is om gen knockouts en medium samenstellingen te vinden die de productie van target-metabolieten maximaliseert. Daarnaast levert OptMAP ook inzichten in de relaties tussen verschillende karakteristieken van de gevonden oplossingen aangezien OptMAP gebaseerd is op een Quality-Diversity of Illumination algoritme.

Kernwoorden: Evolutionaire algoritmen, Quality Diversity algoritmen, MAP-Elites, *In Silico* Metabolic Engineering, Genome-Scale Metabole Modellen

SUMMARY

The phenotype of a microorganism is determined by both its genes and environmental factors. The combinatorial explosion associated with all the possible assortments of genome configurations and environments makes the metabolic engineering of microorganisms in the laboratory a tedious undertaking. Optimizing the growth of microorganisms or the production of metabolites in a laboratory can thus take up a lot of time and resources. Navigating the search space created by all possible genetic variants and environmental factors using trial-and-error in the wet lab could therefore be aided by *in silico* optimization of the organism before it is grown and tested in the wet lab. In this thesis, Quality-Diversity Evolutionary Algorithms are used to evolve metabolic networks of organisms towards specific metabolic engineering goals such as the growth on certain substrates and the production of specific target metabolites. We focus on the use of one particular Quality-Diversity Evolutionary Algorithm called Multi-dimensional Archive of Phenotypic (MAP) Elites. The advantage of Quality-Diversity optimization is that while searching for a well-performing solution, also diversity is promoted. This way, a balance between exploration and convergence is maintained during the search, escaping local minima and finding a set of high-performing solutions distributed over the search space. The latter is in contrast with traditional evolutionary algorithms that search for a single best solution. Throughout this thesis, genome-scale metabolic models are used to represent microbial metabolism *in silico*. These models can be simulated and manipulated using the Cobrapy framework, which provides a simple interface to metabolic constraint-based reconstruction and analysis. Here, the MAP-Elites algorithm allows for the intelligent exploration of this search space of all possible variants. OptMAP, the framework obtained by integrating genome-scale metabolic models with the MAP-Elites algorithm is applied to several case studies. We will start by exploring the evolution of *Escherichia coli* to maximize the production of succinate, acetate, ethanol and flavanones. Afterwards, OptMAP is expanded to consider different medium compositions that can influence growth and the production of target metabolites. The results obtained for the production of succinate, acetate, ethanol and flavanones show that OptMAP is able to discover gene knockouts and medium compositions that maximize the production of target metabolites. Additionally, OptMAP is able to illuminate relationships between different dimensions of the feature space and the associated fitness potential to provide (novel) scientific insights.

Key Words: Evolutionary algorithms, Quality-Diversity algorithms, MAP-Elites, *In Silico* Metabolic Engineering, Genome-Scale Metabolic Models

ACRONYMS

4CL	4-coumarate-CoA ligase
ACOATA	Acetyl-CoA ACP transacylase
AKGDH	2-Oxogluterate dehydrogenase
ATPS4rpp	Periplasmic ATP synthase
BiGG	Biochemical, Genetic and Genomic knowledgebase of large scale metabolic reconstructions
BPCY	Biomass-product coupled yield
CHI	Chalcone isomerase
CHS	Chalcone synthase
CMA-ME	Covariance Matrix Adaptation MAP-Elites
Cobrapy	COnstraint-Based Reconstruction and Analysis in Python
EAs	Evolutionary Algorithms
FBA	Flux balance analysis
FRUpts2	Fructose transport via PEP:Pyr PTS
G6PDH2r	Glucose 6-phosphate dehydrogenase
GEM	Genome-scale metabolic model
GLCpts	D-glucose transport via PEP:Pyr PTS
GLUSy	Glutamate synthase
GND	Phosphogluconate dehydrogenase
GO	Gene Ontology
ICL	Isocitrate lyase
KAS15	Beta-ketoacyl-ACP synthase
KEGG	Kyoto Encyclopedia of Genes and Genomes
MAP-Elites	Multidimensional Archive of Phenotypic Elites
ME-model	GEM of metabolism and macromolecular expression
M-model	Metabolic model
NADH16	NADH dehydrogenase
NADTRHD	NAD transhydrogenase
NEAT	NeuroEvolution of Augmenting Topologies
NS	Novelty search
PFL	Formate lyase
PGL	6-phosphogluconolactonase
PhPP	Phenotypic phase plane
PPC	Phosphoenolpyruvate carboxylase
PTAr	Phosphotransacetylase
QD	Quality diversity
RPE	Ribulose 5-phosphate 3-epimerase
S matrix	Stoichiometric matrix
SUCDi	Succinate dehydrogenase
SUCOAS	Succinyl-CoA synthetase
THD2	NAD(P) transhydrogenase
TPI	Triose phosphate isomerase
UPDRS	Unified Parkinson's Disease Rating Scale

1. INTRODUCTION

Every corner of planet earth is filled with an unimaginable diversity and number of living organisms (Lecointre and Le Guyader, 2006). Each one trying to stay alive and reproduce. Depending on the conditions the organisms are surrounded by, different inventive solutions to avoid death are discovered via natural selection (Kimura, 2020). These solutions are the result of simple principles underlying evolution. Evolution is based on a simple variation–selection loop. Variation is created via mutations or recombination, the new organisms are then selected based on their properties, the niches they occupy and any competitors that may be present (Castle et al., 2021). This evolutionary process has resulted in the most complex and sophisticated designs known to mankind and unparalleled diversity (Kimura, 2020). It is therefore obvious that scientists from many different fields look to this process for inspiration to solve many complex problems (Mouret, 2020). This is nothing new as bio-inspired computing, where biological models are used to solve computer science problems, has been used in many instances such as neural networks based on our brain, the search for emergence based on ant or bee colonies and of course evolutionary algorithms based on evolution (Yang, 2010). For example, artificial neural networks are part of the machine learning field and are at the center of deep learning algorithms. They are inspired by the structure of the human brain and mimic the way biological neurons signal each another (Education, 2022). Artificial neural networks can be used for pattern recognition and have been used for computer vision applications such as face recognition software (Hopfield, 1982). More recently deep learning has been used to tackle one of the largest problems in biology, namely to accurately predict protein folding (Senior et al., 2020). Ant or bee colony optimization algorithms are based on the behaviour of ants or bees to find good paths through graphs (Monmarché et al., 2010). These types of search algorithms can often be used for internet routing or finding the shortest route for delivery drivers (Waldner, 2013).

In this thesis, evolutionary algorithms based on natural evolution will be used as a search algorithm to find out which genetic knockouts optimize specific metabolic engineering objectives. Since the goal is to discover novel microorganisms for metabolic engineering purposes, microbial evolution can reveal interesting ideas as to which features should be implemented in such algorithms. Microorganisms have had the opportunity to evolve for almost 3.5 billion years (Schopf and Packer, 1987). During all this time they have found ways to capture energy from sunlight (Schopf and Packer, 1987), use basic chemical elements found in the ground and oceans to produce extremely complex molecules such as vitamins (Kang et al., 2022) and antibiotics (Chandra and Kumar, 2017) and in recent times microorganisms have even evolved to breakdown synthetic molecules produced by mankind not present for most of the existence of these microorganisms (Kawai, 1995). Because of the ability of microorganisms to produce and break down all kinds of molecules, they have been used by mankind for millennia for the production of beer, wine, bread, cheese etc (Mustafa et al., 2018). In recent centuries, the use of microorganisms has seen an explosion of use cases and development. To optimize microorganisms towards performing a specific

task, a lot of manipulating and tuning of the microorganisms is carried out in the wet lab. Metabolic engineering is the field that is focused on optimizing genetic and regulatory processes within organisms to increase the production of various target molecules (Yang et al., 1998). Although metabolic engineering has come very far, especially in the last decades, the optimization of organisms in the laboratory often takes up a lot of time, effort and resources (Stephanopoulos, 2012). This is the case because of the combinatorial explosion associated with all the possible assortments of genome configurations and environments that ultimately contribute to the phenotype of the organism. If the optimal genome and environmental cues to produce specific products were known before having to grow the microorganisms in the laboratory, the process could be sped up significantly while exploring even more possibilities (Tomar and De, 2013). Simulating the evolution of microorganisms towards specific metabolic engineering objectives in the wet lab would be an ideal candidate to design novel microorganisms if it were not for the large timescales evolution operates at. Gleizer et al. (2019) attempted to convert an *Escherichia coli* to generate all its biomass carbon from CO_2. The chemostat-based directed evolution led to complete trophic mode change in more than 200 days (Gleizer et al., 2019). Therefore, this thesis proposes to simulate the evolution of microorganisms *in silico*. The organism, its genes, metabolism etc are represented *in silico* using mathematical models of the organism called genome-scale metabolic models (GEMs). They will be evolved using a quality-diversity evolutionary algorithm which is inspired by the process of macroevolution (King et al., 2016; Mouret, 2020). Macroevolution is the hypothesis that life evolves by selecting the fittest individual in each niche as opposed to global selection between all individuals independent of the niche they occupy (Stanley, 1975).

In the next two chapters, the state-of-the-art regarding metabolic networks, genome-scale metabolic network models and evolutionary algorithms is described. This insight allows for understanding why combining evolutionary algorithms with genome-scale modelling can be interesting for metabolic engineering. By combining GEMs and evolutionary algorithms, we want to speed up metabolic engineering. We will start by exploring the *in silico* evolution of *Escherichia coli* to maximize the production of simple metabolites such as succinate. Afterwards, the production of acetate, ethanol and flavanones in *Escherichia coli* is investigated. In order to find out which gene knockouts optimize the production of a specific compound, the workflow in Figure 4.1 is followed. First, the appropriate metabolic model has to be chosen. This model will then evolve *in silico* via the MAP-Elites algorithm, to optimize the model towards the chosen metabolic engineering objectives which might include the production of certain metabolites or the growth on specific media. Details about the MAP-Elites algorithm are explained in Section 3 and 4.

2. METABOLIC MODELS

2.1 Metabolic networks

2.1.1 Definition

Metabolic networks are interconnected pathways of biochemical reactions within living cells through which building blocks or compounds necessary for cellular functioning are assembled (anabolism) or energy and matter are produced by breaking down biomolecules (catabolism) (Walhout et al., 2012). Metabolic pathways often form chains or cycles of reactions coupled to each other, in which the product of one reaction may serve as the substrate of the next one (Dubitzky et al., 2013; Chalancon et al., 2013). Next to the metabolic pathways, a metabolic network also includes all the regulatory interactions that constrain the (bio)chemical reactions. These regulatory interactions can include transcriptional and translational regulation, enzyme kinetics, enzyme inhibition etc (Herrgård et al., 2006). Additionally, the metabolic network also describes the interactions between metabolic pathways and environmental factors such as microorganisms, diet, xenobiotics, etc (Renz et al., 2021). However, thermodynamic gradients within metabolic reactions may favour one specific direction. This consequently results in flux through the metabolic pathways. Furthermore, several parts of the reaction network are temporally and spatially organized by compartments (Dubitzky et al., 2013). This is especially obvious in eukaryotic cells as they have many compartments to separate different processes (Gabaldón and Pittis, 2015). Examples of such compartmentalization include the endoplasmic reticulum, the Golgi apparatus, the nucleus, mitochondria, endosomes, lysosomes and peroxisomes (Martin, 2010). However, even in bacteria, there are examples of compartmentalization such as protein-bounded microcompartments for carbon-fixing carboxysomes and lipid-bounded organelles like photosynthetic membranes in cyanobacteria (Cornejo et al., 2014). Metabolic networks are often visualised by a network consisting of nodes that represent metabolites and edges that represent fluxes of (enzymatic) reactions.

2.1.2 Origin and evolution of metabolic networks

The study of the origin and evolution of metabolic pathways is one of the most important aspects of the search for the origin of life. Fani (2012) discusses the emergence of metabolic routes in primordial cells. The RNA world hypothesis proposes that life on Earth started with a simple self-replicating RNA molecule (Neveu et al., 2013). In the RNA-protein world hypothesis, RNA can subsequently catalyse protein synthesis (Fani, 2012). Primordial heterotrophic cells that were able to synthesize amino acids and other organic molecules had a huge evolutionary advantage, as the exhaustion of the prebiotic supply of amino acids and other molecules that were most likely present on the primordial Earth imposed

an important selective pressure (Fani, 2012). The ability to synthesize amino acids and other organic molecules avoided the dependence of primitive life forms on exogenous sources of organic compounds. Four often referenced hypotheses for the origin of metabolic pathways are the *retrograde hypothesis*, *Granick's hypothesis*, the *patchwork hypothesis* and the *shell hypothesis* (Scossa and Fernie, 2020). The *retrograde hypothesis* states that metabolic pathways originate with sequential gene duplications starting from gene catalyzing the last step of current pathways (Fani, 2012). Once certain compounds were depleted in the primordial soup, selective pressures originated favouring the survival and reproduction of the primordial cells able to produce the depleted compounds. Such a process may have been repeated sequentially, in a backward direction, until the pathways we observe today were established. *Granick's hypothesis* on the other hand states that pathways may have been assembled in a forward direction. Starting with simple precursors and towards more complex products. This hypothesis assumes that the older genes across the evolutionary timescale would be represented by those catalyzing the earlier steps in contemporary pathways (Scossa and Fernie, 2020). The *patchwork hypothesis* states that ancestral genes encoding promiscuous enzymes may have expanded the metabolic capabilities of primordial cells via gene duplication and subsequent divergence. Additionally, subfunctionalization as seen in Figure 2.1 (c) is an example where the catalytic activities of an ancestral gene are divided among the paralogs. Lastly, the *shell hypothesis* proposes that the evolution of metabolism can be traced back to the consecutive additions of distinct metabolic pathways. The metabolic network is divided into shells, shell A - the core central pathway (reductive TCA, fatty acids biosynthesis) - which is predated by the addition of nitrogen metabolism in shell B. Sulphur and cofactor metabolism were later added as shell C (Scossa and Fernie, 2020). The difference between the hypotheses can also be observed in Figure 2.1.

The Krebs cycle is a central part of the metabolism, is conserved in almost all organisms and is therefore of particular interest because it can give insight into the origin of metabolic pathways and thus life. Lane (2015) lays out how and where life might have started. One of the primary aspects in the search for the origin of life is where and in which forms life initially obtained energy. All cells similarly manage energy by relying on an electrochemical gradient across a membrane to power the chemical reactions of life. Lane argues that this electrochemical gradient across a membrane could not have originated in ordinary conditions like open oceans or in what Darwin called "warm little ponds". Rather, he follows a similar train of thought as Günter Wächtershäuser (Huber and Wachtershauser, 1997), that life must have begun in deep-sea hydrothermal vents because, unlike the open oceans, these hydrothermal vents contain chemicals that can store energy that could be used by cells, given that the cells have a membrane to generate the needed gradient by maintaining different concentrations of chemicals on both sides of the membrane. Some steps of the reverse Krebs cycle can be catalyzed by minerals or by metal ions such as iron as reducing agents under acidic conditions. This means that the Krebs cycle is not a set of reactions created by life, but more likely repurposed by primordial cells to synthesize amino acids and other organic molecules to gain an evolutionary advantage by avoiding the exhaustion of the prebiotic supply of amino acids (Lane, 2015).

Next to the origin of metabolic networks, their robustness is also of great importance in the quest for a better understanding of metabolic networks. The robustness of a metabolic network is its capacity to remain unaffected when perturbed. Perturbations can include genomic mutations, that knockout certain reactions in the metabolic network, and changes in the environment. The biochemical pathways in metabolic networks are assembled so that the output of the pathway is relatively insensitive to these

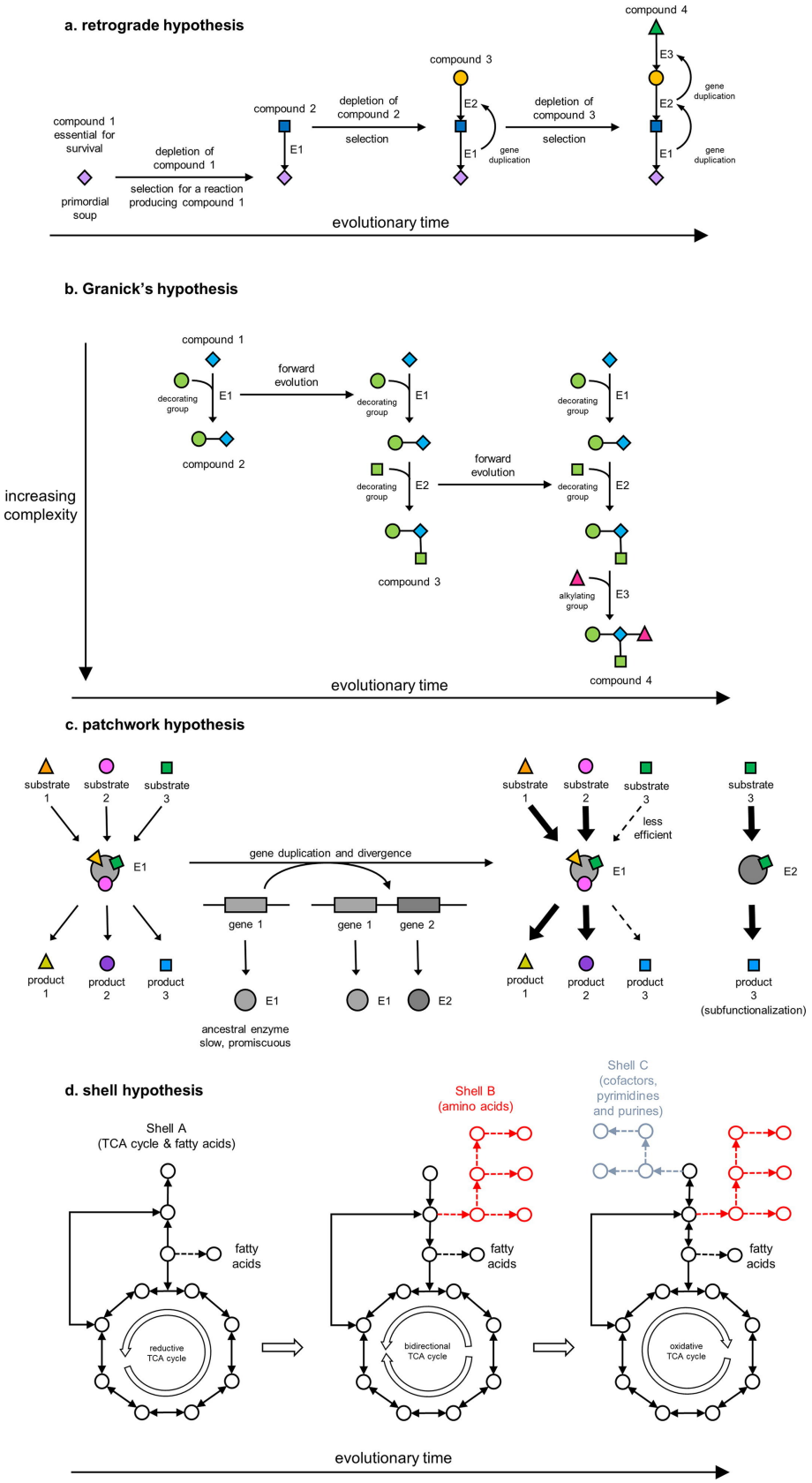


Figure 2.1: Schematic representation of the four main hypotheses for the evolution of metabolic pathways. (a) The *retrograde hypothesis* (b) *Granick's hypothesis* (c) The *patchwork hypothesis* (d) The *shell hypothesis* (Scossa and Fernie, 2020).

perturbations. For organisms that are highly adapted to their environment, robustness to mutations may be lower compared to organisms that are less adapted to their environment. This finding suggests that there is a trade-off between the efficiency of an organism in one specific niche and its ability to cope with change, both internal and external (Marashi et al., 2013).

2.2 Modelling metabolic networks

Metabolic network modelling is a technique that allows for new insights into the molecular mechanisms of a certain organism. Metabolic network modelling makes use of mathematical models of metabolic networks to correlate the genome of an organism with its phenotype (Francke et al., 2005). Metabolic network modelling includes metabolite balancing, which is the basis for metabolic flux analysis (Christensen and Nielsen, 1999). Flux balance analysis (FBA) is a mathematical tool for the analysis of the flow of metabolites through a metabolic network. Calculating the flow of metabolites through the metabolic network allows the prediction of the growth rate of an organism or the rate of production of metabolites of interest. As seen in Figure 2.2, after metabolic network reconstruction and representing the metabolic reactions in a mathematical form called the stoichiometric matrix (S matrix), this matrix can be used as a constraint on the flow of metabolites through the metabolic network. In the most simple case, there are two types of constraints. Mass balance constraints balance reaction inputs and outputs to ensure that the total amount of any compound being produced is equal to the total amount that is consumed. Secondly, each reaction can also have upper and lower bounds. These define the maximum and minimum allowable fluxes of the reactions. Both the balances and bounds create allowable space of flux distributions of a given system. Figure 2.2(d)-(e) shows how the solution space is shrunk by imposing the constraints. Lastly, the objective function allows to find one or multiple optimal solutions within the allowable solution space. A common example of an objective function is the growth rate, represented by a biomass function, which is composed of essential metabolites needed for growth (Fang et al., 2020). The optimal solution(s) are computed via linear programming to solve the set of linear equations formed by the mathematical representations of the metabolic reaction and the objective function. Simulating different conditions using FBA can be done by changing the constraints of the model. Environmental conditions like the availability of certain substrates can be changed by altering the bounds of exchange reactions. Gene knockouts are simulated by setting the flux of limiting reactions to zero (Orth et al., 2010).

Metabolic network modelling acts as a scaffold to integrate a lot of different types of (omics) data. FBA on the other hand act as a tool to use this data to gain new insights into metabolism. For example, *In silico* experiments can be done to predict the maximum growth rate of *Escherichia coli* in the presence and absence of oxygen. Likewise, the phenotypes and capabilities of organisms can be analyzed in various other environmental contexts (such as different media) and multiple genetic perturbations can be explored (Orth et al., 2010). The absence of kinetic parameters means that FBA cannot predict metabolite concentrations. It is only capable of determining fluxes at steady state. These fluxes are usually expressed in $mmol.(gDW.h)^{-1}$ or concentration per gram dry weight of cells and hour. More complex FBA can take into account regulatory effects, such as activation of enzymes by protein kinases or regulation of gene expression, in order to improve the accuracy of predictions (Orth et al., 2010). Metabolic

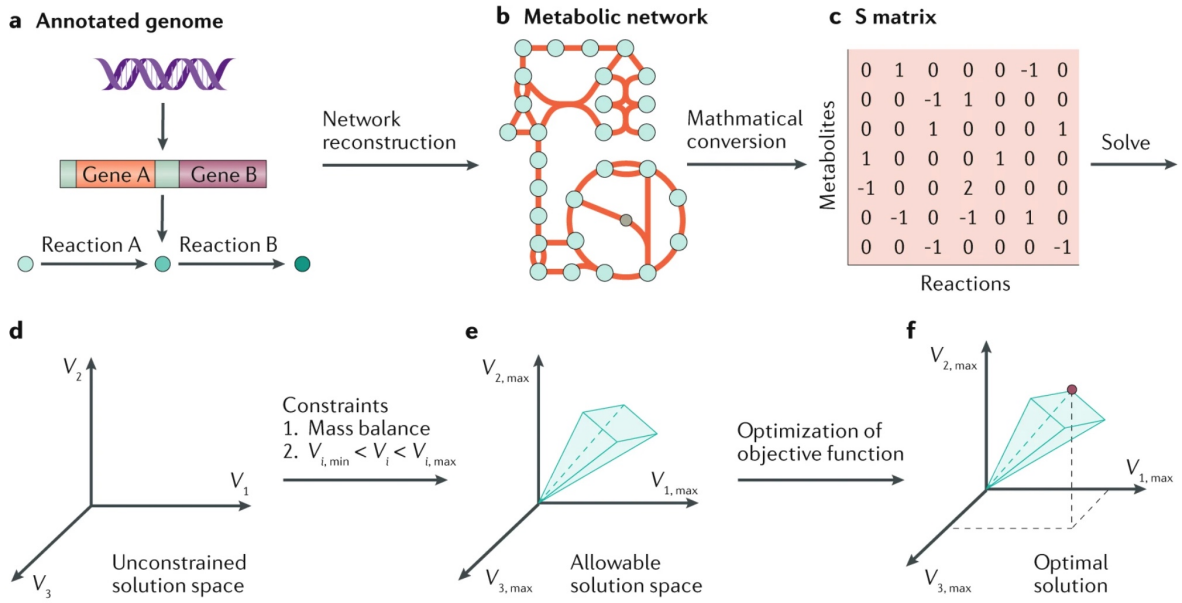


Figure 2.2: (a) Annotated metabolic genes of interest and metabolic knowledge allow to understand the metabolic reactions. (b) The metabolic network for the organism of interest is constructed by integrating all the metabolic reactions and linking shared metabolites. (c) Next, the metabolic network information is converted into a stoichiometric matrix (S matrix). The rows of the S matrix represent metabolites, the columns metabolic reactions. Each entry represents the reaction coefficient of a particular metabolite in a reaction. (d) Using linear programming, the S matrix and an objective function for the model, one can solve for the flux distributions. The solution space consists of all possible solutions of flux distribution. Each axis represents the metabolic flux of a given reaction. (e) Taking additional constraints into account will further narrow the solution space. Background knowledge is used to determine these constraints and include steady-state assumptions and sensible ranges of metabolic flux. (f) The red dot represents the optimal solution that optimizes the objective function of the model, although there might be multiple optimal solutions (Fang et al., 2020).

networks and regulatory effects should be configured in such a way that allows for the comparison of predictions of behaviours to experimental observations.

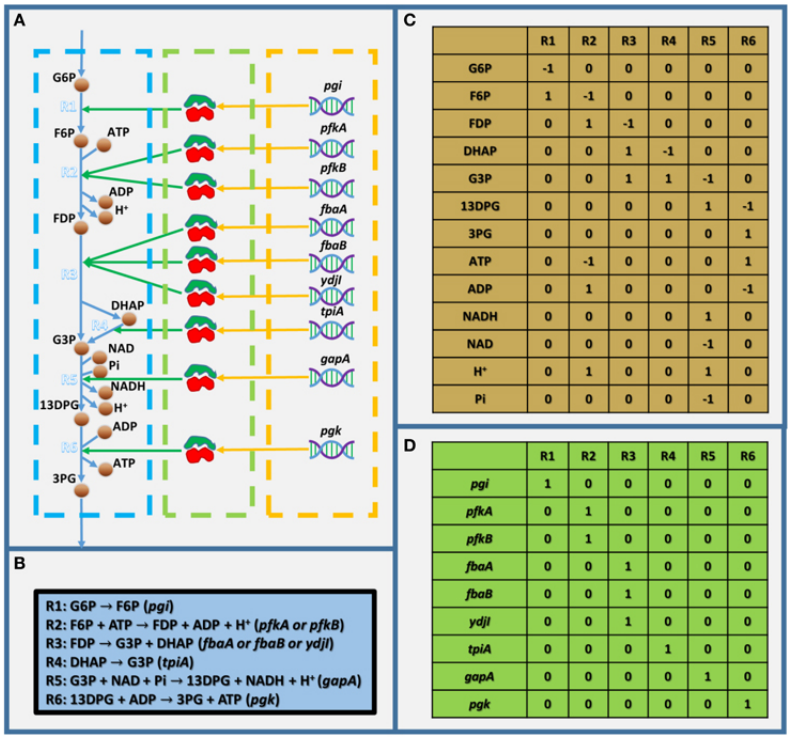
2.3 Genome-scale metabolic models

2.3.1 A systems biology approach

Systems biology connects the study of molecular processes over different scales, from cells to tissues to organs and even whole populations to the phenotype and physiological functions of an organism via computational modelling, high-throughput experiments and quantitative logic. Via these computational and mathematical analyses and modelling of complex biological systems, systems biology can provide systems-level insights into the dynamics within cells, tissues, organs and organisms and interactions at the various scales between them (Tavassoly et al., 2018). The systems-level framework allows for the search for global phenotypic effects of up-and down-regulation of gene expression, gene knockouts

and gene insertion (de Oliveira Dal'Molin et al., 2010). Contrary to the more traditional approach of reductionism, system biology brings a holistic approach to biological research (Tavassoly et al., 2018).

Genome-scale metabolic models (GEMs) are networks connecting all metabolic reactions of a specific organism to their associated metabolites, proteins and genes (Richelle et al., 2020). Genome-scale metabolic network reconstruction provides a mathematical representation of the metabolism of an organism. GEMs are built upon the principles of systems biology in order to obtain a global view of how metabolism works. GEMs are stoichiometric-balanced networks. This implicates a mass balance, energy balance, reduction and proton balance (Zhang and Hua, 2016). The simplest type of GEMs obtain the gene–reaction–metabolite connectivity via two matrices. One matrix associates metabolites to reactions, while another matrix links reactions to corresponding enzymes and genes (Wang et al., 2021). They can subsequently be simulated via flux balance analyses. The two matrices can be seen schematically in parts (c) and (d) of Figure 2.3. More complex GEMs can incorporate transcription and translation. These models are called GEMs of metabolism and macromolecular expression or ME-models and are elaborated on in Section 2.3.3 (Dahal et al., 2021). GEMs are made up of collections of existing knowledge of the metabolism of a specific organism. Most of the time, it is also assumed that the metabolic network is complete. Gap finding and gap filling are two functionalities used in software written specifically for GEMs to solve any exceptions. It is also assumed that there is no (unforeseen) accumulation of metabolites within the metabolism (Zhang and Hua, 2016).



	R1	R2	R3	R4	R5	R6
G6P	-1	0	0	0	0	0
F6P	1	-1	0	0	0	0
FDP	0	1	-1	0	0	0
DHAP	0	0	1	-1	0	0
G3P	0	0	1	1	-1	0
13DPG	0	0	0	0	1	-1
3PG	0	0	0	0	0	1
ATP	0	-1	0	0	0	1
ADP	0	1	0	0	0	-1
NADH	0	0	0	0	1	0
NAD	0	0	0	0	-1	0
H+	0	1	0	0	1	0
Pi	0	0	0	0	-1	0

	R1	R2	R3	R4	R5	R6
pgi	1	0	0	0	0	0
pfkA	0	1	0	0	0	0
pfkB	0	1	0	0	0	0
fbaA	0	0	1	0	0	0
fbaB	0	0	1	0	0	0
ydjI	0	0	1	0	0	0
tpiA	0	0	0	1	0	0
gapA	0	0	0	0	1	0
pgk	0	0	0	0	0	1

Figure 2.3: Toy example of the basic structure of a GEM. (a) Schematic toy model showing the genes that code for enzymes, which in their turn catalyze the biochemical reactions of the metabolic network of the GEM. (b) The biochemical equations of the toy model. (c) The stoichiometric matrix of the toy model that associates metabolites with the reactions. (d) The gene-reaction association matrix that links the biochemical reactions to corresponding enzymes and genes. In (a), the dashed blue, green, and orange frames are the metabolic reactions, enzymes, and genes, respectively (Zhang and Hua, 2016).

GEMs can be downloaded from databases such as modelSEED or the Biochemical, Genetic and Genomic knowledgebase of large scale metabolic reconstructions (BiGG) (Henry et al., 2010; King et al., 2016). The BiGG database will be elaborated on further in Section 4. There are also many automated tools to create genome-scale models such as AutoKEEGRec which makes use of the Kyoto Encyclopedia of Genes and Genomes (KEGG) databases (Karlsen et al., 2018).

2.3.2 Potential of genome-scale metabolic models and integration of omics data

The developments of sequencing technologies in the last 20 years have made it possible to sequence complete genomes of many organisms going from bacteria to humans and subsequently reconstruct their metabolic networks. This is further aided by the rise of the omics field as a whole. Everything from genomics, transcriptomics, proteomics, metabolomics, fluxomics, metabolomics, metagenomics and even thermodynamics can be used to get a better idea of how a metabolic network functions (Zhang and Hua, 2016). Simultaneously, the reconstructed network can provide a relevant biological functional context to the visualization of omics data to improve the interpretation of those data (Francke et al., 2005).

The gene-protein-reaction associations within GEMs are composed based on genome annotation data and information obtained by experiments. Omics data plays a large role in making these models as accurate as possible with respect to the simulations and predictions of the actual cell metabolism (Zhang and Hua, 2016). Additionally, GEMs can also act as scaffolds for systematic integration and analysis of omics data and enzyme kinetic data to get a better overview and understand how different observations correlate to each other (Kim et al., 2015). The integration of additional biochemical information not only yields insight into the metabolism of an organism but also provides answers to questions about the cellular processes within a cell that goes beyond the metabolism and enables the reconstruction of the relationships among genes, enzymes and metabolism (Zhang and Hua, 2016). Examples of these types of additional biochemical information that have already been used in GEMs include protein allocation, cellular macromolecular composition and protein structural information. However, other biochemical properties, like enzyme-substrate interactions, the structure of protein-protein complexes, and post-translational modification could be of interest in future developments (Gu et al., 2019). The integration of this data improves the accuracy of metabolic phenotypes predictions by using omics data as additional constraints (Kim et al., 2015). This can be seen in Figure 2.4.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) is one of the best-known metabolic network databases (Kanehisa and Goto, 2000). In contrast to general metabolic pathway databases such as KEGG, GEMs allow for system-level metabolic response analyses and flux simulations (Zhang and Hua, 2016). Because of the evolution of genome sequencing and other omics analyses, the quality of newly developed GEMs increases and new opportunities to use GEMs open up. Together, genome sequencing, omics analyses and GEMs have already contributed to an increased understanding of metabolism in various organisms such as the model organisms *Escherichia coli*, and *Saccharomyces cerevisiae*, but also many multicellular organisms like plant and human metabolic networks have been reconstructed. The reconstruction of their metabolism has allowed for the construction of a broad spectrum of metabolic

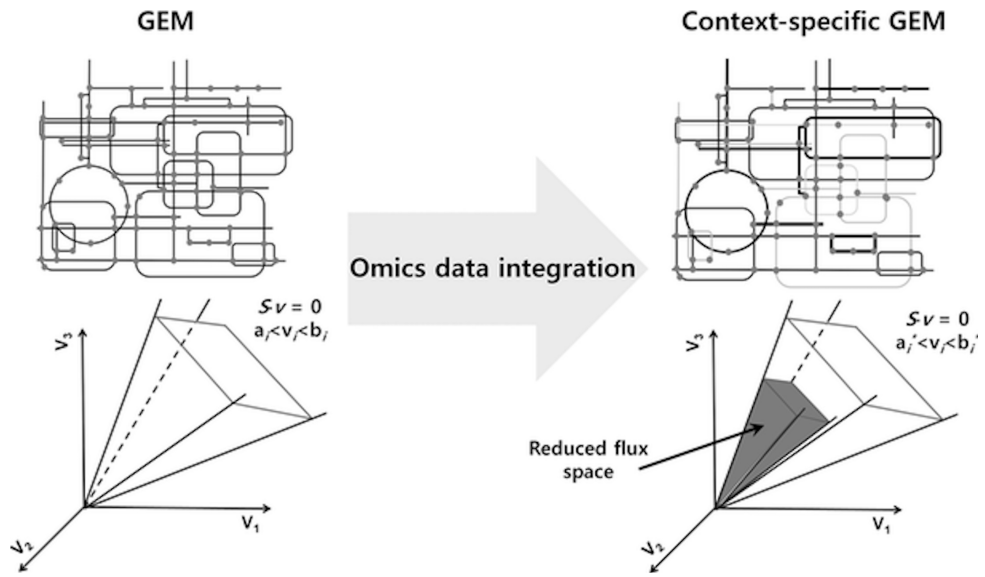


Figure 2.4: The integration of omics data provides a context-specific genome-scale metabolic network model (S) under specific genetic and environmental conditions with improved accuracy of metabolic phenotype predictions by using omics data as additional constraints and thus reducing the space of possible flux distribution (Kim et al., 2015).

studies by generating model-driven hypotheses and implementing various context-specific simulations using GEMs (Gu et al., 2019).

2.3.3 Genome-scale metabolic models of metabolism and macromolecular expression

GEMs are excellent scaffolds for the integration of all different types of data such as omics data, enzyme kinetics, regulatory information etc. While models from databases such as BiGG are interesting because of their standardisation, accessibility and ease of use, sometimes more complex models are required, as the *in silico* results may differ from real-world findings. BiGG models and similar models that focus on their metabolism are often referred to as metabolic (M-)models. These metabolic models are designed to compute the metabolic states of a metabolic network or organism subject to diverse genetic perturbations and environmental cues. An expansion of M-models towards the prediction of both metabolic and proteomic states of an organism by the addition of an expression (E-)matrix resulted in the creation of GEMs of metabolism and macromolecular expression or simply ME-models (Dahal et al., 2021). The expression matrix is comprised of the macromolecular biosynthesis pathways of transcription and translation (Dahal et al., 2020). The incorporation of multiple levels of knowledge going from transcription, and translation to macromolecular resource allocation is a clear advantage of ME-models over the M-models. The additional information allows ME-models to predict the metabolic and protein expression states and the optimal proteome of a cell under various (growth) conditions. Additionally, ME-models can also be used to research the metabolic outcomes of changes in membrane proteins (Dahal et al., 2021). By doing so, ME-models widen the range of applications for GEMs, while simultaneously delivering even better predictive capabilities in general and also specifically for gene expression under numerous conditions such as variation in temperature, external pH and oxygen availability (Dahal et al., 2020). Both the GEMs discussed before and ME-models are based upon optimization and compu-

Table 2.1: Comparison between M- and ME-models for multiple features (Dahal et al., 2020).

Features	**M-models**	**ME-models**
Biological scope	Metabolism	Metabolism and macromolecular expression
Growth simulation capability	Nutrient-limited growth	Nutrient- and proteome-limited (batch) growth
Macromolecular (DNA, RNA, protein) composition of biomass	Constant: fixed stoichiometric coefficients in the biomass synthesis reaction	Variable: optimal macromolecular composition is computed by the model
Metabolic pathway usage prediction capability	Unable to distinguish between pathways that differ in protein expression cost	Accounts for cost of expressing pathway enzymes to correctly predict pathway usage under a given condition
Additional predictive capabilities	Context-/tissue-specific flux states, flux constraints due to enzyme mass and catalytic efficiency, dynamics of metabolism, gene expression regulation	Protein translocation, response to thermal stress, oxidative stress, acid stress, dynamics of metabolism and gene expression, thermodynamics-compliant intracellular fluxes
Computational time	Faster: solved using double-precision solvers	Slower: requires quad-precision solvers, or double-precision MILP solver

tation of the optimal cellular (steady) state of an organism considering all physicochemical constraints such as mass balance, thermodynamics and data-driven constraints like transcriptional regulation, enzyme kinetics etc (Zhang and Hua, 2016). Furthermore, the coupling between metabolic networks and proteome allocation in ME-models allows the exploration of the molecular mechanisms underlying the phenotypic alterations of environmental and genetic variations (Dahal et al., 2020). Moreover, the ability of ME-models to predict the proteome investment in the metabolism under any given condition is a significant improvement in the quantitative predictions of gene expression compared to M-models such as BiGG models (Dahal et al., 2020). In table 2.1, a comparison between M- and ME-models is made for multiple features.

Similarly to M-models, ME-models have been applied for extending the knowledge of high throughput data, expanding the knowledge of biological systems, biomedical applications, metabolic engineering applications, bioremediation and even interactions between different organisms (Dahal et al., 2020). Nonetheless, the ability to predict both metabolic and proteomic states opens up many new opportunities. E. coli ME-models have been used to predict deletions of global regulators that lead to proteome reallocation causing increased fluxes toward metabolic processes and decreased fluxes toward translation machinery (Iyer et al., 2020). M- and ME-models have been used to find resource allocation trade-offs for growth-coupled product secretion. After which, *in silico* solutions were compared to their *in vivo* results and a significant amount of common knockout combinations were found. This further indicated the improved accuracy of ME-models for the prediction of growth-coupled production. Fundamental trade-offs between growth rate, yield and biological and physical constraints have been explored using ME-models. Together with experimental data, a set of essential genes was formed to find a minimal set of genetic manipulations needed to reduce resources of the proteome reserved for nonessential functions. Altogether, the predictions were validated using *in vivo* experiments that showed significantly higher production of the target molecule compared to the wild-type strain (Dahal et al., 2021).

2.4 Applications of metabolic models in metabolic engineering

Some examples that have benefitted from the use of GEMs include strain development for the production of bio-based chemicals and materials (Campodonico et al., 2018), drug targeting in pathogens (Mienda et al., 2018), the prediction of enzyme functions (Oberhardt et al., 2016), pan-reactome analysis (Kim et al., 2020), modelling interactions among multiple cells or organisms (Kim et al., 2017) and understanding human diseases (Mardinoglu et al., 2013). From the list above it is clear that besides the scientific insight, GEMs can also aid the development of biotechnological applications (Gu et al., 2019).

This first GEM was developed in 1999 and was a model for *Haemophilus influenzae Rd* metabolic genotype (Edwards and Palsson, 1999). Since then, more GEMs have been developed for an increasing number of organisms across the three domains of life: bacteria, archaea and eukarya (Kim et al., 2017). In the domain of the bacteria, *Escherichia coli*, and *Bacillus subtilis* are two examples of model organisms for which GEMs have been developed and used for optimizing the production of various enzymes and proteins in industrial biotechnology (Kim et al., 2017). One possible application for industrial biotechnology is the *in silico* simulations based on GEMs that guide the rational design of industrial microorganisms (Zhang and Hua, 2016). GEMs can be used for *in silico* metabolic engineering by predicting gene modification strategies for the overproduction of desired compounds and in so doing, speed up the metabolic engineering process as a whole (Kim et al., 2015). Many *in silico* metabolic engineering methods follow a similar approach. First, the preferred strain is defined and secondly, approaches that bring the wild-type strain closer to the desired strain are identified. Examples of approaches that have been used in *in silico* metabolic engineering include reaction or gene knockouts (Ruckerbauer et al., 2014), overexpression and suppression of genes (Chowdhury et al., 2014), knock-ins of foreign pathways (Pharkya et al., 2004) and exchanging co-factors for a chosen enzyme such as NADH to NADPH (King and Feist, 2013). Knockout identification *in silico* is nevertheless easier and therefore more used than up-or down-regulation of genes (Zhang and Hua, 2016). These software frameworks incorporating similar strategies include OptGene (Rocha et al., 2008), OptKnock (Burgard et al., 2003) and OptDesign (Jiang, 2021). Alternatively, reconstructing metabolic networks of microorganisms can help researchers figure out which media are best suited for growth or which pathways could improve the production yield of a specific metabolite. Moreover, studying the metabolic networks of organisms that live today and lived in the past might help us get closer to figuring out what role chemical networks played in the origin of life as discussed in Section 2.1.2 (Seckbach, 2012).

Although the *in silico* possibilities seem promising, there is still a big disadvantage over *in vivo* metabolic engineering. The production of a target product predicted *in silico* is rarely achieved *in vivo* (Zhang and Hua, 2016). One possible explanation is the increased complexity of the behaviour of strains *in vivo* that is not yet able to be reproduced by GEMs. The key assumption behind every method or model should never be forgotten and the results obtained should be used as instructions. GEMs are nevertheless quite suitable for qualitative applications like essentiality analysis and synthetic lethality analysis (Zhang and Hua, 2016). Essentiality analysis tries to identify all the essential genes or reactions whose knockout will disable a specific biological function. It is able to do this through FBA. All single gene or reaction knockouts are enumerated and for each knockout, the chosen biological objective(s) are checked to see if they are still operative. Similarly, synthetic lethality analysis searches for combinations of knockouts

of multiple reactions or genes that lead to a blocked biological function of the target. As GEMs are considered to be complete metabolic networks, they can be of use for gene or reaction essentiality analysis (Joyce and Palsson, 2008).

GEMs for *Mycobacterium tuberculosis* on the other hand have helped in the fight against microbial pathogens and in understanding the condition-specific metabolism of pathogens. Gaining insight into the metabolism of a pathogen at specific lifecycle points at a systems-level has been especially valuable for the discovery of effective drug targets. When the principle of holism from systems biology is applied to medicine, it is referred to as systems medicine, meaning researchers look at the systems of the human body as part of an integrated whole, incorporating biochemical, physiological, environmental interactions and the patient's genomics in order to gain insight into more complex interactions within the human body and develop methods to restore impaired functioning of the human body (Federoff and Gostin, 2009). Moreover, in the domain of the archaea, the GEM of Methanosarcina acetivorans has proven itself to be a valuable resource for metabolic studies on the unique characteristics of archaea in an expansive coverage of habitats such as extreme environments e.g. geysers, hydrothermal vents, but also in the human gut (Gu et al., 2019).

In the domain of the Eukarya, *Saccharomyces cerevisiae*, which is a model organism for eukaryotic microorganisms is also used for various biotechnological applications and has multiple GEMs. Eukaryotes are different from Bacteria and Archaea in the fact that Eukaryotes have multiple compartments including the nucleus in which the DNA resides (Gabaldón and Pittis, 2015). Additionally, Eukaryotes are interesting for the production of proteins with specific post-translational modifications that can not be produced in Bacteria or Archaea (Amann et al., 2019). Plants, despite their greater biological complexity, are also being used. *Arabidopsis thaliana* for example has GEMs that serve as a model for plant metabolisms (de Oliveira Dal'Molin et al., 2010). Maybe the most interesting GEMs are those for Homo sapiens, as traditional research on humans is limited and gaining insight into the biological mechanisms behind various human diseases and subsequently designing suitable disease treatments is of great interest to the scientific community and society as a whole (Gu et al., 2019). Another application GEMs can be used for is to simulate interactions between multiple tissues, multiple (micro)organisms and even between microbiota and human tissues. This allows for the discovery of anti-cancer, anti-viral and anti-parasite drugs and discover more about the relationships between different organisms in health and disease (Zhang and Hua, 2016).

3. EVOLUTIONARY ALGORITHMS

3.1 Introduction

The evolutionary mechanisms of natural selection are based on the principles of variation, inheritance, selection and adaptation (Bard, 2016). It is the apparent simplicity of the process of evolution and the ability to create the most complex and sophisticated designs found on earth that fascinates so many (Adami et al., 2000). It might therefore also be useful to look at these principles when trying to design new systems ourselves. Computer algorithms that make use of evolutionary principles are termed evolutionary algorithms. They belong to the field of evolutionary computation, which in part belongs to the field of computational intelligence. Evolutionary algorithms are population-based metaheuristic optimization algorithms that are often used to help a computer learn a specific task from experimental observations or generated data (Vikhar, 2016). The following three steps are almost always looped over in evolutionary algorithms. First, the fitness of each individual in the population is evaluated via a fitness function that measures their performance at the task. Secondly, the individuals are ranked and afterwards selected using their fitness value. Lastly, variation operators are applied to the best individuals to create a new population before the loop is started again. The initial population is generated randomly. This can be seen in the top part of Figure 3.1. The two most used variation operators are mutation and crossover. Mutation means that random variation is added to a single genome, for example. If the genome is represented as a list of real numbers, mutation can be applied by adding Gaussian noise to some of the numbers. Crossover on the other hand is obtained by combining two genomes of the population. The goal of crossover is the merging of the features of the two genomes. For the example of genomes as a list of numbers, a crossover operation can be implemented by adding a linear combination of the elements of the two parental genomes. Depending on how genomes are represented in the algorithm, however, crossover might not be used (Mouret, 2020). There are many different implementations of evolutionary algorithms including genetic algorithms (Zhi and Liu, 2019), genetic programming (Kovačič and Župerl, 2020), evolutionary programming (Tian et al., 2020), neuroevolution (Jalali et al., 2020) and differential evolution (Deng et al., 2021). Most different implementations differ in the way they represent the genome and the problem they try to solve. For example, in genetic algorithms, the problem is encoded in a series of bit strings that are subsequently tweaked by the algorithm. In evolutionary algorithms on the other hand, the decision variables and problem functions are used as such (Systems, 2017). Evolutionary algorithms can be applied to many single and multi-objective optimization problems, but also for scheduling, planning, design, and management problems (Slowik and Kwasnicka, 2020).

From a computer science perspective, artificial evolution is considered to be a mathematical optimization algorithm. This implies that the algorithm tries to optimize a given objective. These algorithms can

therefore be used for many applications in fields such as engineering, bioinformatics, logistics and machine learning as there are problems in these fields that can be formalized as the maximization or minimization of a numerical objective. The field of optimization algorithms is vast, but one of the advantages of evolutionary algorithms is that they can be used for functions for which the analytical gradient cannot be computed. These functions are also referred to as black-box functions (Mouret and Clune, 2015). In addition, evolutionary algorithms have also been used in the field of Artificial Life (AL) to simulate evolution based on crossover, mutation and selection for the creation of artificial organisms (Komosinski and Ulatowski, 1998). If we now look at evolutionary algorithms from the standpoint of biology, they model natural evolution in that all individuals compete against each other, the fittest can reproduce and the least individuals fit will disappear (Back, 1996).

Unlike in nature, the repeated variation–selection loop of natural selection leads these algorithms to converge on a single genome instead of the large variety of genomes that can be observed in nature. During microevolution, a fitness gradient in one specific niche is followed, while during macroevolution multiple niches with a variety of species are filled. The process of microevolutionary or intra-niche evolution is, therefore, more suited when describing the convergence of traditional evolutionary algorithms where only the fittest individual remains. Microevolution is defined as evolution within a single ecological niche in which all individuals compete together, often until there is only one dominant species. Macroevolution, however, is the creation of new species that colonise diverse niches (Mouret, 2020). While global competition governs microevolution, macroevolution is governed by local competition. During local competition, solutions that are similar to each other compete for particular niches (Clune J, 2019). Macroevolution is able to explain the diversity of species as a consequence of the diversity of niches found on earth. This also brings us to the point of a new variety of evolutionary algorithms, namely those that follow the principles of macroevolution instead of microevolution (Mouret, 2020). Some scientists view that the wonderful creations of biological evolution are less a result of the perfect optimization of each species and more in the diversity between species. The ability of evolution to create such diverse solutions is also the reason that the modern view of artificial evolution is moving away from microevolution to macroevolution. Macroevolution algorithms have already been successfully applied to the evolution of gait repertoires (Cully et al., 2015), diverse designs for aerodynamic bikes (Gaier et al., 2018) and designing modular robots (Nordmoen et al., 2021).

3.2 Novelty search

The fitness function has long been a user-defined input to evolutionary algorithms used as a compass to guide the algorithms towards the objective. Nonetheless, in some cases, fitness or objective function may be a false compass, which is called deception. One example of deceptive problems are trap functions (Deb and Goldberg, 1993). Trap functions are characterized by points in their solution space with large basins of attraction that correspond to local optimal solutions and by points in the solution space with relatively small basins of attraction that correspond to global optimal solutions (Blum and Dorigo, 2004). Deception is therefore obviously a fundamental problem in search problems as it can cause search algorithms to get stuck in attractive local optima (Stanley and Lehman, 2015). Local optima are solutions that are optimal within a neighbouring set of possible solutions as opposed to a global op-

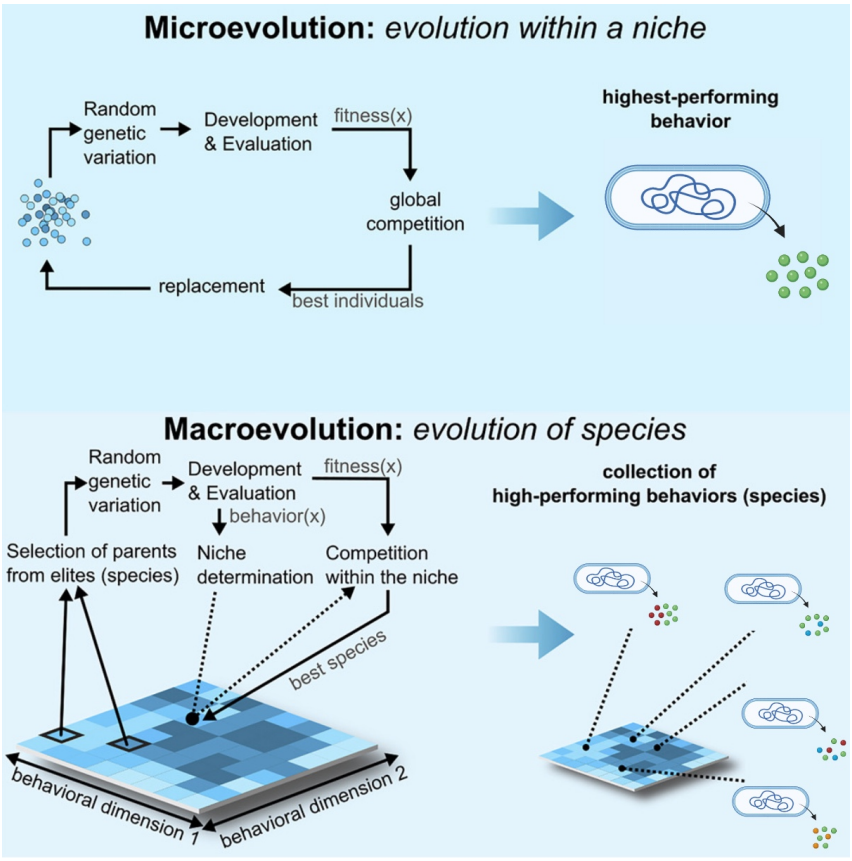


Figure 3.1: Schematical representation of the difference between microevolution and macroevolution. Modified from (Mouret, 2020).

timum, which is the optimal solution between all possible solutions (Lehman and Stanley, 2011a). The stepping stone problem is the problem of how the intermediate steps during the search for a specific solution are frequently very different and often seemingly unrelated to the objective (Clune J, 2019). One of the main questions posed to Darwin was, “What use is half a wing?" (Mouret, 2020). Similarly, cars were not discovered while searching for a way to improve horses. The main problem is that the objective function may not recognize the stepping stones in the search space that ultimately lead to the objective (Lehman and Stanley, 2011a).

As an answer to the stepping stone problem and in order to avoid local optima and deception, a new idea was proposed, namely novelty search (NS). Novelty search is the idea of looking past the fitness function and focusing on generating novel solutions, suggesting counter-intuitively that ignoring the objective in this way could benefit the search for the objective (Stanley and Lehman, 2015). The idea behind the novelty-driven heuristic is that when evolution has exhausted low hanging fruit, it will have to increase the complexity of solutions that will create potential stepping stones towards an optimal design (Clune J, 2019).

In terms of computer programming, novelty search proposes to substitute the fitness function with a novelty score, in order to obtain as many diverse solutions as possible. The novelty score is obtained by comparing the solutions to all the others or a subset of individuals generated so far. Here, diversity is characterized by the feature or behavioural space to which the solutions belong. High-dimensional spaces can also be visualized in low dimensional representations to reveal interesting properties (Mouret

and Clune, 2015). Counterintuitively, novelty search is able to come up with higher quality solutions or find solutions faster than objective-based searches (Risi et al., 2009). The ability to avoid local optima during optimization is one of the main benefits of novelty search.

Novelty search has been successfully used to optimize the design of cancer-targeted drug delivery systems by alleviating the difficulties of deceptive objective function landscapes (Tsompanas et al., 2020). Additionally, novelty search combined with NeuroEvolution of Augmenting Topologies (NEAT) was used for the evolution of neural controllers for homogeneous swarms of robots (Gomes et al., 2013). Gomes et al. (2013) showed that novelty search is able to find solutions with lower complexity compared to traditional fitness-based evolution and can also find a broad diversity of solutions for the same objective. The last example of novelty search is the successful creation of diverse and feasible game levels (Liapis et al., 2013). Liapis et al. (2013) were also able to show that algorithms using novelty search can produce larger and more diverse sets of feasible strategy game maps than existing algorithms.

3.3 Quality diversity algorithms

As innovations often come from combining existing principles, new optimization methods have been created that combine novelty search with objective-based optimization. These algorithms enjoy the benefits of novelty search while still driving the population toward the preferred goal. In doing so, wasteful computing is avoided when search spaces are too large to be explored in their entirety. However, it is still debated how exactly novelty search and fitness functions should be integrated. On the one hand, it has been demonstrated in experiments that fitness functions can have a detrimental influence on the evolutionary process of novelty search (Lehman and Stanley, 2011c). On the other hand, most of the time diversity is not the actual goal, but merely a tool to explore more widely. On top of that, computational resources are most often limited, so exploring the solution space in all directions using novelty search is most likely to be infeasible in many interesting problems (Mouret, 2020). Quality diversity (QD) algorithms or Illumination algorithms are an example of algorithms that combine a novelty search approach with objective-based optimization. They achieve this by searching for a set of solutions that are diverse as possible while still being extraordinarily high performing. QD algorithms are therefore similar to macroevolution as the "ecological" niches are explicitly implemented in the algorithms. Consequently, QD algorithms explore ways of maximizing the objective or fitness function, similarly to how natural evolution searches for multiple paths to maximize reproductive success or fitness in each distinct ecological niche. Nevertheless, QD algorithms are still a rough abstraction of macroevolution as they still require a function to decide to which niche each solution belongs (Mouret, 2020). The main idea is that sometimes, diversity in how a problem is solved is more noteworthy than simply finding the single most efficient solution (Clune J, 2019). A downside of QD algorithms compared to real open-endedness is that it can not observe what it was not designed for initially, it might miss unique innovations (Lehman et al., 2008). If two-dimensional trajectories of robots are investigated, for instance, a flying robot wouldn't be able to be recognized as only the projection of the trajectory to the ground is captured (Mouret, 2020). However, the fact that QD algorithms are modelled after macroevolution instead of microevolutions makes them more capable of generating fascinating stepping stones inspired by the astonishing creativity of natural evolution (Cully and Demiris, 2017; Mouret, 2020).

The main difference between traditional optimization algorithms and QD algorithms is that while optimization algorithms search for the highest-performing solution in a search space, QD algorithms search for the highest-performing solution at different regions of the feature space (Pugh et al., 2016). QD or illumination algorithms can illuminate the fitness potential of each region of the feature space. The illumination of the fitness potential of each region of the feature space in biological terms means the illumination of the phenotype-fitness map (Mouret and Clune, 2015). QD algorithms can nonetheless also be used as optimization algorithms in addition to their ability to illuminate the fitness potential of each region of the feature space.

3.4 MAP-Elites for exploring *in silico* evolution of organisms

The Multidimensional Archive of Phenotypic Elites (MAP-Elites) algorithm is an example of a Quality Diversity (QD) algorithm. The MAP-Elites algorithm is based on the following principles. A fitness or objective function that returns a fitness value. Additionally, solutions are also characterized by an n-dimensional behavioural descriptor or feature descriptor. Considering solutions with n features, the feature space is an n-dimensional space that contains all values for your features solutions. Each solution is characterized by an n-dimensional descriptor, that contains a value for each of the n features. The solution space however is the feasible region defining the set of all possible solutions. The feature or behaviour space is divided into behavioural or ecological niches. The feature descriptor describes which niche the specific solution belongs to. The collection of niches is also called an *archive*. In each niche, the best-known solution for that specific niche is stored. This solution is called an *elite*. In the next step, elites are randomly chosen from the archive and variation operators such as mutation and crossover are applied. The new solution now competes with the existing elite in the relevant ecological niche. An important observation to make is the fact the result of the MAP-Elites algorithm and other Quality Diversity algorithms is a set of high performing solutions (i.e., a multidimensional archive of phenotypic elites) and not a single optima solution as is the case with traditional genetic algorithms (Mouret, 2020). The bottom left part of Figure 3.1 visualises the main loop of the MAP-Elites algorithm.

Search algorithms are used in almost all engineering and science domains to automatically explore a search space to find high-performing solutions (Koziel and Yang, 2011). Traditionally these search algorithms return the single highest-performing solution in a search space (Youssef et al., 2001). MAP-Elites however provides a global view of how the high performing solutions are distributed in the search space. The ability for solutions to jump from niche to niche without being the overall best solution allows MAP-Elites to find even better-performing solutions compared to state-of-the-art algorithms for certain problems (Lehman and Stanley, 2011b). This phenomenon of jumping between niches is also described as goal switching and is a way for algorithms to capture serendipitous discoveries and allows for good (fundamental) ideas to spread. New solutions are built upon the fundamental idea to solve new problems (Clune J, 2019). The MAP-Elites algorithm is an abstract model of the evolution of species or macroevolution as introduced in Section 3.1.

Observations in nature show that each individual niche is colonised by a single species (Baquero et al., 2021). This species is able to outperform all the other species in this specific niche. Tigers don't compete with sharks for example. They are elites in their own specific niche. Tigers don't have a higher fitness

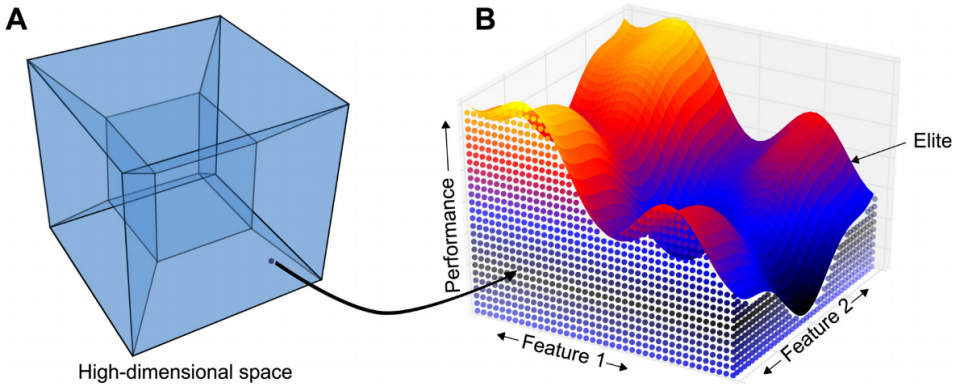


Figure 3.2: The search space of MAP-Elites is often high-dimensional. The MAP-Elites algorithm tries to find the highest-performing solution for each niche in a low-dimensional feature space. The dimensions of variation of interest of that low dimensional space are chosen by the user. This type of algorithm is called an illumination algorithm as it illuminates the fitness potential of each area of the feature space (Mouret and Clune, 2015).

value than sharks, they simply occupy a different niche. When considering tigers and lions, however, they could replace each other in their niche provided that they undergo a few adaptations. Likewise, if the variation operators in the MAP-Elites algorithm improve the solution without altering the behaviour of the solution, it will replace the current elite in that niche. Otherwise, if the behaviour or features and thus their niche becomes different, the solution will compete with the elite in another niche and replace it if the fitness of the new solution is better than that of the current one (Lehman and Stanley, 2011b). Similarly, as to how different parts of the genome of different organisms are highly conserved, high-performing solutions of QD algorithms might be concentrated in a specific elite hypervolume in the genotypic space. This entails that their there genes are similar and makes the cross-over variation operator particularly effective in the MAP-Elites and other QD algorithms (Mouret, 2020). MAP-Elites can identify the performance peaks in a vast search space by actively searching for them. Sampling random solutions is often not a good strategy to find fitness peaks, as the probability of randomly finding such a peak is very unlikely for extensive search spaces (Mouret and Clune, 2015).

The MAP-Elites algorithm has multiple advantages over the current state-of-the-art search algorithms. Firstly, the MAP-Elites algorithm is able to create a lot of diversity between possible solutions in the chosen niches as well as an improved optimization performance compared to the state-of-the-art search algorithms of today. Searching for better solutions within one niche also helps the parallel search for better-performing solutions in different niches, as the probability of generating a solution for a given niche by mutating an elite from another niche or by crossing over two elites from different niches is larger than generating a random solution. This should result in higher-performing solutions compared to the case in which a separate search is conducted for each niche. Additionally, the algorithm returns the archive of phenotypic elites that can illuminate the fitness potential of the whole feature space and not only the high-performing areas. In doing so, new relationships between dimensions of interest and performance can be explored. This is why MAP-Elites is also called an Illumination or QD algorithm.

Notably, there is no guarantee that every niche of the archive will be filled. The first reason is that there may not be a solution that maps to that particular site of the feature space for example. Secondly, even

if there is a possible solution for that niche, MAP-Elites might fail to produce it. Since there are many solutions or genotypes that map to a single niche in the feature space it is not feasible to explore directly in the feature space (Mouret and Clune, 2015).

Multiple variations have been suggested to improve the MAP-Elites algorithm or adapt it for new applications, such as Covariance Matrix Adaptation MAP-Elites (CMA-ME) for derivative-free optimization in single-objective continuous domains for applications including design, testing and reinforcement learning (Fontaine et al., 2020). While currently only one solution is stored per niche, the elite, storing multiple per niche might promote diversity even more. Additionally, instead of choosing random elites to produce new mutants or crossovers, a bias could be implemented towards high- or low- performing areas or niches with empty neighbours (Mouret and Clune, 2015).

Even though QD algorithms such as MAP-Elites try to steer evolutionary algorithms towards open-ended evolution as is present in the natural world, MAP-Elites does not create new niches over time not present in the original feature space. It is therefore by definition, not open-ended evolution (Mouret and Clune, 2015). In nature, however, new niches can be created by organisms that occupy an existing niche. One of the best examples is all the new niches on earth for oxygen-consuming organisms that were created when oxygen began accumulating at a global scale thanks to photosynthetic microorganisms (Berkner and Marshall, 1965; Baquero et al., 2021).

4. MATERIALS AND METHODS

In the following paragraphs, we describe how the MAP-elites algorithm was adapted for *in silico* evolution of metabolic networks. First, a toy problem concerning feature selection is examined, in order to illustrate the benefit of MAP-elites to offer a diverse set of high performing solutions to a combinatorial optimization problem. Next, the MAP-Elites algorithm is applied for simulating evolution in the context of metabolic engineering. First, the optimization of succinate production in *E. coli* by executing combinatorial gene knockouts will be tackled. Next, we elaborate on the case of acetate overproduction and ethanol overproduction. Lastly, the case where a heterologous pathway for the production of flavones is virtually introduced in *E. coli* is discussed. Most code is written in Python 3.9.1, except for a small number of scripts which are written in Julia 1.7.1. All the code used in this master thesis is available on GitHub[1].

4.1 MAP-Elites for feature selection as a toy example

Feature selection is a combinatorial optimization problem, where the number of input variables of a predictive model is optimized for obtaining a maximal predictive performance for a minimal number of input variables (Kuhn et al., 2013). Here, we use MAP-Elites for the task of feature selection, to serve as an alternative to statistical tests and standard genetic algorithms (Hussein et al., 2001). This toy example is meant to illustrate how MAP-Elites can provide a diverse set of solutions with varying complexity to a combinatorial optimization problem. While the original MAP-Elites algorithm from Mouret and Clune (2015) was designed for real valued inputs, this implementation deals with binary input variables.

4.1.1 MAP-Elites implementation

Each solution, consisting of an optimal feature set, is represented as a vector with the indices of the selected features. The number of selected attributes in a given solution is used to determine the niche of each solution. The MAP-Elites for feature selection takes as input the training and test data, together with the number of iterations (I) and the number of initial random solutions (G). The algorithm is initiated by randomly generating G solutions and determining the performance and niche of each one. Those solutions are then placed into the correct niche of the feature space. If multiple solutions map to the same niche, the highest-performing solution - also called the *elite* - is retained. The following part is repeated I times. In each iteration, two random niches in the archive are selected and the elites in those niches are used to create new offspring using variation operators such as crossover and mutation. The niche of the new offspring is determined based on the number of features. If the new offspring performs better than the current elite in its specific niche, the new offspring will become the current elite in its niche. If the niche is empty, the new offspring will automatically become the current elite in that niche.

[1] `https://github.com/shvhoye/OptMAP`

The archive with the best solution found for each niche in the feature space and their corresponding fitness values are ultimately returned (Quinonez et al., 2019).

4.1.2 Data sources

Datasets were obtained from (López et al., 2006) and (Dua and Graff, 2017). The data was preprocessed and split into a training and test dataset for the MAP-Elites algorithm. Four datasets were used, the first three for classification and the last one for regression. The Ionosphere dataset contains radar data, which is used to predict a binary "Good" or "Bad" signal. "Good" radar returns are those showing evidence of some type of structure in the ionosphere, while "Bad" returns are those that do not. "Bad" signals pass through the ionosphere. The Glass dataset is a set of physical characteristics such as refractive index and weight percent of Sodium, Magnesium, Aluminum etc. of seven different glass types. The type of glass is predicted based on its characteristics. The attributes of the Cancer dataset are real-valued features computed for each cell nucleus obtained from a fine needle aspirate of a breast mass, including radius, texture and smoothness. The breast mass is classified as either malignant or benign. The Parkinsons dataset is made up of information about patients suffering from Parkinson's disease such as age and gender and multiple biomedical voice measurements to predict the patients' Unified Parkinson's Disease Rating Scale (UPDRS) score. UPDRS can be used for remote symptom progression monitoring.

4.2 OptMAP: MAP-Elites for *in silico* metabolic engineering

MAP-Elites was implemented for *in silico* evolution of the metabolic network of *E. coli* towards specific metabolic engineering goals, i.e., the production of target metabolites. An overview of the workflow can be seen in Figure 4.1.

4.2.1 Genome-scale metabolic models

Genome-scale metabolic models of *E. coli* were downloaded from the University of California San Diego's BiGG database of GEMs. BiGG is short for Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. BiGG is a collection of genome-scale metabolic network reconstructions comprised of more than 70 published genome-scale metabolic networks. They are grouped into a single database with BiGG IDs which are a set of standardized identifiers. Mapping of the genes in the BiGG models is done using NCBI genome annotations. The metabolites are linked to multiple external databases such as KEGG, PubChem etc (King et al., 2016). High quality curated BiGG metabolic models are accessible via the BiGG website. One can browse the website to search for specific metabolites, reactions, genes or organisms related to GEMs, visualize metabolic pathway maps and export models as an SBML file for further analysis by external software packages. BiGG brings data uniformity and ease of use to the table at a time of increasing numbers of reconstructions and analysis methods. Published genome-scale metabolic networks are integrated into BiGG with standard nomenclature. This allows for the comparison of components across different organisms (Schellenberger et al., 2010).

BiGG models are also compatible with the *Cobrapy* software. The *Cobra* Python library is used to simulate and manipulate GEMs (Ebrahim et al., 2013). Existing genes or reactions in the metabolic models

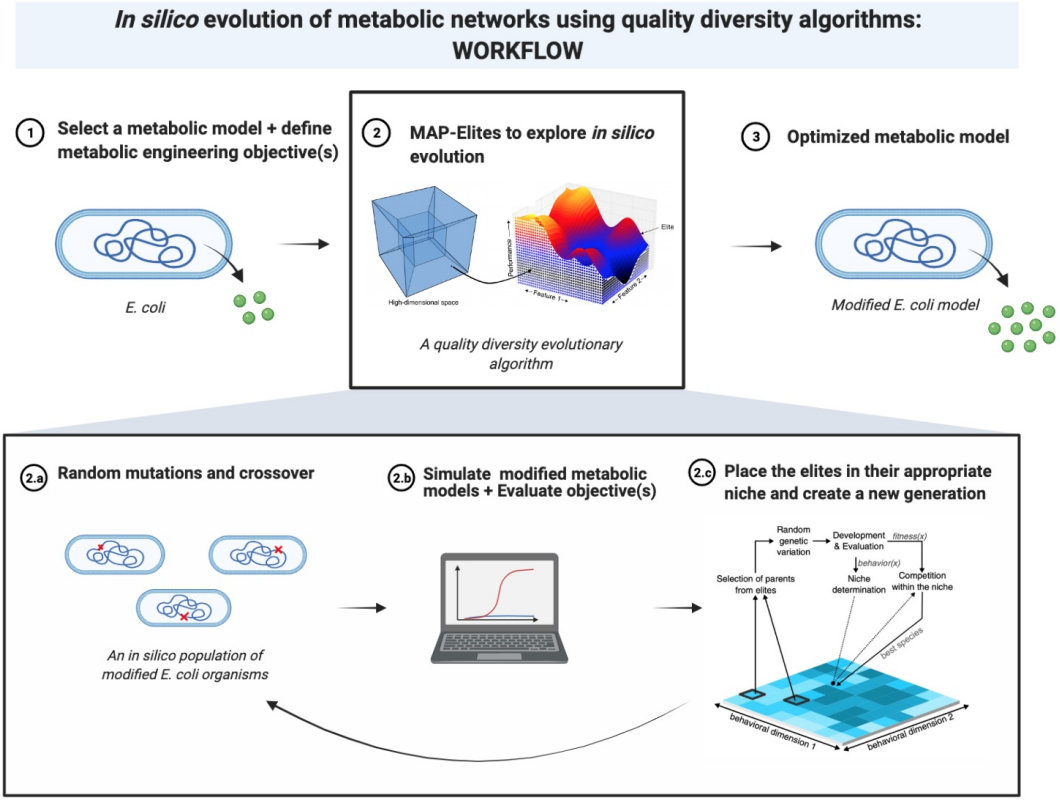


Figure 4.1: A proposed workflow for *in silico* evolution of metabolic networks.

can be knocked out and new reactions can be knocked in for example. *Cobra* provides a simple interface to metabolic constraint-based reconstruction and analysis.

Visualisation of BiGG models is achieved via Escher pathway maps. Escher is a tool for building and visualizing biological pathways. Escher maps are an excellent way to contextualize metabolic data (King et al., 2015). One can visualise metabolites, reactions and labels. In this thesis, Escher is used to visualise knockouts and show how they affect downstream reactions. Curated metabolic maps are available for various organisms and BiGG models (Schellenberger et al., 2010).

For the case studies of succinate, acetate and ethanol production, the *e-coli-core* model for *Escherichia coli str. K-12 substr. MG1655* was employed. This GEM contains 72 metabolites, 95 reactions and 137 genes. Flavanone production modelling was performed by adding the heterologous pathway manually in the iML1515 genome-scale model for *Escherichia coli str. K-12 substr. MG1655*. The heterologous flavanone pathway in *Escherichia coli* is shown in Figure A.1 and is mainly made up of 4-coumarate-CoA ligase (4CL), Chalcone synthase (CHS) and Chalcone isomerase (CHI). The specific reactions that make up the heterologous flavanone pathway that are added to the GEMs are also visualized in Table 4.1. Next to 4CL, CHS and CHI, there are also exchange reactions for coumaric acid (cma) and naringenin (narg) namely EX_cma_e and EX_narg_e respectively. These two reactions are exchange reactions and are conceptual reactions, created only for modelling purposes called pseudoreactions (Courtot et al., 2011). These reactions do not have a physical correspondence, but allow for matter influx or efflux to a model (Courtot et al., 2011). Even though the reactions CMAt and NARGt are also not chemical or en-

Table 4.1: The reactions of the heterologous flavanone pathway added to GEMs iML1515 and iJR904.

Abbreviation reaction name	Reaction formula	Subsystems
EX_cma_e	cma_e $\Longleftrightarrow$	Flavonoid biosynthesis
CMAt	cma_e $\Longleftrightarrow$ cma_c	Flavonoid biosynthesis
4CL	cma_c + atp_c + coa_c $\Longrightarrow$ amp_c + cmcoa_c + pi_c	Flavonoid biosynthesis
CHS	3 malcoa_c + cmcoa_c $\Longrightarrow$ 4 coa_c + chal_c + 3 co2_c	Flavonoid biosynthesis
CHI	chal_c $\Longrightarrow$ narg_c	Flavonoid biosynthesis
NARGt	narg_c $\Longleftrightarrow$ narg_e	Flavonoid biosynthesis
EX_narg_e	narg_e $\Longleftrightarrow$	Flavonoid biosynthesis

zymatic reactions, they do have a physical correspondence as they represent the transport of coumaric acid or naringenin across a membrane into a different compartment or across the cell membrane as is the case here. The iML1515 is the latest genome-scale metabolic model for *Escherichia coli str. K-12 substr. MG1655* and contains 1877 metabolites, 2712 reactions and 1516 genes. The *e-coli-flavanone* model, which is the iML1515 genome-scale model with the heterologous flavanone pathway, contains 1883 metabolites, 2720 reactions and 1521 genes. The later optimization for malonyl-CoA as a major building block for flavanones was done using the iJR904, model which also represents *Escherichia coli str. K-12 substr. MG1655* and has 761 metabolites, 1075 reactions and 904 genes. iJR904 was used for further optimization and troubleshooting as this is a smaller version of iML1515. The increase in size from the *e-coli-core* GEM directly results in a more computationally intensive and longer runtime. This is true for both OptMAP and existing python packages aiding the strain design process in metabolic engineering projects such as OptGene. OptMAP is run multiple times during troubleshooting, so it was better to use the smaller iJR904 model as runtimes were shorter.

4.2.2 Representation of organisms *in silico*

We consider L to be the number of genes that is eligible for genetic modification, i.e., the number of genes that can be knocked out during the *in silico* evolution procedure. While GEMs are used to model *E. coli* and predict its fluxes *in silico*, the genotype of *E. coli* is represented as a binary string of length L, in which each position is linked to a specific gene. Positions with zeros indicate unmodified genes and positions with ones correspond to gene knockouts. Therefore, the initial metabolic network of the *in silico* organism is composed of all genes set at the value of zero in the genotype string.

4.2.3 Representation of evolutionary processes *in silico*

Evolutionary processes in the MAP-Elites algorithm are based on the scheme in the bottom part of Figure 3.1. Individuals are randomly selected from the archive. Elites from these different niches are then used to generate offspring. The variation operators mutation and crossover are used to model the evolutionary processes within MAP-Elites. In k percent of the cases, two elites are randomly selected from the archive and a crossover of the two elites is made. A random position p in the binary string of length L is chosen. The crossover event will then result in a mutant whose first p values are the same as for the first elites, while its remaining values are the same as for the second selected elite. In $(100 - k)$ percent of the cases, however, the binary string of length L of a selected elite is mutated to generate a new mutant. The event of a mutation in a gene of the metabolic network is modelled as converting the value of a specific gene in the binary string from 0 to 1 or from 1 to 0. A random position in the binary string corresponding with one specific gene is selected. If the current value in that position is 0, it will

be changed to 1. The latter corresponds to that knocking out a specific gene. If the current value at the random position is a 1, it will be changed to 0, which reverses the knockout of the gene. Both variation operations are visualised in Figure 4.2. By changing values in the binary string of length L, we adapt the metabolic network of *E. coli*. All genes that correspond to 1 in the binary string will be knocked out in the GEM. Afterwards, the fluxes are predicted using *Cobra* for the given metabolic network. The best mutant in each niche is selected in the variation-selection loop of evolution, to end up with an archive with elites that each is the best performing mutant in their niche.

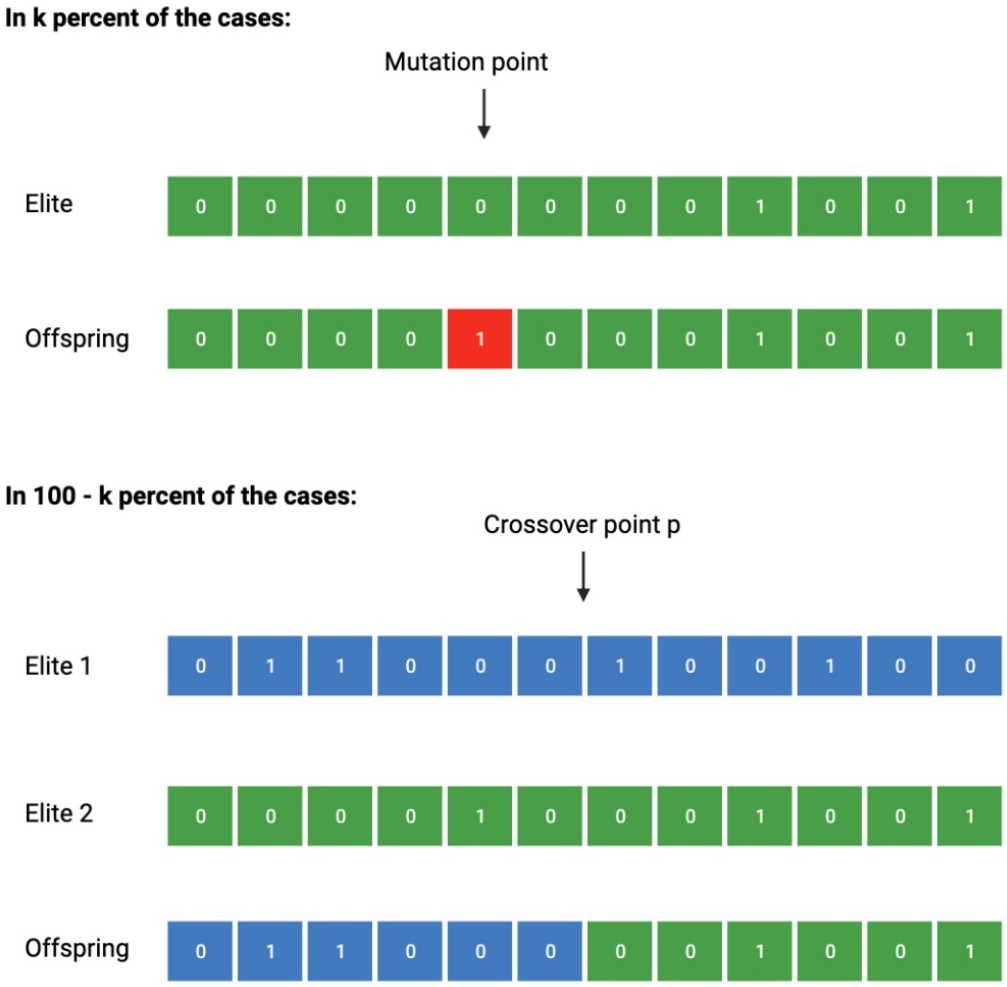


Figure 4.2: Schematical representation of a mutation- and crossover event in MAP-Elites.

4.2.4 Specific implementation of the MAP-Elites algorithm for the combinatorial optimization of metabolic networks

Niche definition

Niches are defined as specific regions in the feature space. The feature space is formed by two behavioural dimensions as visualized in Figure 3.1. The first dimension is the number of knockouts. The number of knockouts is simply obtained by taking the sum of the binary string of length L. The second dimension is the type of metabolism that is influenced by the knockouts. In order to know which type

of metabolism is affected by a gene knockout, a parser was built to search for the gene in the Ecocyc database (Keseler et al., 2017) and to extract Gene Ontology (GO) numbers link to the gene. Next, the GO knowledgebase is consulted to check how many times keywords related to metabolism type such as "sugar", "nucleotide" and "lipid" are mentioned in relation to each GO term (Ashburner et al., 2000; knowledgebase, 2021). The keyword that occurs the most for a specific solution determines the type of metabolism that is most affected by executing the gene knockout, defining the second behavioural dimension of the niche that the solution belongs to. After OptMAP was validated for multiple case studies, it was expanded to incorporate medium compositions as niches. Six different minimal media compositions are considered and shown in Table 5.10. These media mainly differ in carbon source, but also the presence of ammonium and oxygen for example. The medium compositions considered here are calculated by *Cobrapy* to be minimal media, but one could also use different medium compositions used in the wet lab for specific applications if the medium compositions are known.

Fitness criteria

The fitness or performance of a mutant is determined by (1) the flux through the metabolic reaction that yields the target metabolite, (2) biomass production and (3) the carbon source that is consumed. All three are calculated using *Cobrapy*, performing a flux balance analysis on the genome-scale model of the organism in which the reactions that correspond to a gene knockout are deleted. The resulting fitness function is equal to the Biomass-product coupled yield (BPCY) and is defined as:

$$BPCY = \frac{[Biomass] * [Succinate]}{[D\text{-}Glucose]}$$

or

$$BPCY = production\,yield * growth\,rate$$

where BPCY is expressed in gram product per gram glucose per hour. BPCY is a widely used fitness function in *in silico* microbial strains optimization used in for example OptGene, OptKnock and now OptMAP (Choon et al., 2014).

Input parameters

MAP-Elites has four input parameters: the number of generations (I), the genome-scale metabolic model, the metabolite for which the production should be maximized and k the percentage of variation operations that should be crossover. The probability that a variation consists of mutations is therefore $100-k$ percent. Suppose exploration is more important than finding a solution as close to the optimum as possible. In that case, one might opt to choose a larger k compared to when finding the optimal solution is most important. In all further results, a value of 0.05 is used for k to keep a balance between the exploration crossover provides and a gradual approach of the optimum due to mutations. Figure 4.3 illustrates that the performance increases when the algorithm is run for more generations, up until a certain point

after which the performance stagnates. Depending on which metabolite should be maximized and the model that is used, the exact number may vary.

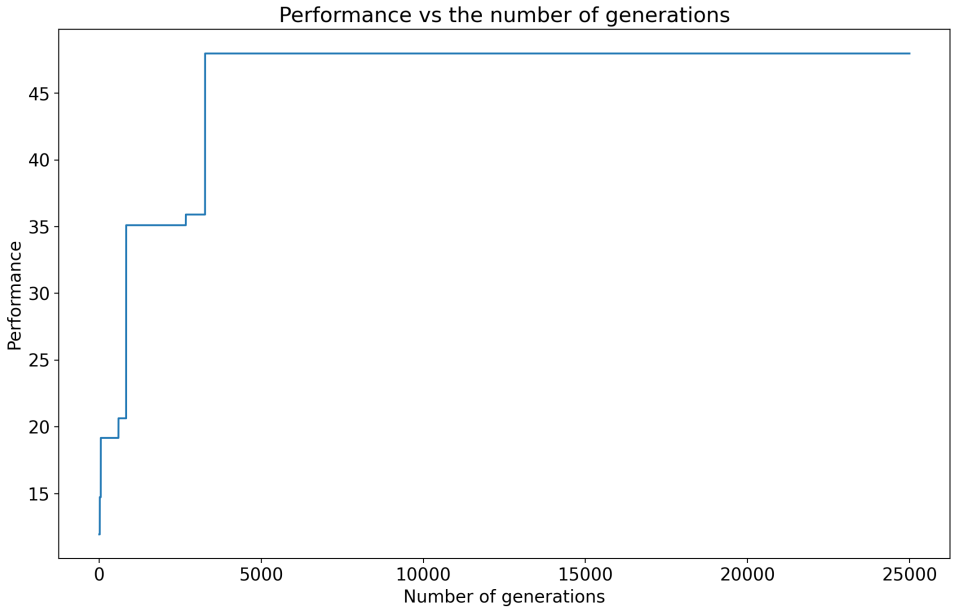


Figure 4.3: An illustration of how the performance evolves with an increasing number of generations. The performance illustrates the BPCY and is measured in gram product per gram glucose per hour.

Initial population

Unlike MAP-ELites for feature selection, OptMAP does not take an input (G) for the number of initial random solutions. Rather, the initial population is generated by executing all possible single-gene knock-outs, after which they are evaluated and compete in their respective niches.

4.2.5 Experimental setup

OptMAP was run on a MacBook with a 2.7 GHz dual-core Intel Core i5-processor and 8 GB RAM. The scripts used to connect genes to the type of metabolism they affect, by searching the EcoCyc and GO database were run on Kaggle (Kaggle, 2022).

5. RESULTS AND DISCUSSION

Metabolic engineering in the wet lab is limited by the number of possible combinations of knock-ins, knockouts, media compositions etc. that can be tested (Lawson et al., 2021). The search or design space created by all the possible combinations of manipulations that are possible in metabolic engineering is so large that navigating it in the wet lab is a long and resource consuming journey (Lawson et al., 2021). Therefore, in this thesis, we introduce a novel computational tool, called OptMAP, which makes use of the quality diversity evolutionary algorithm called MAP-Elites to predict strain design strategies for microbial production of target metabolites. First, a demonstration of the MAP-Elites algorithm applied to feature selection is given as a toy example. Later, four case studies are discussed where OptMAP is applied to the production of industrially relevant compounds.

5.1 MAP-Elites for feature selection as a toy example

The MAP-Elites algorithm can be used for many applications involving combinatorial optimization, as discussed in Section 3.4. A simple example that can illustrate the strengths of MAP-Elites is the feature selection problem. Feature selection is considered to be a combinatorial optimization problem which seeks to reduce the number of input variables when creating a predictive model (Kuhn et al., 2013). Feature selection allows for both a decrease in computational cost associated with modelling and an increase in the performance of the model, which is crucial, particularly with the increasing amounts of data that are becoming available (Brownlee, 2020). A simple brute force algorithm that tests every possible subset of features to find the one that minimizes the error rate is not desirable especially for large datasets, as there are 2^n solutions for n attributes.

Given the combinatorial complexity of the feature selection problem, a variation on the Map-Elites algorithm was implemented for feature selection (Quinonez et al., 2019). This implementation both serves as an example to show the usefulness of the Map-Elites algorithm, but also as a step up to the problem of (*in silico*) metabolic engineering which can also be considered as a combinatorial optimization problem. Four experiments were conducted using different datasets from real scenarios obtained from (López et al., 2006) and (Dua and Graff, 2017). Table 5.1 shows the performance of the Map-Elites algorithm where solutions were evaluated using Bayes Classifier for classification problems or linear regression for regression problems. For classification, fitness is defined as accuracy, which means the percentage of correctly classified data points. For regression, on the other hand, fitness is defined as the mean squared error (MSE) between a test dataset and the predicted values. The results show that the Map-Elites algorithm is able to find high performing solutions in a large search space while simultaneously reducing the number of features to a minimum. The difference in performance between datasets is most likely due to the datasets themselves as the results are consistent with the findings of Quinonez et al. (2019). As

Table 5.1: Map-Elites results for feature selection

Dataset	Application	All features	Fitness	Number of selected features
Ionosphere	classification	34	0.91	14
Glass	classification	9	0.57	6
Cancer	classification	30	0.97	10
Parkinsons	regression	21	30808	19

the search spaces and feature spaces become more complex, as is the case with metabolic engineering, the advantage of Map-Elites is expected to grow compared to other search algorithms. Additionally, MAP-Elites is able to create a wide variety of high performing solutions, which are diverse by design as they are each optimized in their respective niches. This advantage becomes especially interesting when searching for novel metabolic strains as you could test a batch of potentially high performing strains. This results in a higher possibility that at least one of the *in silico* suggested strains also perform well in the wet lab. While having a diverse batch of high performing results thus creates a buffer to increase the chance of success, the MAP-Elites also allows for the user to control the degree of complexity of the solutions. If resources in the wet lab are quite limited, one might opt to test *in silico* suggested strains with a low number of knockouts in the wet lab. The same is true for the feature selection problem, if resources are available, more features might be selected for regression applications as it would likely increase the performance as discussed before. However, MAP-Elites returns a variety of solutions with different complexities which allows users to select the solution that fits their situation. Figure 5.1 illustrates that, as expected for regression, selecting more attributes leads to a lower MSE. By illuminating the search space using MAP-Elites, we can conclude that thirteen attributes might be almost as good as twenty-one attributes. For very large datasets, this kind of additional information on top of high performing solutions returned by the algorithm could prevent the unnecessary use of extra resources that does not improve the performance significantly.

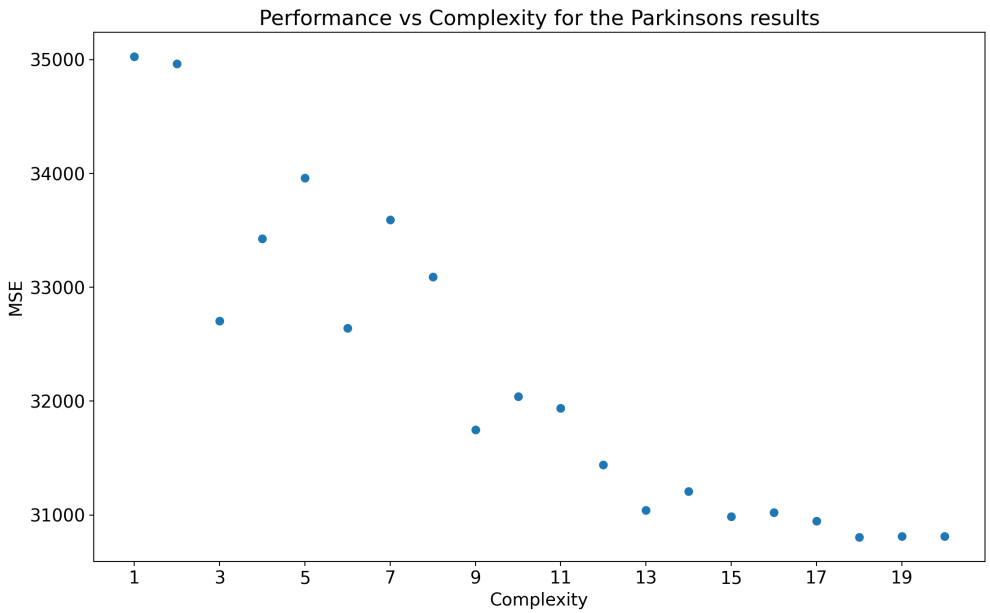


Figure 5.1: A visualisation of the relationship between the complexity of solutions (the number of selected attributes) and the inverse of the performance of those solutions (MSE).

5.2 OptMAP: MAP-Elites for *in silico* metabolic engineering

OptMAP is a novelty search based framework that combines quality diversity evolutionary algorithms with genome-scale modelling for intelligent *in silico* metabolic engineering. The main algorithm is inspired by the QD algorithm called MAP-Elites and aims to predict strain design strategies for the maximal production of specific compounds. We use an evolutionary algorithm in order to model the evolution of metabolic networks *in silico* in the hope to discover novel high performing strains. Metabolic engineering manipulations suggested by OptMAP include gene knockout strategies for microbial strain optimization and later on also medium compositions, with the possibility to expand to up-and down-regulation of genes, the addition of heterologous pathways and much more.

The OptMAP framework was validated by identifying metabolic manipulations for the production of four industrially relevant biochemicals using the genome-scale metabolic models for *Escherichia coli str. K-12 substr. MG1655*. OptMAP was applied to the overproduction of succinate, acetate and ethanol and the production of flavanones. In this thesis, we seek to answer multiple research questions.

Some questions pertain to the inner workings of the algorithm itself, how it is able to find sets of high performing solutions and which role the addition of niches plays in search algorithms. Additional questions refer to the optimization of the production of specific compounds as discussed in the case studies. The main questions posed in the case studies are which gene knockouts increase the production of compounds of interest, which reactions these genes are involved in, which media are best for production and how the results compare to existing software frameworks for predicting metabolic engineering strategies.

5.2.1 Succinate overproduction

Succinate overproduction was chosen as the first case to test the OptMAP algorithm as succinate is both relevant as a product that is used as a supplement for example and is also intricately involved in the central part of the *E. coli* metabolism namely the Krebs cycle. Specifically, we aim to discover which genes are good knockout targets and how these knockouts are distributed in the metabolic network. Figure 5.2 visualises the archive that is returned by the MAP-Elites algorithm. This archive contains all the elites in the various niches created by the number of knockouts and the type of metabolism the knockouts influence. The performance, measured in gram succinate per gram glucose per hour, of each elite, is shown using a colour gradient. When analysing the archive and Figure A.2 in the appendix, it can be observed that performance increases with an increasing number of knockouts up until six knockouts. The knockouts do seem to be spread somewhat equally over the different types of metabolism, as is shown in Figure A.3.

Next, we compare OptMAP with OptGene, a framework for *in silico* metabolic engineering developed previously by Rocha et al. (2008). OptGene uses traditional Evolutionary Algorithms (EAs) and Simulated Annealing to search for gene knockouts in specific microorganisms to maximize the production of a given compound (Rocha et al., 2008). OptGene is an established framework to suggest gene knockouts that has been used in the past for multiple applications including enhancing sesquiterpene production in

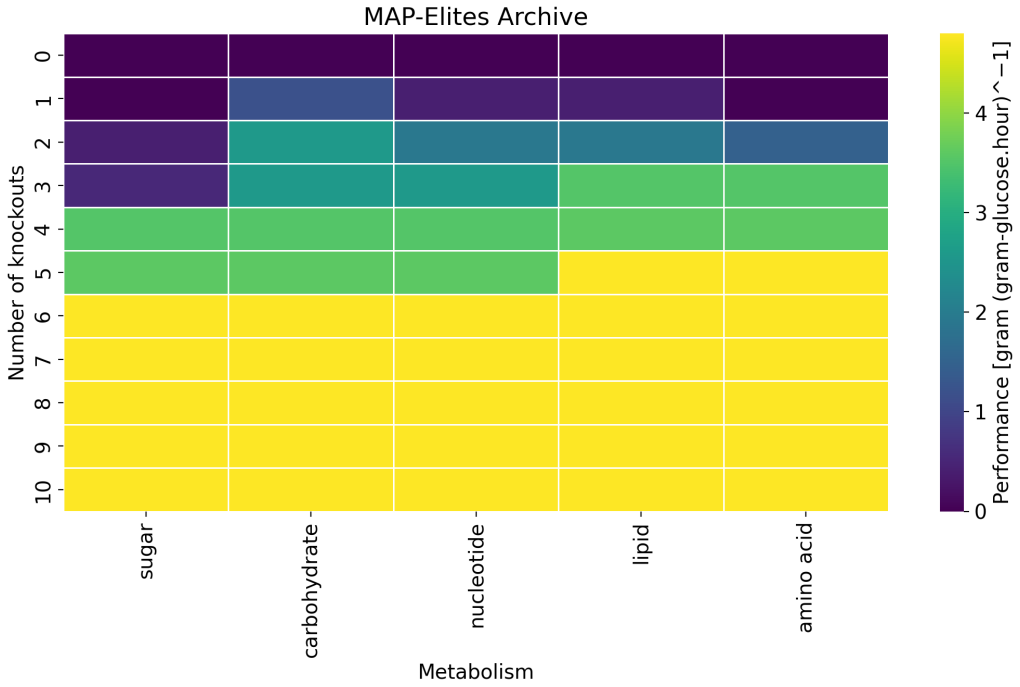


Figure 5.2: The archive containing the elite in each niche returned by OptMAP for the overproduction of succinate. Performance is measured in gram succinate per gram glucose per hour.

Saccharomyces cerevisiae (Asadollahi et al., 2009), biofuel production by *Synechocystis* (Shabestary and Hudson, 2016), improving l-phenylacetylcarbinol production in *Saccharomyces cerevisiae* (Iranmanesh et al., 2020) and the improved production of vanillin in baker's yeast (Brochado et al., 2010). It is therefore interesting to compare the results for the overproduction of succinate obtained with OptMAP to those obtained via OptGene. When running OptGene for the overproduction of succinate using the *e-coli-core* model for *Escherichia coli str. K-12 substr. MG1655*, four solutions are returned and are visualized in Table 5.2. Previously, it was mentioned that OptGene uses traditional EAs, which return only one optimal solution. However, in Tabel 5.2 multiple results are shown, this is because OptGene runs its algorithm multiple times to come up with various solutions. It is able to do this because the algorithm uses randomly generated mutations which then result in different solutions. However, as seen in Tabel 5.2, 5.4 and 5.6 this often results in solutions that are very similar to each other, while OptMAP

Table 5.2: OptGene results for succinate overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b1852*, *b0722*, *b1852*, *b0721*	SUCDi[a], G6PDH2r[b]	10.62	0.62	0.66
2	*b2029*, *b0721*, *b2029*, *b0724*, *b2029*, *b0722*, *b2029*, *b0723*	SUCDi, GND[c]	10.62	0.62	0.66
3	*b0767*, *b0722*, *b0767*, *b0724*	SUCDi, PGL[d]	10.62	0.62	0.66
4	*b2029*, *b4015*, *b0726*	AKGDH[e], GND, ICL[f]	10.62	0.62	0.66

[a] Succinate dehydrogenase, [b] Glucose 6-phosphate dehydrogenase, [c] Phosphogluconate dehydrogenase, [d] 6-phosphogluconolactonase, [e] 2-Oxogluterate dehydrogenase, [f] Isocitrate lyase

Table 5.3: The best three elites in the archive returned by OptMAP for succinate overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b3733*, *b0902*, *b3952*, *b2297*, *b2458*	ATPS4r[a], PFL[b], PTAr[c]	128.20	0.37	4.80
2	*b3733*, *b1603*	ATPS4r, NADTRHD[d], THD2[e]	51.22	0.37	1.92
3	*b3734*, *b3919*	ATPS4r, TPI[f]	109.34	0.13	1.47

[a] ATP synthase, [b] Pyruvate formate lyase, [c] Phosphotransacetylase
[d] NAD transhydrogenase, [e] NAD(P) transhydrogenase, [f] Triose phosphateisomerase

is able to provide a diverse set of solutions as seen in Table 5.3 and Figure 5.2. Additionally, this way of running the traditional algorithm for multiple cycles is also less efficient than running MAP-Elites once. MAP-Elites is usually also able to return more solutions compared to OptGene, but only the three best performing solutions are shown to keep everything comprehensible. Table 5.3 shows the best three elites in the archive returned by OptMAP for succinate overproduction. The resulting performance of OptGene and OptMAP can be compared directly as they implement the same fitness function. When comparing these two outputs, we can conclude that at least for succinate overproduction, OptMAP is able to find better solutions than OptGene. Not only is OptMAP able to find higher-performing solutions, but it is generally also done with fewer gene knockouts than the solutions of OptGene. A possible explanation for this is that neither OptGene nor OptMAP directly minimize the number of knockouts, but since OptMAP has defined the number of knockouts as part of the niches and the solutions of OptMAP are optimized within each niche, we can select the best performing solutions that occupy the niche with the smallest number of knockouts that result in that performance. While the fitness of the OptMAP solutions are higher, the biomass flux is significantly lower compared to the OptGene solutions.

While at first glance the gene knockout suggestions obtained with OptMAP might seem different from those suggested by OptGene, previous studies have also shown that the knockout genes that result in the deletion of the enzymes pyruvate formate lyase (PFL), phosphotransacetylase (PTAr) (Wahid et al., 2016), periplasmic ATP synthase (ATPS4rpp) (Jiang, 2021), triose phosphate isomerase (TPI) (Burgard et al., 2003), NAD transhydrogenase (NADTRHD) and NAD(P) transhydrogenase (THD2) (Mienda and Shamsir, 2015) could be beneficial for the overproduction of succinate. By comparing different *in silico* frameworks for metabolic engineering, Wahid et al. (2016) has shown that there are multiple ways to achieve overproduction of succinate, with some resulting in higher production than others. Importantly, it should be noted that every run of the OptMAP algorithm results in slightly different solutions. In Figure 5.3, the gene knockouts of the best three elites in the archive of one specific new run are visualized under the form of a Venn diagram. While some gene knockouts return, e.g., *b3733* and *b3734* which block ATPS4rpp, *b3952* and *b0902* which blocks PFL, *b1602* which blocks NADTRHD and THD2 and *b3919* which blocks TPI, also new knockouts are proposed, e.g., *b3956* which blocks Phosphoenolpyruvate carboxylase (PPC), *b3212* which blocks Glutamate synthase (GLUSy), *b1101* which blocks D-glucose transport via PEP:Pyr PTS (GLCpts) and *b1817* which blocks GLCpts and Fructose transport via PEP:Pyr PTS (f6p generating) (FRUpts2). The latter knockouts suggestions have also been put forward in previous studies (Oh et al., 2009) directly or indirectly. For example, knocking out FRUpts2 might lead to an increased availability of PEP, which has been shown to increase succinate production (Mienda et al., 2016). Interestingly, here OptMAP suggests the knockout of PPC, while other studies have put forward PPC as an up-regulation target for increased succinate production (Mienda et al., 2016). Figure 5.4 visualises the

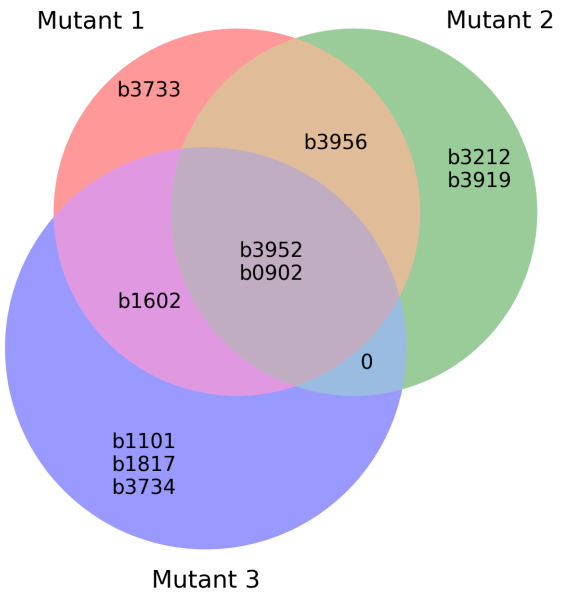


Figure 5.3: A Venn diagram of the gene knockouts in the three best elites in a second run op OptMAP for succinate overproduction.

phenotypic phase plane (PhPP) or production envelope of the best performing mutant, compared to the wild type. The production envelope is actually a specific type of PhPP between growth and a product of interest (Cardoso et al., 2018). This allows for the inspection of the limitations of the suggested mutants compared to the wild type. The production envelope can also show the coupling between growth and production. Figure 5.4 illustrates that the best performing mutant is not a growth-coupled mutant as there is no stoichiometric coupling between growth and production of succinate in this example. Here, succinate production is completely decoupled from growth and it is theoretically possible to produce more succinate by decreasing the growth rate. The production envelope thus represents the trade-off between the production of the desired product and growth. Figure 5.5 shows some suggested design strategies identified by OptMAP that are likely to improve succinate production in the wet lab.

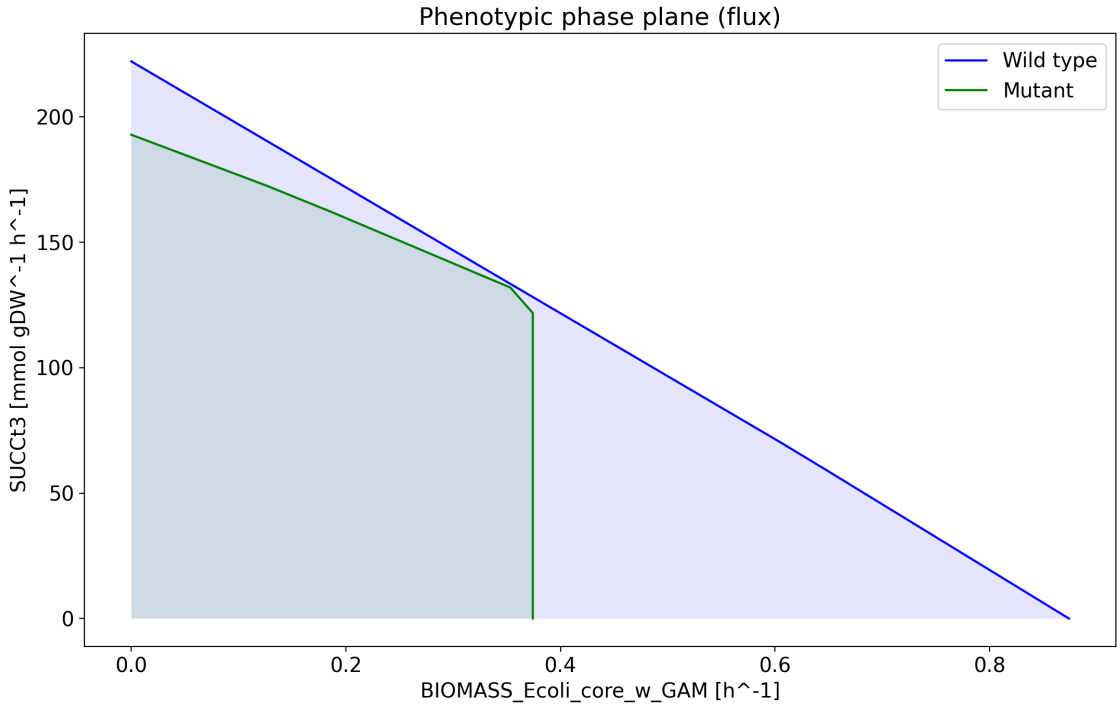


Figure 5.4: Phenotypic phase planes or production envelopes of the most performant mutant suggested by OptMAP for succinate overproduction. The production envelope illustrates the minimum and maximum production rates a production strain can achieve at different growth rates compared to the wild type. The blue production envelope belongs to the wild type model, while the green production envelope belongs to the mutant. In this particular mutant, the genes *b3733*, *b0902*, *b3952*, *b2297* and *b2458* are knocked out. Reaction names are consistent with the *e-coli-core* model for *Escherichia coli str. K-12 substr. MG1655*.

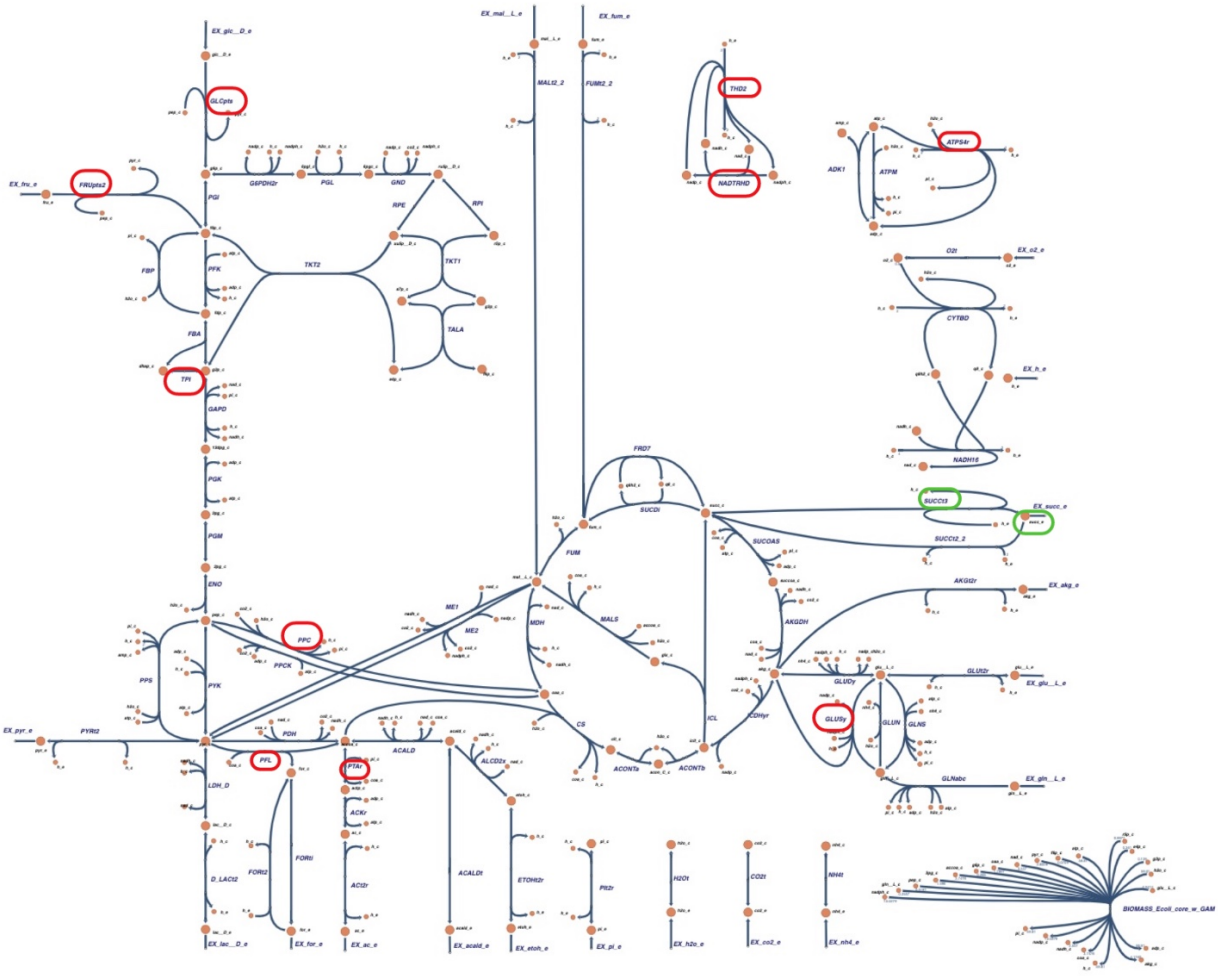


Figure 5.5: Reaction knockouts suggested by OptMAP to improve succinate production visualized on the metabolic network of the *e-coli-core* model for *Escherichia coli str. K-12 substr. MG1655*. The green circles point to the reaction and metabolite of interest, while the red circles indicate possible knockout targets. Modified from the *e-coli-core* map from Escher (King et al., 2015).

5.2.2 Acetate overproduction

Acetate overproduction was chosen as a second case to test the OptMAP algorithm. Although acetate production is often minimized during *E. coli* fermentations such as for the production of recombinant proteins, it is an example often used in tutorials for OptGene or OptKnock (Heirendt et al., 2019). The archive returned by OptMAP for acetate overproduction is visualized in Figure 5.6.

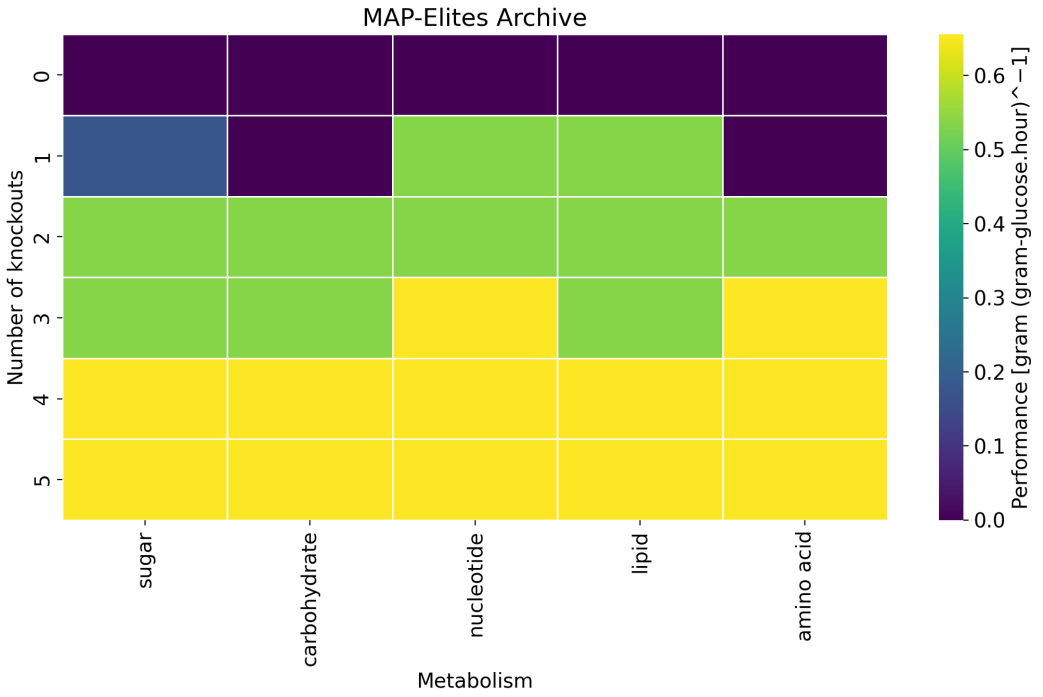


Figure 5.6: The archive containing the elite in each niche returned by OptMAP for the overproduction of acetate. Performance is measured in gram acetate per gram glucose per hour.

When comparing the three highest performing elites to the results obtained by OptGene for acetate production, respectively seen in Table 5.5 and Table 5.4, we see that both OptMAP's and OptGene's highest performing solutions perform the same, although different gene knockouts are suggested. Similarly as before, OptMAP seems to be suggesting high performing solutions, obtained via fewer knockouts than OptGene even though OptMAP does not directly minimize the number of gene knockouts. Although software frameworks such as OptMAP can suggest interesting and helpful strain design choices, critical reflection of the results is still necessary. For example, the knockout of gene *s0001* does result in an improved acetate flux while also maintaining biomass flux, but when looking at the affected reactions, we see that reactions such as *O2t*, are not enabled by enzymes and thus not encoded in real genes. *O2t* is the reaction that allows for O_2 transport via diffusion, this is not a gene or reaction which be turned off in the wet lab. However, these results might suggest that anaerobic growth of *E. coli* could improve the production of acetate. Creating conditions with limited oxygen availability is something that can be achieved in the wet lab. The gene knockouts suggested by OptMAP for the highest performing elite have previously been shown to improve acetate production using various *in silico* techniques (Long and Antoniewicz, 2019).

Table 5.4: OptGene results for acetate overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b1852*, *b0722*	SUCDi[a], G6PDH2r[b]	10.62	0.62	0.66
2	*b2029*, *b0721*, *b2029*, *b0722*, *b2029*, *b0723*	GND[c], SUCDi	10.62	0.62	0.66
3	*b3738*, *b3733*, *b3734*, *b3736*, *b3731*, *b3737*, *b3735*	ATPS4r[d]	13.98	0.37	0.52

[a] Succinate dehydrogenase, [b] Glucose 6-phosphate dehydrogenase, [c] Phosphogluconate dehydrogenase, [d] ATP synthase

Table 5.5: The best three elites in the archive returned by OptMAP for acetate overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b2029*, *b4015*, *b0728*	GND[a], ICL[b], SUCOAS[c]	10.62	0.62	0.66
2	*b3734*	ATPS4r[d]	14.31	0.37	0.53
3	*s0001*	ACALDt, CO2t, H2Ot, NH4t, O2t	8.32	0.21	0.18

[a] Phosphogluconate dehydrogenase, [b] Isocitrate lyase, [c] Succinyl-CoA synthetase, [d] ATP synthase

Figure 5.7 visualises the PhPP of the best performing mutant, compared to the wild type for acetate production. In contrast to the previous example, OptMAP was able to discover a growth-coupled mutant for acetate production. In for the mutant visualized in Figure 5.7, an increase in acetate production can be achieved simultaneously with an increase in biomass production. These types of phenotypes are called growth-coupled phenotypes.

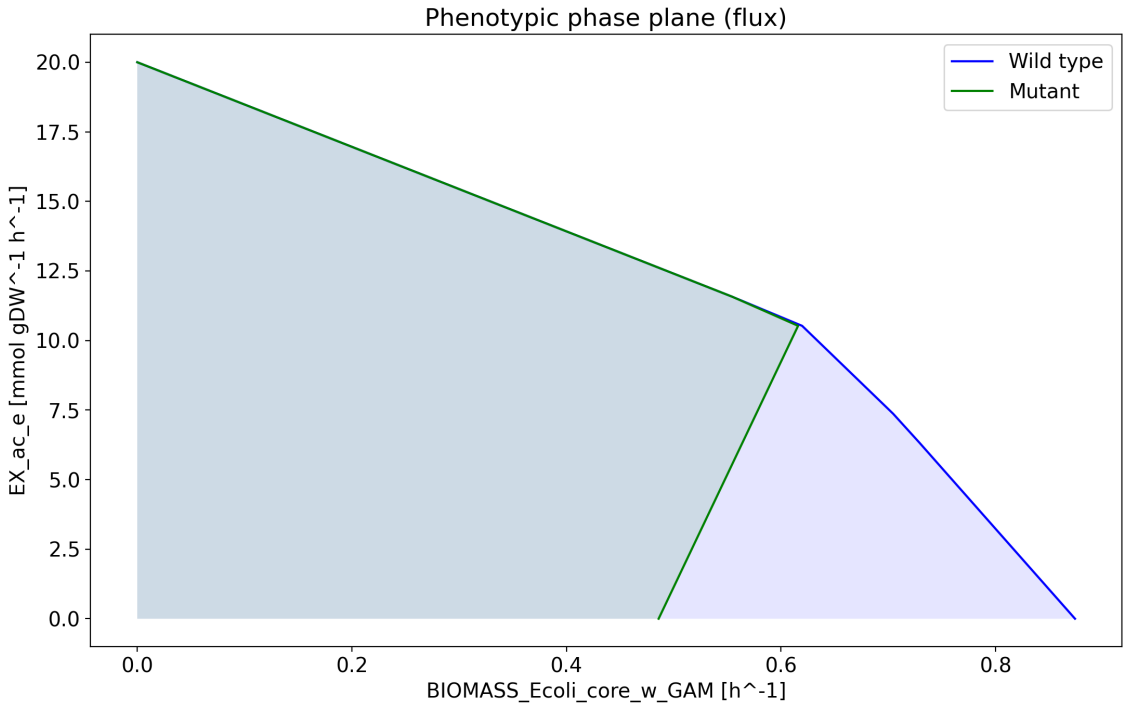


Figure 5.7: Phenotypic phase planes or production envelopes of the most performant mutant suggested by OptMAP for acetate overproduction. The production envelope illustrates the minimum and maximum production rates a production strain can achieve at different growth rates compared to the wild type. The blue production envelope belongs to the wild type model, while the green production envelope belongs to the mutant.

5.2.3 Ethanol overproduction

Ethanol overproduction was chosen as a third case to test the OptMAP algorithm. Figure 5.8 visualises the archive that is returned by the MAP-Elites algorithm for ethanol production by *E. coli*.

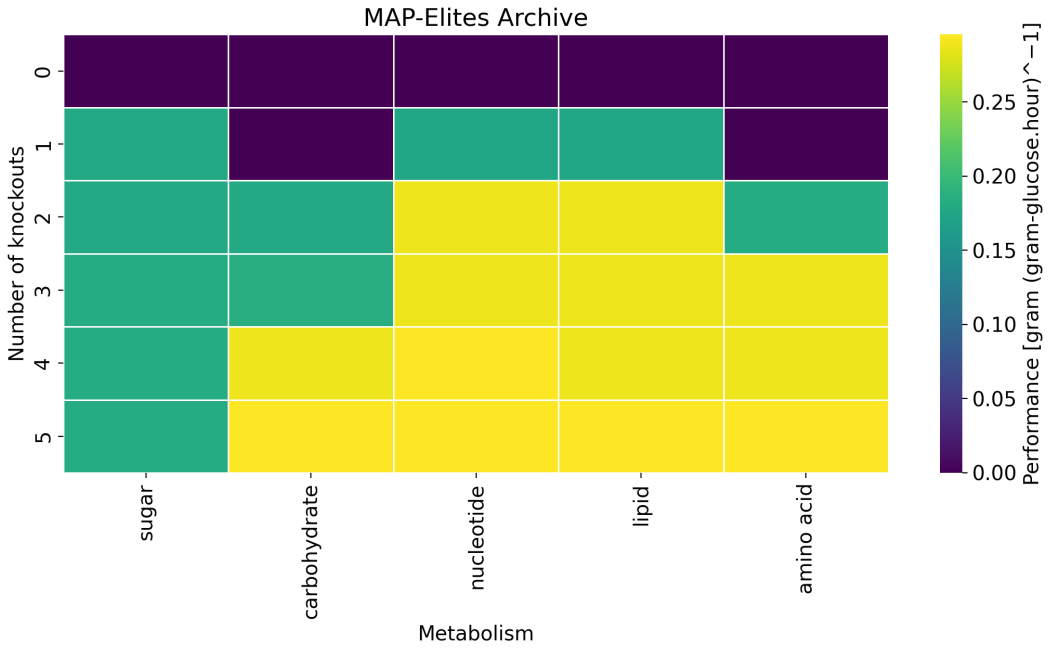


Figure 5.8: The archive containing the elite in each niche returned by OptMAP for the overproduction of ethanol. Performance is measured in gram ethanol per gram glucose per hour.

Table 5.7 illustrates the gene knockouts and affected reactions suggested by OptMAP. When compared to the results given by OptGene in Table 5.6, we conclude that both algorithms are able to find similarly performing solutions even though OptMAP is generally able to do this by implementing fewer gene knockouts compared to OptGene. This observation is consistent with the results for succinate and acetate overproduction. The gene knockouts and subsequent reaction knockouts offered by OptMAP can also be found throughout the different solutions of OptGene. Additionally, similar results have been reported both for the gene knockouts and the resulting fluxes using other bio-inspired algorithms for the optimization of ethanol production as discussed by Liew et al. (2016).

Figure 5.9 visualises the phenotypic phase plane or production envelope of the best performing mutant, compared to the wild type. Similarly to the example of acetate production, OptMAP was able to discover a growth-coupled mutant for ethanol production. In for the mutant visualized in Figure 5.9, an increase in ethanol production can be achieved by increasing biomass production. Figure 5.10 shows which genes are conserved between the solutions found by OptMAP.

Table 5.6: OptGene results for ethanol overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b2029*, *b0729*, *b2287*, *b3731*	SUCOAS[a], NADH16[b], ATPS4r[c], GND[d]	15.35	0.19	0.30
2	*b0767*, *b2287*, *b3731*, *b0722*, *b0767*, *b2287*, *b3731*, *b0723*	PGL[e], NADH16[f], ATPS4r, SUCDi[g]	15.35	0.19	0.30
3	*b2029*, *b3956*, *b2287*, *b3731*	PPC[h], NADH16, ATPS4r, GND	15.11	0.20	0.30
4	*b1852*, *b3737*, *b2281*, *b1852*, *b2284*, *b3737*	NADH16, ATPS4r, G6PDH2r[i]	14.09	0.20	0.28

[a] Succinyl-CoA synthetase, [b] NADH dehydrogenase, [c] ATP synthase, [d] Phosphogluconate dehydrogenase,
[e] 6-phosphogluconolactonase, [f] NADH dehydrogenase [g] Succinate dehydrogenase, [h] Phosphoenolpyruvate carboxylase,
[i] Glucose 6-phosphate dehydrogenase

Table 5.7: The best three elites in the archive returned by OptMAP for ethanol overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b3732*, *b2286*, *b0767*, *b3956*, *b4301*, *b0728*	ATPS4r[a], NADH16[b], PGL[c], PPC[d], RPE[e], SUCOAS[f]	15.51	0.19	0.30
2	*b0727*, *b3732*, *b1852*, *b2286*	AKGDH[g], ATPS4r, G6PDH2r[h], NADH16	15.35	0.19	0.30
3	*b3734*, *b2286*	NADH16, ATPS4r	14,20	0.20	0.29

[a] Periplasmic ATP synthase, [b] NADH dehydrogenase, [c] 6-phosphogluconolactonase, [d] Phosphoenolpyruvate carboxylase,
[e] Ribulose 5-phosphate 3-epimerase, [f] Succinyl-CoA synthetase, [g] 2-Oxogluterate dehydrogenase, [h] Glucose 6-phosphate dehydrogenase

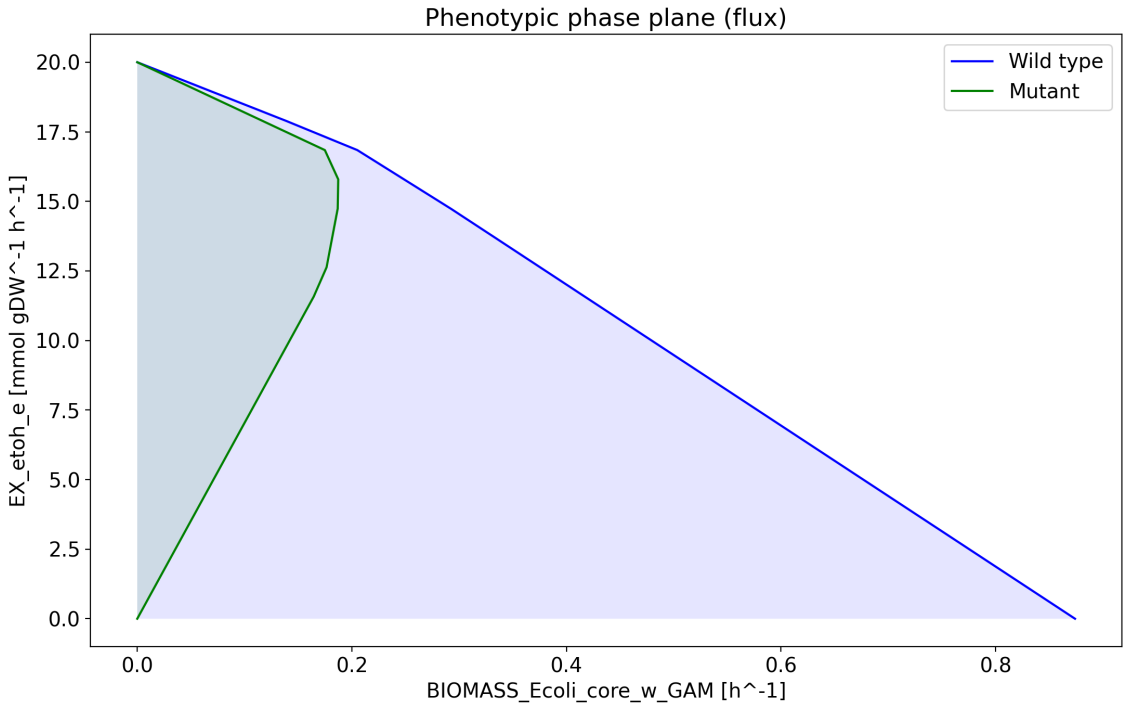


Figure 5.9: Phenotypic phase planes or production envelopes of the most performant mutant suggested by OptMAP for ethanol overproduction. The production envelope illustrates the minimum and maximum production rates a production strain can achieve at different growth rates compared to the wild type. The blue production envelope belongs to the wild type model, while the green production envelope belongs to the mutant.

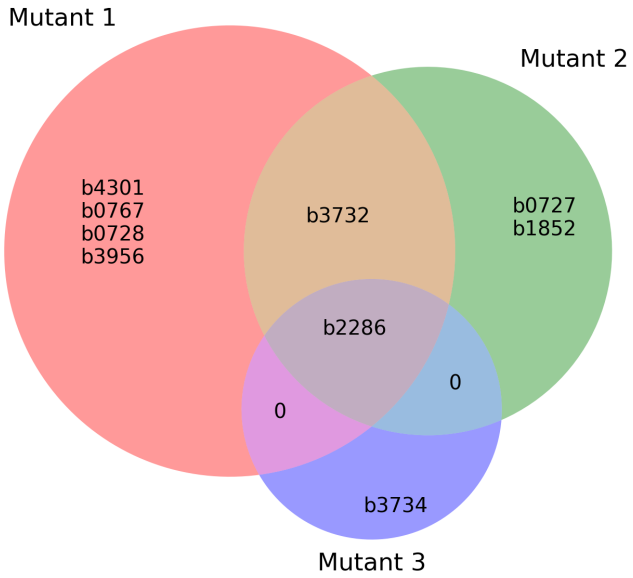


Figure 5.10: A Venn diagram of the gene knockouts in the three best elites in a second run op OptMAP for ethanol overproduction.

5.2.4 Flavanone production

Once multiple positive results from OptMAP were achieved for relatively simple metabolites such as succinate, acetate and ethanol that are naturally produced by *E. coli*, we decided to optimize the heterologous production of the flavanone naringenin. Recently, yet another *in silico* metabolic engineering framework was built called OptDesign. OptDesign is a two-step strain design strategy in which regulation candidates that have a noticeable flux difference between the wild type and production strains are selected, after which the optimal design strategies with limited manipulations (combining regulation and knockout) that lead to high production of a metabolite of interest are computed (Jiang, 2021). The OptDesign paper from Jiang (2021) also discussed three cases namely succinate, naringenin and lycopene production using the iML1515 genome-scale model for *Escherichia coli str. K-12 substr. MG1655*. While OptMAP found similar results for succinate as described by OptDesign, we were unable to find gene knockouts for naringenin. As this came as a surprise after the success of OptMAP with the previous challenges, the gene knockouts suggested by OptDesign were manually tested on iML1515 to check the improvement in naringenin production. To our surprise, even though the suggested knockouts seemed sensible and were suggested in the OptDesign paper, they did not result in improved naringenin production in our scripts. As described in Section 4, the same iML1515 model from BiGG was used and the same homologous flavanone pathway was added. These findings were checked multiple times, yet it is still unclear as to what could explain the difference between our results and the findings reported by Jiang (2021). On the other hand, there are some discrepancies in the Supplemental Information of the OptDesign paper (Jiang, 2021) where they state that the reaction *NARGt* is defined as *chal_c* $\Longleftrightarrow$ *narg_e* while another paper from Fowler et al. (2009) defined the reaction as *narg_c* $\Longleftrightarrow$ *narg_e*. We also implemented the *NARGt* reaction as is done by Fowler et al. (2009) and assume that the difference compared to Jiang (2021) is due to a typographical error. Alternatively, OptDesign is coded and run in MATLAB while we work in Python which might also account for a difference in outcomes. Interesting however is also that the knockouts suggested by OptDesign seem to decrease the flux towards malonyl-CoA as opposed to increasing the flux toward that major building block for naringenin, which we found to be quite remarkable. In this thesis, there was no further investigation into whether the differences could be due to the programming language used or other possible explanations, as the main focus was to test OptMAP. Therefore, OptMAP was applied to the search for gene knockouts that increase the main precursor of naringenin namely malonyl-CoA. As seen in Table 5.9 and Table 5.8, both OptMAP and the previously mentioned OptGene found the same gene knockout that increases flux through the *ACCOAC* reaction, which produces malonyl-CoA.

Table 5.8: OptGene results for malonyl-CoA overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b1091*	KAS15[a], ACOATA[b]	2.42	0.92	2.22

[a] Beta-ketoacyl-ACP synthase, [b] Acetyl-CoA ACP transacylase

Table 5.9: The best elite in the archive returned by OptMAP for malonyl-CoA overproduction.

Solution	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [$g.(g.h)^{-1}$]
1	*b1091*	KAS15[a], ACOATA[b]	2.42	0.92	2.22

[a] Beta-ketoacyl-ACP synthase, [b] Acetyl-CoA ACP transacylase

5.2.5 Incorporating medium compositions as niches

Until now, the same default medium composition has been used to search for gene knockouts that optimize the production of a specific compound. However, the media composition might also influence the production of a given metabolite and a specific gene knockout could also result in different outcomes depending on the substrate the microorganism grows on. OptMAP is therefore expanded to consider medium compositions as one of the behavioural dimensions that make up the feature space and thus also influences the niches. The six different minimal media compositions that are considered are shown in Table 5.10. This added feature of optimizing strains on different media is something that is not often explored in other strain design frameworks such as OptKnock, OptGene or OptDesign. OptCouple is a framework that can predict medium modifications, but is mostly focused on strategies for coupling production to growth. The expanded version of OptMAP is validated for the overproduction of succinate. While you could in theory run another type of software package like OptGene for all medium compositions, this is much more tedious. Furthermore, OptGene only takes in one substrate such as *glc__D_e*, so multiple carbon sources and other substrates can not be adjusted in OptGene. When we tried to optimize *E. coli* on substrates other than *glc__D_e* using OptGene, no solutions were found. The opposite is true for OptMAP as even more and better-performing solutions were found when compared to Section 5.2.1. Figure 5.11 visualizes the archive returned by the expanded version of OptMAP for the overproduction of succinate, with the performance value for each elite.

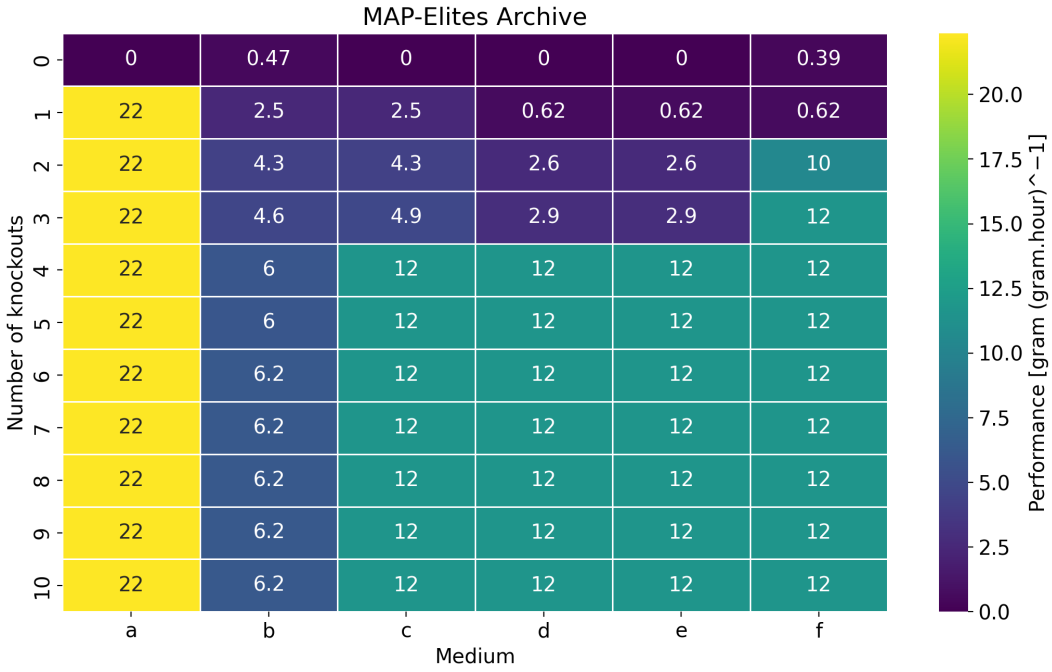


Figure 5.11: The archive containing the elite in each niche returned by the expanded version of OptMAP which includes media as niches for the overproduction of succinate. Performance is measured in gram succinate per gram carbon source per hour.

Table 5.10: Six medium compositions used to expand MAP-Elites to consider media as niches. The upper bound for each import flux are given and expressed in mmol.(gDW.h)$^{-1}$.

Components	Medium a	Medium b	Medium c	Medium d	Medium e	Medium f
EX_fru_e	0.00	0.00	520.11	309.46	228.41	0.00
EX_glc__D_e	0.00	520.11	0.00	0.00	0.00	0.00
EX_gln__L_e	0.00	0.00	0.00	2.18	0.00	0.00
EX_glu__L_e	278.65	115.54	115.54	0.00	0.00	4.41
EX_mal__L_e	0.00	0.00	0.00	0.00	0.00	206.01
EX_nh4_e	0.00	0.00	0.00	0.00	4.36	0.00
EX_o2_e	500.00	0.00	0.00	0.00	0.00	0.00
EX_pi_e	47.39	55.67	55.67	2.94	2.94	2.94

A first observation that can be made is that while the wild type did not produce any succinate on the default medium, it does produce some succinate on media b and f. Moreover, we can conclude that while the maximal BPCY found in Section 5.2.1 was 4.80 g.(g.h)$^{-1}$, here we find various higher values of 6.2, 10, 12 and 22 g.(g.h)$^{-1}$. This immediately shows the importance of the medium on the production and growth of an organism and illustrates the added benefit of using media as niches in which organisms are evolved towards the chosen metabolic engineering objectives.

Table 5.11: The best three elites in the archive returned by the extended version of OptMAP that includes media as niches for succinate overproduction.

Solution	Medium	Genes	Reactions	Target Flux	Biomass flux [h^{-1}]	Performance [g.(g.h)$^{-1}$]
1	a	*b3734*	ATPS4r[a]	1000	6.23	22.35
2	f	*b3734*, *b2280*, *b0728*	ATPS4r, NADH16[b], SUCOAS[c]	0.72	0.11	11.62
3	f	*b3734*, *b2280*	ATPS4r, NADH16	0.64	0.11	10.43

[a] ATP synthase, [b] NADH dehydrogenase, [c] Succinyl-CoA synthetase

Table 5.11 shows the genes that are knocked out in the best three elites. Overall, the suggested gene knockouts are situated in the same parts of the metabolism as the knockouts suggested in Section 5.2.1. While some gene knockouts such as *b3734* are exactly the same, knockouts that affect the reactions NADH dehydrogenase (NADH16), Succinyl-CoA synthetase (SUCOAS) are not suggested in Section 5.2.1, but also play a role in the succinate and NADH metabolism, similar to knockout suggestions found in Table 5.3.

5.2.6 OptMAP compared to existing strain design frameworks

On top of OptMAP's ability to predict gene knockout targets and predict growth-coupled production phenotypes, it also has multiple more interesting features. OptMAP utilizes novelty search and niches to maximally explore the search space and return a diverse archive of high performing elites, it can illuminate the fitness potential of each region of the feature space to provide (novel) scientific insights and it optimizes strains on different medium compositions. Additionally, there are also some functionalities seen in other tools that are not (yet) present in OptMAP such as the ability to suggest up-and down-regulation of genes and the ability to predict heterologous pathways via knock-ins. Table 5.12 illustrates the differences and similarities between OptMAP and various *in silico* metabolic engineering frameworks for the prediction of strain design strategies for microbial production of target metabolites. We can conclude that even though OptMAP has some interesting features compared to existing frame-

works and has proven that it can discover high performing design strategies with various characteristics, further development is still possible to improve OptMAP.

Table 5.12: A comparison of OptMAP with similar *in silico* metabolic engineering frameworks for the prediction of strain design strategies for microbial production of target metabolites.

	OptGene	**OptKnock**	**OptDesign**	**OptCouple**	**OptMAP**
Novelty Search	✖	✖	✖	✖	✓
Illuminate the fitness potential of each region of the feature space	✖	✖	✖	✖	✓
Predict medium modification	✖	✖	✖	✓	✓
Predict gene knockout targets	✓	✓	✓	✓	✓
Predict growth-coupled production phenotypes	✓	✓	✓	✓	✓
Predict up- and down-regulation targets	✖	✓	✓	✓	not yet
Predict heterologous pathways via knock-ins	✖	✓	✓	✓	not yet

6. CONCLUSION

The search space created by all the possible combinations of manipulations that are possible in metabolic engineering such as gene knockouts is so large that navigating it in the wet lab is a long and resource consuming journey. To that end, multiple software tools have been created to aid that search. In this thesis, OptMAP, a novel computational tool to predict strain design strategies for metabolic engineering is developed. OptMAP is based on the Quality Diversity evolutionary algorithm called MAP-Elites as the main optimization algorithm. OptMAP is mainly made up of an adapted version of the MAP-ELites algorithm, combined with genome-scale metabolic modelling. The main goal was to build a framework that predicts strain design strategies for metabolic engineering as a tool to explore more possible solutions compared to metabolic engineering in the wet lab. OptMAP has the unique ability to return a diverse set of high performing solutions which can be tested in the wet lab as opposed to other frameworks that usually provide one optimal solution or multiple solutions that are very similar. The fact that OptMAP makes use of various niches in the feature space allows it to find both better-performing solutions as well as more diverse ones. In the current version, OptMAP is able to predict gene knockout strategies to maximize the production of various metabolites.

OptMAP was validated using four case studies, namely the production of succinate, acetate, ethanol and flavanones. As observed in the obtained results, we were able to create a working version of OptMAP which is able to provide solutions that are as good or better than comparable *in silico* metabolic frameworks. Our initial goal of further aiding the explorations of all possible solutions for the development of strains with specific characteristics has been achieved. This indicates that the OptMAP could be a useful tool to help researchers in the wet lab. It reduces unnecessary resource allocation by limiting the number of experiments that should be run in the wet lab and provides an improved computational tool to do so.

One of the most interesting emergent features of the MAP-ELites algorithm and a likely explanation for the increased performance is that elites are able to jump from one niche to another. This allows elites to explore many different roads towards a high performing solution, while not having to be the absolute best-performing mutant throughout the whole search. This is in contrast to a traditional evolutionary algorithm as the one on which OptGene is based, where only the best mutants are kept to search for a single best solution. Novelty search has therefore shown its potential in the search for strain design strategies. Additionally, since OptMAP returns an archive with the elite in each niche and their performance, OptMAP is able to illuminate the relationship between performance and dimensions of interest in solutions. These additional insights could be of help both for rethinking the niches as they are currently still defined by the user and not emergent and to make better-informed decisions when selecting a set of mutants to test for specific properties in the wet lab.

Finally, we can conclude that the further development of software frameworks based on Quality Diversity algorithms for the prediction of strain design strategies is likely to help discover new mutant strains for a range of different applications, not only for the production of certain metabolites, but also the production of recombinant proteins when ME-models are used or creating microorganisms able to metabolize novel toxic compounds or developing microorganisms for microbial therapeutics. As Quality Diversity algorithms have been proven to return a set of high-quality solutions and additionally also improve the state-of-the-art for finding a single, best solution for applications including designing robots and now designing novel strains, it would be interesting to how these types of algorithms would allow for new advances throughout different scientific and engineering domains. Drug design, protein engineering and plant breeding are just some examples of possible applications where a framework similar to that of OptMAP could be used to create new breakthroughs using the principles of Quality Diversity algorithms.

7. FUTURE PERSPECTIVES

While OptMAP has shown to be helpful for predicting gene knockouts that maximize the growth-coupled production of multiple metabolites, there are a variety of additions that would improve OptMAP. While currently only knockouts are predicted, predicting regulation targets for the up-or downregulation of certain genes would help to broaden the flexibility of the algorithm to find high performing solutions. Sometimes the knockout of a reaction might allow for an increase through another reaction, but it does not improve the production because the knockout target might be essential for growth. Down-regulation of that gene could therefore be a way to increase production while still allowing for biomass growth. Additionally, as upregulation of genes is a technique that is an option to use in the wet lab, it would be ideal to incorporate this in OptMAP. Predicting heterologous pathways via knock-ins is a great addition for OptMAP for applications such as searching for mutant strains that are able to metabolise certain toxic compounds for bioremediation proposes for example. Furthermore, ME models should be integrated both for more accurate predictions, but also to predict strain development strategies for the production of (recombinant) proteins.

In order to gain even more insight into the way the algorithm is able to find its solutions, the phylogeny of elites could be recorded. One obvious conclusion this kind of information could provide would be that as a phylogenic tree covering a large part of the feature space, it could be interpreted as having a large number of different stepping stones (Nordmoen et al., 2021).

Another interesting idea would be to try and find a synthetic metabolism for a given metabolic engineering objective similar to retrosynthesis in chemistry. An *in silico* synthetic metabolism could be created by adding enzymes to the metabolic network to improve the production or breakdown of certain compounds on top of the existing chemical dynamics. This way, minimal metabolic networks could be discovered for a variety of applications. This is also becoming increasingly relevant in the industry as cell-free production of (biological) compounds is gaining traction. For example, FabricNano is a company designing high spatial precision DNA scaffolds combined with enzymes in flow reactors (Burns, 2021; FabricNano, 2022).

In addition to the further development of OptMAP, new applications for Novelty Search and Quality Diversity algorithms could improve current search strategies. Problems in various scientific and engineering domains such as drug design, protein engineering and plant breeding could be formulated in such as way that Quality Diversity algorithms can find novel solutions and additional insights.

BIBLIOGRAPHY

Adami, C., Ofria, C., and Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463–4468.

Amann, T., Schmieder, V., Faustrup Kildegaard, H., Borth, N., and Andersen, M. R. (2019). Genetic engineering approaches to improve posttranslational modification of biopharmaceuticals in different production platforms. *Biotechnology and Bioengineering*, 116(10):2778–2796.

Asadollahi, M. A., Maury, J., Patil, K. R., Schalk, M., Clark, A., and Nielsen, J. (2009). Enhancing sesquiterpene production in saccharomyces cerevisiae through in silico driven metabolic engineering. *Metabolic Engineering*, 11(6):328–334.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Baquero, F., Coque, T. M., Galán, J. C., and Martinez, J. L. (2021). The origin of niches and species in the bacterial world. *Frontiers in Microbiology*, 12:566.

Bard, J. (2016). *Principles of evolution: Systems, species, and the history of life*. Garland Science.

Berkner, L. V. and Marshall, L. (1965). On the origin and rise of oxygen concentration in the earth's atmosphere. *Journal of Atmospheric Sciences*, 22(3):225–261.

Blum, C. and Dorigo, M. (2004). Deception in ant colony optimization. In *International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 118–129. Springer.

Brochado, A. R., Matos, C., Møller, B. L., Hansen, J., Mortensen, U. H., and Patil, K. R. (2010). Improved vanillin production in baker's yeast through in silico design. *Microbial cell factories*, 9(1):1–15.

Brownlee, J. (2020). How to choose a feature selection method for machine learning.

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

Burns, J. R. (2021). Introducing bacteria and synthetic biomolecules along engineered dna fibers. *Small*, 17(25):2100136.

Campodonico, M. A., Sukumara, S., Feist, A. M., and Herrgård, M. J. (2018). Computational methods to assess the production potential of bio-based chemicals. In *Synthetic Metabolic Pathways*, pages 97–116. Springer.

Cardoso, J. G., Jensen, K., Lieven, C., Lærke Hansen, A. S., Galkina, S., Beber, M., Ozdemir, E., Herrgård, M. J., Redestig, H., and Sonnenschein, N. (2018). Cameo: a python library for computer aided metabolic engineering and optimization of cell factories. *ACS synthetic biology*, 7(4):1163–1166.

Castle, S. D., Grierson, C. S., and Gorochowski, T. E. (2021). Towards an engineering theory of evolution. *Nature Communications*, 12(1):1–12.

Chalancon, G., Kruse, K., and Babu, M. M. (2013). Metabolic networks, structure and dynamics. *Encyclopedia of Systems Biology. Springer, New York*.

Chandra, N. and Kumar, S. (2017). Antibiotics producing soil microorganisms. In *Antibiotics and antibiotics resistance genes in soils*, pages 1–18. Springer.

Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., Omatu, S., and Corchado, J. M. (2014). Differential bees flux balance analysis with optknock for in silico microbial strains optimization. *PloS one*, 9(7):e102744.

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

Christensen, B. and Nielsen, J. (1999). Metabolic network analysis. *Bioanalysis and Biosensors for Bioprocess Monitoring*, pages 209–231.

Clune J, Lehman J, S. K. (2019). Icml 2019 tutorial: Recent advances in population-based search for deep neural networks.

Cornejo, E., Abreu, N., and Komeili, A. (2014). Compartmentalization and organelle formation in bacteria. *Current opinion in cell biology*, 26:132–138.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. *Molecular systems biology*, 7(1):543.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Cully, A. and Demiris, Y. (2017). Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259.

Dahal, S., Zhao, J., and Yang, L. (2020). Genome-scale modeling of metabolism and macromolecular expression and their applications. *Biotechnology and Bioprocess Engineering*, 25(6):931–943.

Dahal, S., Zhao, J., and Yang, L. (2021). Recent advances in genome-scale modeling of proteome allocation. *Current Opinion in Systems Biology*, 26:39–45.

de Oliveira Dal'Molin, C. G., Quek, L.-E., Palfreyman, R. W., Brumbley, S. M., and Nielsen, L. K. (2010). Aragem, a genome-scale reconstruction of the primary metabolic network in arabidopsis. *Plant physiology*, 152(2):579–589.

Deb, K. and Goldberg, D. E. (1993). Analyzing deception in trap functions. In *Foundations of genetic algorithms*, volume 2, pages 93–108. Elsevier.

Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., and Xu, J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*, 25(7):5277–5298.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dubitzky, W., Wolkenhauer, O., Cho, K.-H., and Yokota, H. (2013). *Encyclopedia of systems biology*, volume 402. Springer New York.

Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013). Cobrapy: constraints-based reconstruction and analysis for python. *BMC systems biology*, 7(1):1–6.

Education, I. C. (2022). What are neural networks?

Edwards, J. S. and Palsson, B. O. (1999). Systems properties of the haemophilus influenzaerd metabolic genotype. *Journal of Biological Chemistry*, 274(25):17410–17416.

FabricNano (2022). Dna nanotechnology.

Fang, X., Lloyd, C. J., and Palsson, B. O. (2020). Reconstructing organisms in silico: genome-scale models and their emerging applications. *Nature Reviews Microbiology*, 18(12):731–743.

Fani, R. (2012). The origin and evolution of metabolic pathways: why and how did primordial cells construct metabolic routes? *Evolution: Education and Outreach*, 5(3):367–381.

Federoff, H. J. and Gostin, L. O. (2009). Evolving from reductionism to holism: is there a future for systems medicine? *Jama*, 302(9):994–996.

Fontaine, M. C., Togelius, J., Nikolaidis, S., and Hoover, A. K. (2020). Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 94–102.

Fowler, Z. L., Gikandi, W. W., and Koffas, M. A. (2009). Increased malonyl coenzyme a biosynthesis by tuning the escherichia coli metabolic network and its application to flavanone production. *Applied and environmental microbiology*, 75(18):5831–5839.

Francke, C., Siezen, R. J., and Teusink, B. (2005). Reconstructing the metabolic network of a bacterium from its genome. *Trends in microbiology*, 13(11):550–558.

Gabaldón, T. and Pittis, A. A. (2015). Origin and evolution of metabolic sub-cellular compartmentalization in eukaryotes. *Biochimie*, 119:262–268.

Gaier, A., Asteroth, A., and Mouret, J.-B. (2018). Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation*, 26(3):381–410.

Gleizer, S., Ben-Nissan, R., Bar-On, Y. M., Antonovsky, N., Noor, E., Zohar, Y., Jona, G., Krieger, E., Shamshoum, M., Bar-Even, A., et al. (2019). Conversion of escherichia coli to generate all biomass carbon from co2. *Cell*, 179(6):1255–1263.

Gomes, J., Urbano, P., and Christensen, A. L. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2):115–144.

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

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

Henry, C. S., DeJongh, M., Best, A. A., Frybarger, P. M., Linsay, B., and Stevens, R. L. (2010). High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature biotechnology*, 28(9):977–982.

Herrgård, M. J., Lee, B.-S., Portnoy, V., and Palsson, B. Ø. (2006). Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in saccharomyces cerevisiae. *Genome research*, 16(5):627–635.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Huber, C. and Wachtershauser, G. (1997). Activated acetic acid by carbon fixation on (fe, ni) s under primordial conditions. *Science*, 276(5310):245–247.

Hussein, F., Kharma, N., and Ward, R. (2001). Genetic algorithms for feature selection and weighting, a review and study. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 1240–1244.

Iranmanesh, E., Asadollahi, M. A., and Biria, D. (2020). Improving l-phenylacetylcarbinol production in saccharomyces cerevisiae by in silico aided metabolic engineering. *Journal of biotechnology*, 308:27–34.

Iyer, M. S., Pal, A., Srinivasan, S., Somvanshi, P. R., and Venkatesh, K. (2020). Global transcriptional regulators fine-tune the translational and metabolic machinery in escherichia coli under anaerobic fermentation. *bioRxiv*.

Jalali, S. M. J., Ahmadian, S., Khosravi, A., Mirjalili, S., Mahmoudi, M. R., and Nahavandi, S. (2020). Neuroevolution-based autonomous robot navigation: a comparative study. *Cognitive Systems Research*, 62:35–43.

Jiang, S. (2021). Optdesign: Identifying optimum design strategies in strain engineering for biochemical production. *bioRxiv*.

Joyce, A. R. and Palsson, B. Ø. (2008). Predicting gene essentiality using genome-scale in silico models. In *Microbial Gene Essentiality: Protocols and Bioinformatics*, pages 433–457. Springer.

Kaggle (2022). Kaggle. https://www.kaggle.com.

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

Kang, M.-J., Baek, K.-R., Lee, Y.-R., Kim, G.-H., and Seo, S.-O. (2022). Production of vitamin k by wild-type and engineered microorganisms. *Microorganisms*, 10(3):554.

Karlsen, E., Schulz, C., and Almaas, E. (2018). Automated generation of genome-scale metabolic draft reconstructions based on kegg. *BMC bioinformatics*, 19(1):1–11.

Kawai, F. (1995). Breakdown of plastics and polymers by microorganisms. *Microbial and enzymatic bioproducts*, pages 151–194.

Keseler, I. M., Mackie, A., Santos-Zavaleta, A., Billington, R., Bonavides-Martínez, C., Caspi, R., Fulcher, C., Gama-Castro, S., Kothari, A., Krummenacker, M., et al. (2017). The ecocyc database: reflecting new knowledge about escherichia coli k-12. *Nucleic acids research*, 45(D1):D543–D550.

Kim, B., Kim, W. J., Kim, D. I., and Lee, S. Y. (2015). Applications of genome-scale metabolic network model in metabolic engineering. *Journal of industrial microbiology and biotechnology*, 42(3):339–348.

Kim, W. J., Kim, H. U., and Lee, S. Y. (2017). Current state and applications of microbial genome-scale metabolic models. *Current Opinion in Systems Biology*, 2:10–18.

Kim, Y., Gu, C., Kim, H. U., and Lee, S. Y. (2020). Current status of pan-genome analysis for pathogenic bacteria. *Current opinion in biotechnology*, 63:54–62.

Kimura, M. (2020). Diversity of organisms and views on evolution. In *My Thoughts on Biological Evolution*, pages 1–13. Springer.

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

King, Z. A. and Feist, A. M. (2013). Optimizing cofactor specificity of oxidoreductase enzymes for the generation of microbial production strains—optswap. *Industrial Biotechnology*, 9(4):236–246.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2016). Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522.

knowledgebase, G. O. (2021). The gene ontology resource: enriching a gold mine. *Nucleic acids research*, 49(D1):D325–D334.

Komosinski, M. and Ulatowski, S. (1998). Framsticks-artificial life. *ECML'98 Demonstration and Poster Papers, Chemnitzer Informatik Berichte*, pages 7–9.

Kovačič, M. and Župerl, U. (2020). Genetic programming in the steelmaking industry. *Genetic Programming and Evolvable Machines*, 21(1):99–128.

Koziel, S. and Yang, X.-S. (2011). *Computational optimization, methods and algorithms*, volume 356. Springer.

Kuhn, M., Johnson, K., et al. (2013). *Applied predictive modeling*, volume 26. Springer.

Lane, N. (2015). *The vital question: energy, evolution, and the origins of complex life*. WW Norton & Company.

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

Lecointre, G. and Le Guyader, H. (2006). *The tree of life: a phylogenetic classification*. Harvard University Press.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218.

Lehman, J. and Stanley, K. O. (2011c). Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*, pages 37–56. Springer.

Lehman, J., Stanley, K. O., et al. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336. Citeseer.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2013). Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 343–350.

Liew, M. J., Salleh, A. H. M., Mohamad, M. S., Choon, Y. W., Deris, S., Samah, A. A., and Majid, H. A. (2016). In silico gene deletion of escherichia coli for optimal ethanol production using a hybrid algorithm of particle swarm optimization and flux balance analysis. *Jurnal Teknologi*, 78(12-3).

Long, C. P. and Antoniewicz, M. R. (2019). Metabolic flux responses to deletion of 20 core enzymes reveal flexibility and limits of e. coli metabolism. *Metabolic Engineering*, 55:249–257.

López, F. G., Torres, M. G., Batista, B. M., Pérez, J. A. M., and Moreno-Vega, J. M. (2006). Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research*, 169(2):477–489.

Marashi, S.-A., Kouhestani, H., and Mahdavi, M. (2013). Studying the relationship between robustness against mutations in metabolic networks and lifestyle of organisms. *The Scientific World Journal*, 2013.

Mardinoglu, A., Gatto, F., and Nielsen, J. (2013). Genome-scale modeling of human metabolism–a systems biology approach. *Biotechnology journal*, 8(9):985–996.

Martin, W. (2010). Evolutionary origins of metabolic compartmentalization in eukaryotes. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1541):847–855.

Mienda, B. S., Salihu, R., Adamu, A., and Idris, S. (2018). Genome-scale metabolic models as platforms for identification of novel genes as antimicrobial drug targets. *Future Microbiology*, 13(4):455–467.

Mienda, B. S. and Shamsir, M. S. (2015). Bioscience and bioengineering communications.

Mienda, B. S., Shamsir, M. S., and Illias, R. M. (2016). Model-guided metabolic gene knockout of gnd for enhanced succinate production in escherichia coli from glucose and glycerol substrates. *Computational biology and chemistry*, 61:130–137.

Monmarché, N., Guinand, F., and Siarry, P. (2010). *Artificial ants*. Wiley-iste Hoboken.

Mouret, J.-B. (2020). Evolving the behavior of machines: from micro to macroevolution. *Iscience*, 23(11):101731.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Mustafa, M. G., Khan, M. G. M., Nguyen, D., and Iqbal, S. (2018). Techniques in biotechnology: Essential for industry. In *Omics Technologies and Bio-Engineering*, pages 233–249. Elsevier.

Neveu, M., Kim, H.-J., and Benner, S. A. (2013). The "strong" rna world hypothesis: Fifty years old. *Astrobiology*, 13(4):391–403.

Nordmoen, J., Veenstra, F., Ellefsen, K. O., and Glette, K. (2021). Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8.

Oberhardt, M. A., Zarecki, R., Reshef, L., Xia, F., Duran-Frigola, M., Schreiber, R., Henry, C. S., Ben-Tal, N., Dwyer, D. J., Gophna, U., et al. (2016). Systems-wide prediction of enzyme promiscuity reveals a new underground alternative route for pyridoxal 5'-phosphate production in e. coli. *PLoS computational biology*, 12(1):e1004705.

Oh, Y.-G., Lee, D.-Y., Lee, S. Y., and Park, S. (2009). Multiobjective flux balancing using the nise method for metabolic network analysis. *Biotechnology progress*, 25(4):999–1008.

Orth, J. D., Thiele, I., and Palsson, B. (2010). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.

Pharkya, P., Burgard, A. P., and Maranas, C. D. (2004). Optstrain: a computational framework for redesign of microbial production systems. *Genome research*, 14(11):2367–2376.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Quinonez, B., Pinto-Roa, D. P., García-Torres, M., García-Díaz, M. E., Núnez-Castillo, C., and Divina, F. (2019). Map-elites algorithm for features selection problem. In *AMW*.

Renz, A., Mostolizadeh, R., and Dräger, A. (2021). Clinical applications of metabolic models in sbml format. *Systems Medicine*.

Richelle, A., David, B., Demaegd, D., Dewerchin, M., Kinet, R., Morreale, A., Portela, R., Zune, Q., and von Stosch, M. (2020). Towards a widespread adoption of metabolic modeling tools in biopharmaceutical industry: a process systems biology engineering perspective. *NPJ systems biology and applications*, 6(1):1–5.

Risi, S., Vanderbleek, S. D., Hughes, C. E., and Stanley, K. O. (2009). How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 153–160.

Rocha, I., Maia, P., Rocha, M., and Ferreira, E. C. (2008). Optgene: a framework for in silico metabolic engineering.

Ruckerbauer, D. E., Jungreuthmayer, C., and Zanghellini, J. (2014). Design of optimally constructed metabolic networks of minimal functionality. *PLoS One*, 9(3):e92583.

Schellenberger, J., Park, J. O., Conrad, T. M., and Palsson, B. Ø. (2010). Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics*, 11(1):1–10.

Schopf, J. W. and Packer, B. M. (1987). Early archean (3.3-billion to 3.5-billion-year-old) microfossils from warrawoona group, australia. *Science*, 237(4810):70–73.

Scossa, F. and Fernie, A. R. (2020). The evolution of metabolism: How to test evolutionary hypotheses at the genomic level. *Computational and Structural Biotechnology Journal*, 18:482–500.

Seckbach, J. (2012). *Genesis-in the beginning: precursors of life, chemical models and early biological evolution*, volume 22. Springer Science & Business Media.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710.

Shabestary, K. and Hudson, E. P. (2016). Computational metabolic engineering strategies for growth-coupled biofuel production by synechocystis. *Metabolic engineering communications*, 3:216–226.

Slowik, A. and Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32(16):12363–12379.

Stanley, K. O. and Lehman, J. (2015). *Why greatness cannot be planned: The myth of the objective*. Springer.

Stanley, S. M. (1975). A theory of evolution above the species level. *Proceedings of the National Academy of Sciences*, 72(2):646–650.

Stephanopoulos, G. (2012). Synthetic biology and metabolic engineering. *ACS synthetic biology*, 1(11):514–525.

Systems, F. (2017). Genetic algorithms and evolutionary algorithms - introduction.

Tavassoly, I., Goldfarb, J., and Iyengar, R. (2018). Systems biology primer: the basic methods and approaches. *Essays in biochemistry*, 62(4):487–500.

Tian, H., Chen, S.-C., and Shyu, M.-L. (2020). Evolutionary programming based deep learning feature selection and network construction for visual data classification. *Information Systems Frontiers*, 22(5):1053–1066.

Tomar, N. and De, R. K. (2013). Comparing methods for metabolic network analysis and an application to metabolic engineering. *Gene*, 521(1):1–14.

Tsompanas, M.-A., Bull, L., Adamatzky, A., and Balaz, I. (2020). Novelty search employed into the development of cancer treatment simulations. *Informatics in Medicine Unlocked*, 19:100347.

Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE.

Wahid, N. S. A., Mohamad, M. S., Salleh, A. H. M., Deris, S., Chan, W. H., Omatu, S., Corchado, J. M., Sjaugi, M. F., Ibrahim, Z., and Yusof, Z. M. (2016). A hybrid of harmony search and minimization of metabolic adjustment for optimization of succinic acid production. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 183–191. Springer.

Waldner, J.-B. (2013). *Nanocomputers and swarm intelligence*. John Wiley & Sons.

Walhout, M., Vidal, M., and Dekker, J. (2012). *Handbook of systems biology: concepts and insights*. Academic Press.

Wang, H., Robinson, J. L., Kocabas, P., Gustafsson, J., Anton, M., Cholley, P.-E., Huang, S., Gobom, J., Svensson, T., Uhlen, M., et al. (2021). Genome-scale metabolic network reconstruction of model animals as a platform for translational research. *Proceedings of the National Academy of Sciences*, 118(30).

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

Yang, Y.-T., Bennett, G. N., and San, K.-Y. (1998). Genetic and metabolic engineering. *Electronic Journal of Biotechnology*, 1(3):20–21.

Youssef, H., Sait, S. M., and Adiche, H. (2001). Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence*, 14(2):167–181.

Zhang, C. and Hua, Q. (2016). Applications of genome-scale metabolic models in biotechnology and systems medicine. *Frontiers in physiology*, 6:413.

Zhi, H. and Liu, S. (2019). Face recognition based on genetic algorithm. *Journal of Visual Communication and Image Representation*, 58:495–502.

APPENDIX A

SUPPLEMENTARY FIGURES

A.1 MAP-Elites for *in silico* metabolic engineering

A.1.1 Materials and methods

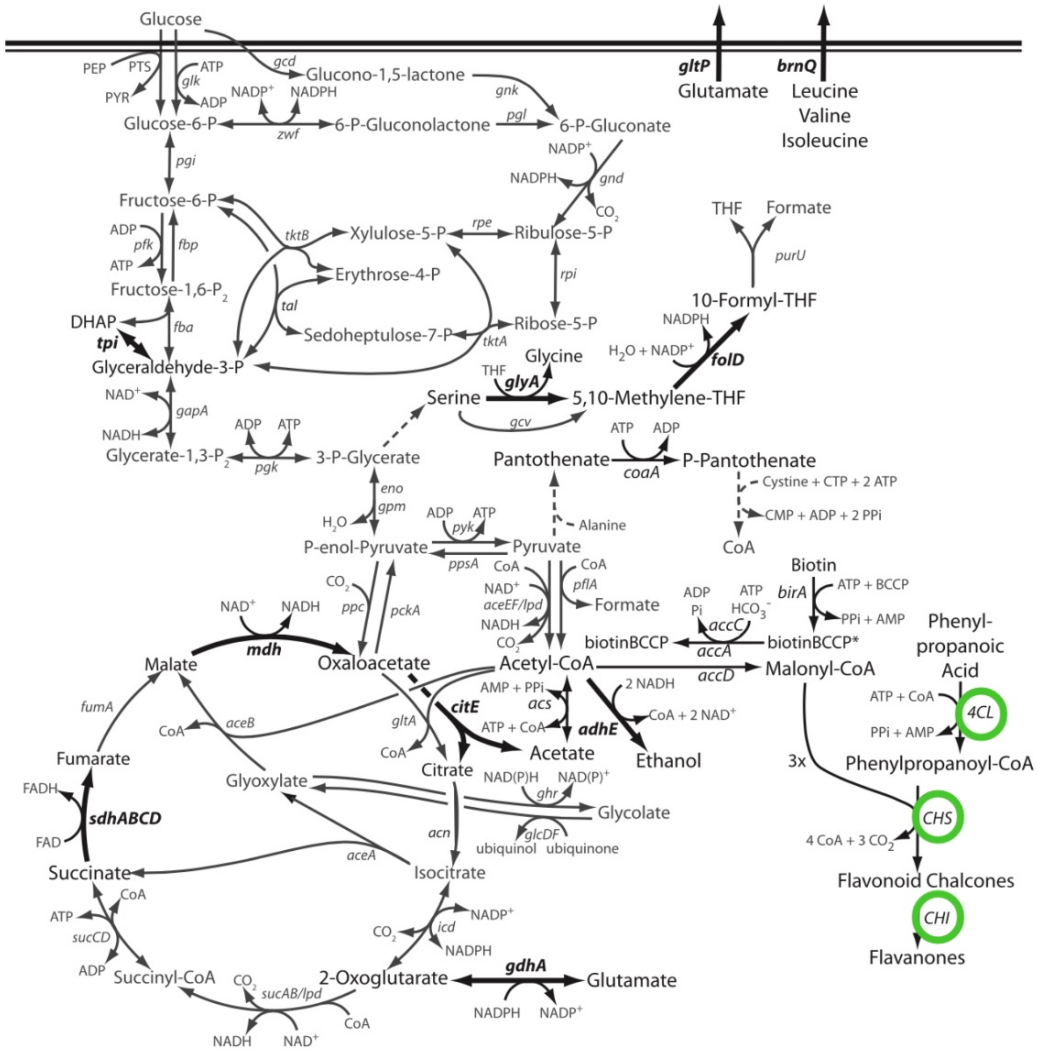


Figure A.1: A partial view of the metabolic network of *Escherichia coli str. K-12 substr. MG1655* with the reactions making up the heterologous flavanone pathway circled in green. These reactions include 4-coumarate-CoA ligase (4CL), Chalcone synthase (CHS) and Chalcone isomerase (CHI). Modified from Fowler et al. (2009).

A.1.2 Results and discussion

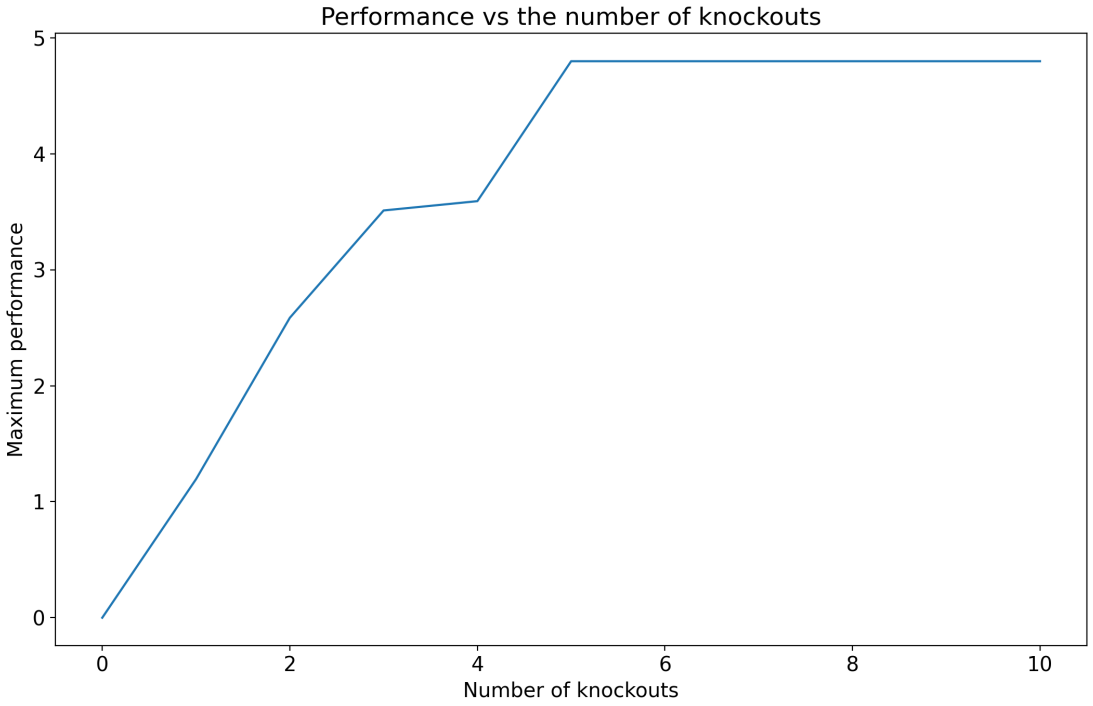


Figure A.2: Plot showing the relation between the number of knockouts and the maximum performance over all the different types of metabolism in the case of succinate overproduction. The performance illustrates the BPCY and is measured in gram product per gram glucose per hour.

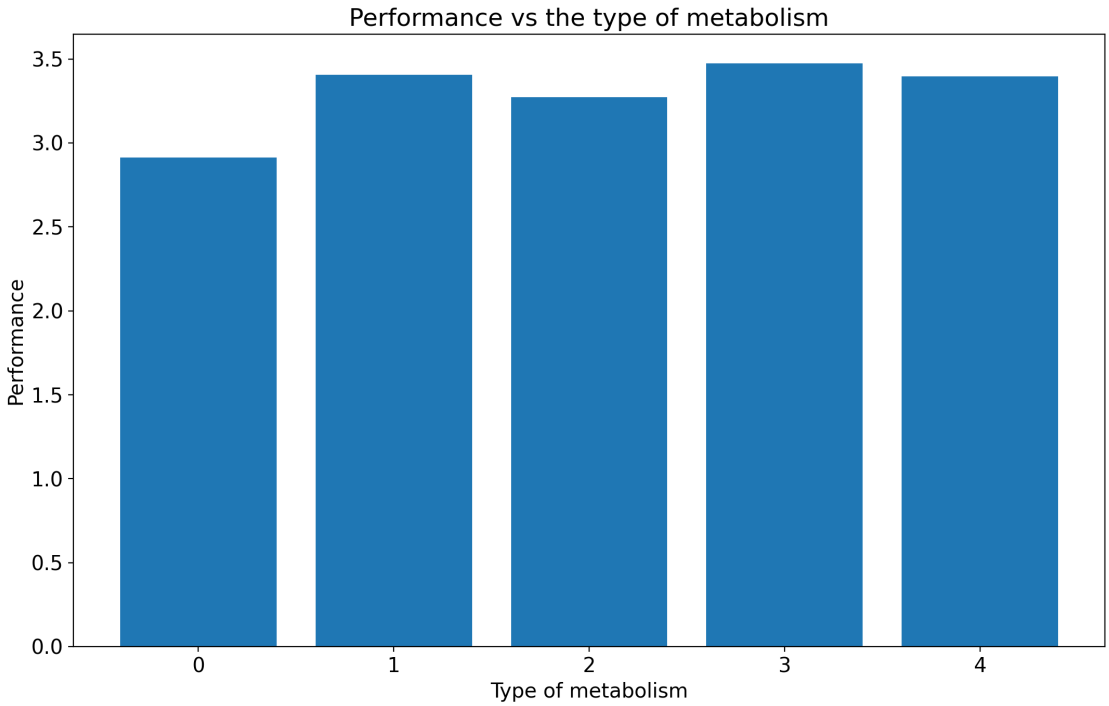


Figure A.3: Plot showing the relation between the type of metabolism the genes are related to and the average performance over all the number of knockouts in the case of succinate overproduction. The performance illustrates the BPCY and is measured in gram product per gram glucose per hour.