Design automation in synthetic biology: a dual evolutionary strategy

Sungshic Park

School of Computing Science

Newcastle University

A thesis submitted for the degree of

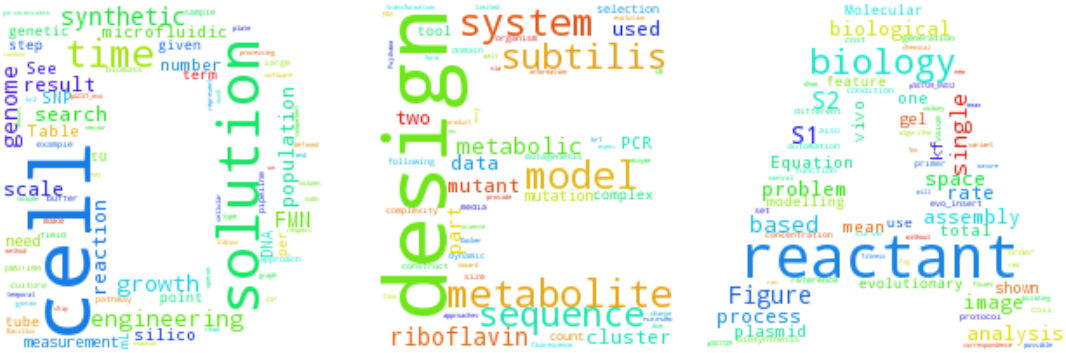
Doctor of Philosophy

June 2019

Abstract

Synthetic biology offers a new horizon in designing complex systems. However, unprecedented complexity hinders the development of biological systems to its full potential. Mitigating complexity via adopting design principles from engineering and computer science fields has resulted in some success. For example, modularisation to foster reuse of design elements, and using computer assisted design tools have helped contain complexity to an extent. Nevertheless, these design practices are still limited, due to their heavy dependence on rational decision making by human designers. The issue with rational design approaches here arises from the challenging nature of dealing with highly complex biological systems of which we currently do not have complete understanding. Systematic processes that can algorithmically find design solutions would be better able to cope with uncertainties posed by high levels of design complexity. A new framework for enabling design automation in synthetic biology was investigated. The framework works by projecting design problems into search problems, and by searching for design solutions based on the *dual-evolutionary approach* to combine the respective power of design domains *in vivo* and *in silico*. Proof-of-concept ideas, software, and hardware were developed to exemplify key technologies necessary in realising the dual evolutionary approach. Some of the areas investigated as part of this research included single-cell-level microfluidics, programmatic data collection, processing and analysis, molecular devices supporting solution search *in vivo*, and mathematical modelling. These somewhat eclectic collection of research themes were shown to work together to provide necessary means with which to design and characterise biological systems in a systematic fashion.

Keywords:


synthetic
genetic
microfluidic
step
time
given
genome
See
number
term
result
SNP
search
Table
scale
cell
reaction
solution
population
DNA
per
FMN
need
growth
point
engineering
tube
measurement
silico
design
system
selection
used
tool
subtilis
two
metabolic
PCR
data
model
mutant
mutation
complex
part
metabolite
sequence
riboflavin
cluster
Molecular
biological
biology
feature
S2
one
gel
single
vivo
S1
Equation
space
problem
rate
modelling
based
mean
use
assembly
total
reactant
Figure
evolutionary
shown
process
image
plasmid
analysis

A string of light pierced into a space filled with what seemed like eternal darkness. I chased the light, chased the light, and chased the light. Still out of reach, but never out of sight was the light of wisdom. Better not be an illusion, nor hallucination, as I have begun to run out of breath. The Plato's cave, where I have been locked up against my will for life, pulls me back into the safe harbour, into the eternal bliss of ignorance. I have been adrift in the void for so long; I wish to tell my olde days so long. I erected my petrified soul and hurled it against the air in a paddling motion. Out of desperation goes my wishful thought of making butter out of thin air; so one day, I may be able to stand on the butter of wisdom and strut out of this cave. For thee who shalt not doubt, making air taste like cream would be as miraculous an act as making wine out of water. For philosophers like I who question and reason, phenomena exist not in miracle but in causality. So I keep on paddling my soul as yet, by tuning into a coxswain's shout, toasting once to the art of doubting, and twice to doubting my own doubts.

But there are some things I am determined never to doubt: that I owe a debt of gratitude to my parents, my brother and my wife, and that I love them.

I would like to dedicate this thesis to my parents Soonja and Juhnguen who shaped me to dream my dream, to my brother Moonshic who introduced to me the art of computing for chasing my dream with, and to my wife Yeji who made the tortuous marathon run of beating the butter of wisdom a bearable journey.

Shakespeare did a great job distilling my prose into poetry:

'Doubt thou the stars are fire;
Doubt that the sun doth move;
Doubt truth to be a liar;
But never doubt I love.'

Acknowledgements

Throughout the project, many people have provided me with help of all sorts. It just so happened that many of them simply shared similar passion in research. What amazed me though is that these great minds were willing to provide help *pro bono*. I would like to acknowledge their generosity and express my gratitude to my contemporaries, listed in an arbitrary order: Catherine McAndrew, Keith Flanagan, Pawel Widera, Wendy Smith, Lucy Eland, Martin Sim, Laurence Orr, John Hedley, Neil Keegan, Susanne Pohl, Beth Lawry, Chris Taylor, Curtis Madsen, Katherine James, Goksel Misirli, Joe Mullen, Matthew Collison, Matthew Pocock, Owen Gilfellon, Jeff Errington, Richard Daniel, Jacob Biboy, Harold Fellermann, Aurelie Guyet, Colin Harwood, Seann Keith, Birgit Koch, Jurek Zozyra, Jad Sassine, Tom Richardson, Sheryl Rowland, Daniela Vollmer, Ling Juan Wu, Heath Murray, Nicola Lazzarini, Alessandro Ceccarelli, Fedor Shmarov, Charles Winterhalter, Jonny Naylor, Omer Markovitch, Sol Lim, Peter Andras, Jeremy Revell, James Mclauphlin, James Skelton, Michael Bell, Jennifer Warrender, Roman Bauer, Marcus Kaiser, Nunzia Lopiccolo, Phil Lord, Clare Smith, Paolo Zuliani, Jens Kristian Geyti, Tarique Perera, Andre Fenton, Eun Hye Park, Mario Luoni, Trevor Koob, and Joey Resignato.

I would like to give my special hat tip to my research mentors Anil Wipat and Jennifer Hallinan who have recognised my passion and shared with me their superb research vision.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most useful characteristics of humanity, differentiating us from other high-order animals, is our capability to design complex systems. We have the mental ability to abstract ideas, which has helped us create and expand sciences and meta-sciences [59, 90]. Our abstracting and design capability is at the core of how we could have built the civilisation as we know it today. Technological and scientific advancements achieved so far have begun to open ways for us to read and write the code of life - deoxyribonucleic acid (DNA) sequences. There is an old saying that sums up quite nicely what is going to unravel with this newfound capability of ours. The proverb, with a little twist, goes like this: where there is a way, there is a will (to design). Brian Cox inadvertently proved the general trueness of this proverbial eventuality, in his quantum theoretic perspective on questioning the universal phenomenon: "why anything that can happen, does" [43]. The field of synthetic biology has been born, and there are wills galore to walk the way to take humanity to whole new levels of design feats. Designing complex synthetic biological systems is, undoubtedly, the next frontier in science, technology, engineering, and mathematics.

The primary thesis being set out here for discussion is about design in synthetic biology. The topic's key concept, with respect to its semantics, needs to be clarified, before the discussion can proceed any further though. It is the deceivingly simple, yet difficult, question of what it means by "design." There are a plethora of partial definitions on the meaning of design. Nevertheless, there

has not been any universally agreeable formal definition of design existing to date [141]. In an article about the etymology of design [174], Terzidis reflects on the meaning of design the word's Greek root, "σχεδόν (pronounced schedon), meaning nearly, almost, about, or approximately." Terzidis explains, "From its Greek definition, design is about incompleteness, indefiniteness, or imperfection, yet it also is about likelihood, expectation, or anticipation. In its largest sense, design signifies not only the vague, intangible, or ambiguous, but also the strive to capture the elusive." This etymological definition associates design to some activity that is intrinsically vague, trying to capture the vague. While such a definition may inspire some philosophical thoughts, it is a rhetoric far from providing a tangible definition as to what design really is in a practical sense.

According to Freeman and Hart [65], "Design encompasses all the activities involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems." This modern definition of software systems design exhibits uncanny relevance to design in synthetic biology, and is connoting the idea of engineering in its definition. Yet another interesting perspective on the meaning of design relevant to synthetic biology comes from the field of engineering design. In a book on engineering design [57], Dym explains design as part of a central activity in engineering focused on the final goal of creating artefacts, and argues that "(the meaning of) design incorporates both representation of the artifact being designed as well as the process by which design is completed". This definition suggests that engineering design is more than just providing a conceptual blueprint for the construction of artefacts or the artefacts per se. Dym's argument, by saying "design incorporates ... the process by which design is completed," is defining the scope of design to be inclusive of implementation level details, or documentation unequivocally delineating how desired artifacts can be constructed.

That synthetic biology is a field of biological science harnessed with engineering principles is an argument we often hear [5, 82, 99, 115]. This argument entails synthetic biology's unique need for modifying existing systems, contrasting other bio-science fields primarily focused on characterising existing systems. Modification as a purpose is what differentiates engineering from science. Such an engineering-centric view of systems design in synthetic biology is also in line with the prevalent use of specifications in discussing synthetic biological designs. The

initiatives in Synthetic Biology Open Language (SBOL) [69], for instance, epitomise the importance of specifications in the field. The role of specifications in design can be related to the following excerpt on differentiating engineering from science. Kroes argues [103], "Whereas in science our ideas and beliefs are adjusted to how things are in the world, the engineering attitude is precisely the opposite, namely to adapt the world to our ideas, desires and needs." The medium by which "our ideas, desires and needs" can be captured is what a specification is to design.

Within the context of this thesis, I would like to adopt the systems engineering perspective on the meaning of design, where design encompasses all the activities and artefacts needed in the realisation of functional properties. In this view, design contains not only the 'what' of the implementation but also the 'how,' via unequivocally documenting and constructing the mechanisms with which designed artifects, whether they be ideas, aesthetics, data, energy, materials, or genetic sequences, can possess intended functional properties. Such a definition of design is also relevant to the repeatability of design, a quality necessary in achieving automation.

To design is to solve design problems. Conventionally, the process of solving design problems follows its Greek root in being nebulous, requiring highly cognitive brain tasks by humans, and involving such obscure activities as brainstorming. I believe the lack of clarity in this process, at least in part, stems from having to deal with highly complicated data, leading to analysis paralysis. Choices made by the human brain are heavily influenced by how options are presented [50]. According to Herbert Simon's theory [75, p.1], the human mind has "bounded rationality" in that "people reason and choose rationally, but only within the constraints imposed by their limited search and computational capacities." For problems with incomplete or complex information, rational decision making by the human brain relies on simplified heuristics or rule-of-thumb principles that rather deviate from the laws of probability [75, p.1].

Is complexity the major culprit in causing obscurity in design activities? If so, can we make design problems more trivial to solve by making the process of dealing with highly complex data more straightforward? Would there be ways to employ a set of simple, well-defined processes, to make easily manageable any given task and corresponding data a human designer or a computerised logic deals

with? Would it ever be possible to replace the good old-fashioned, enigma-laden, and human-intelligence-driven design approach with a mindless systematic design approach more amenable to automation? Ultimately, is achieving full automation in designing synthetic biological systems a possibility?

In an attempt to shed light on these questions, my research aims to build supporting technologies to enable design automation in synthetic biology. To this end, the work presented here explores some of the key areas in the design process that would greatly benefit from automation. This work explores genetic and microfluidic means to provide an interface to facilitate data exchange between the *in vivo* and *in silico* domains. Among other important aspects of design automation, the data exchange interface would allow for the programmatic execution of tasks such as gathering and analysing measurement data, and would serve as the cornerstone in building a full design automation stack.

1.1 How design is done

Traditionally, approaches to designing complex systems can be classified into being either top-down or bottom-up [44], depending on the flow of design processes. The use of such classification is widespread across various design domains ranging from integrated circuits [98], to software systems [137, 175], and to trade agreements [2]. This dichotomous classification provides conceptual frameworks for understanding how knowledge is organised in performing design. The former refines a design through decomposition: breaking down broader concepts into narrower details. This flow can offer relatively easy a transition from the initial specification into a design. There are clear downsides to this approach in that designed systems may end up with no parts to fulfill the need, and that testing cannot start until the systems are decomposed into testable parts. The latter starts from identifying available parts, and builds a system through composition, hence testing can start early on in the design process. However, the final system may not necessarily reflect what was initially intended in the specification. Neither approach cannot be declared to be outright superior to the other. Often, designing complex systems would need to incorporate both approaches. While these classifications can give some indication as to the directionality of a design flow, they do not provide clear

methodological steps to be followed in terms of accomplishing a design.

Designing complex systems, synthetic biological systems included, requires strenuous analyses involving various domains of knowledge. Without having a system of methodologies to define and govern such analyses, it would not be possible to wield the complexity in design. Software engineering principles, such as those advocated by the Agile Manifesto [14], provide insightful ideas about managing the processes involved in the design of complex systems. Agile software development uses iterative and incremental design approaches by working with manageably small analyses and implementation tasks. Extreme programming (XP) [15] is a type of agile software development methodologies that particularly stresses the importance of testing as part of design and implementation cycles. According to the proponents of XP, how software systems design is done depends on the cost of change. One of the common assumptions about the cost of change taken into consideration in design projects is that the cost would rise exponentially as projects progress. XP proponents claim that trying to account for the future cost of change in design would impose burdensome constraints in making design decisions. Unnecessary design decisions are made early-on in the design process in order to avoid any late changes. Such a strategy would be especially detrimental to design problems with which only partial information is available upfront, as are the cases in most synthetic biology design problems. If there is a strategy that can make the cost of change a non-issue, critical decisions can be made as late in the design process as possible. XP finds the answer to the issue of cost of change in systematic testing. Having a solid testing scheme would mean that frequent changes can be applied to a design with relative ease. How costly it is to introduce late design changes would dictate the applicability of iterative and incremental design approaches to building complex systems. The more iterations of changes a design is subject to, the more feedbacks the design can receive towards satisfying the specification. XP takes this argument further by adding that no systems design can even start without first defining how to test the systems, and that no design is complete without testing. Testing in XP is what makes possible the decomposition of a complex systems design problem into as many manageably small design elements and/or iterations as possible without losing the integrity of the whole system. The notion that testing is an integral part of design is key to

taking a systematic approach to design in synthetic biology. Testing is where intended functionalities are verified, where design criteria can be defined, and where measurements and analyses can be translated into design elements. Furthermore, systematic testing keeps the cost of change contained at a reasonably low and predictable level in order to allow for late changes inevitable in synthetic biology design problems.

1.2 Design by abstraction

In dealing with complex design tasks, it is not an option but a necessity to use computer aided or automated design processes. Synthetic biology is certainly not an exception to having the need for computerised design processes. Complex designs have a lot to benefit from model-driven engineering (MDE) approaches. MDE is an approach pioneered in computer science with the aim of facilitating computer-aided systems engineering through abstraction offered by the use of models [64]. The model checking community has long been advocating the value of using abstraction in validating complex system designs. Abstraction in design allows its systems or subsystems to be tested in isolation, which in turn facilitates the achievement of higher degrees of design complexity [37]. MDE has enabled, via abstraction, large-scale design projects in software [64] and automotives [23], among others. MDE has also been adopted in synthetic biology with some success in designing RNA devices [27] and engineering metabolic pathways [97]. MDE enables the inspection of complex systems being designed via the versatile lens of *in silico* models. Models allow the simulation of physical realities at the granularity as fine as required. Models, for instance, can be used to represent physical reality at the molecular, or even lower levels, if need be. Models also provide a highly organised and functionally active means for capturing experimental data that tend to be structurally flat and functionally inactive in their original forms. Without doubt, using MDE in synthetic biology will offer to be of great help in mitigating the challenges associated to design complexity. However, the current lack of availability of well characterised biological parts from which to produce models is a hindrance to the use of MDE in practice. Adopting a MDE approach that would require minimal *a priori* knowledge is, therefore, crucial to the success

of modelling in synthetic biology.

1.3 Design in synthetic biology

The field of synthetic biology arose from the development of technological advances in molecular biology, electrical engineering and computer science. Many of the principles of electrical engineering and computer science have been adopted in synthetic biology, so much so that use of the term 'biological engineering' in the design of novel biological systems is prevalent in the field. The synthetic biology engineering life cycle includes many of the stages familiar to software engineering: specification, design, modelling, implementation, testing and maintenance. Some of the technological developments concerning this field, including Next Generation Sequencing and DNA synthesis, have begun to outpace Moore's law as witnessed in the computer industry. It is not unimaginable to foresee the design complexity of synthetic biological systems to eventually surpass that of electrical engineering and the reasoning capacity of the human brain.

Natural evolution employs iterative selection and randomness as universal means to solve complex genome-scale biological design problems. Nature's system of finding design solutions has inspired me to pursue the investigation into an evolutionary approach to achieving genome-scale design in synthetic biology. The challenging nature of genome-scale design is intensified by the computational complexity and the large parameter space inherent to modeling approaches [139], heavily dependent on kinetic [147, 176, 177], stochastic [143, 152], and cybernetic elements [161, 167]. Furthermore, parameters in models could largely be unknown, and are often costly or impossible to be characterised [148]. The dual-evolutionary approach (DEA) as suggested by Hallinan and her colleagues [80] is an attempt to provide a genome-scale design framework that can overcome the current lack of availability of well characterised parts and computational power to handle complex systems design. The approach employs solutions from two iterative search domains, namely the *in silico* and the *in vivo* counterparts. The marriage of the two poses promising capabilities with regards to facilitating large-scale engineering of genome design.

1.4 Motivation for design automation in synthetic biology

Engineering microbial organisms under the paradigm of synthetic biology has a huge potential for igniting a new era of industrial revolution. Designed organisms will have impacts on our daily lives in the future soon to arrive. In the short run, the idea of harnessing cell factories will be indispensable to the economical production of pharmaceutical substances and renewable energy resources [127]. In the long run, designed organisms will eventually open ways to provide general biocomputing platforms that can be programmed to solve problems deemed too difficult or costly for conventional silicon-based computing architectures.

Designing biological organisms *de novo* would require highly challenging engineering feats. One of the major hurdles in the design of biological systems is that molecular interactions at the genome-scale level are extremely complicated [182]. The lack of availability of well-characterised biological parts and tools in synthetic biology further restricts the design flexibility at such complex levels. Most design in synthetic biology today is done on an *ad hoc* and manual basis by domain experts [115]. This practice not only hampers genome-scale design but also involves trial and error far too costly for achieving the economy of scale. For designing large-scale biological systems, synthetic biologists can no longer rely on manual, impromptu processes. Current design practice in synthetic biology, heavily reliant on the intuition of domain experts, can only fulfill the field's market potential to a limited extent. Impending in synthetic biology, therefore, is a demand for systematic design approaches, that are more amenable to automation.

1.5 Hypothesis, aims, and objectives

Given many uncertainties associated with engineering complex biological systems, the synthetic biology engineering cycle would need to be highly iterative. Rational design decisions would be inefficient for any such iterative tasks, and would hinder large-scale design. More promising is the use of an inherently iterative, heuristic approach such as DEA, which would potentially be much easier to be automated

and be more capable in handling design complexity. Having speculative arguments thus far given about DEA, this research was based on the following hypotheses: that design complexity would correspond to the time taken to find design solutions or vice versa; that the DEA framework would be an effective means to shorten time to design, subsequently mitigating design complexity; and that the framework would be amenable to automation.

With respect to these hypotheses, the overarching aim of this project was to establish a groundwork for DEA with which to further the investigation and the development of methodologies for accomplishing design automation in synthetic biology. One of the objectives, stemming from this aim, was to investigate the feasibility of employing DEA as an effective framework to facilitate the design of complex systems in synthetic biology. Another objective was to investigate the duality of *in silico* and *in vivo* design domains, and to suggest how the two can be integrated, minding the cross-domain gap. The final objective as part of this research work was to investigate and suggest how process automation can be achieved for streamlining the highly iterative design processes of DEA.

1.6 Contributions

Some of the original works done as part of this research contributed to the following publications. The application of automated, iterative, dual evolutionary strategies, which formed the basis of many ideas put together in this research work under the umbrella term of DEA, was initially published as a positional piece [80] in which I participated as a co-author. My work on modelling genome-scale enzymatic pathways and analysing the flux balance of riboflavin biosynthesis in *B. subtilis* was applied in the prediction of key metabolisms out of the pentose phosphate pathway for riboflavin production [156]. Some parts of my work on microfluidics, including the application of novel methods on fabrication, single-cell-level measurement and analysis, were published as part of a book chapter [61].

1.7 Thesis outline

Due to the interdisciplinary nature of the field of synthetic biology, multiple scientific disciplines contributed to the writing conventions adopted as part of structuring this thesis. Chapter 2 briefs on the subject field and the prior arts of synthetic biology, and Chapter 3 gives an overview of the DEA framework. Immediately following this is Chapter 4 showing the wetlab methods used as part of this research work. Placing the methods chapter before the body chapters (Chapter 5, 6, and 7) was for abiding by the bioscience writing convention. This way of organising information was deliberately chosen as a gentle reminder of the importance of wetlab works in synthetic biology. The methods text taking precedence in the positional order would also be more practical in serving its purpose as an immediate reference point for any hands-on works in the wetlab.

The body part comprises three chapters. The first of which (Chapter 5) covers the *in vivo* design domain, followed by the work on the *in silico* design domain (Chapter 6). These two juxtaposed body chapters convey the details underpinning the duality of the DEA framework. The last body chapter to follow (Chapter 7) covers the research work on accomplishing process automation in synthetic biology via using microfluidics. Chapter 8 provides concluding remarks, in reflection of the project aims and objectives, with a comprehensive discussion about potential future works, based on specific details presented across the preceding body chapters.

Chapter 2

Background

2.1 Synthetic biology

Synthetic biology is an interdisciplinary research field that applies engineering principles to the design and construction of novel biological systems [5, 82, 99, 115]. Having the word "engineering" as part of describing this emerging biology field signifies that it is not just about observation but more about modification that matters about the field. To pursue the modification of biological systems in order to make systems that can perform useful tasks is the defining characteristic of synthetic biology. In fact, such an endeavour to alter biological systems is not new to biology. Throughout history, we have long been using selective breeding techniques to improve the yield, flavours, or aesthetics of various biological organisms such as crops [117], cattle [145], and flowers [192]. More recently, the field of genetic engineering has already opened ways to genetically modify biological systems. What would then make synthetic biology distinguished from these pioneering attempts of other fields in modifying the phenotypes and the genotypes of biological systems? The answer to this question can be found by drawing a parallel line in history and looking at the Industrial Revolution.

Steam engines are attributed to be one of the key elements that precipitated the Industrial Revolution [125]. However, the idea of using steam power to make a rotary movement had already been contemplated by ancient Greeks in the 1st

century AD (see Figure 2.1[1]). Yet, harnessing steam power to its full potential was not materialised till the 18th century. What differentiated industrialists and engineers of the 18th century from ancient Greeks was the prevalent use of standardisation. Standardised parts during the Industrial Revolution made it easy to document designs and to build artifacts from documentations in turn [53]. This consequently allowed many engineers to easily share their ideas and build ideas on top of one another. Standardised parts lowered technological barriers and fostered modularisation and reuse of functional units. Engineers in the 18th century could see further by standing on the shoulders of giants, by not having to build things from scratch.

Figure 2.1: An illustration of the ancient Greek steam engine, aeolipile [101]

The idea of fully embracing standardisation is at the heart of what distinguishes synthetic biology from the old school of genetic engineering. This nascent

[1]The aeolipile illustration is an excerpt from Knights American Mechanical Dictionary([101]), and its copyright is in the public domain.

bio-engineering field is still premature and has yet to unravel its full potential. Nevertheless, there have already been exciting achievements that exemplify the revolutionary nature of synthetic biology. Perhaps, synthetic biology is the first field to have the true potential of becoming the engineering counterpart of biology, much like chemical engineering has been for the field of chemistry. That is to say synthetic biology will bring about a new era of revolution to humanity.

2.2 Building blocks in synthetic biology: parts and devices

There has been a recurring research theme in synthetic biology skewed to implementing proof-of-concept biological equivalents of logic gates or the building blocks of electronic engineering [124, 165, 170]. The popularity in building biological logic gates [4] exemplifies the current trend of parts-based and bottom-up design approaches prevalent in synthetic biology. Such design approaches, if taken in an attempt to replicate, verbatim, the functional complexity of electronic devices in biological systems, would be fundamentally flawed. Unlike electronic systems, biological systems cannot be hardwired to construct networks of dense logic gates functioning together. The use of chemical signals as a means to wiring gates, as shown by Tamsir and his colleagues [170], can only be scaled to a limited design complexity owing to the lack of available orthogonal signals that do not interfere each other. Regot and others [142] have suggested a way to reduce wiring constraints in biological gate designs, using the concept of "distributed computing" via multiple cells working in collaboration. Their approach allowed a successful implementation of a 1-bit adder. Nevertheless, their approach would still suffer from the scalability issue in building complex systems with practical real-world applications.

2.3 Bio-design automation in synthetic biology

Bio-design automation (BDA) is critical to achieving the level of design complexity necessary in solving real-world problems. There have been initiatives in synthetic

biology communities with a consensus goal towards automating the process of biological systems design. However, the majority of these initiatives have so far been centered in computer-aided design (CAD), as versus to BDA in its true sense [115]. CAD evolves around software tools that can assist human designers with their design tasks often in terms only of managing information in a bottom-up fashion. BDA practices in synthetic biology have mainly been focused on what would be considered low-hanging fruits, only capable of providing rudimentary forms of "automation" as would CAD tools. Some of the software tools in this endeavour include TinkerCell [33] and GenoCAD [48]. TinkerCell provides a graphical user interface where a human designer can drag and drop genetic parts such as promoters, RBS, CDS, and terminators to put together a design construct. TinkerCell allows the study of the dynamics of molecular interactions by allowing the definition of cellular boundaries and types of molecular reactions using kinetic parameters. GenoCAD took a little more formalised approach to integrating genetic parts by working with rules with which matching parts from databases such as the Registry of Standard Biological Parts can be found. Tools such as GEC [135] and Eugene [19] lean towards the programmatic exploration of genetic designs by working with the concept of programming languages and abstraction to specify genetic circuit designs. These tools can be used to convert abstract low-level designs into genetic circuits with specific parts. However, they are still limited to offering parts-based and bottom-up approaches to accomplishing genetic design.

Parts-based, bottom-up approaches would offer design perspectives relatively more straight-forward, not only in terms of developing software tools but also in using such tools to document rationalised ideas by designers. However, the subject in this design process is still the human experts. This means that rational decision making is required on every critical details of a design on a manual basis. Regardless of whether all the mental somersaults in rational decision making were performed in perfection or not, the resulting design is still subject to errors in the expert knowledge and limits in the reasoning capacity of humans. The current bias toward bottom-up design approaches in synthetic biology software tool bases is perhaps attributed in part by that fruit harvest competition and heavy reliance on *ad hoc*, rational design processes. Lux and his collaborators have addressed the concerns regarding the *ad hoc* nature of current design practice in synthetic biology

[115], and urged for "the formalization of genetic design rules that determine the complex relationships between genotype and phenotype."

2.4 Metabolic engineering as an application of synthetic biology

Metabolic engineering is an attempt to design and manipulate cellular metabolism in order to facilitate the biosynthesis of molecules of interest, usually of high commercial value [187]. Metabolism is an integral cellular process involving complex sequences of chemical reactions that transport, convert and harness carbon and nitrogen fluxes for the maintenance of biological systems [36]. Metabolism provides cells with pathways to convert nutrients into energy and base chemical species necessary for making everything cells need. The kinds of chemical species or metabolites required by cells, consequently made available by their metabolism, differ depending on the environmental niches to which the cells of any given organism evolved to adapt [92].

While there are highly conserved metabolic constituents such as nucleic acids, amino acids and various hydrocarbon chains that make up some of the essential cellular components including DNA, RNA, proteins and lipids, there are whole other spectrums of esoteric secondary metabolites unique only to a handful of organisms in their wildtype conditions [93].

Metabolic engineering and recombinant DNA technologies have enabled us to make cellular factories that can produce nearly any organic molecules [47]. In that capacity, metabolic engineering seems to have a lot of common aspects with synthetic biology. Metabolic engineering could be considered a subset of areas in which synthetic biology can be applied. Nielsen and Keasling expressed in their opinion piece [127] that there currently is some degree of disconnect as well as overlap between three different disciplines of metabolic engineering, systems biology, and synthetic biology. They opined that different endeavours from these three fields will eventually have to merge together in order to facilitate progress in the engineering of biological systems.

2.5 Current affairs in synthetic biology

The field of synthetic biology is still at its infancy. While the big picture of this nascent field is portrayed as being aimed at the genome-scale design of synthetic organisms, the reality now is far from achieving designs of such a grandiose scale. Many synthetic biology projects to date have investigated the bottom-up composition of genetic circuitries out of a handful of genetic elements such as promoters, ribosome binding sites, coding sequences, and terminators. Even computer aided design tools developed for applications in synthetic biology lean towards the same bottom up approach, and are focused on providing visual means for building and simulating models that are often represented by a serial juxtaposition of a limited number of genetic elements. This approach has been moderately successful insofar as domain experts with in-depth knowledge of the biological system of interest can manually design the system. However, the interim success of CAD-based manual design approaches is going to be short-lived for the following reason. The manual, expert-driven practice of designing synthetic biological systems is not only expensive but also limited in the level of achievable design complexity.

Large-scale engineering of entire genomes has been attempted, including the *de novo* syntheses of poliovirus cDNA [31], bacteriophage genome [159], bacterial genome [74], and a chromosome of *Saccharomyces cerevisiae* [6]. However, these works have focused on the reconstruction of existing genomes. While the work of Gibson and colleagues [74] did involve modifications to the existing genome, they were confined to non-functional modifications, such as inserting watermark sequences into non-essential genes. There have been some synthetic biology projects taking genome-scale modification more seriously, rather than just tackling verbatim synthesis of wildtype genome. One faction of synthetic biologists has been tinkering with the idea of eliminating nonessential genes for genome reduction in *Escherichia coli* [138], and in *Bacillus subtilis* [172], while other faction has dealt with genome refactoring in bacteriophage T7 [32], and in *Klebsiella oxytoca* [173]. Genome refactoring attempts are especially noticeable in that functional genes are completely revamped, via forward engineering, to make synthetic strains that are phenotypically better, and genotypically more organised. To date, however, true genome-scale engineering for introducing appreciable, novel, and nontrivial

functionality in a biological system has not been achieved yet. The difficulty in reaching such a high level of engineering feat is due, in part to the heavy reliance on domain experts' manual labour, and in part to the lack of a viable design automation framework like that envisioned by DEA.

Chapter 3

A dual-evolutionary strategy for synthetic biology

There is an inherent limit to the use of domain expertise or rational decisions in design, since some design problems can harbour solution space much larger than our mind can hold. A systematic framework that can harness evolutionary strategies to design genome-scale synthetic biological systems was proposed to improve upon the limit of rational approaches [80]. The framework, called dual-evolutionary approach (DEA), comprises two complementary evolutionary processes, respectively running in *in silico* and *in vivo* domains. The idea of using DEA in exploring the design space of biological systems initially stood on the speculation that differences between *in vivo* and *in silico* evolutionary domains can be leveraged to bring about a complementary effect towards increasing each domain's design capability. The initial advocates of the idea warned against a potential gap between the two domains due to the dichotomy of the inter-domain differences, and called for research that can close the gap.

Design is a search problem. To be more specific, it is a search problem that has an indefinitely large solution space to be explored. Evolutionary processes *in vivo* have built-in mechanisms for finding solutions to design problems encoded in DNA as directed by environmental conditions. Solutions found by the *in vivo* search operation are robust, so much so that they are guaranteed to work as advertised. On the flip side, they tend to be rather too specific, and may not easily offer the

generality required. Given enough time and the right prescription of environmental conditions, solutions satisfying any criteria, as long as they conform to the law of physics, will emerge. Difficulties arise when we try to harness evolutionary processes in engineering biological systems. Firstly, there is a timescale dilemma. The normal evolutionary timescale is not something for which synthetic biologists trying to engineer genetic systems would have patience to wait. Also troublesome is that there is no framework with which to manipulate environmental conditions for directing evolution to find its ways to desired solutions.

DEA can come as a rescue in terms not only of reducing the evolutionary timescale, but also of serving as a framework for genome-scale biological systems design. Any data generated as a result of directed evolution *in vivo* can be stored digitally, to serve as reference ingredients to be filtered, mixed, and recombined in variable combinations as directed by evolutionary processes *in silico*. The dual evolutionary domains can run in iterative cycles towards reaching a design goal, as depicted in Figure 3.1.

The solution space for a genome-scale design is hugely vast. The norm in the current practice of engineering biological systems is still heavily dependent on the use of scarce domain experts and manual labour. This manual approach unnecessarily imposes on the designer a large portion of the burden of exploring solution space, hence does not scale well to genome-scale designs hidden in complex, high-dimensional solution space. The adoption of evolutionary algorithm (EA) or EA-like principles in the exploration of solution space in DEA bodes well with the framework's requirements for supporting the following aspects: general applicability, flexibility, and scalability in handling a wide range of different types of problems. These aspects would offer to be an effective arsenal against problems lacking *a priori* knowledge, as is often the case in synthetic biology. DEA can lessen designer's burden by providing a viable framework with which much of the solution exploring endeavour can be delegated, in a standard manner, to the dual, mutually complementary search domains. Due to its heavy reliance on the EA-like principles, DEA would also be affected by the same drawbacks of EA. These include the issue of premature settlement on local optima, or the lack of means to determine if solutions at hand are local or global.

The DEA framework can overcome the issue of local optima, via supporting

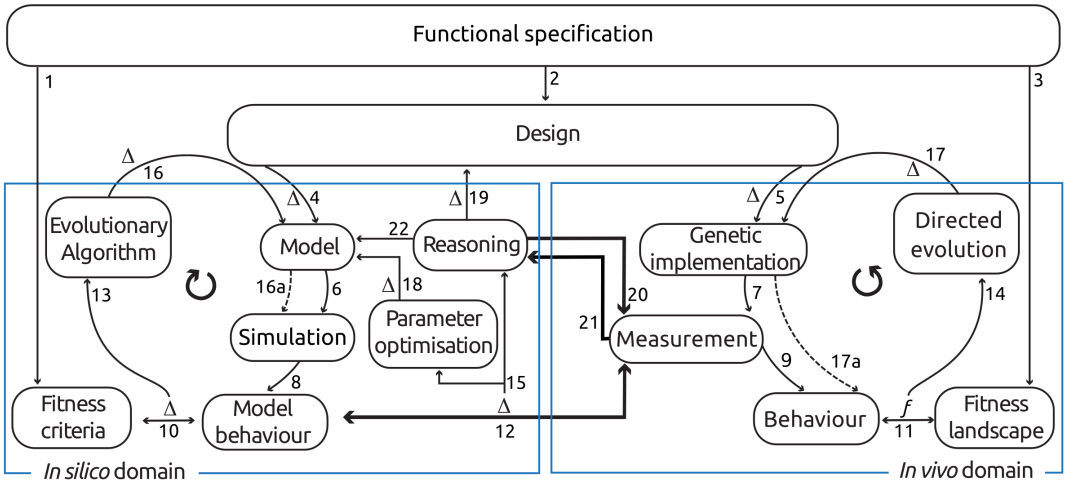


Figure 3.1: Overview of the DEA framework for synthetic biology

Functional specification is used **1** to define a desired model behaviour for *in silico* evolution, **2** to come up with an initial starting point of a design, or **3** to define environmental incentives for *in vivo* evolution. A design can be converted, as necessary, to **4** *in silico* models, or **5** *in vivo* genetic implementations. **6** A model is simulated. **7** The characteristics of genetic implementations, including any changes in the genome sequences and in key phenotypes, are measured. **8** Model simulation predicts the potential behaviour of a design. **9** Measured characteristics may reveal the *in vivo* behaviour (phenotype) of a genetic implementation. **10** Model behaviour is compared to the desired behaviour. **11** *In vivo* behaviour is affected by environmental incentives defined by fitness landscapes and vice versa. **12** *In vivo* behaviour is compared to *in silico* behaviour. **13** Differences between the desired and model behaviours feed back into the Evolutionary Algorithm (EA). **14** The dynamics between environmental incentives and phenotypes is the driving force (f) for directing evolutionary changes. **15** The comparison between *in silico* and *in vivo* behaviours can be used as a basis for parameter optimisation, and reasoning about changes to the design, the model, or the measurement criteria. **16** The EA can affect the model view of the design, and **16a** its subsequent simulation. **17** Any evolutionary changes are directly recorded in the genome, serving as a medium for genetic implementations *in vivo*. **17a** Genetic changes may result in behavioural changes. **18** Parameter optimisation, via studying the differences of the two domains can help refine the model. Reasoning based on the comparison of the two domains can **19** lead to design changes or **20** affect measurement criteria. **21** Information on meaningful mutations in genetic implementations due to evolutionary changes can be used in the reasoning process to refine the design, or **22** to refine the model. **Note**: The Δ symbol denotes difference or comparison in 10 and 12, and change or modification in 4, 5, 16, 17, 18 and 19. The **thick arrows** (20, 21, and 12) constitute cross-domain information exchange, indicating the areas of concern for building a cross-domain data interface.

effective means to assess the fitness of solutions in each domain (See Note 1, 3, 10, and 11 in Figure 3.1). Successful fitness assessment would subsequently grant the possibility of determining the validity of design solutions, and make the uncertainty of whether a solution is local or global a nonissue.

There are key elements necessary in achieving genome-scale designs through the dual evolutionary approach, as depicted and noted throughout Figure 3.1. First, we need modeling schemes that would require as little parameterisation (Note 15 and 18) as possible, as each parameter costs a substantial amount of computation. Also needed are means by which solutions are encoded (Note 4 and 5), generated (Note 16 and 17), explored (Note 8 and 9), assessed (Note 10 and 11), and measured (Note 6 and 7) in either or both side(s) of the two domains. And finally, we need means to integrate the findings from the two domains (Note 15 and 21) in order to perform reasonably efficient exploration of the solution space via refining the model (Note 22); The integration of which requires a cross-domain data interface as noted in thick arrows in Figure 3.1.

As hinted by the thick arrows being the only links constituting the cross-domain data interface, the *Measurement* element, depicted in Figure 3.1, is of paramount importance to the realisation of the *in vivo* domain, in terms of closing the *in silico*-*in vivo* gap. Regarding the *Measurement* element, affecting the measurement criteria (Note 20) includes making changes not only in the measurement apparatus level, but also in the growth condition level directly affecting mutant cells. The latter case may induce some conditional behaviours in mutant cells, intended as part of the controlling process of DEA in the evolutionary processes. It seems as though, on the depiction in Figure 3.1, the comparison of *in silico* and *in vivo* behaviours (Note 12) takes place directly between the *Measurement* and the *Model behaviour* elements. However, the comparison in actuality takes place inside the *Reasoning* element. So the cross-domain data interface, in essence, only comprises the two links between the *Reasoning* and the *Measurement* elements (Note 20 and 21).

Chapter 4

Methods

4.1 Materials

Table 4.1: Bacterial strains used in this study:

Strain	Genotype	Source*
E. coli DH5alpha	F- λ- Φ80*lacZ*ΔM15 Δ(*lacZYA-argF*)U169 *recA1 endA1 hsdR17* Δ*phoA8 supE44 thi-1 gyrA96 relA1*	CBCB
E. coli MC1061	str. K-12 F- λ- Δ(*ara-leu*)7697 [*araD139*]B/r Δ(*codB-lacI*)3 *galK16 galE15* e14- *mcrA0 mcrA0 relA1 rpsL150*(Str^R) *spoT1 mcrB1 hsdR2*	CBCB
B. subtilis 168	*trpC2*	BGSC, CBCB
B. subtilis BSB1	str. 168 trp prototroph	CBCB
B. subtilis EVOt2	str. BSB1 *amyE*::pEVOt2	This study

* BGSC: Bacillus Genetic Stock Center at Ohio State University. CBCB: Centre for Bacterial Cell Biology at Newcastle University. Among the CBCB acquired strains, *E. coli* DH5alpha [171] was a gift from Wendy Smith, *E. coli* MC1061 [30] from Ling Juan Wu, *B. subtilis* 168 [194] from Heath Murray, Aurelie Guyet and Wendy Smith, and *B. subtilis* BSB1 [126] from Wendy Smith.

Table 4.2: Plasmids used in this study:

Plasmid	Description	Usage	Source*
pUC57	*bla*, ColE1/pMB1/pBR322/pUC ori, *lacZα*	transformation	Genscript
pMUTIN4	*bla*, *erm*, Pspac, *spoVG*_RBS-*lacZ*, *lacI*, ColE1 ori	integration vector	CBCB
pSG1729	*spc*, *bla*, ColE1 ori, *amyE'-'amyE*, Pxyl *gfpmut1*	integration vector	CBCB
pET28	Kan, T7 promoter, His-Tags T7-Tag, *lacI*, pBR322 ori, f1 ori	expression	CBCB
pUC57_evo	pUC57::evo_insert	transformation	This study
pMUTIN4_evo	pMUTIN::evo_insert	integration vector	This study
pEVOt2	pSG1729::EVOt2	integration vector	This study

* CBCB: Centre for Bacterial Cell Biology at Newcastle University. Among the CBCB acquired plasmids, pMUTIN4 [180] and pSG1729 [112] were a gift from Ling Juan Wu, and pET28 from Jad Sassine.

Table 4.3: Reagents used in this study:

Reagents	Source
dNTP	Promega
PCR DNA oligos	IDT
Q5 DNA polymerase	NEB
Restriction digest enzymes	NEB
Gibson assembly cloning kit	NEB
QIAquick PCR purification kit	QIAGEN
QIAprep Spin Miniprep kit	QIAGEN
DNA gel extraction kit	QIAGEN
DNeasy Blood & Tissue kit	QIAGEN
Ethidium Bromide	Sigma-Aldrich
Bacto-trypton	BD Biosciences
Bacto Yeast extract	BD Biosciences
Bacto Agar	BD Biosciences
Starch	Fisher

4.2 The construction of pMUTIN4_evo plasmid vector

The FMN sensor construct together with other constructs for inducible hypermutation, collectively dubbed the name the *evo_insert* as discussed in chapter 5, were synthesised by Genscript (NJ, USA). The evo_insert sequence (See Appendix B.1 for sequence details) was designed to be flanked by two restriction sites, BamHI and HindIII, and was inserted into the corresponding sites in the multiple cloning site (MCS) of pUC57 (Genscript, USA). This pUC57 plasmid carrying the insert sequence was named pUC57_evo (Figure 4.1).

4.2.1 Subcloning the evo_insert construct from pUC57_evo into pMUTIN4

Using pUC57_evo as the donor plasmid, the insert sequence was subcloned into pMUTIN4 [180]. This recipient plasmid with the insert in between its BamHI and HindIII was named pMUTIN4_evo (Figure 4.2).

4.2.1.1 Restriction digests to prepare evo_insert and pMUTIN4 backbone fragments

A double restriction digest was performed as described in Table 4.4. A 100 mL agarose gel (0.7 %) was made using the standard protocol (Section 4.5.3) with the exception of not adding EtBr in the gel. A comb with the largest available well size was taped to make two wells, large enough to accomodate 96 µL of solutions in each well. 80 µL of each of the reactants ID'ed **i** and **ii** from Table 4.4 was mixed with 16 µL of 6X loading dye. An electrophoresis tray was prepared with 1X TAE buffer without EtBr, and the gel was placed in. Each of the 96 µL of reactant solutions mixed with dye were loaded onto the gel. 80 V of electricity was applied for 1.5 h. The gel was transferred into MilliQ water with 0.5 µg mL^{-1} EtBr and was stained for 15 min, followed by resting 15 min in fresh MilliQ water for destaining. The gel was then UV inspected, on low dose, long wavelength light, to quickly excise the pUC57_evo's evo_insert band at around 3585bp region, and the

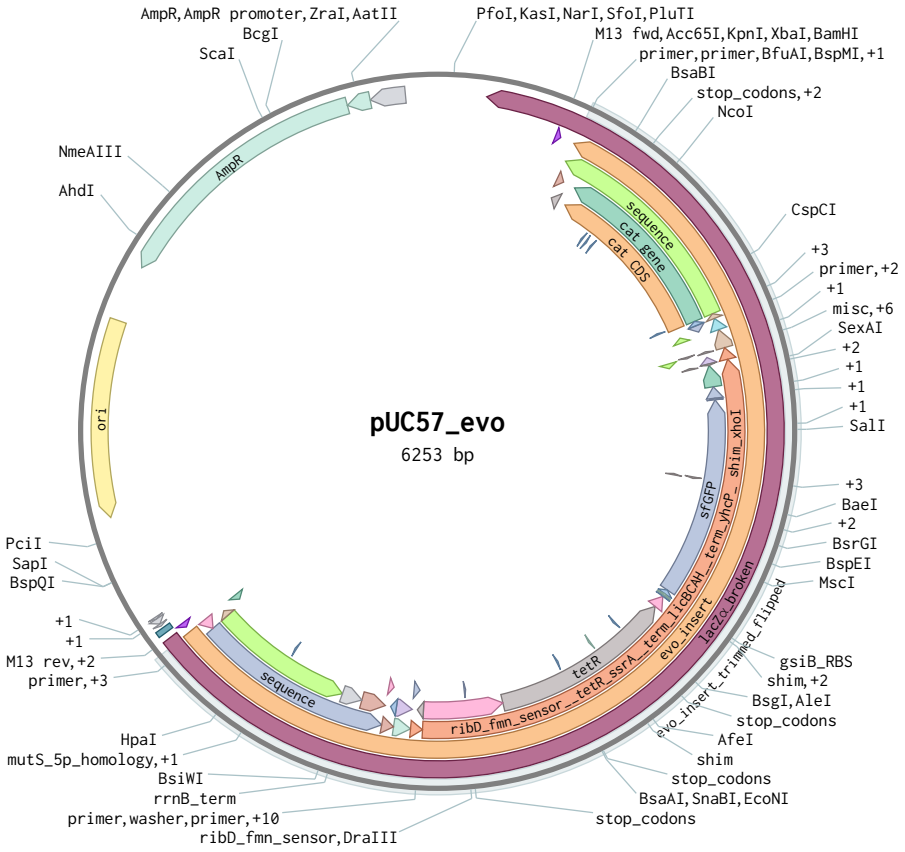


Figure 4.1: The pUC57_evo plasmid map.

pMUTIN4 backbone band at around 8600bp region out of the gel, using a sterile scalpel. The pUC57_evo and pMUTIN4 gel slices were weighed to be 160 mg and 100 mg respectively. The two gel slices were subject to the DNA gel extraction protocol as detailed in Section 4.5.1. Nanodrop runs on the gel extracted DNA resulted in:

4.2.1.2 Double Restriction Enzyme Digestion

E. coli DH5α cultures respectively transformed with pUC57_evo and pMUTIN4 were grown overnight on a shaking incubator at 37.0 °C. The overnight cultures

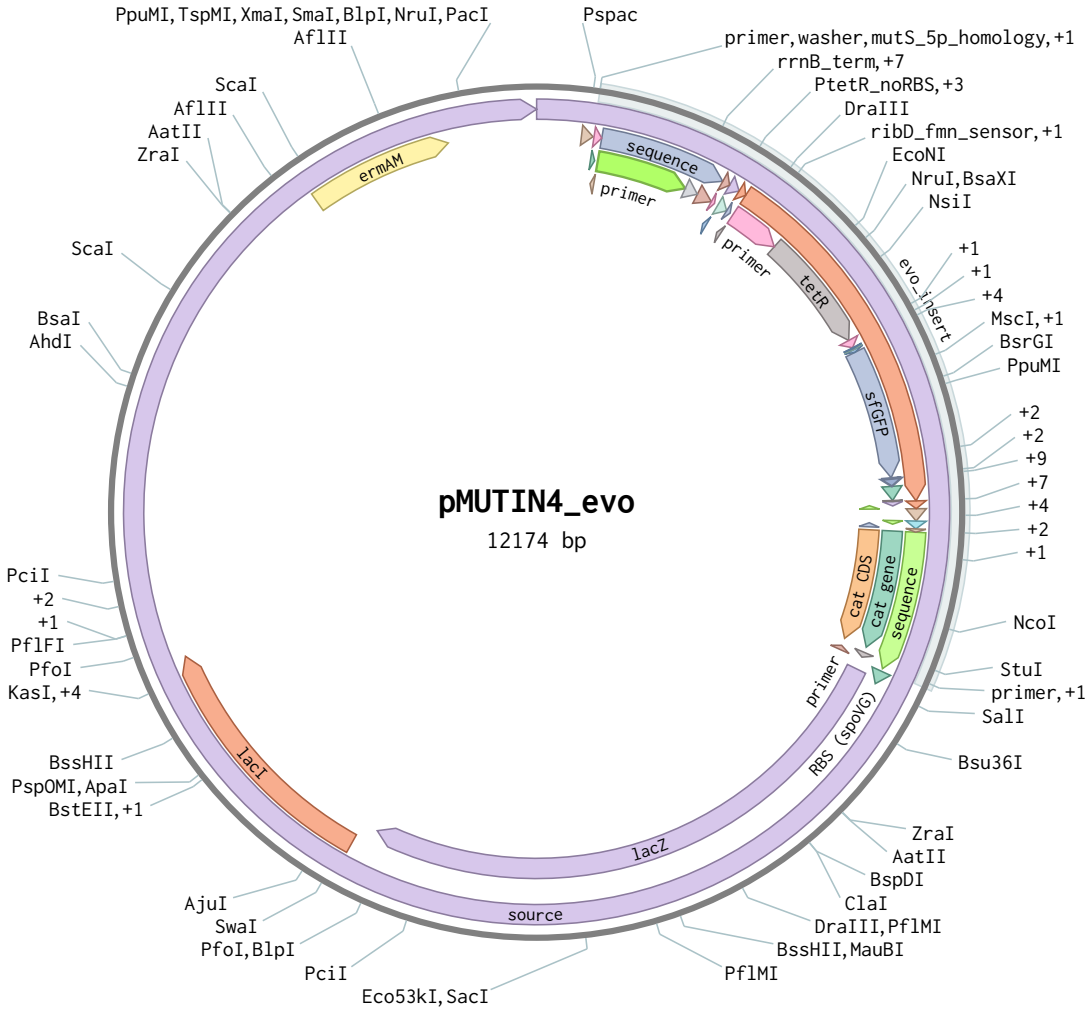


Figure 4.2: The pMUTIN4_evo plasmid map.

were Miniprep'ed using the standard protocol described in Section 4.5.8 The double digest was performed as summarised in Table 4.4.

A diagnostic gel electrophoresis was performed to verify the result of the restriction digest reactions. 0.7 % agarose gel was made by following the standard protocol (Section 4.5.3). The gel was placed into an electrophoresis buffer tray, loaded with reaction samples (with 6X loading dye), and applied with 100 V for 1 h. The gel was UV imaged on a gel dock afterwards. Figure 4.3 shows that pUC75_evo and pMUTIN4 were digested as expected, and resulted in correct

	evo_insert	pMUTIN4 backbone
ng/µL	36.5	26.6
260/280	1.71	1.66
260/230	0.13	0.14

Table 4.4: Double restriction enzyme digest reaction, as a prestep to subcloning evo_insert into pMUTIN4

	Doner plasmid (main) (pUC57_evo)	Vector backbone (main) (pMUTIN4)	Doner plasmid (diagnostic) (pUC57_evo)	Vector backbone (diagnostic) (pMUTIN4)
Rxn ID	i	ii	iii	iv
Total rxn volume	100µL	100µL	20µL	20µL
pUC57_evo (288.9ng/µL)	38µL (11µg)	N/A	1.7µL (0.5µg)	N/A
pMUTIN4 (276.7ng/µL)	N/A	39.8µL (11µg)	N/A	1.8µL (0.5µg)
10X rxn buffer	10µL	10µL	2µL	2µL
dH2O	50.0µL	48.2µL	15.8µL	15.7µL
BamHI (NEB)	1µL	1µL	0.5µL	0.5µL
HindIII (NEB)	1µL	1µL	N/A	N/A
Incubation	37°C for 4 hours	37°C for 4 hours	37°C for 4 hours	37°C for 4 hours
Inactivation	80°C for 20 mins	80°C for 20 mins	N/A	N/A

number of bands with right sizes.

4.2.1.3 Ligation assembly of evo_insert and pMUTIN4 vector backbone

A ligation reaction to assemble the evo_insert and pMUTIN4 backbone fragments was performed as summarised in Table 4.5. Competent *E. coli* DH5alpha was

Table 4.5: The reaction setup of the ligation of evo_insert and pMUTIN4 fragments:

	evo_insert+pMUTIN4 backbone
Total rxn volume	20µL
300ng insert	8.2µL (36.5ng/µL)
100ng vector	3.8µL (26.6ng/µL)
10X rxn buffer	2µL
T4 ligase	2µL
dH2O	4µL
Incubation	25°C for 2 hours

transformed (See the protocol in Section 4.5.7), by adding 15 µL of the above ligate in lieu of adding circular plasmid DNA. The transformant *E. coli* culture was spread on ampicillin (100 µg mL^{-1}) LB agar plates pre-treated with a 200 µL solution made of 20 µL 1 M IPTG, 140 µL MilliQ water, 40 µL 4 % Xgal (40 µg mL^{-1}). After overnight growth, two colonies survived and passed the blue/white colony

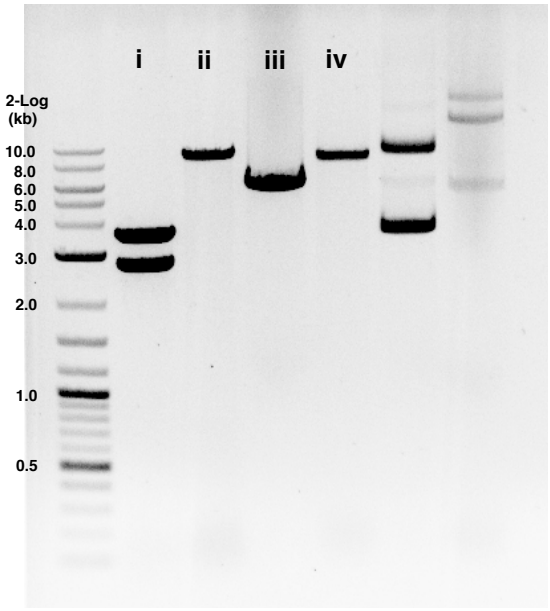


Figure 4.3: A diagnostic gel electrophoresis image for the restriction digest from Table 4.4.

test[1]. Two overnight cultures in LB with 100 µg mL^{-1} ampicillin were set up from the two colonies. Only one of the two cultures turned turbid. This culture was minipreped as per Section 4.5.8, and the resulting plasmid DNA was subject to diagnostic restriction enzyme digest and gel electrophoresis as follow.

Table 4.6: Diagnostic restriction enzyme digest reactions, to determine the validity of the putative pMUTIN4_evo

	Rxn1	Rxn2	Rxn3
Total rxn volume	20µL	20µL	20µL
DNA (372.5ng/µL)	1.4µL (0.5µg)	1.4µL (0.5µg)	1.4µL (0.5µg)
10X rxn buffer	2µL	2µL	2µL
dH2O	15.7µL	16.0µL	16.3µL
BamHI (NEB)	0.3µL	0.3µL	0.3µL
HindIII (NEB)	0.3µL	0.3µL	N/A
SalI (NEB)	0.3µL	N/A	N/A

The diagnostic gel electrophoresis was run using 0.7 % gel, and the resulting bands indicated the size of pUC57_evo, not that of pMUTIN4_evo (See Figure

[1]A successful insertion of the evo_insert fragment into the multiple cloning sites of the pMUTIN4 backbone would result in the disruption of *lacZ* and result in white colonies.

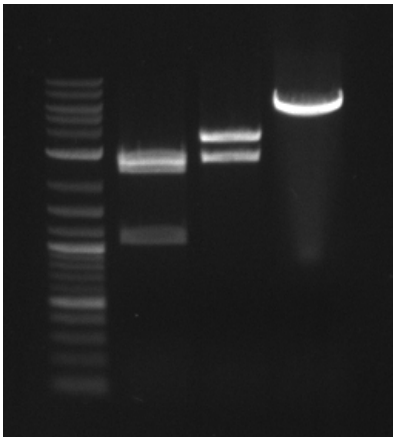


Figure 4.4: A diagnostic gel electrophoresis image for the above restriction digest reactions: the bands are indicative of pUC57_evo, not pMUTIN4_evo.

4.4).

Transformation using the ligation product out of the gel-purified fragments (Section 4.2.1.1 and 4.2.1.3) revealed poor efficiency resulting in only a single transformant worthy of diagnostic inspection. In the following ligation method, column purification (QIAquick spin columns) instead of gel purification was used for preparing the restriction digested DNA fragments.

A triple restriction digest was performed as described in Table 4.9. The digest solution was subject to column purification as described in Section 4.5.9. Nanodrop runs on the column purified DNA resulted in:

	evo_insert	pMUTIN4 backbone
ng/µL	85.3	31.7
260/280	1.93	1.94
260/230	2.33	1.99

The assembly of pMUTIN4_evo using ligation was performed in a second trial. DNA fragments resulting from restriction digests followed by column purification were used in the ligation assembly of pMUTIN4_evo. The second ligation reaction was run as summarised in Table 4.7. Competent *E. coli* DH5alpha was transformed (See the protocol in Section 4.5.7), by adding 15 µL of the above ligate in lieu of adding circular plasmid DNA. The transformant *E. coli* culture was spread on

Table 4.7: The reaction setup of the second ligation of evo_insert and pMUTIN4 fragments:

	evo_insert+pMUTIN4 backbone
Total rxn volume	20µL
300ng insert	3.5µL (85.3ng/µL)
100ng vector	3.2µL (31.7ng/µL)
10X rxn buffer	2µL
T4 ligase	2µL
dH2O	9.3µL
Incubation	25°C for 2 hours

ampicillin ($100\,\mu g\,mL^{-1}$) LB agar plates pre-treated with a 200 µL solution made of 20 µL 1 M IPTG, 140 µL MilliQ water, 40 µL 4 % Xgal ($40\,\mu g\,mL^{-1}$). After overnight growth, there was an abundance of colonies surviving and passing the blue/white colony test. Overnight cultures in LB with $100\,\mu g\,mL^{-1}$ ampicillin were set up from nine colonies. The overnight cultures were minipreped as per Section 4.5.8, and their resulting plasmid DNAs were subject to diagnostic restriction enzyme digests and gel electrophoresis as respectively shown in Table 4.8 and Figure 4.5.

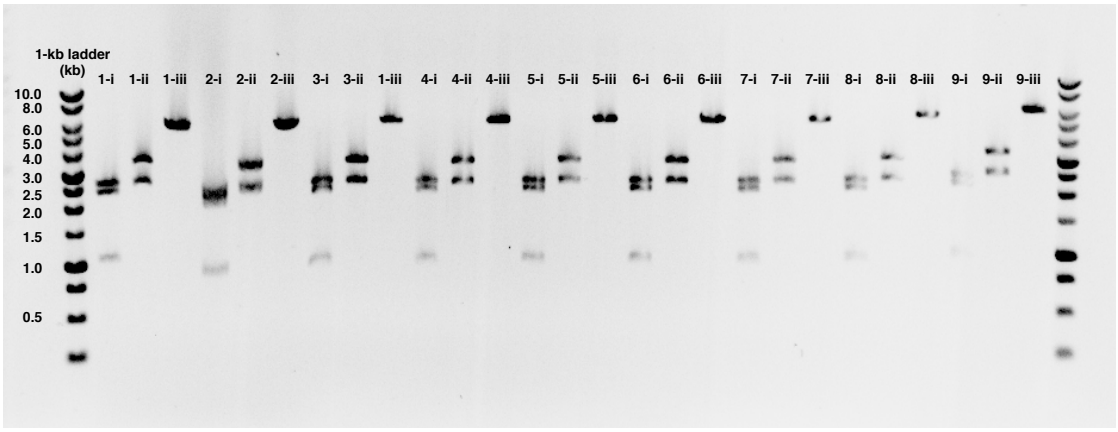


Figure 4.5: The diagnostic restriction digest gel electrophoresis of putative pMUTIN4_evo clones: all the screened colonies showed negative results.

According to the diagnostic gel electrophoresis (Figure 4.5), all of the nine colonies were shown to be habouring pUC57_evo instead of the much anticipated plasmid of pMUTIN4_evo. The ligation protocol without gel purification could

Table 4.8: Diagnostic restriction enzyme digest reactions, to determine the validity of the putative pMUTIN4_evo, assembled from column purified fragments:

colony ID	1			2			3		
rxn ID	i	ii	iii	i	ii	iii	i	ii	iii
0.4µg DNA	1.6µL (253.7ng/µL)			2.4µL (115.1ng/µL)			1.4µL (289.1ng/µL)		
Total rxn volume	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL
10X rxn buffer	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL
dH2O	15.5µL	15.8µL	16.1µL	14.7µL	15.0µL	15.3µL	15.7µL	16.0µL	16.3µL
BamHI (NEB)	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL
HindIII (NEB)	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A
SalI (NEB)	0.3µL	N/A	N/A	0.3µL	N/A	N/A	0.3µL	N/A	N/A
Incubation	37°C for 4 hours								
colony ID	4			5			6		
rxn ID	i	ii	iii	i	ii	iii	i	ii	iii
0.4µg DNA	4.2µL (93.5ng/µL)			1.3µL (296.5ng/µL)			1.5µL (273.0ng/µL)		
Total rxn volume	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL
10X rxn buffer	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL
dH2O	12.9µL	13.2µL	13.5µL	15.8µL	16.1µL	16.4µL	15.6µL	15.9µL	16.2µL
BamHI (NEB)	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL
HindIII (NEB)	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A
SalI (NEB)	0.3µL	N/A	N/A	0.3µL	N/A	N/A	0.3µL	N/A	N/A
Incubation	37°C for 4 hours								
colony ID	7			8			9		
rxn ID	i	ii	iii	i	ii	iii	i	ii	iii
0.4µg DNA	4.6µL (86.6ng/µL)			1.5µL (271.6ng/µL)			2.8µL (142.7ng/µL)		
Total rxn volume	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL	20µL
10X rxn buffer	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL	2µL
dH2O	12.5µL	12.8µL	13.1µL	15.6µL	15.9µL	16.2µL	14.3µL	14.6µL	14.9µL
BamHI (NEB)	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL	0.3µL
HindIII (NEB)	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A	0.3µL	0.3µL	N/A
SalI (NEB)	0.3µL	N/A	N/A	0.3µL	N/A	N/A	0.3µL	N/A	N/A
Incubation	37°C for 4 hours								

have retained quite a large quantity of the unwanted pUC57 backbone as part of the final ligation. As a counter-measure to having such impurity in the reaction, pUC57_evo was digested with an extra enzyme XbaI (See Table 4.9) to remove one of the two sticky ends in the backbone required for ligating it to evo_insert fragments. However, there might have been enough intact unwanted backbone fragments remaining to dominate the reactions.

One of the difficulties in subcloning pUC57_evo into pMUTIN4_evo was that they both rely on a beta-lactamese-based antibiotics marker for screening *E. coli* transformants. An intermediate plasmid offering a different class of antibiotics

marker could make the selection for valid clones easier. Among others, pET28 befitted such a need as the plasmid has a kanamycin resistance cassette, and has BamHI as well as HindIII as part of its multiple cloning site.

4.2.1.4 Triple Restriction Enzyme Digestion

Table 4.9: triple restriction enzyme digest reaction, as a prestep to subcloning evo_insert into pMUTIN4

	Doner plasmid (main) (pUC57_evo)	Vector backbone (main) (pMUTIN4)
Rxn ID	i	ii
Total rxn volume	50µL	50µL
pUC57_evo (277.1ng/µL)	18µL (5µg)	N/A
pMUTIN4 (253.3ng/µL)	N/A	19.7µL (5µg)
10X rxn buffer	5µL	5µL
dH2O	25.5µL	24.3µL
BamHI (NEB)	0.5µL	0.5µL
HindIII (NEB)	0.5µL	0.5µL
XbaI (NEB)	0.5µL	N/A
Incubation	37°C for 4 hours	37°C for 4 hours
Inactivation	80°C for 20 mins	80°C for 20 mins

E. coli DH5α transformed with pUC57_evo and pMUTIN4 were grown overnight on a shaking incubator at 37.0 °C The overnight cultures were Miniprep'ed using the standard protocol described in Section 4.5.8 The triple digest was performed as summarised in Table 4.9. pUC57_evo was triple digested, in order to obtain a larger separation between evo_insert fragments and pUC57 backbone fragments. XbaI would break the pUC57 backbone into smaller pieces.

4.2.2 Subcloning the evo_insert construct from pUC57_evo into pET28

In the following procedure, the evo_insert from pUC57_evo was subcloned into pET28 to construct the intermediate plasmid named pET28_evo. The evo_insert fragments were sourced from the gel-purified double-restriction digest of pUC57_evo (rxn i) as described in Table 4.4. The pET28 backbone was prepared by double digesting it with BamHI and HindIII, followed by column purification. A ligation reaction was prepared based on a 1:3 molar ratio of the backbone (5.3 kb) and the insert (3.5 kb) as in Table 4.10. Competent *E. coli* DH5alpha was transformed (See the protocol in Section 4.5.7), by adding 5 µL of the above ligate in lieu of adding

Table 4.10: The reaction setup of the ligation of evo_insert and pET28 backbone fragments:

	evo_insert+pET28 backbone
Total rxn volume	10µL
198ng insert	5.4µL (36.5ng/µL)
100ng vector	3.2µL (49.3ng/µL)
10X rxn buffer	1µL
T4 ligase	1µL
dH2O	0.6µL
Incubation	25°C for 3 hours

circular plasmid DNA. The transformant *E. coli* culture was spread on kanamycin ($50\,\mu g\,mL^{-1}$) LB agar plates. After overnight growth, there was an abundance of colonies surviving. Overnight cultures in LB with $50\,\mu g\,mL^{-1}$ kanamycin were set up from nine randomly chosen colonies. The overnight cultures were minipreped as per Section 4.5.8, and their resulting plasmid DNAs were subject to diagnostic restriction enzyme digests and gel electrophoresis as follow (See Table 4.11 and Figure 4.6).

Table 4.11: Diagnostic restriction enzyme digest reactions, to determine the validity of putative pET28_evo clones:

rxn ID	1	2	3	4	5	6	7	8	9
Plasmid conc.(ng/µL)	79.5	114.4	85.8	79.5	126.6	200.6	101.4	98.5	114.5
0.2µg plasmid DNA (µL)	2.5	1.8	2.3	2.5	1.6	1.0	2.0	2.0	1.8
Total rxn volume (µL)	20	20	20	20	20	20	20	20	20
10X rxn buffer (µL)	2	2	2	2	2	2	2	2	2
dH2O (µL)	14.5	15.2	14.7	14.5	15.4	16.0	15.0	15.0	15.2
BamHI (µL)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
HindIII (µL)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Incubation	37°C for 4 hours								

The lanes labelled 2 and 9 in Figure 4.6 showed two bands approximately at about 5.3 kb and 3.5 kb. These two bands were highly likely to be respectively representative of pET28 and evo_insert, and indicated that the corresponding clones may habour pET28_evo.

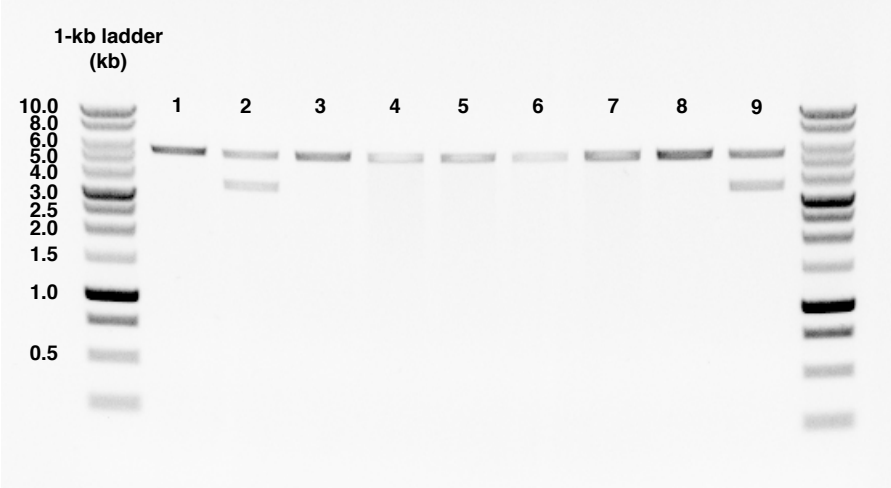


Figure 4.6: The diagnostic restriction digest gel electrophoresis of putative pET28_evo clones: lanes 2 and 9 showed positive results.

4.2.3 Subcloning the evo_insert construct from pET28_evo into pMUTIN4

In the following procedure, the evo_insert from pET28_evo was subsequently subcloned into pMUTIN4 to construct pMUTIN4_evo. Double restriction digests were performed to source the ligation fragments as shown in Table 4.12.

Table 4.12: Restriction enzyme digest reactions, to obtain the evo_insert and pMUTIN4 backbone fragments:

	Doner plasmid (pET28_evo)	Vector backbone (pMUTIN4)
Rxn ID	1	2
Plasmid conc.	114.5ng/µL	253.3ng/µL
Total rxn volume	60µL	20µL
DNA amount	52µL (6µg)	11.8µL (3µg)
10X rxn buffer	6µL	2µL
dH2O	0.0µL	5.2µL
BamHI (NEB)	1µL	0.5µL
HindIII (NEB)	1µL	0.5µL
Incubation	37°C for 4 hours	37°C for 4 hours

The restriction enzyme digest product of pET28_evo was gel purified as per Section 4.5.1, and the digest product of pMUTIN4 was column purified as per Section 4.5.9.

Nanodrop readings of the respective purified products were obtained as follow:

	evo_insert	pMUTIN4 backbone
ng/µL	20.8	26.7
260/280	2.04	1.99
260/230	0.10	1.79

A ligation reaction was prepared and was incubated at 25 °C for 3 hours, based on a 1:3 molar ratio of the backbone (8.6 kb) and the insert (3.5 kb) as follows:

	evo_insert+pMUTIN4 backbone
Total rxn volume	12µL
122ng insert	5.9µL (20.8ng/µL)
100ng vector	3.7µL (26.7ng/µL)
10X rxn buffer	1.2µL
T4 ligase	1.2µL
dH2O	0.0µL

Competent *E. coli* DH5alpha was transformed (See the protocol in Section 4.5.7), by adding 5 µL of the above ligate in lieu of adding circular plasmid DNA. The

	pMUTIN4_evo candidate
Rxn ID	1
Plasmid conc.	122.2ng/µL
Total rxn volume	20µL
DNA amount	1.7µL (0.2µg)
10X rxn buffer	2µL
dH2O	15.3µL
BamHI (NEB)	0.5µL
HindIII (NEB)	0.5µL
Incubation	37°C for 4 hours

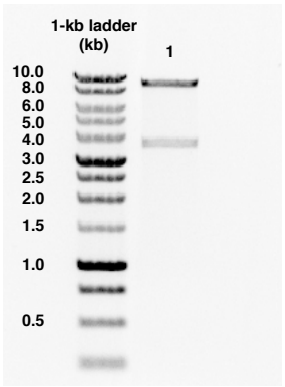


Figure 4.7: A diagnostic restriction digest gel electrophoresis of the putative pMUTIN4_evo clone: the bands indicate a successful pMUTIN4_evo clone.

transformant *E. coli* culture was spread on ampicillin (100 µg mL^{-1}) LB agar plates pre-treated with a 200 µL solution made of 20 µL 1 M IPTG, 140 µL MilliQ water, 40 µL 4 % Xgal (40 µg mL^{-1}). After overnight growth, there was a single colony surviving and passing the blue/white colony test. An overnight culture in LB with 100 µg mL^{-1} ampicillin was set up from that colony. The overnight culture was

minipreped as per Section 4.5.8, and the resulting plasmid DNA was subject to a diagnostic restriction enzyme digest followed by a diagnostic gel electrophoresis as shown in Figure 4.7.

Figure 4.7 at lane 1 showed two bands approximately at about 8.6 kb and 3.5 kb. These two bands were highly likely to be respectively representative of pMUTIN4 and evo_insert, and indicated that the corresponding clone may habour pMUTIN4_evo.

4.2.4 Transformation of pMUTIN4_evo into *B. subtilis* BSB1

The overnight culture of *E. coli* DH5alpha with pMUTIN4_evo was minipreped. Competent *B. subtilis* BSB1 cells were transformed by following the *B. subtilis* transformation protocol (see 4.5.6). The transformation culture was spread onto LB agar plates with $5\,\mu g\,mL^{-1}$ erythromycin and 1 mM IPTG. The plates were incubated at 37 °C overnight. One transformant plate had three colonies, another transformant plate two colonies, and the negative control plate had zero colonies. Five colonies were picked for colony PCR to verify their insert sizes. These colonies were labelled *i*, *ii*, *iii*, *iv*, and *v*, together with a wildtype negative control (*vi*) and a PCR negative control (*vii*). The primers were designed to amplify a 4.7 kbp long band across the evo_insert between the 5-prime end and the 3-prime end of target locus in the chromosome of *B. subtilis*. See Table B.4 for the information on PCR primers used.

Table 4.13: pMUTIN4_evo *B. subtilis* transformation: colony PCR

Colony PCR table	
evo_insert_fwd1 primer	1.25 µL
evo_insert_rev1 primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	13.75 µL
template DNA	3.0 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	98°C (30s)
Denaturation	98°C (10s)
Annealing	72°C (30s)
Extension	72°C (200s)
Cycle count	30
Final extension	72°C (2min)

A diagnostic gel electrophoresis was performed to verify the result of the colony PCRs. A 0.7 % agarose gel was made by following the standard protocol (Section

4.5.3). The gel was placed into an electrophoresis buffer tray, loaded with reaction samples (with 6X loading dye, Promega), and applied with 100 V for 1 h. The gel was UV imaged on a gel dock (BioRad) afterwards. Figure 4.8 shows that the

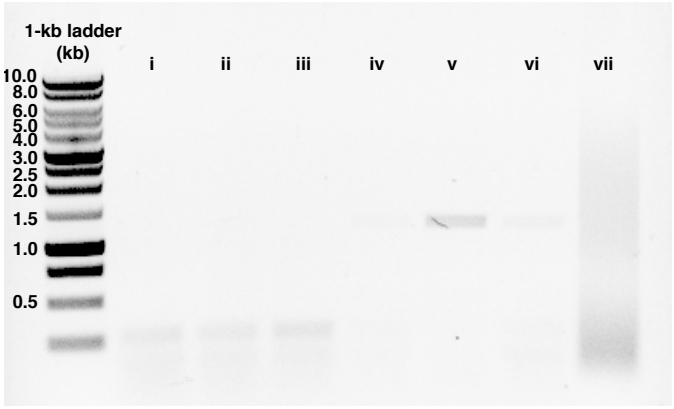


Figure 4.8: A diagnostic gel electrophoresis image for the colony PCRs of putative *B. subtilis* pMUTIN4_evo clones.

sample on lane *v* had a band appear little below the 1500 bp ladder, quite distant from the expected size of successfully amplified DNA fragments (i.e. 4.7 kbp). Following the unsuccessful result, numerable attempts to transform *B. subtilis* using pMUTIN4_evo had failed. Some of the likely explanations to this failure in transforming *B. subtilis* with pMUTIN4_evo are as follows. Firstly, the size of the plasmid is over 12 kb, and this could be too large for the transformation method using the natural competency of *B. subtilis*. Transformation using electroporation (results not shown) was tried with no avail either. Another potential culprit might be to do with the 500 bp long sequence homologous to the chromosomal insertion locus in that the homology might be too short with respect to the 12 kb vector to be inserted as a whole using a single cross-over. The third possibility is that the insertion of pMUTIN4_evo to the target locus might be fatal to the cell. After all, I had to move on without finding any definitive answers to these suggested possibilities due to time constraints in the project.

4.3 Constructing pSG1729_EVOt2 using Gibson assembly

Using pUC57_evo as the donor plasmid, two parts of the evo_insert sequence from the donor plasmid were PCR amplified and subcloned into the pSG1729 backbone via Gibson assembly. The two-part PCR amplifications of the evo_insert sequence were aimed at excluding the sfGFP element in the evo_insert sequence. This recipient plasmid with the two inserts assembled into the backbone's locus under the regulatory control of the PxylR promoter was named pSG1729_EVOt2 or pEVOt2 in short (Figure 4.9).

Table 4.14: Gibson primers used for the assembly of pSG1729_EVOt2

Primer ID	Oligo sequence
pSG1729 FWD	ACGAAAAGGAGGAATTCAAAATGAGTAAAGGAGAAGAACTTTTC
pSG1729 REV	CCAACAACCACTCGCCGACGAGATGCATTTTATGTCATATTGTAAG
assembled_seq FWD 1	ATATGACATAAAATGCATCTCGTCGGCGAGTGGTTGTTGG
assembled_seq REV 1	CCATTACAGGCCGGCTTTTTGAATGCTTATTAAACAGCGTCTGCT
assembled_seq FWD 2	ACGCTGTTTAATAAGCATTCAAAAAGCCGGCCTGTAATGG
assembled_seq REV 2	AGTTCTTCTCCTTTACTCATTTTGAATTCCTCCTTTTCGTC

Primer ID	T_m^{whole}(°C)	T_m^{anneal}(°C)	T_m^{diff}(°C)	Binding(bp)	Overhang(bp)	$\Delta G_{hairpin}$(kcal)
pSG1729 FWD	71.0	58.9	0.49	24	20	-0.9
pSG1729 REV	75.3	58.4	0.49	26	20	-5.3
assembled_seq FWD 1	74.9	68.7	4.30	20	20	-4.4
assembled_seq REV 1	76.9	64.4	4.30	25	20	-2.8
assembled_seq FWD 2	74.6	63.7	4.92	20	20	-1.7
assembled_seq REV 2	70.8	58.8	4.92	21	20	0

The pSG1729 and pUC57_evo plasmids were diluted to the concentrations of $1\,\mathrm{ng\,\mu L^{-1}}$ PCRs were run as specified in Table 4.15 and Table 4.14.

PCR samples were column purified using QIAGEN PCR Purification Kit. Nanodrop readings were obtained for PCR results (see Table 4.16) DpnI digestion was performed on PCR samples (see Table 4.17) A 3-fragment Gibson assembly was done (see 4.5.2) as per Table 4.18 Colony PCR was performed to identify transformants with successful plasmid assembly. Five colonies were picked (samples labelled i through v, in addition to a negative sample (vi)) using sterile pipette tips and each picked colony was suspended in 20 µL of cold MilliQ H2O. 1 µL from each suspended sample was used as template DNA in PCR set up as in the following Table 4.19. See Table B.4 for the information on PCR primers used.

Figure 4.9: The pSG1729_EVOt2 plasmid map.

A diagnostic gel electrophoresis was performed to verify the result of the colony PCRs. A 0.7 % agarose gel was made by following the standard protocol (Section 4.5.3). The gel was placed into an electrophoresis buffer tray, loaded with reaction samples (with 6X loading dye), and applied with 100 V for 1 h. The gel was UV imaged on a gel dock afterwards. Figure 4.10 shows that samples *i* and *ii* had bands appear in between the 1500 bp and 2000 bp ladders very close to the 2000 bp ladder. The expected size of successfully amplified DNA fragment, as a result of the colony PCR, is 1918 bp, and these two bands seem to be in correct sizes.

Table 4.15: pSG1729_EVOt2 assembly PCR details:

PCR #1: backbone	
pSG1729 FWD primer	1.25 µL
pSG1729 REV primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	16.25 µL
pSG1729 plasmid (1ng/µL)	0.5 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	98°C (30s)
Denaturation	98°C (10s)
Annealing	61°C (30s)
Extension	72°C (154s)
Cycle count	30

PCR #2: insert 1	
assembled_seq FWD 1 primer	1.25 µL
assembled_seq REV 1 primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	16.25 µL
pUC57_evo plasmid (1ng/µL)	0.5 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	98°C (30s)
Denaturation	98°C (10s)
Annealing	61°C (30s)
Extension	72°C (20s)
Cycle count	30

PCR #3: insert 2	
assembled_seq FWD 2 primer	1.25 µL
assembled_seq REV 2 primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	16.25 µL
pUC57_evo plasmid (1ng/µL)	0.5 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	98°C (30s)
Denaturation	98°C (10s)
Annealing	61°C (30s)
Extension	72°C (20s)
Cycle count	30

Table 4.16: pSG1729_EVOt2 assembly: PCR Nanodrop results

Nanodrop readings			
	PCR #1	PCR #2	PCR #3
ng/µL	40.6	66.1	41.2
260/280	1.88	1.88	1.83
260/230	1.52	2.26	1.81

Table 4.17: pSG1729_EVOt2 assembly: DpnI digestion

Restriction Enzyme reaction table			
	PCR #1	PCR #2	PCR #3
DNA sample	8 µL (325ng)	6.9 µL (455ng)	6.3 µL (257.7ng)
RE buffer (10X)	1 µL	1 µL	1 µL
MilliQ H2O	0 µL	1.1 µL	1.7 µL
DpnI (NEB)	1 µL	1 µL	1 µL
Total rxn volume	10 µL	10 µL	10 µL
Gibson: sample molar value	1X molar (100ng)	3X molar (45.5ng)	3X molar (12.6ng)
Gibson: sample volume	3.1 µL	1 µL	0.5 µL

Table 4.18: pSG1729_EVOt2 assembly: Gibson reaction

Gibson reaction table	
PCR sample #1	3.1 µL (100ng: 1X molar)
PCR sample #2	1.0 µL (45.5ng: 3X molar)
PCR sample #3	0.5 µL (12.6ng: 3X molar)
MilliQ H2O	5.4 µL
Gibson master mix (2X)	10 µL
Total rxn volume	20 µL

Table 4.19: pSG1729_EVOt2 assembly: colony PCR

Colony PCR table	
pEVOt2_colpcr FWD primer	1.25 µL
pEVOt2_colpcr REV primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	15.75 µL
template DNA	1.0 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	95°C (6min)
Denaturation	98°C (10s)
Annealing	59°C (30s)
Extension	72°C (58s)
Cycle count	30
Final extension	72°C (2min)

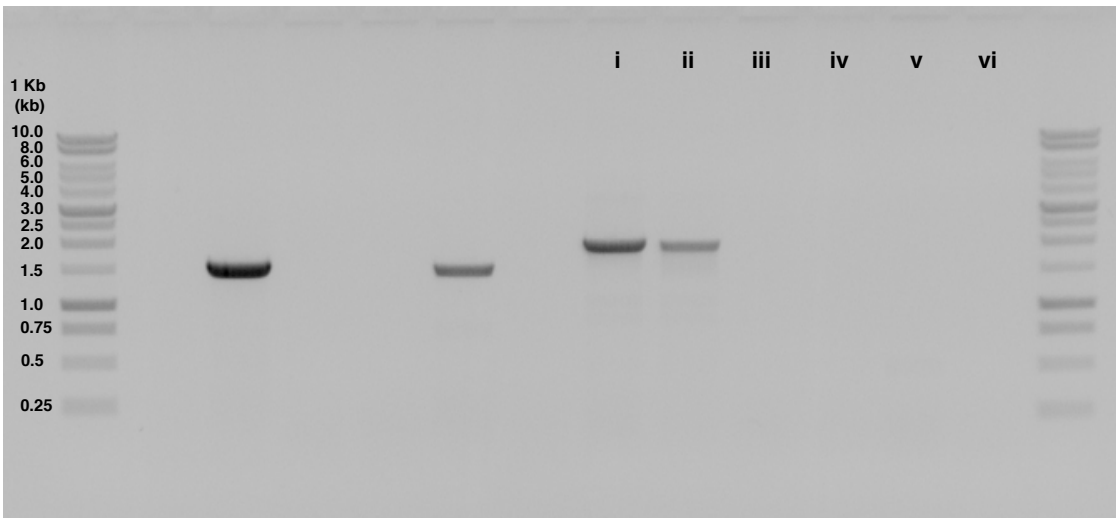


Figure 4.10: A diagnostic gel electrophoresis image for the colony PCRs from Table 4.19.

4.3.1 Resuspension of lyophilised DNA (Revised from the Genscript protocol)

The vial containing lyophilised DNA was centrifuged at 6000x g for 1 minute. 20 µL of deionised water (MilliQ) was added to the vial. The solution was heated at 50 °C for 15 minutes.

4.3.2 Amplification of plasmid DNA using *E. coli* transformation

Plasmid vectors (e.g. pMUTIN4 or pUC57_evo) were used in transforming chemically competent *E. coli* culture (DH5alph or MC1061) via the heat-shock protocol (see 4.5.7). Ampicillin (100 µg µL^{-1}) LB agar plates were used for selecting *E. coli* transformants of pMUTIN4 and pUC57_evo. Approximately about 200 ng of circular plasmid DNA was used in each transformation reaction. 3 µL of pMUTIN4 in 83 ng µL^{-1} was added per 100 µL of competent cells. 1 µL of pUC57_evo in 200 ng µL^{-1} was added per 100 µL of competent cells. *E. coli* transformants were cultured overnight and their plasmids were extracted. 10 mL of LB with 100 µg mL^{-1} of ampicillin was innoculated with a single colony from the *E. coli* transformant plates (see 4.5.7). Three colonies from the plate of pMUTIN4 transformants and three colonies from the plate of pUC57_evo transformants were picked, and used for respectively innoculating 10 mL ampicillin LB in 50 mL polypropylene conical centrifuge tubes. The innoculated culture tubes were incubated for 16 hrs at 37 °C on a shaker. Plasmid DNA from each overnight culture was extracted using Miniprep (see 4.5.8).

4.3.3 Transformation of pSG1729 into *B. subtilis* BSB1

B. subtilis BSB1 was transformed using the same protocol as shown in Section 4.4 with the exception of using pSG1729 instead of pSG1729_EVOt2.

4.4 Transformation of pSG1729_EVOt2 into *B. subtilis* BSB1

The overnight culture of *E. coli* MC1061 with pSG1729_EVOt2 was minipreped. Competent *B. subtilis* BSB1 cells were transformed by following the *B. subtilis* transformation protocol (see 4.5.6). The transformation culture was spread onto LB Spec agar plates. The plates were incubated at 37 °C overnight. Colonies were picked and replica plated on a starch plate for checking the loss of amylase activity on correct transformants. Colonies that have lost amylase activity were picked for colony PCR to verify their insert sizes. These colonies were labelled *i*, *ii*, *iii*, *iv*, *v* and *vi*, together with a negative control (*vii*). The primers were designed to amplify a 1464 bp long band carrying a portion of the gene for spectinomycin resistance (from pSG1729_EVOt2) and a portion of the 3-prime end of the *amyE* gene (from the chromosome of *B. subtilis*). See Table B.4 for the information on PCR primers used.

Table 4.20: pSG1729_EVOt2 *B. subtilis* transformation: colony PCR

Colony PCR table	
colpcr_pSG_spec_spec_cds_fwd primer	1.25 µL
colpcr_amyE_3p_end_rev primer	1.25 µL
5X PCR buffer	5 µL
dNTP	0.5 µL
Q5	0.25 µL
MilliQ H2O	15.75 µL
template DNA	1.0 µL
Total rxn volume	25 µL

PCR Thermocycler program	
Initialisation	98°C (30s)
Denaturation	98°C (10s)
Annealing	58°C (30s)
Extension	72°C (44s)
Cycle count	30
Final extension	72°C (2min)

A diagnostic gel electrophoresis was performed to verify the result of the colony PCRs. A 0.7 % agarose gel was made by following the standard protocol (Section 4.5.3). The gel was placed into an electrophoresis buffer tray, loaded with reaction samples (with 6X loading dye), and applied with 100 V for 1 h. The gel was UV imaged on a gel dock afterwards. Figure 4.11 shows that samples *iii*, *iv* and *vi* had bands appear little below the 1500 bp ladder, very close to the expected size of successfully amplified DNA fragment (i.e. 1464 bp). The mutant colony used for the sample *iv* was named *B. subtilis* EVOt2.

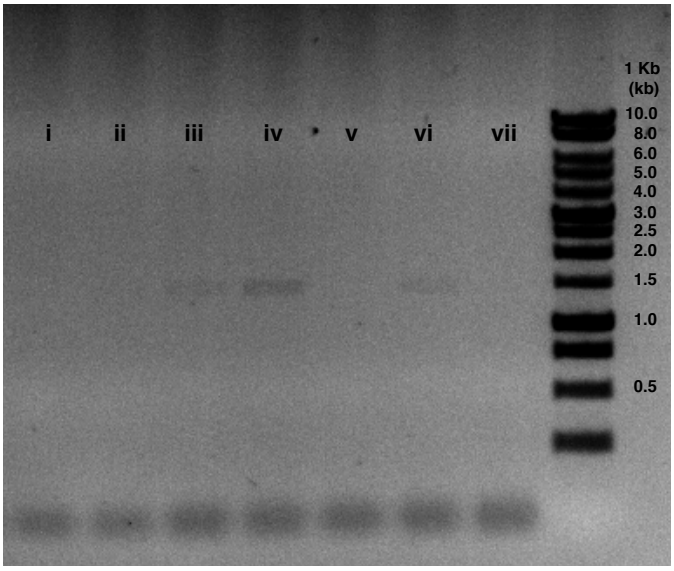


Figure 4.11: A diagnostic gel electrophoresis image for the colony PCRs from Table 4.20.

4.4.1 Sequence verification of the *B. subtilis* EVOt2 mutant

The chromosomal DNA of *B. subtilis* EVOt2 was prepared using DNeasy Blood & Tissue kit (QIAGEN), and was sequenced using Illumina MiSEQ at the sequencing facility in CBCB. The raw sequence reads were stored in FASTQ format, and were assembled using the sequence assembly pipeline shown in Section 5.3.2.1. The EVOt2 sequence was blasted against the nucleotide sequences of contigs generated from the *de novo* assembly in the pipeline, to find a contig that carried a chromosomally inserted copy of the EVOt2 sequence. The chromosomal copy of EVOt2 was compared against the original EVOt2 sequence for any anomalies. As shown in Figure 4.12, the chromosomal EVOt2 sequence and its *amyE* insertion locus in the *B. subtilis* BSB1 genome were examined to be 100 % accurate.

4.4.2 Flow cytometry of the *B. subtilis* EVOt2 mutant

B. subtilis EVOt2 clone was cultured overnight in LB with $50\,\mu g\,mL^{-1}$ spectinomycin. Four different growth media as shown in Table 4.21 were innoculated from the overnight culture and were incubated at 37 °C on a shaker. At 150 min into the

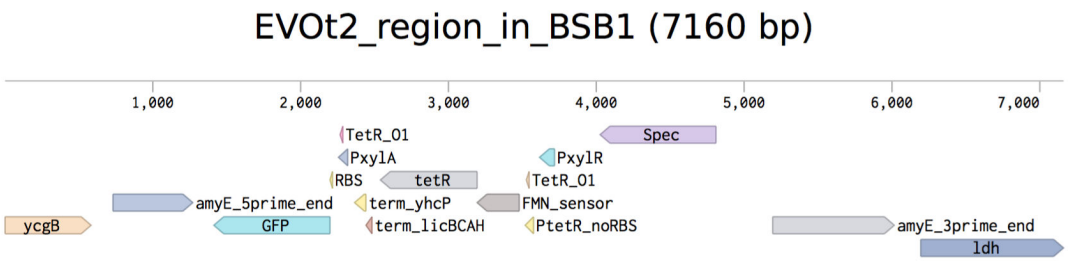


Figure 4.12: An annotated sequence showing the region in and around the chromosomally inserted copy of EVOt2.

incubation, 10 µL of culture from each growth media was harvested and diluted in 1 mL of chilled PBS. The PBS diluted samples were immediately placed in ice. Flow cytometry was performed on each of the four samples. Flow cytometry was repeatedly performed on samples harvested at 60 min intervals till 390 min using the same PBS dilution ratio. The resulting raw cytometry data readings were processed by the Python script[1] to automatically generate Figure 5.8.

Table 4.21: *B. subtilis* EVOt2 growth conditions and expected outcomes.

Media ID	Composition	Expected fluorecence level
m0	LB with 1% xylose	lowest
m1	LB with riboflavin and 1% xylose	higher than m0 and lower than m3
m2	LB	higher than m0 and lower than m3
m3	LB with riboflavin	high

[1] Appendix A.3 shows an instruction to invoke a Docker container to use the Python script.

4.5 Standard lab protocols

4.5.1 DNA gel extraction using QIAGEN kits (modified from the original instructions)

An empty microcentrifuge tube was used to tare a scale. A gel slice was placed into the microcentrifuge tube and weighed. Given the rough estimate of 100 µg = 100 µL, 3X gel volume equivalent of QG buffer was added to the tube. Without any vortexing, the tube was incubated at 50 °C for 5 min to dissolve the gel. 1X gel volume equivalent amount of isopropanol was added to the tube and gently mix by flicking with a finger. The disolved solution was added to QIAGEN spin column(s) to be centrifuged at 13 000 RPM for 1 min. The flow-through was discarded. Each spin column was topped up with fresh 500 µL QG buffer, and centrifuged to discard the flow-through. 750 µL Buffer PE was added to each spin column, to be incubated at room temperature for 4 min. All columns were centrifuged and the flow-throughs discarded. The columns were centrifuged for one more run of 13 000 RPM to rid of residual buffer. Each column was placed in a sterile 1.5 µL microcentrifuge tube. 25 µL EB, warmed to 50 °C, was added to the center of each column's membrane. The columns were incubated at room temperature for 5 min, centrifuged at 13 000 RPM for 1 min. 25 µL of fresh EB was added again before the tubes get incubated for 5 min in room temperature, and centrifuged at 13 000 RPM for 1 min. The 50 µL of elute in each collection tube was run through the same column again, to be incubated and centrifuged under the same conditions as before. The final step was repeated one more time to increase the yield.

4.5.2 Gibson Assembly

Assembly reactions were set up as explained in Table 4.22[1]. Reaction samples were incubated at 50 °C for 1 hour using a thermocycler with the lid temperature set at 99 °C. Chemically competent *E. coli* cells were transformed using the heat shock protocol (see 4.5.7).

[1] Based on NEB Gibson assembly protocol

Table 4.22: Gibson Assembly reaction setup chart.

	2–3 fragment assembly	4–6 fragment assembly
Total amount of fragments	X µL (0.02 – 0.5 pmols)	X µL (0.2 – 1 pmols)
Gibson Assembly master mix (2X)	10 µL	10 µL
MilliQ H2O	10 - X µL	10 - X µL
Total Volume	20 µL	20 µL

4.5.3 Making 0.7 % agarose gel for electrophoresis

For a 40 mL volume, 0.28 g of agarose was measured out, and poured into a 150 mL flask. 40 mL of 1X TAE was added into the flask and microwaved up to about 1.5 min till the solution started to boil. The flask was gently swirled until the molten agarose solution was clear. The flask was rested on a bench, till it cooled to a level, warm to touch. 1.0 µL of stock EtBr solution (10 mg mL^{-1}) was added to the molten agarose and was gently swirled to mix. A well comb of chosen size was placed onto an appropriately sealed casting tray. The agarose solution was poured into the tray. A sterile pipette tip or a toothpick was used to remove any impurities and bubbles. The gel was left in room temperature for about 30 min till it firmly solidified.

4.5.4 Freezing *B. subtilis* competent cells

The starvation culture was centrifuged down and was concentrated by 10X. The supernatant was used for resuspending the spun-down cells. Glycerol was added in the culture to the 15 % (v/v) concentration (e.g. 300 µL of 50 % glycerol in 700 µL culture). 100 µL aliquots of the glycerol culture werepipetted to Eppendorf tubes. The tubes were snap-frozen in liquid nitrogen, and stored in −80 °C.

4.5.5 *B. subtilis* transformation using frozen competent cells

Frozen competent cells would result in a lower transformation efficiency compared to freshly prepared competent cells. Competent *B. subtilis* cells in 100 µL aliquot was thawed at room temperature. 500 ng to 2 µg of plasmid DNA was added to the cells. The tube was incubated at 37 °C for 1 hr. 200 µL of the incubated culture was plated on a selection plate, and incubated overnight at 37 °C.

4.5.6 *B. subtilis* transformation using natural competency

This protocol worked for 168 and BSB1 with high efficiency. A single colony was scraped from a plate of *B. subtilis* wildtype strain, and was used to inoculate 5 mL of MM competence media (see B.5.2) in a 15 mL Falcon tube. The culture was incubated at 37 °C, 180 RPM overnight. 0.3 mL of the overnight culture was transferred into fresh 5 mL MM competence media in a 50 mL Falcon tube, and incubated for 3 hrs at 37 °C, 180 RPM. 5 mL of pre-warmed (37 °C) Starvation media (see B.5.3) was added, and incubated for 2 more hours at 37 °C, 180 RPM[1]. 0.4 mL of the culture was transferred to a 1.5 mL micro-centrifuge tube, and 1 µg DNA was added. Nanodrop was used to measure the DNA concentration of the plasmid Miniprep. Negative controls, such as a separate 0.4 mL culture without adding plasmid DNA were included. The tubes were incubated at 37 °C, 180 RPM for 1 hr. For each incubation culture, 200 µL was plated to a selection plate, and was incubated overnight. Plates with an appropriate antibiotics were used as an initial screening. Replica plating on starch plates was performed to test for transformants with the successful integration into the *amyE* locus.

4.5.7 *E. coli* transformation using heat shock

A heat block was preheated to 42 °C. LB agar plates containing appropriate antibiotics (selection plates) were rested in room temperature, so the plates were not too cold before plating transformed cells. 1.5 mL microcentrifuge tubes containing 100 µL aliquots of chemically competent *E. coli* cells (e.g. DH5alpha or MC1061) that had been kept frozen at −80 °C were taken out immediately before use. The competent cell tubes were placed on ice and were incubated for 10 minutes to slowly thaw. Approximately about 200 ng of circular plasmid DNA was added into each tube. The tubes were incubated on ice for further 20 minutes. The tubes were heated at 42 °C for 50 seconds using the preheated heat block. The tubes were placed back onto ice, and were incubated for 2 minutes, in order to reduce the heat shock damage. 1 mL of LB was added to each tube. Then the

[1] Competent cells were frozen after the starvation incubation. See the protocol for freezing *B. subtilis* competent cells, 4.5.4. See 4.5.5 for the transformation protocol using frozen competent cells.

tubes were incubated in 37 °C on a shaker for 45 minutes. 100 µL of the resulting culture from each tube was pipetted onto a selection plate, and spreaded using sterile beads. 400 µL of the remaining culture from each tube was pipetted onto another selection plate and spreaded using sterile glass beads. This second high-cell-count plating was used as a contingency measure in case of low transformation efficiency. The plates were incubated at 37 °C overnight.

4.5.8 QIAprep Spin (Qiagen) Miniprep

The overnight cultures were pelleted by centrifugation at 6800x g for 3 minutes at room temperature. After discarding the supernatant, the pelleted cell mass in each tube was resuspended in 500 µL resuspension buffer (Buffer P1), and vortexed. The composition of the resuspension buffer was 50 mM Tris-HCl at pH 8.0, 10 mM EDTA, and RNAse A in 100 µg mL^{-1} concentration. Due to the presence of RNAse in the buffer, the buffer had been kept in 4 °C, and was chilled on ice prior to use. Each 500 µL cell resuspension was transfered to two 1.5 mL microcentrifuge tubes in aliquots of 250 µL. To each microcentrifuge tube, 250 µL lysis buffer (Buffer P2), with a composition of 200 mM NaOH and 1 % SDS, was added. All tubes were inverted back and forth until the solutions were mixed well and turned clear. The tubes were incubated at room temperature for 3 minutes. Immediately following the previous step, 350 µL neutralisation buffer (Buffer N3), with a composition of 4.2 M GuHCl and 0.9 M KAc at pH 4.8, was added to each tube, and thoroughly mixed by inverting 5 times. The solution formed white precipitates out of SDS, lipid and protein, and exhibited gooey consistency mostly due to the presence of chromosomal DNA. The tubes were spun down for 10 minutes at 13000 rpm on a benchtop centrifuge. The supernatant was transfered to respectively labelled QIAprep spin columns. The spin columns were centrifuged for 60 seconds, and the flow-through was discarded. 500 µL wash/binding buffer (Buffer PB), with a composition of 5.0 M GuHCl and 30 % isopropanol, was added to each spin column. The columns were centrifuged for 60 seconds at 13000 rpm to discard the flow-through. 750 µL wash buffer (Buffer PE), with a composition of 10 mM Tris (pH 7.5), and 80 % EtOH, was added to each spin column. The columns were centrifuged for 60 seconds at 13000 rpm to discard the flow-through. The columns

were centrifuged for 1 additional minute to remove residual wash buffer. Each column was moved to a sterile 1.5 mL microcentrifuge tube, then was added 50 µL elution buffer (Buffer EB), with a composition of 10 mM Tris-HCl, pH 8.5. After incubating the columns in room temperature for 5 minutes, they were centrifuged for 1 minute at 13000 rpm to elute plasmid DNA. Then aliquots of the plasmid DNAs were stored in 4 °C for a short-term storage, and in −20 °C for a long-term storage.

4.5.9 Purification of restriction digested DNA fragments using QIAquick spin columns

250 µL Buffer PB was added to each of 50 µL restriction digested solution A spin column was mounted per solution mix into a collection tube, and the mix was pipetted into the spin column. Spin columns were centrifuge at 13000 RPM for 1 minute, and the flow through was discarded. 750 µL Buffer PE was added to each tube. Tubes were incubated at room temperature for 2 minutes, and were centrifuged at 13000 RPM for 1 minute. The flow through was discarded afterwards. One more centrifugation was performed at 13000 RPM for 1 minute to rid fo residual wash buffer. Each spin column was moved to a sterile 1.5 mL centrifuge tube. 30 µL elution buffer was added to the centre of each spin column's filtering membrane. Spin columns were incubated at room temperature for 5 minutes, and centrifuged at 13000 RPM for 1 minute to collect the elute.

Chapter 5

The *in vivo* evolutionary design process

5.1 Introduction

Rationality is one of the underlying assumptions used in game theory to explain how people make decisions [18]. Endowed with the propensity to reason, human mind is inclined to make rational decisions. Our mind likes to secure the sense of being in control so much so that all decisions taken need to make rational sense, if at all possible. The pitfall in such a perfectionist's approach to decision making is that it has heavy reliance on *a priori* knowledge. To make rational sense of something requires having the knowledge to determine so [155]. For problem domains that are novel and complex, the extent to which rational decisions can hold as valid is severely limited. In the biomedicine field, for instance, drug design based on trial and error is more likely to succeed than rational design approaches [181]. The issue with rational design approaches here arises from the challenging nature of dealing with highly complex biological systems of which we currently do not have complete understanding. As such, rational decision making is only capable of playing limited roles in engineering complex biological systems.

That a search process taking random walks in the solution space can accomplish design is a notion wholly adopted by natural evolution. Nature employs DNA as a medium to encode the solution space, and harnesses random mutagenesis as a

vehicle to explore the space in search for sample solutions. Environmental niches would provide selection pressure to reinforce small subsets of solutions that happen to meet the needs of interest. The net result of which processes is so powerful that there exist a plethora of genetic designs encoding proteins with incredibly complex and diverse functions in nature. Charles Darwin, during his journey around the world onboard HMS Beagle, marvelled at the diversity of species, and their remarkably adapted functional features suitable for specific environmental needs [49].

In the design of synthetic biological systems, the use of random solutions offers a viable alternative to rational approaches [80]. The hurdle in going random, however, is finding ways to cope with the problem of search space explosion. The volume of a given search space would exponentially expand and quickly become unwieldy as more and more genes are combinatorially considered as part of a design. Richard Bellman, a mathematician, summed up this search space expansion problem so elegantly by coining the term, "the curse of dimensionality," to explain the difficulty of dealing with high-dimensional solution spaces [16].

To a degree, the curse has negatively affected numerical sciences in that use of the term is often associated with the justification for going against the idea of dealing with high dimensional spaces. In synthetic biology, with respect to harnessing random solutions in design, the curse would need to be a subject not to be avoided but to be embraced. In order to make significant progresses in the field, dealing with high-dimensional spaces is inevitable. To this end, there are interesting research questions to be asked. Would it be possible to contain the curse of dimensionality as a result of harnessing random design strategies in synthetic biology? If so, how can this containment be articulated? What would then be the limit to the complexity of genetic systems design, practically achievable by employing random design?

5.1.1 Establishing the scope of design in synthetic biology

Often, the term "design" is perceived only by the final outcome or artefact of design [59, 60]. With respect to the current discussion of design, the focus is not on the design artefact per se but on the art of engineering design or the processes

that lead into producing design artefacts. Crucial in the art of design engineering is design documentation with which to capture information sufficient enough to specify how the functional properties of final design artefacts can be achieved, as explained in Chapter 1.

Design artefacts of biological systems in the context of synthetic biology are functional properties of the systems being designed that can carry out meaningful cellular processes. Therefore, biological systems design should ideally convey unequivocal documentations on how to achieve desired cellular functions of the systems being designed. In agreement with the central dogma, generally regarded as true [45], is the idea that cellular functions of biological systems are emergent properties of relevant genetic elements. This proposition makes genetic sequences well positioned to be the documenting medium for design implementation in synthetic biology. However, design in synthetic biology is not simply about documenting genetic sequences. It is more importantly about how to evaluate the fitness level of genetic sequences in fulfilling desired cellular functionalities. Design in synthetic biology should, therefore, be inclusive of the means with which a particular genetic sequence can be tested for the possession of intended functional properties.

5.1.2 Solution generation via random mutagenesis

Random mutagenesis of cellular DNA sequences can be induced from various sources including chemicals [84, 166, 188], transposons [109, 157, 185], and stress factors such as aging [20], starvation [26], UV irradiation [67], or oxidation [20]. Mutations can also occur spontaneously due to errors in the DNA replication machinery and the mismatch repair mechanism [56, 169, 179]. Aspects concerning safety and automation were considered in this study as part of adopting a mutagenesis strategy for use in the iterative design cycle of the dual-evolutionary approach. Chemically induced mutagenesis strategies were excluded due to safety concerns, and strategies involving transposons and stress factors were excluded due to difficulties in automating their induction procedures. Adopting the spontaneous mutagenesis strategy befitted the criteria as the strategy offered to be both safe and amenable to automation.

Evaluating the spontaneous mutation of microbes provided some useful insight

into the feasibility of using this mutagenesis strategy in directed evolution. Spontaneous mutation in *E. coli* for instance, is known to occur at a rate of about 1×10^{-3} per genome and generation [179]. The probability of a single point mutation was assumed to be roughly equal to the mutation rate. This is a valid assumption given the small value for the rate. This means that a single point mutation is expected to occur every 1000 cells being replicated. Provided that the bacterial genome has 4.22×10^6 base pairs, the mutation rate per base pair per replication is $4.22^{-1} \times 10^{-9}$. This number is largely based on the error rate of the DNA replication machinery when cells are dividing. In order to find a cell that ends up with a specific point mutation, it is expected that 4.22×10^9 cells need screening[1].

An *E. coli* culture at OD_{600} of 1.0 would roughly amount to a cell density of about 8×10^8 cells/mL [153]. The $\Delta mutSL$ mutation in *Listeria monocytogenes* was shown to result in 100- to 1000-fold increase in the spontaneous mutation rate [119]. Assuming the same OD_{600} readings to cell density conversion, and a 1000-fold increase in mutagenesis via $\Delta mutSL$ in *B. subtilis*, 4.22×10^6 cells, or roughly about 5.28 uL of *B. subtilis* cell culture at OD_{600} of 1.0 would contain a cell with a specific point mutation. Extrapolating this thought experiment, 22.3 L culture at OD_{600} of 1.0 would amount to approximately about 1.78×10^{13} cells, a cell population large enough to expect a cell with a specific double SNPs.

In nature, the hypermutation state resulting from the lack of *mut* genes responsible for DNA mismatch repair was shown to have significant consequences in facilitating bacterial adaptation, such as enhancing virulence and surviving in hostile environments [13, 132, 189]. The possibility of successful bacterial adaptation cases demonstrated in nature corroborates the idea that viable solutions to complex problems can be generated from random solution pools driven by spontaneous mutagenesis.

[1]Let a specific point mutation be an event that occurs with probability $p = 4.22^{-1} \times 10^{-9}$, then the mathematically expected number of cells (E) to be screened before seeing the first occurrence of a cell with the specific point mutation is $E = 1/p$

5.1.3 Feasibility of finding solutions in randomness

There are numerable factors, considerable, affecting the feasibility of using directed evolution and random mutagenesis in genetic systems design. These include design complexity, microbial sample size, selection throughput and mutation rate. Design complexity would be positively correlated to the volume of the searchable solution space - the more complex a design is, the further the solution space exponentially expands. As the solution space begins to explodingly expand, a proportionately large population of cells is necessary to support enough random solutions to harbour a fit solution. Having a large population of cells subject to random mutagenesis, hence a large pool of solution candidates, would be beneficial to counteracting the curse of dimensionality associated to complex designs. However, there is a downside to having a large cell count. The larger the cell count is, the more critical the selection throughput becomes, as the process of screening and sorting cells based on measurements has costly overhead [35].

Fluorescence-activated cell sorting (FACS) devices, touted to offer an ultra-high throughput screening in today's standard, can perform at a rate of approximately about 10^7 particles per hour [193]. While it seems as though a large number, such a throughput is still far from being sufficient, at least for the purpose of employing directed evolution in complex designs. For example, it would require screening through a minimum of 4.22×10^6 cells in order to ensure that a specific point mutation can be found in *B. subtilis*, assuming a mutation rate of $4.22^{-1} \times 10^{-6}$ per base pair per replication (See Section 5.1.2). This means it would need about half an hour[1] of continuous operation of a FACS device, before a desired single point mutation can be isolated, provided that the FACS device is error-free. In reality, FACS-based screening and sorting would take longer.

The search time worsens exponentially as we enter the territory of multiple point mutations. In screening for a specific double point mutations, the minimum search space is as large as 1.78×10^{13} cells. This would amount to about 1.78 million hours of continuous operation of a single FACS device to ensure a desired mutant to be found. Using multiple FACS machines for parallel search operations would only be able to diminish the numbers in a linear fashion. While limited,

[1] 4.22×10^6 cells divided by $10^7/60$ particles per minute = 25.32 minutes

increased mutation rates can take some load off of the screening and sorting process by reducing the total number of cells per point mutation. Increased mutation rates would amount to higher *solution densities* per unit cell population, hence would reduce the total number of cells to be screened per desired solution. Nonetheless, the improvement offered by increased mutation rates would be limited, certainly not meaningful enough to warrant the adoption of mechanical sorting in directed evolution.

The process of random mutagenesis is indifferent to which way changes are made in the genome. It is only that the selection process is intrinsically biased by the constant pressure of the need to sustain life, and ends up favouring one mutation to another. Directed evolution can harness random mutagenesis and extrinsic evolutionary pressure to drive random changes towards meeting certain fitness criteria [9, 193], unessential to sustaining life. The biosynthesis of secondary metabolites accounting for many commercially significant high-value chemicals [11, 92] are good examples relevant to the application of such *extrinsic fitness criteria*. Whether intrinsic or extrinsic, evolutionary pressure is a type of a selection process. In case of the intrinsic pressure, the selection process is the survival of the fittest, the end result of which is the enrichment of traits deemed advantageous or necessary for sustaining life in the given environmental conditions. The selection process concerning extrinsic pressure, on the other hand, can enrich groups with traits, normally considered unessential for the purpose of sustaining life.

How can we enrich groups with unessential traits then? Cherry picking cells after screening for desirable traits is one obvious way, albeit such a selection process is limited by the throughput of the screening and sorting procedures as well as the feasibility of trait quantification. Alternatively, extrinsic selection processes can take place by being coupled to the intrinsic selection process, so that the traits of interest, that are otherwise unessential, become conditions for survival. Making survival dependent on certain unessential traits can be challenging on its own right. Nevertheless, the coupling approach offers an attractive means for effectively exploring a large design space via random mutagenesis, since the approach does not suffer as much from the bottleneck imposed by screening and sorting throughput as the cherry picking approach.

What does it mean by being throughput limited in terms of the selection pro-

cesses in directed evolution? Given a finite search time, it amounts to having a limited coverage in the solution space, therefore having a reduced likelihood of running into a solution. Adding more equipment, let alone the problem of doing so not being economically viable an option, would not help much in an attempt to increase the throughput, as the throughput gain in doing so only goes up linearly. Simply put, such a linear gain cannot deal with solution space explosion.

What if each cell had the capability to screen itself in or out? Suffice it to say, the selection process would then no longer be the limiting factor. The bottleneck would then be at the mutagenesis rate, or the rate of churning out solution candidates. The work described in Section 5.2 was an attempt to provide a viable solution to the problem of developing an *in vivo* genetic device that can couple the biosynthesis of a non-essential metabolite to a survival condition. The genetic device was designed to confer the capability of self-screening or selecting for given fitness criteria on individual cells.

5.1.4 Application of the *in vivo* evolutionary approach to a metabolic engineering case in bacteria

A specific application case was needed, in order to explain implementation-level details of the concept of harnessing the *in vivo* evolutionary approach to design. A metabolic engineering case, more specifically the biosynthesis of riboflavin, was considered in this study as an example.

Riboflavin is essential to the production of biologically critical coenzymes such as FMN and FAD. Since higher organisms, such as mammals, have lost their ability to naturally synthesize the metabolite *in vivo*, higher organisms rely on dietary intakes for the supply of this essential nutrient, commonly known as vitamin B2. Only microorganisms and plants can produce riboflavin. The industrial riboflavin production is estimated to have exceeded 3000 metric tons per year, and about 80% of which is produced by employing microorganisms [114].

Taking the evolutionary design approach to the biosynthesis of riboflavin would offer to be an exemplary synthetic biology application having potentials for immediate industrial implications. Also, there already exist known genetic mutations that would result in riboflavin overproduction [40, 164] in a number of different or-

ganisms including *B. subtilis*. Those known mutations can act as reference points for comparing potential solutions generated by the evolutionary approach.

5.1.5 Minimising testing overheads via automation

As previously pointed out, testing is one of the most important elements in design. Its importance is especially pronounced in taking the evolutionary approach to design where solutions are found in randomness. The evolutionary approach is an iterative design scheme in which random solutions get repeatedly tested for their fitness.

The *in vivo* device shown in Section 5.2 was in fact designed to embed a molecular testing mechanism into individual cells. It is a form of automated testing achieved at the cellular level. There are other levels of testing that necessitate benchwork and data analysis in order to better understand putative genetic solutions at hand. Tests of such kind have costly overheads. The work shown in Section 5.3 addressed the subject of automating the analysis of data acquired from routinely performed laboratory procedures such as cytometry and genome sequencing.

5.2 Designing a genetic system to support *in vivo* evolutionary design

If the phenotypic change introduced by a point mutation of interest confers resistance to some substance that is otherwise fatal to the organism, its screening is as easy as using intrinsic selection or conventional molecular biology techniques in which survivors are picked from plates with selective growth conditions. In extrinsic selection cases, such as finding the trait of riboflavin biosynthesis, the same plate screening technique is not a viable option, as the production of riboflavin is normally considered unessential to survival.

In order to address this issue, an *in vivo* genetic system was designed to couple intrinsic selection pressure to extrinsic selection. The trait of riboflavin biosynthesis can be coupled to the survival condition of *B. subtilis* via using a molecular sensor capable of detecting the presence of riboflavin in the organism. *Bacillus*

subtilis possesses a negative feedback mechanism by which riboflavin production *in vivo* in the wildtype organism is regulated, as the microorganism only needs a trace amount of riboflavin in normal conditions.

FMN is a phosphorylated form of riboflavin that is involved in the negative feedback regulation of riboflavin biosynthesis in some microoragnisms including *B. subtilis*. *B. subtilis* uses a form of riboswitch that can have conformational changes depending on the presence of FMN in the cytosol [110, 186]. In the presence of FMN, the riboswitch forms a rho-independent termination loop at the 5-prime-end untranslated region of the *rib* operon, resulting in the premature termination of the operon's transcriptional events. In the absence of FMN, the termination loop structure is disrupted to a level not stable enough to halt transcriptional events. The endogenous FMN riboswitch of *B. subtilis*, used in the following coupler design, was adopted from a part of the 5-prime-end untranslated region of the *rib* operon[1] of *Bacillus subtilis* 168.

5.2.1 Building a FMN sensor construct

The FMN riboswitch sequence was obtained from the 285bp leader sequence upstream of the start codon of *ribD*[2] in *B. subtilis* 168. The strain's reference genome sequence from which the leader sequence was obtained is accessible via Genbank accession ID AL009126.3 [105]. The 285bp leader sequence should carry the native promoter and RBS of *ribD*. The sequence's FMN dependent riboswitch property provides a mechanism for the inhibition of the expression of any downstream genes concatenated to the sequence in tandem.

Two different ways to apply selection pressure was considered using this sensor construct. One was to use selective activation of a kill switch, and the other was to use selective activation of an antibiotic resistance. The former would need to actively promote a death mechanism for the unfit, while the latter would need a survival mechanism for the fit. Given the inhibitory regulation of the FMN riboswitch, it would seem that concatenating a death mechanism downstream of the sensor construct can be an effective measure to select for the population producing

[1] The *rib* operon expresses enzymes for metabolising nutrients into riboflavin and FMN.
[2] The *ribD* gene is synonymous to *ribG* in *B. subtilis*.

(or with the cytosolic presence of) FMN. It is true in a logical sense that "inhibiting death" of the fit is equivalent to promoting the fit. However, this approach would drive the fate of the unfit to plunge into death by default. This means that initially unfit mother cells (lacking the target phenotype) would be deprived of any chances to evolve. Therefore, it is imperative that the default status of mother cells not be death but survival. This design requirement was achieved by concatenating a NOT gate followed by an antibiotic resistance. By opting for the survival mechanism via antibiotic resistance, the time point in which selection takes place can be controlled at will. Mother cells can then take as much time as needed before being subject to antibiotic screening that selectively kills off the unfit.

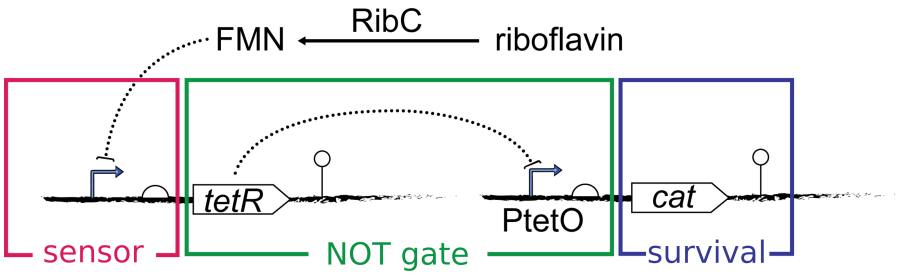


Figure 5.1: The sensor construct coupled to a survival mechanism via a NOT gate

5.2.2 A NOT gate coupled to the FMN sensor for positive regulation

A NOT gate was introduced to reverse the logic of the inhibitory regulation of the FMN riboswitch, so that the presence of FMN can be tied to survival (See Figure 5.1). The 285bp FMN riboswitch sequence was joined together with the coding sequence of *tetR* from transposon Tn10 (BBa_C0040), sourced from the Registry of Standard Biological Parts. The -35 and -10 regions of the *xylA* promoter of *B. subtilis* 168 [71] were used together with O_1 TetR operator sequence [72] in building a *B. subtilis* promoter negatively regulated by TetR.

Table 5.1: The truth table of FMN vs antibiotic resistance for survival.

FMN available	TetR expressed	resistance
True	False	True
False	True	False

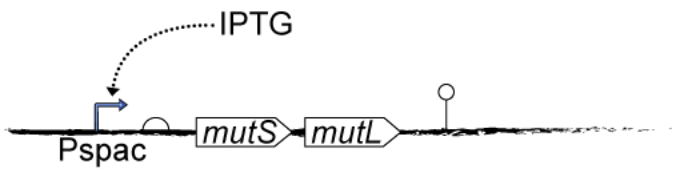


Figure 5.2: An inducible *mutSL* operon to programmatically regulate mutagenesis rates

5.2.3 A construct to regulate the mutagenesis rate

The mutagenesis rate of an organism is one of the most critical factors governing the speed of evolution. For organisms in the wild, mutagenesis is a dual-edged blade in that it is beneficial for survival during the times of trouble, while it can also destruct the integrity of genes. Evolution has worked out optimal mutagenesis rates that best suit the varying natural needs of different organisms. The usual rule-of-thumb strategy taken by nature is to inhibit mutagenesis when cells are happy, and to encourage mutagenesis when cells are under stress. In *B. subtilis*, for example, there are molecular mechanisms in place to regulate mutagenesis rates depending on cellular conditions via the *mutS mutL* operon [76, 158].

In light of directed evolution, mutagenesis is one of the critical bottlenecks in the process of generating and finding design solutions (See the discussion in Section 5.1.3). The wildtype spontaneous mutation rate would be too slow and inefficient in exploring large solution spaces of complex designs. For this reason, a genetic device that can induce a hyper-mutant state was devised (See Figure 5.2). The wildtype promoter of the *mutS mutL* operon in *B. subtilis* was replaced with an inducible promoter that can be regulated by IPTG. Rudimentary though it may be, the genetic device is a good example of how simple genetic modifications as such can make *in vivo* cellular systems capable of interfacing *in silico* systems, via a microfluidics device programmatically modulating the IPTG concentration. The ramification of having such an interface is quite significant in terms of achieving

design automation; this interface can enable programmatic modulation of the speed of evolution in the host chassis.

5.2.4 The evo_insert construct

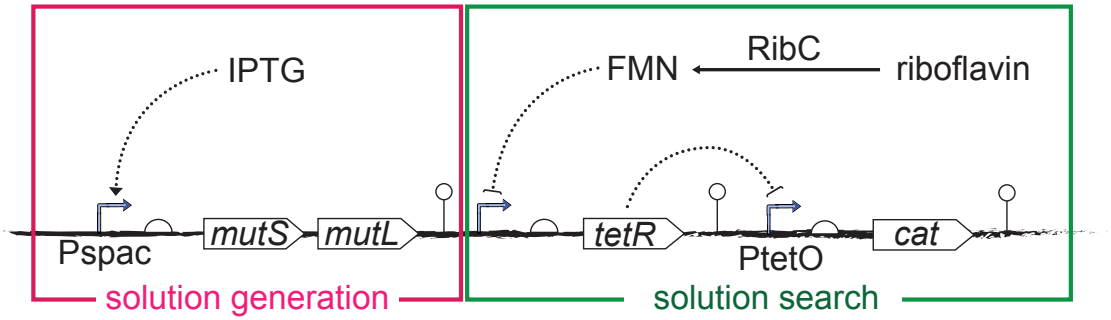


Figure 5.3: The evo_insert construct for insertion into pMUTIN4
See Appendix B.2 for the plasmid map of pMUTIN4.

The evo_insert construct was designed to have two functional modules mutually independent from each other (Figure 5.3). The first module is for regulating the expression of *mutSL* operon, thus functions to switch random solution generation on or off. The second module is for detecting the presence of FMN and for coupling the detection event to the expression of an antibiotics resistance marker (e.g. *cat*[1]). The second module would function to evaluate the fitness of randomly generated solutions, and to select the fitter ones.

5.2.5 Testing the coupler system: the EVOt2 construct

The EVOt2 construct was designed for testing the FMN sensor together with the NOT gate (Figure 5.4). The regulatory region of the FMN sensor in this construct had an additional xylose inducible promoter so that the strength of the downstream FMN-regulated promoter can be modulated by varying xylose concentration. The construct was designed to detect the presence of FMN and couple the detection

[1]The *cat* gene is for the expression of chloramphenicol acetyltransferase which confers resistance to the antibiotics, chloramphenicol.

event to the expression of a fluorescence marker to assist in the characterisation of the coupler system.

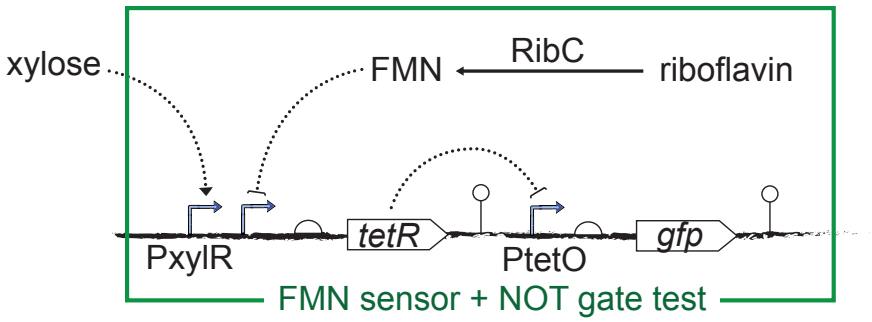


Figure 5.4: The EVOt2 construct for insertion into pSG1729

5.2.6 Cloning strategy

In molecular biology, the term "cloning" refers to a series of processes involved in introducing an exogenous genetic construct into a host organism. Various factors were needed to be taken into consideration in deciding on the appropriate cloning strategy. These included the host strain of choice, the target insertion locus (chromosome vs plasmid), the availability of vectors compatible for the chosen locus, etc. The evo_insert construct, being cloned into the host strain *B. subtilis* BSB1, was accounted for many strain-specific design factors such as the use of promoters and Shine-Dalgarno sequences (ribosome binding sites) compatible with *B. subtilis*. Initially, the evo_insert construct's two functional modules were designed to be inserted into a single chromosomal locus, at the 5-prime-end of *mutS*. The construct was synthesised by Genscript, assembled into the plasmid vector pUC57, and was collectively named pUC57_evo. pUC57 is a plasmid vector with a replication origin of *E. coli*, and is not intended for used in *B. subtilis*. The evo_insert construct, or the payload without the plasmid vector, was subcloned into the *B. subtilis* plasmid vector pMUTIN4 [180] (Appendix B.2), and was named pMUTIN4_evo (Section 4.2.1). pMUTIN4 was chosen as the backbone because the vector allows insertion at an arbitrary chromosomal locus depending

on the sequence homology built into the payload sequence. The evo_insert payload was designed with a sequence homologous to the 500bp sequence immediately following the promoter region of *mutS*. Please refer to Section 4.2.3 for details on how pMUTIN4_evo was constructed.

The EVOt2 construct was assembled using Gibson techniques (See Section 4.3). Its assembly fragments were amplified out of the evo_insert (Appendix B.1) in pUC57_evo (Figure 4.1), using the PCR primers listed in Table 4.14. As part of the assembly, the EVOt2 construct was inserted into the plasmid backbone, pSG1729 (Appendix B.3), and the Gibson product was accordingly named pSG1729_EVOt2 (Figure 4.9).

The EVOt2 construct was chromosomally integrated into the *amyE* locus of *B. subtilis* BSB1 using pSG1729_EVOt2, and the mutant was named *B. subtilis* BSB1 EVOt2 (Section 4.4). For use as an experimental control, the barebone pSG1729 was used to transform *B. subtilis* BSB1 for the chromosomal insertion of the vector's xylose inducible GFP expression circuitry at the *amyE* locus. The control mutant was named *B. subtilis* BSB1 SG0.

5.3 Automation of the analysis of measurement data

The evo_insert and EVOt2 constructs described so far are molecular devices designed to facilitate information exchange between the *in vivo* and *in silico* domains. By enabling single-cell-level phenotypic measurement coupled to selection, these molecular devices can increase the information entropy (or information density) per mutant population, consequently opening ways to increase the transfer rate of useful information to the *in silico* domain. They are examples of how directed evolution can be employed for finding solutions in light of the DEA framework. Solution search can take place in a massively parallel fashion given the power of the selection coupling (See Section 5.1.3). However, such molecular devices, albeit capable of achieving high throughput screening, are still subject to noise, resulting in multitudes of putative solutions that require further scrutiny.

Systematic characterisation required to scrutinise mutants at this stage is costly

and time-consuming. At the core of the DEA framework is the idea of harnessing iterative design cycles, where it becomes necessary to repeat laborious characterisation procedures over and over again. In this framework, the versatility offered by the following two characterisation means was promising: flow cytometry and genome sequencing. They can generate copious amounts of data in raw format. These data also need to be subject to postprocessing as well as downstream analyses in order to extract any useful information. Dealing with experimental data can be one of the most time consuming elements out of the characterisation effort. The burden of characterising clones can be lessened to a large extent by employing automation in this process.

Software pipelines were developed for postprocessing and analysing data obtained in their raw formats from cytometry and genome sequencing. These data pipelines were built as proof-of-concept examples of developing software components that can readily participate in the design cycle of DEA, automated to involve as little human intervention as possible. The pipelines were built as management layers to invoke various open-source libraries [12, 21, 38, 108, 113, 118, 190], each of which is good at handling specific data processing needs. There are useful GUI-based software tools [8, 66] to assist the postprocessing and analysis of such experimental data, primarily intended for occasional use by people on an *ad hoc* basis. The pipelines developed in this study were meant not to be used directly by people but to be used programmatically as part of an automated loop running in parallel backend servers.

Firstly, a software script that can programmatically process raw flow cytometry data was built and tested on *B. subtilis* clones labelled BSB1 EVOt2 and BSB1 SG0, respectively transformed as per Section 4.4 and Section 4.3.3. Secondly, software pipelines to automate the assembly and the analysis of whole-genome sequence data were developed and tested on the *B. subtilis* 168 and BSB1 wildtype strains as well as on the BSB1 EVOt2 clone.

5.3.1 Processing flow cytometry data for systematic analysis

Flow cytometry offers an efficient way to gather fluorescence-marker-based experimental data. Data of this kind are useful in testing whether relevant genetic parts function as intended. Due to large volumes of data typical cytometry experiments would produce, one of the most cumbersome steps in using cytometry is processing raw data into a form ready to be interpreted. Besides, postprocessing raw cytometry data is often influenced by inaccurate gating polygons hand-drawn at the whim of people. A Python script, that can programmatically postprocess raw experimental cytometry data, was developed. It is a rudimentary script consisting of procedures for simple data array manipulations and plotting operations. Yet, this test case was introduced to serve as an example contrasting the efficiency of using process automation versus manual labour in performing downstream post-experiment data analyses.

B. subtilis EVOt2 clone was cultured and subject to flow cytometry as explained in Section 4.4.2. The resulting raw cytometry data readings were processed by the Python script[1] to automatically generate Figure 5.8 in a matter of seconds. This process, if performed manually in a conventional wet-lab setup, would take hours, if not days.

5.3.2 Post-evolution whole-genome sequence analysis

Whole-genome sequence analysis provides a way of faithfully capturing and inspecting genetic solutions offered by selected clones. As part of such analysis, clones with potential solutions can be evaluated against relevant fitness criteria before their solutions can be accepted. The evo_insert circuitry, for instance, has a FMN sensor with which fitness measurements can be taken *in vivo* at the single cell level. The circuitry was designed for coupling measurement results with the application of antibiotics selection pressure. The significance of inspecting how clones survived selection in this instance is to differentiate valid selections from invalid selections. There are mainly two kinds of invalid selections: false negatives

[1] Appendix A.3 shows an instruction to invoke a Docker container to use the Python script.

and false positives. False negatives are missed opportunities that would deem valid solutions as invalid. False positives are the ones that would deem invalid solutions as valid.

One possible cause to these problems is measurement error in the screening process. Should measurement error be the culprit, these problems can be mitigated either by improving the quality of measurement systems (e.g. a better FMN sensor) or by altering the fitness threshold (e.g. antibiotics concentration). While improving measurement quality can reduce both kinds of problems, altering the fitness threshold may only reduce one of the two depending on which way the threshold is altered. Lowering the fitness threshold allows more sample solutions to be screened in, and would result in the reduction of false negatives at the expense of increased false positives. On the other hand, raising the fitness threshold restricts sample solutions from being screened in, and would result in the reduction of false positives at the expense of increased false negatives.

Another possibility causing these selection problems is that the clones being selected have anomalies in the evo_insert circuitry leading to conferring immunity to selection pressure without having the primary trait of concern (i.e. FMN biosynthesis). False positives as a result of the latter cause would need to be accompanied by a secondary screening scheme involving sequence analysis. As far as synthetic biology and its application of DEA in metabolic engineering are concerned, a secondary fitness evaluation via using whole-genome sequence analysis is promising, owing to the ever-dropping Next Generation Sequencing (NGS) cost. The caveat is that the secondary evaluation process would then need to be braced for the challenges posed by the inundation of genome-scale sequence data.

The challenges posed by having to deal with overwhelming amounts of DNA sequence data can be alleviated by using automation. The discussion to follow is about automating post-evolution whole-genome sequence data maneuvers, such as sequence data assembly and analysis. It is also worth noting that the discussion of the following automated pipelines will be a preamble to discussing, in Section 6.4.3, how evolutionary solutions *in vivo* can be integrated to *in silico* solutions to bridge the gap between the two domains.

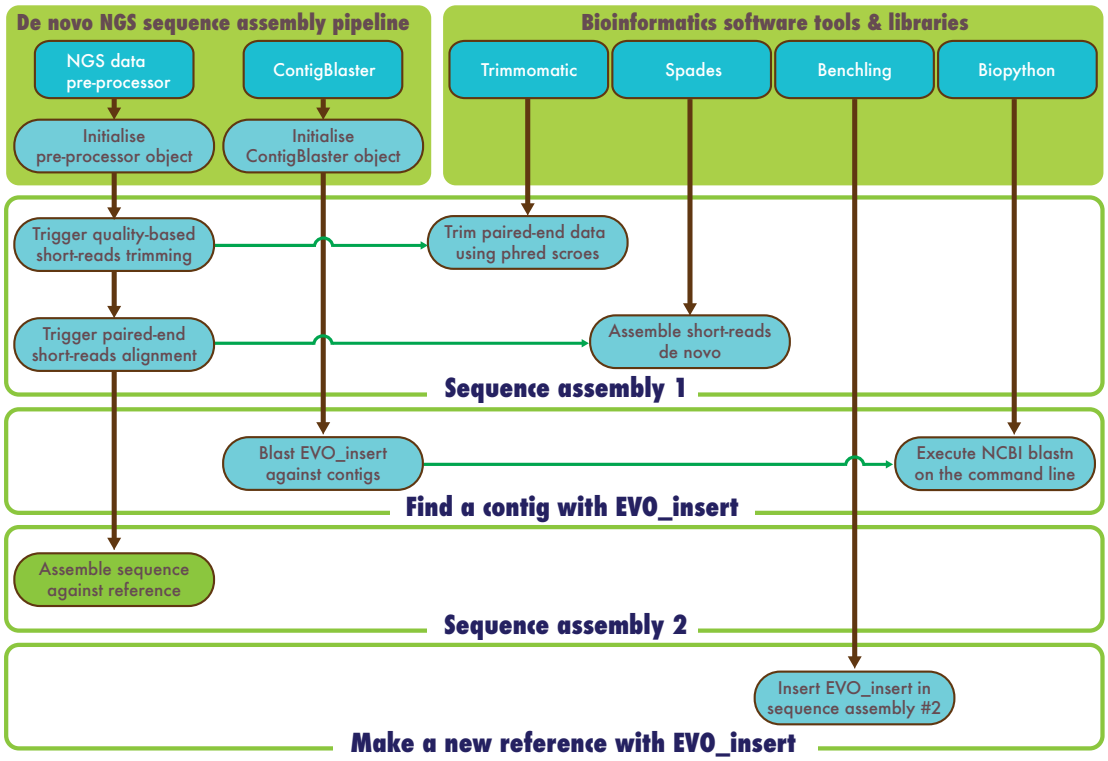


Figure 5.5: Overview of the hybrid sequence assembly pipeline

The pipeline consists of four main parts: sequence assembly *de novo*, locating the assembled sequence of EVO_insert, sequence assembly against reference (assembly #2), and combining assembly #2 and the assembled EVO_insert sequence.

5.3.2.1 NGS assembly pipeline

Automated NGS assembly pipelines can greatly reduce the burden of downstream genomic analysis processes. The lessened burden would subsequently result in an increase in the analysis throughput, key to success in the DEA framework. NGS devices generate short sequence reads collated together into raw unassembled sequence data formats, such as FASTQ. There are a series of steps involved in converting the raw data into some form ready to be consumed by other high-level bioinformatic analyses. Software pipelines were built, in order to automate the assembly of NGS sequence data. Here, sequence assemblies can primarily be done against a reference sequence of *B. subtilis* 168. However, *de novo* sequence assemblies are also needed to incorporate the *evo_insert* construct as part of the

final variant analysis. Figure 5.5 shows a sequence diagram of the hybrid assembly pipeline to build a reference sequence using a sequence assembly done against a gold-standard *Bacillus subtilis* 168 sequence (i.e. AL009126.3) together with an *evo_insert* sequence assembled *de novo*. The hybrid assembly pipeline, developed in Python 2.7, was packaged with all its dependent libraries to be provisioned as a Docker image, ready to be deployed and tried out[1]. The Docker image has an example demonstrating the hybrid assembly pipeline to build an annotated sequence assembly file out of NGS data obtained from a *Bacillus subtilis* BSB1 EVOt2 clone (Section 5.2.5).

The same hybrid assembly pipeline can also be used to build a base annotated sequence for a DEA mother cell with the evo_insert construct (Section 5.2.4). Such a base annotated sequence would function as a reference sequence in the variant analysis pipeline, shown in Figure 5.6, for investigating the SNPs in random mutants stemming off of the mother cell.

5.3.2.2 Variant analysis pipeline

For the sake of using DEA in finding genetic solutions to problems such as riboflavin biosynthesis in *B. subtilis*, it is imperative that experimental findings resulting from evolution *in vivo* be bridged onto the EA *in silico*. Variant analysis is appropriate for bridging such a gap, as the data made available by the analysis convey information on genetic variations among solution candidates readily compatible to the type of data required in exploring the solution space of the EA *in silico*. A further benefit to the use of variant analysis is providing a solution to decrease false positives. Information regarding SNP variants, specifically in the chromosomally integrated evo_insert construct, can be leveraged in having solution candidates checked against false positives. Candidates with significant mutations in the solution search module (Figure 5.3), as part of the evo_insert construct, are prone to become false positives, therefore can be precluded from further analysis. Significant mutations include non-silent SNPs[2] in the coding re-

[1]Please see the instructions in Appendix A.2 to gain access to the Docker image and to execute the pipeline as part of a Docker container.

[2]Non-silent mutations are significant mutations with possible alterations in the phenotypes of the protein in which they occur. They occur in coding regions and involve changes to the amino

gions of the solution search module, and mutations in the module's regulatory elements such as promoters, operators, FMN riboswitch, and ribosome binding sites. SNPs occurring in loci other than evo_insert can be used to help with the *in silico* analysis as described in Section 6.4.2.

A software pipeline was developed (Figure 5.6) to enable variant analysis on mutants in order to associate their SNPs to desirable mutant phenotypes such as increase in the riboflavin biosynthesis. In order to test the variant analysis pipeline, *Bacillus subtilis* 168 and BSB1 strain collections independently maintained by different research groups at the Centre for Bacterial Cell Biology in Newcastle University were acquired, and sequenced via the in-house NGS service using Illumina MiSeq. The resulting raw sequence data for each strain sample were uniquely labelled, assembled against the reference (AL009126.3), and processed through the variant analysis pipeline to respectively generate EMBL files with SNP annotations (See Table 5.2).

The test clone described in Section 5.2.5, a *B. subtilis* BSB1 with the EVOt2 construct chromosomally integrated at its *amyE* locus, was sequenced using a NGS device (Illumina MiSEQ). A docker image (see Appendix A.2) was built to demonstrate the application of the workflow shown in Figure 5.6 for checking if the transformation of the clone was done successfully. A reference sequence was constructed by inserting the EVOt2 sequence at the *amyE* locus of the 168 reference sequence (AL009126.3), and was named AL009126_EVOt2[1].

acid sequence, or in non-coding regions and involve changes to critical regulatory sequences.

[1]The AL009126_EVOt2 sequence in Genbank format was made available via accessing the Docker image `sungshic/dea_hybrid_assembler:1.1` from Docker Hub. The Genbank file was named `AL009126_EVOt2.gb` and was placed in `/root/workspace/evot2`, accessible as part of the Docker container upon instantiation. See Appendix A.2 for details on the Docker image.

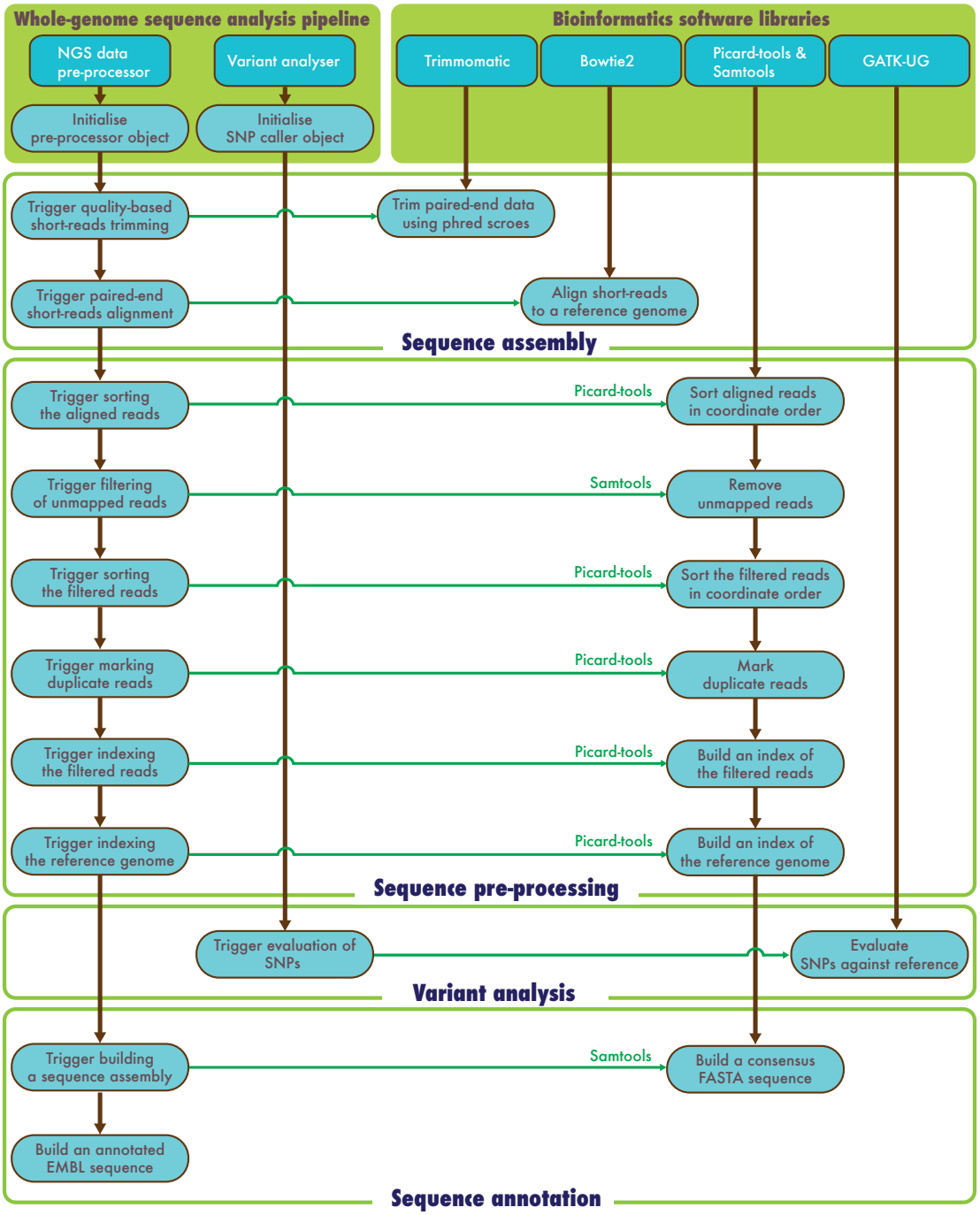


Figure 5.6: Overview of whole-genome sequence analysis pipeline

The pipeline consists of four main parts: sequence assembly, sequence pre-processing, variant analysis, and sequence annotation.

5.4 Results from running the pipelines for sequence assembly, variant analysis, and plotting cytometry data

Table 5.2: Variant analysis of genome sequencing samples against reference genomes

Sample ID	Sample 1	Sample 13	Sample 14	Sample 16	Sample 21	BSB1_wendy	BSB1_wendy
Sequencing platform	MiSeq	MiSeq	MiSeq	MiSeq	MiSeq	MiSeq	MiSeq
Description	168 strain from Heath	MSBS168, Pasteur strain from Colin	WSBSB1, a BSB1 strain from Etienne	168 strain from Aurelie	1A1, 168 strain from BGSC	BSB1 strain from Wendy	BSB1 strain from Wendy
Annotated embl filename	sample1_vs_AL009126_v11_annotated.embl	sample13_vs_AL009126_v11_annotated.embl	sample14_vs_AL009126_v11_annotated.embl	sample16_vs_AL009126_v11_annotated.embl	sample21_vs_AL009126_v11_annotated.embl	BSB1wendy_vs_AL009126_v11_annotated.embl	BSB1wendy_vs_SLR16.1_annotated.embl
Ref Seq.	AL009126	AL009126	AL009126	AL009126	AL009126	AL009126	SLR16.1
non-silent SNPs	1	6	1	2	20	1	too many
total SNPs except trp operon	6	12	8	6	36	7	1588
total SNPs	6	12	84	6	36	60	1646
Genes with non-slient mutations (in samples marked 'X'), except the trp operon							
uvrX	X	X	X	X	X	X	
ytkK		X					
gerAA		X(2)			X		X(2)
sdpB		X					
yxbD		X			X		X(2)
ypqP				X			
scoC					X		
oppD					X		
sigI					X		
mswC					X		
sepF					X		
rluD					X		
trmD					X		
ymfD					X		
gltA					X		X(2)
yoqA					X		
ypiB					X		
yqxL					X		
sftA					X		
comP					X		X(4)
yutE					X		
epsC					X		
sacA					X		X
							...

Table 5.2 summarises a SNP analysis result made available by the NGS analysis pipeline discussed in Section 5.3.2.2 and Figure 5.6. The example raw sequence data used here were chosen for the purpose of demonstrating the capability of the

software pipeline. Samples labelled 1, 13, 16 and 21 were *B. subtilis* 168 strains. It was shown that their sequences did not deviate much from the *B. subtilis* 168 reference AL009126, with the exception of Sample 13 and Sample 21. Sample 13 was a 168 strain from Pasteur Institute, and Sample 21 was a 168 strain acquired from Bacillus Genome Stock Center.

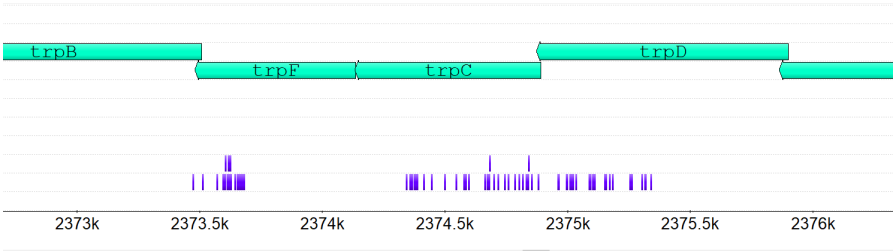


Figure 5.7: The *trpDCF* region of Sample 14 carrying the majority of its SNPs
Indicated in purple bars below the green CDS arrows were the SNPs skewed in the *trpDCF* region, contributing to 75 of a total of 84 SNPs reported in Sample 14 (Table 5.2). The SNP profile and the corresponding EMBL annotated sequence generated by the pipeline developed in this study was visualised using Unipro UGENE [131].

It was evident from the pipeline's variant analysis result that the 168 strains of Samples 13 and 21 had many non-silent SNP variants with respect to the reference. Samples labelled 14, 21 and BSB1_wendy were *B. subtilis* BSB1 strains, derived from *B. subtilis* 168 by replacing its *trp* operon to fix 168's tryptophan auxotrophy. The two *trp* operons were clearly different, as the total SNP counts of BSB1 strains exhibited striking differences between the counts including and excluding the *trp* operon. As a control measure, the BSB1 strain (BSB1_wendy) was compared against the old *B. subtilis* 168 reference sequence (SLR16.1) from Pasteur Institute. Immediately noticeable in this comparison was how different the old 168 reference (SLR16.1) was from the new 168 reference (AL009126). It was also interesting to see that the *uvrX* SNPs recurrent on other samples against AL009126 were no longer there against SLR16.1. After all, the pipeline was able to successfully automate the processes of assembling raw short sequence reads and analysing SNP variants. The correctness of the variant analysis pipeline was evident in its ability to detect the tryptophan auxotrophic allele of 168 (i.e. *trpC2* [3]). Sample 14 shown in Table 5.2, for instance, was a BSB1 strain which is a prototrophic derivative of 168. As visualised in Figure 5.7, Sample 14's SNP profile with respect

to 168 had a great majority of them skewed within the *trp* operon, more specifically within *trpDCF*. The *trpDCF* region carried 75 SNPs out of a total of 84 SNPs reported in Sample 14.

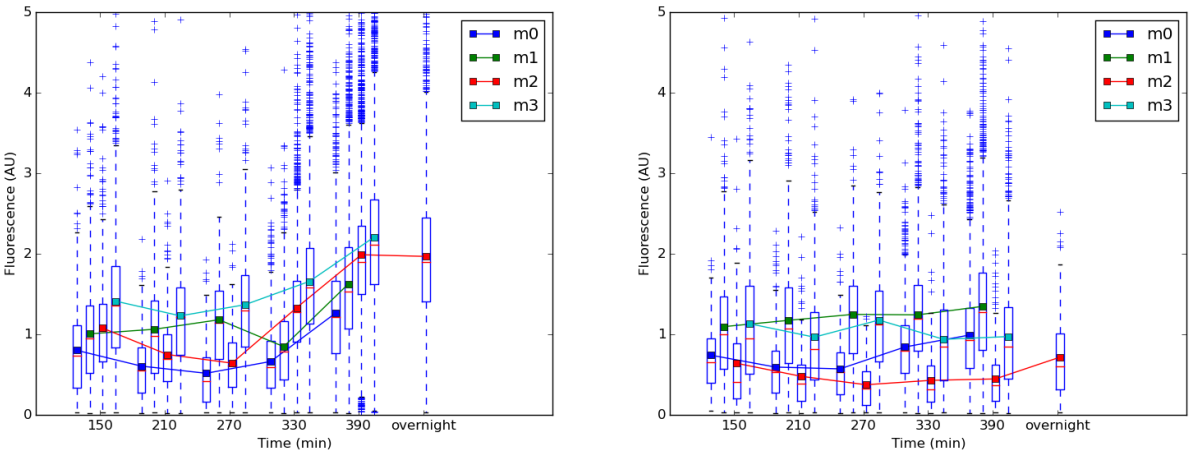


Figure 5.8: Time-lapse flow cytometry results

There are four different growth media: m0, m1, m2, m3. **m0**: LB 1 % (w/v) xylose. **m1**: LB 1 % (w/v) xylose + riboflavin. **m2**: LB. **m3**: LB + riboflavin. Fluorescence is calculated by dividing the fluorescence values measured by the flow cytometer by individual particles' forward scatter values. This division offsets the high fluorescence values due to bacteria forming chains. (a) BSB1 EVOt2 in four different media conditions. (b) BSB1 SG0 in the four different media conditions.

The results in Figure 5.8, showing time-series boxplots of the fluorescence of individual cells, were programmatically produced from raw cytometry data (See Section 5.3.1). Figure 5.8a was based on the BSB1 EVOt2 clone, and Figure 5.8b on the BSB1 SG0 clone[1].

The expected behaviour of the EVOt2 clone in the m0 media was to have the lowest fluorescence level. The lack of riboflavin in the growth media would mean the lack of FMN in the milieu. This would make the riboswitch in the EVOt2 circuitry not to halt the expression of downstream genes (*tetR* in this case). This ensues TetR to be available in the milieu, inhibiting the expression of

[1]BSB1 SG0 was cloned by transforming *B. subtilis* BSB1 with pSG1729 (See Appendix B.3) to confer the clone with xylose inducible fluorescence (GFP).

the fluorescent protein downstream of the promoter guarded by a *tet* operon. In addition, having xylose in the media would increase the expression of *tetR*, further intensifying its inhibitory role of the fluorescence level. The flow cytometry result (labelled m0 shown in blue) in Figure 5.8a confirmed this expected behaviour by exhibiting the lowest fluorescence out of the four conditions.

The m1 media given the presence of riboflavin in addition to the media composition of m0 was expected to have fluorescence higher than that of m0. A high concentration of riboflavin would contribute to the rise of FMN concentration in the milieu which consequently would inhibit the expression of *tetR*. A decrease in TetR concentration would lower the inhibition of the fluorescence expression, hence resulting in increased fluorescence.

The m2 media had neither riboflavin nor xylose. The lack of riboflavin, as explained above, would contribute towards decreasing fluorescence, while the lack of xylose would contribute towards increasing fluorescence. It was speculated that the fluorescence due to m2 would be higher than that of m0, yet lower than that of m3. This was confirmed to be the case in the experimental results summarised in Figure 5.8a.

The m3 media had riboflavin (hence FMN) as the sole contributing factor of fluorescence. The presence of which would increase fluorescence. Given the lack of any inhibition factors, this media composition would result in the highest fluorescence level out of the four conditions. This was also confirmed to be the case in Figure 5.8a.

While the expected behaviours versus growth conditions were shown to be maintained throughout the observed growth time span, all four conditions exhibited gradual increase in fluorescence over time. The likely explanation to this result is that the host system's native *rib* operon would, over time, produce riboflavin *in vivo* contributing to the regulation of the EVOt2 circuitry. However, the verification of this claim requires a negative control such as a separate EVOt2 clone with a deletion of the riboflavin synthase gene (*ribB* in Figure 6.3) involved in producing riboflavin *in vivo*. Due to time constraints in the project, making the clone for negative control was not pursued.

In fact, the EVOt2 construct had a glitch in the design. The emission spectrum of riboflavin overlaps that of GFP, the choice of fluorescent protein in the EVOt2

design. This clash attributed to the basal fluorescence levels in the measurements. The side-effect of this design glitch was shown in Figure 5.8b generated from the cytometry data of the BSB1 SG0 clone grown under the four growth conditions. The BSB1 SG0 clone's fluorescence expression circuitry transformed from pSG1729 should normally be inducible only via xylose. Xylose was able to induce the fluorescence expression of BSB1 SG0 to increase over time as shown in the m0 and m1 conditions in Figure 5.8b. However, it was also evident that the m1 and m3 conditions, given the presence of riboflavin, elevated the basal fluorescence level compared to the non-riboflavin conditions (i.e. m0 and m2). With the benefit of hindsight, this design mistake helped highlight the importance of having a means to make design amendments easier: that is to have an automated testing to appease the pain of repeating experiments as exemplified in this chapter.

5.5 Discussion

Evolution is a powerful problem-solving framework. Harnessed with random mutagenesis and selection, evolution is universally applicable and elegantly simple a genetic solver that can give rise to an amazing array of genetic solutions with diverse functionalities. In natural evolution, as Charles Darwin had speculated in his seminal work [49], the selection process is intrinsically welded into the survival of organisms amid the peculiarities of given environmental niches. Provided with the survival as its ultimate goal, how that goal is achieved by organisms is not a matter of concern. In directed evolution, with respect to DEA, the survival of organisms, extrinsically coupled to the selection process *in vivo*, is not the ultimate goal but an intermediary step to isolate clones bearing potential genetic solutions. Here, how organisms managed to survive the selection pressure does matter, necessitating the clones or solutions selected as part of the *in vivo* selection process to be subject to further scrutiny. The research work presented here addressed the question of how the solution search can be facilitated in the immense pool of random *in vivo* solutions. Screening for mutants at the molecular level via genetic devices *in vivo*, and streamlining the characterisation of mutants bearing putative solutions via automation were this study's attempts to provide an answer to this non-trivial question. As yet, this only constituted half an answer. How *in silico*

modeling and virtual evolution can improve on the validity of solutions found *in vivo* was as difficult a question to be answered. Working on answering this resulted in a significant section on its own, expansive enough to warrant Chapter 6.

One of the primary challenges in the dual-evolutionary design framework was how *in vivo* and *in silico* design domains can be integrated. DNA was one of the most obvious common denominators between the evolutionary solution spaces of the two design domains. The DNA sequence analysis hitherto shown in this study was an effective means with which transitions to and from the two solution domains can be achieved. The variant analysis of *in vivo* solutions, for instance, provided information at an abstraction level suitable for fitting into *in silico* solutions in a manner that is computationally simple yet analytically meaningful.

Chapter 6

In silico model based design to bridge the gap in the dual-evolutionary domains

6.1 Introduction

To date, proponents of design automation have contributed in the research and development of building various means to achieve automated assistance in designing processes. As a result of such endeavour, the use of computerised design assistance tools has become a routine practice across varying fields of engineering [144]. These tools, collectively termed Computer Aided Design (CAD) software, have made it possible for humanity to achieve new levels of heights in wielding design complexity [96]. Field experts claim that "No product is designed today without the use of computer-aided design (CAD) technology" [22]. They are the means with which the modern civilisation as we know it today has been technologically innovated.

CAD tools have helped raise the bar in the level of complexity manageable by designers to a certain extent. Nevertheless, CAD tools are limited to providing "islands of automation" [46], and are never intended for replacing human designers. The use of automated assistance in CAD is still bottlenecked by the reasoning capacity of designers. Given high degrees of complexity and uncertainty involved in engineering biological designs, the validity of CAD tools [19, 33, 48, 135] will

eventually reach a saturation point.

Biological systems, whose cryptic mechanisms have come about as a result of natural evolution, are complex because of "the bewildering diversity of interactions and regulatory networks" [182]. Our current lack of a total understanding of this intricacy poses unprecedented challenges in engineering the design of biological systems.

One of the primary hurdles in designing complex biological systems is the current lack of methodological paradigms that can handle the scale of design complexity in a significantly meaningful manner. In fact, nature has already been mindlessly carrying out design tasks. Nature, without relying on any intelligent agents to mastermind design tasks, has always been able to precipitate design solutions reaching complexity well above the reasoning capacity of human minds. The making of *Homo sapiens* by nature [29], for instance, is an epitome in reaching the height of design complexity far beyond that achievable by our own capability to design at the moment. Nature's vehicle for such a design feat and autonomy is evolution [29, 55]. Evolution employs randomness and selection [107] as the universal tools to explore and search for solutions to design problems. Natural evolution is a living proof that complex biological designs can be achieved autonomously without human interventions.

In Chapter 5, the idea of adopting the methodological paradigm offered by evolution was explored in the *in vivo* design domain. At the core of this evolutionary approach was seeing design as a search problem that can be systematically maneuvered via randomness and selection. It was hypothesised that a combination of molecular devices and programmatically executed analytic pipelines can facilitate the search for design solutions in random *in vivo* pools, consequently opening up ways to enable design automation in synthetic biology.

In light of seeing design as a search problem, the size of solution spaces to be explored is proportionate to design complexity. Complex designs hosting larger solution spaces would hence require more time before solutions can be found. With respect to the dual evolutionary design approach, it was suggested [80] that delegating some of the *in vivo* domain's search burdens to the *in silico* domain would result in a significant reduction in time to solution. This claim was verified in the work described in Section 6.4.3, following the presentation of details on how

cellular systems were modelled to serve as an interface between the two domains in an attempt to reduce their gap.

In engineering, design complexity can be mitigated to an extent by introducing modularity in design [134, 154] or by using mathematical modelling [63]. Modularity helps break larger problems into smaller manageable ones and facilitate the reuse of designs. The complexity of systems design can be reduced by modules decoupling unnecessary interdependencies of underlying subsystems. Mathematical modelling, on the other hand, helps quantitatively describe and assess complex designs in a systematic fashion. Using modularity and modelling has already been a practice well established in software engineering for conquering complexity [86]. The idea of using these mitigators in engineering biological designs has also been attempted with some success [39, 62].

In modelling the example of riboflavin biosynthesis, two approaches, dynamic versus static modelling, were respectively investigated in Section 6.3.2 and 6.3.3. Each approach revealed different pros and cons, but the static model was more promising to be used as a fitness function for *in silico* evolution, and offered to provide a more compatible interface between the dual-evolutionary domains.

In addition to this finding, the extent to which the use of *in silico* evolution can expand the searchable limit in the solution space was investigated in Section 6.4. This work confirmed the initial hypothesis by Hallinan and her colleagues [80] in proposing the dual-evolutionary framework as an effective measure to mitigate design complexity in synthetic biology. As part of this work, a novel method that can model the population-level dynamics of mutagenesis was necessary and was developed (See Section 6.3.4).

6.2 The significance of modelling

What is the significance of *in silico* modelling in terms of seeing design as a search problem? Models *in silico* are virtual reconstructions of certain real-world functions [7]. Out of such functional reconstructions, computers can calculate the predicted output as a consequence of a given input.

Mathematical modelling allows relatively cost-effective grounds for estimating whether certain design choices can meet given design goals. It is a well estab-

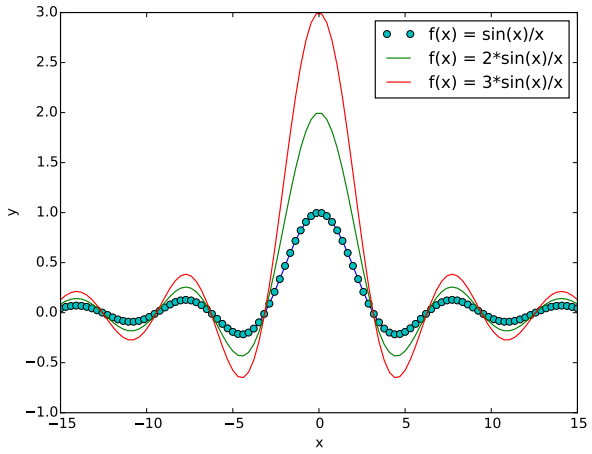


Figure 6.1: Some examples of the archetypal function of the form $y = f(x)$

lished methodology in affecting design choices in the software field of requirements engineering [129], for example. The practice of using mathematical abstraction in this work was particularly concerned with the establishment of a systematic naming convention. Achieving a mathematical abstraction permits the conceptual labelling of a physical reality such that different aspects of the physical reality can be systematically referred to by *in silico* processes.

Before things can be named anew, however, they would have to be defined first [111]. How things can be defined is an open-ended question, highly dependent on the context within which the definitions are to be used. Mathematical abstraction offers an universal system of vocabularies, or a language, to allow things to be defined in an unequivocal and consistent manner. It is a linguistic framework for engineering [79]. Following is an archetypal mathematical equation that represents the output y as a function f of the input x. Some arbitrary functions were plotted in Figure 6.1 to exemplify this archetype.

$$y = f(x) \tag{6.1}$$

Drawing a parallel with this functional archetype, biological systems can be

defined with respect to the same three constituents, namely the input x, the output y, and the function f defining the relationship between the input and the output. For example, y can be the phenotypic outcome of a biological system or a function f that responds to environmental conditions x. In the context of metabolic reactions, for instance, x could be a set of metabolites subject to an enzymatic reaction f to result in a different set of metabolites y.

6.2.1 Modelling as a tool for design documentation and exploration

Integral to the design process is the means with which real-world properties can be documented. The old-fashioned way of documenting designs, as exemplified by Leonardo da Vinci's schematic drawings [96], has been in written and drawn forms using carbon copies. These manual practices continued well into the first half of the 20th century [96], while they evolved to offer more accurate renderings of engineering drawings compared to those of the Renaissance. CAD tools, since the influential development of Sketchpad [168] in the sixties, have helped digitise design documentations [25]. Often, CAD tools just provide authoring means to help document design properties in digital formats, only allowing as much details as carbon copies would contain. Digitising information, even at such a rudimentary level, has immediate benefits, including the ease of archiving, sharing, editing, and finding information.

More substantial a benefit from digitisation would be that designs can be represented using much more dynamic context via *in silico* models. Modelling allows the documentation of a rich array of dynamic design properties unattainable by the carbon-copy driven documentation of designs. However, the *in silico* modelling of complex systems is inherently limited by finite computer resources. This means that the minute details of real-world properties cannot be represented in full by models. It is imperative that only the properties pertinent to the modelling goal of interest need consideration using the parsimonious principle of Occam's razor [160]. Mathematical abstraction is a great ally in abiding by such principles [106], playing an important role in the documentation of complex designs.

It is argued that facilitating the re-use of up-to-date design knowledge, infor-

mation and data can help increase the efficiency of producing novel designs [46]. In the dual-evolutionary approach, as demonstrated in this work, functional *in silico* models by expanding from Equation 6.1 were developed to serve as a means to document and to share the complex fabric of biological systems and their designs. This effort is in agreement with the efficiency argument, in that it not only bolsters the important role of documentation in design but also of mathematical models in documenting the insight into the functions of design.

Dealing with *in silico* models is cost-effective in that models can be evaluated repeatedly against various conditions. As such, models offer ways to explore the solution spaces of design problems without needing to implement solutions in reality. Therefore, models are great means with which to search for fitting solutions in design spaces. Being able to programmatically search for solutions is especially useful in terms of mitigating design complexity.

Furthermore, *in silico* models can enable introspection into the realm that are normally out of reach in reality. Such capability of models is a perfect fit for exploring the solution spaces of designs involving aspects that are physically infeasible to be measured in reality. It is a promising notion to use modelling as an exploration tool in the quest to finding design solutions.

6.3 Modelling cellular systems

Cellular systems have underlying molecular mechanisms that enable the systems to function. The law of physics governing those molecular mechanisms can be encapsulated using *in silico* modelling. Modelling allows the reconstruction of functional behaviours of the systems in light of mathematical abstraction. Models can be used to further the understanding of how cellular systems work, and to predict their behaviours. *In silico* models can be broadly categorised into two classes - dynamic and static models. Dynamic models address temporal changes in systems, while static models evolve around time-invariant steady-state conditions of systems in equilibrium. In the following, riboflavin biosynthesis was used as an example to explain how different aspects of cellular systems can be modelled using these two classes.

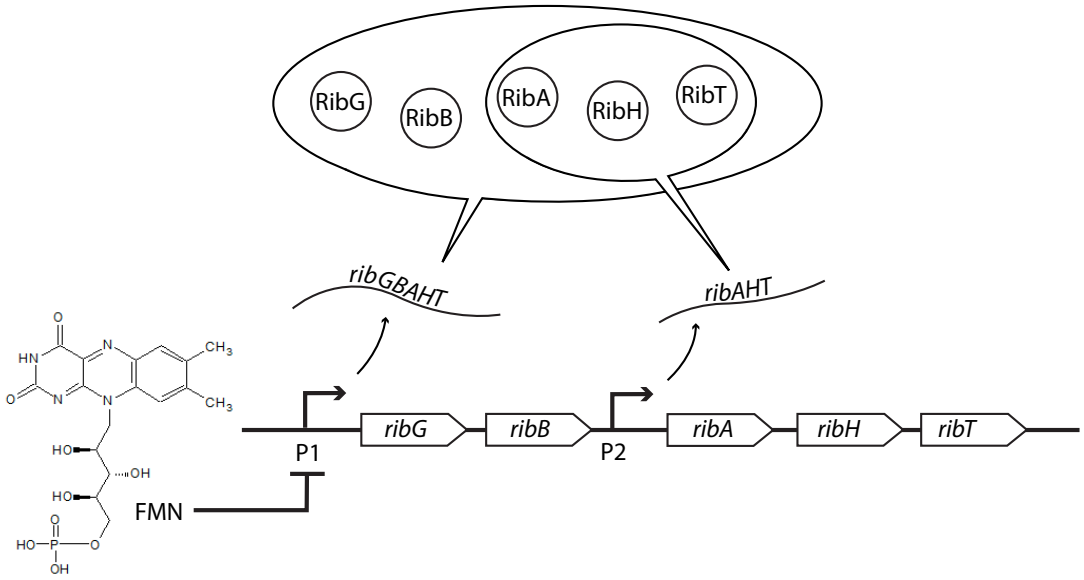


Figure 6.2: The *rib* operon and its transcriptional regulation in *B. subtilis*

6.3.1 Riboflavin biosynthesis and metabolic pathways in bacteria

In *Bacillus subtilis*, the riboflavin biosynthesis and its metabolic pathway are largely affected by a set of five genes (*ribGBAHT*, Figure 6.2) comprising the *rib* operon [121, 184]. It has been postulated that the metabolism and transport of riboflavin are regulated via transcriptional attenuation in Gram-positive bacteria, as opposed to the translational level regulation of riboflavin in most Gram-negative bacteria [184]. The five *rib* genes encode catalytic enzymes for converting one molecule of guanosine-5-triphosphate (GTP) and one molecule of ribulose-5-phosphate (R5P) into one molecule of 6,7-dimethyl-8-ribityl-lumazine (DMRL), and for subsequently converting two molecules of DMRL into one molecule each of riboflavin and 5-amino-6-ribitylamino-2,4(1H,3H)-pyrimidinedione [10, 114] (See Figure 6.3). The two precursor metabolites, GTP and R5P, are sourced through two distinct pathways, GTP via the purine metabolic pathway and R5P via the pentose phosphate pathway. These precursors go through a series of enzymatic reactions catalysed by the protein products of the *rib* operon, leading to DMRL, the immediate precursor of riboflavin in these pathways. The DMRL to riboflavin

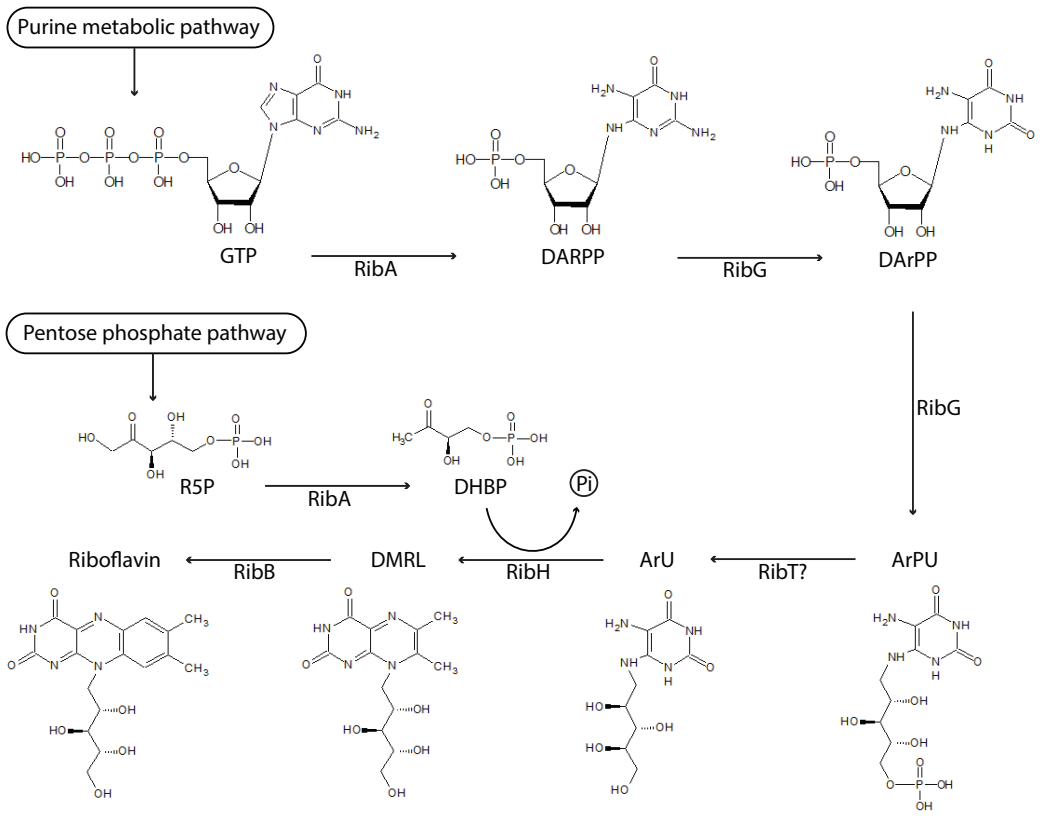


Figure 6.3: Riboflavin biosynthesis pathway in *B. subtilis*

GTP: guanosine-5-triphosphate **DARPP**: 2,5-diamino-6-ribosylamino-4(3H)-pyrimidinone-5'-phosphate **DArPP**: 2,5-diamino-6-ribitylamino-4(3H)-pyrimidinone-5'-phosphate **ArPU**: 5-amino-6-(5'-phosphoribitylamino)uracil **ArU**: 5-amino-6-ribityl-aminouracil **DMRL**: 6,7-dimethyl-8-ribityl-lumazine **R5P**: ribulose-5-phosphate **DHBP**: 3,4-dihydroxy-2-butanone-4-phosphate

reaction is catalysed by riboflavin synthase (RibB in *B. subtilis*). The *B. subtilis* genes *ribA* (DHBP synthase), *ribG* and *ribB* have different names in *E. coli*, respectively called *ribB*, *ribD* and *ribE* [184].

Most bacteria produce less than 10 mg/L of riboflavin, while some clostridia can yield up to 100 mg/L, some yeast such as *Candida flareri* up to 600 mg/L, and molds such as *Eremothecium ashbyii* and *Ashbya gossypii* can yield well over 1000 mg/L depending on the fermenting conditions [52].

It has been reported that some flavinogenic species overproduce riboflavin when

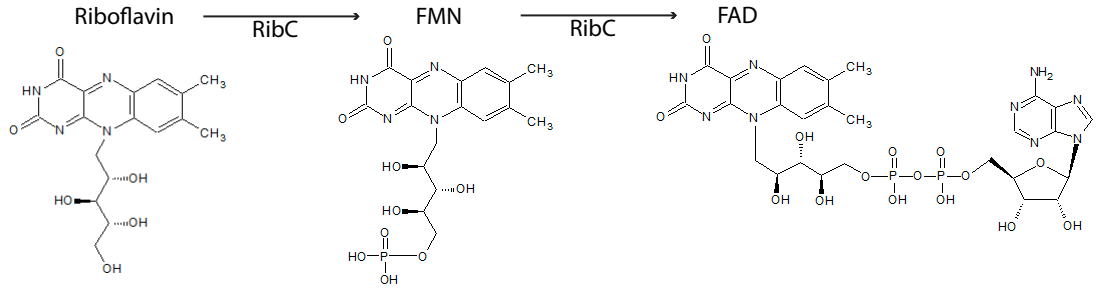


Figure 6.4: The bifunctional enzyme RibC in *B. subtilis*
Riboflavin to FMN and FAD are catalysed by RibC. **FMN**: flavin mononucleotide **FAD**: flavin adenine dinucleotide

purine derivatives are in abundance [10]. *B. subtilis* with mutations in the *ribC* gene (putatively encoding flavokinase and FAD synthetase) was shown to overproduce riboflavin [40, 116, 136]. The combination of mutations in *ribC* as well as in the leader sequence (i.e. regulatory region) of the *rib* operon could boost riboflavin production in *B. subtilis* beyond 10 g/L [136]; that is three orders of magnitude higher a level of production compared to those of most naturally occurring bacteria.

The effector molecule directly involved in the regulation of riboflavin biosynthesis had not been elucidated until recently. Experimental results have suggested that riboflavin biosynthesis is regulated not via employing riboflavin as an effector molecule but via FMN [10]. It seems FMN has inhibitory effects on the *rib* operon by binding to the conserved regulatory element in the 5' UTR called rfn [128]. Moreover, FMN's being an effector molecule is congruent to the *ribC* mutants exhibiting riboflavin overproduction - The *ribC* mutations would probably have some sort of inhibitory effect on flavokinase (an enzyme for riboflavin to FMN conversion) or activating effect on FAD synthetase (an enzyme for FMN to FAD conversion), resulting in a lower overall intracellular FMN concentration (See Figure 6.4).

6.3.2 Constructing a dynamic model for riboflavin metabolism

Dynamic models can offer exciting possibilities for understanding metabolic pathways by allowing detailed inspection of the temporal dynamics of metabolites and enzymatic reactions. The seminal work of Guldberg and Waage in the law of mass action [78] initiated the developments of chemical kinetics, such as Michaelis-Menten kinetics [120] in the area of enzyme catalysis. Michaelis-Menten kinetics is probably one of the most popular enzyme kinetics models, still in prevalent use after its inception more than a century ago [41].

The premise of the Michaelis-Menten kinetics model, as best described in the work by Briggs and Haldane [73], is based on an enzyme (E) and a substrate (S) associating at a rate of k_f to form a complex (ES), or ES dissociating at a rate of k_r (See Equation 6.2). Formation of ES, subsequently catalyses the irreversible conversion into product(s) (P) at a rate of k_{cat}.

$$\mathrm{E + S \underset{k_r}{\overset{k_f}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E + P} \tag{6.2}$$

$$v = \frac{V_{max}[S]}{K_M + [S]} \tag{6.3}$$

The rate at which P forms ($\frac{d[P]}{dt}$) is given by Equation 6.3, where $V_{max} = k_{cat}[E]_0$ and $K_M = \frac{k_r + k_{cat}}{k_f}$. $[E]_0$ is the initial concentration of the enzyme E. K_M is also known as the Michaelis constant indicating the substrate concentration ($[S]$) at which the reaction rate reaches $\frac{V_{max}}{2}$. The Michaelis-Menten (or Briggs-Haldane) kinetics would be applicable to all enzymatic reactions in Figures 6.3 and 6.4, except the reaction catalysed by RibH. The RibH reaction has two substrates (DHBP and ArU) producing one product (DMRL), which Equation 6.2 does not support.

The following chemical equations show an example of two substrates going through enzymatic reactions to produce one product. It is assumed that S_1 and S_2 are independent of each other in binding to the enzyme E. Hence, the association (k_{f1}) and dissociation (k_{r1}) rates of S_1 to E are unaffected regardless of whether

the other substrate S_2 is bound to E, and vice versa.

$$\begin{aligned}
\mathrm{S_1 + E} &\underset{\mathrm{k_{r1}}}{\overset{\mathrm{k_{f1}}}{\rightleftharpoons}} \mathrm{S_1E} \\
\mathrm{S_2 + E} &\underset{\mathrm{k_{r2}}}{\overset{\mathrm{k_{f2}}}{\rightleftharpoons}} \mathrm{S_2E} \\
\mathrm{S_1E + S_2} &\underset{\mathrm{k_{r2}}}{\overset{\mathrm{k_{f2}}}{\rightleftharpoons}} \mathrm{S_1S_2E} \\
\mathrm{S_2E + S_1} &\underset{\mathrm{k_{r1}}}{\overset{\mathrm{k_{f1}}}{\rightleftharpoons}} \mathrm{S_1S_2E} \\
\mathrm{S_1S_2E} &\xrightarrow{\mathrm{k_{cat}}} \mathrm{P + E}
\end{aligned} \tag{6.4}$$

The law of mass action [78] gives rise to the following ordinary differential equations defining the rate of change of chemical species in Equation 6.4 with respect to time t.

$$\begin{aligned}
\frac{d[S_1]}{dt} &= -k_{f1}[S_1][E] + k_{r1}[S_1E] - k_{f1}[S_2E][S_1] + k_{r1}[S_1S_2E] \\
\frac{d[S_2]}{dt} &= -k_{f2}[S_2][E] + k_{r2}[S_2E] - k_{f2}[S_1E][S_2] + k_{r2}[S_1S_2E] \\
\frac{d[S_1E]}{dt} &= k_{f1}[E][S_1] + k_{r2}[S_1S_2E] - k_{r1}[S_1E] - k_{f2}[S_1E][S_2] \\
\frac{d[S_2E]}{dt} &= k_{f2}[E][S_2] + k_{r1}[S_1S_2E] - k_{r2}[S_2E] - k_{f1}[S_2E][S_1] \\
\frac{d[E]}{dt} &= -k_{f1}[E][S_1] + k_{r1}[S_1E] - k_{f2}[E][S_2] + k_{r2}[S_2E] + k_{cat}[S_1S_2E] \\
\frac{d[S_1S_2E]}{dt} &= k_{f2}[S_1E][S_2] - k_{r2}[S_1S_2E] + k_{f1}[S_2E][S_1] - k_{r1}[S_1S_2E] \\
\frac{d[P]}{dt} &= k_{cat}[S_1S_2E]
\end{aligned} \tag{6.5}$$

Applying the Michaelis-Menten assumption of equilibrium between substrates and substrate-enzyme complexes, the following equations are held true.

$$\begin{aligned}
k_{f1}[S_1][E] &= k_{r1}[S_1E] \\
k_{f2}[S_2][E] &= k_{r2}[S_2E] \\
k_{f2}[S_1E][S_2] &= k_{r2}[S_1S_2E] \\
k_{f1}[S_2E][S_1] &= k_{r1}[S_1S_2E]
\end{aligned} \tag{6.6}$$

The initial enzyme concentration at t_0, denoted $[E]_0$, can be given by the law of enzyme conservation as follows.

$$\begin{aligned} [E]_0 &= [E] + [S_1E] + [S_2E] + [S_1S_2E] \\ \rightarrow [E] &= [E]_0 - [S_1E] - [S_2E] + [S_1S_2E] \end{aligned} \tag{6.7}$$

Combining the equilibrium conditions from Equation 6.6 to 6.7 gives the following equation:

$$[E] = [E]_0 - \frac{k_{r2}}{k_{f2}}\frac{[S_1S_2E]}{[S_2]} - \frac{k_{r1}}{k_{f1}}\frac{[S_1S_2E]}{[S_1]} - [S_1S_2E] \tag{6.8}$$

The following set of equations can be derived from recombining Equation 6.6.

$$\begin{aligned} [S_1S_2E] = \frac{k_{f2}}{k_{r2}}[S_1E][S_2] &= \frac{k_{f1}}{k_{r1}}[S_2E][S_1] \\ [S_1E] &= \frac{k_{f1}}{k_{r1}}[E][S_1] \\ [S_2E] &= \frac{k_{f2}}{k_{r2}}[E][S_2] \\ \rightarrow [S_1S_2E] &= \frac{k_{f1}}{k_{r1}}\frac{k_{f2}}{k_{r2}}[E][S_1][S_2] \end{aligned} \tag{6.9}$$

Substituting $[E]$ from Equation 6.8 into Equation 6.9 gives:

$$\begin{aligned} [S_1S_2E] &= \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}}[E]_0[S_1][S_2] - \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}}\left(\frac{k_{r2}}{k_{f2}[S_2]} + \frac{k_{r1}}{k_{f1}[S_1]} + 1\right)[S_1S_2E][S_1][S_2] \\ \rightarrow [S_1S_2E]\left(1 + \frac{k_{f1}[S_1]}{k_{r1}} + \frac{k_{f2}[S_2]}{k_{r2}} + \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}}[S_1][S_2]\right) &= \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}}[E]_0[S_1][S_2] \\ \rightarrow [S_1S_2E] = \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}} \cdot \frac{[E]_0[S_1][S_2]}{\left(1 + \frac{k_{f1}}{k_{r1}}[S_0] + \frac{k_{f2}}{k_{r2}}[S_1] + \frac{k_{f1}k_{f2}}{k_{r1}k_{r2}}[S_1][S_2]\right)} \\ \rightarrow [S_1S_2E] = \frac{[E]_0[S_1][S_2]}{\frac{k_{r1}k_{r2}}{k_{f1}k_{f2}} + \frac{k_{r2}}{k_{f2}}[S_1] + \frac{k_{r1}}{k_{f1}}[S_2] + [S_1][S_2]} \\ \therefore [S_1S_2E] = \frac{[E]_0[S_1][S_2]}{\left(\frac{k_{r2}}{k_{f2}} + [S_2]\right) \cdot \left(\frac{k_{r1}}{k_{f1}} + [S_1]\right)} \end{aligned} \tag{6.10}$$

The terms $\frac{k_{r1}}{k_{f1}}$ and $\frac{k_{r2}}{k_{f2}}$ in Equation 6.10 respectively represent the dissociation constants for substrate-enzyme complexes S_1E and S_2E. These terms can be interchanged for K_M values in light of the quasi-steady-state assumption made in the Briggs-Haldane kinetics model (See Equation 6.3), giving:

$$\begin{aligned}[S_1S_2E] &= \frac{[E]_0[S_1][S_2]}{\left(\frac{k_{r2}+k_{cat}}{k_{f2}} + [S_2]\right) \cdot \left(\frac{k_{r1}+k_{cat}}{k_{f1}} + [S_1]\right)} \\ \rightarrow [S_1S_2E] &= \frac{[E]_0[S_1][S_2]}{(K_{M2} + [S_2]) \cdot (K_{M1} + [S_1])}\end{aligned} \tag{6.11}$$

Provided with Equation 6.11, the rate of enzymatic reaction, denoted v, can be derived from the ODE of product formation (Equation 6.5):

$$\begin{aligned}v = \frac{d[P]}{dt} &= k_{cat}[S_1S_2E] \\ &= k_{cat}\frac{[E]_0[S_1][S_2]}{(K_{M2} + [S_2]) \cdot (K_{M1} + [S_1])} \\ &= \frac{V_{max}[S_1][S_2]}{(K_{M2} + [S_2]) \cdot (K_{M1} + [S_1])}\end{aligned} \tag{6.12}$$

, where $V_{max} = k_{cat}[E]_0$. In its canonical form supporting an arbitrary number (n) of substrates, the rate equation is

$$v = \frac{k_{cat}[E]_0 \prod_{i=1}^{n} ([S]_i)}{\prod_{i=1}^{n} (K_{Mi} + [S]_i)} \tag{6.13}$$

Equation 6.13 supports irreversible non-modulated non-interacting multi-reactant enzymes. The rate laws expressed by Equation 6.13 and Equation 6.12 respectively correspond to the Systems Biology Ontology (SBO) [42] catalogued terms `SBO:0000150` and `SBO:0000151`, a special case of `SBO:0000150`.

An enzyme database such as BRENDA [34] serves to be a good resource to find out about experimentally verified parameter values for K_{Mi} and k_{cat}. Using the parameter values from BRENDA along with the above derived rate equations, a SBML [91] model encapsulating the riboflavin metabolic pathway in *B. subtilis*, as described in Figures 6.2, 6.3 and 6.4, was built in COPASI [87]. Figure 6.5 summarises the results obtained from simulating the model. It is shown that the

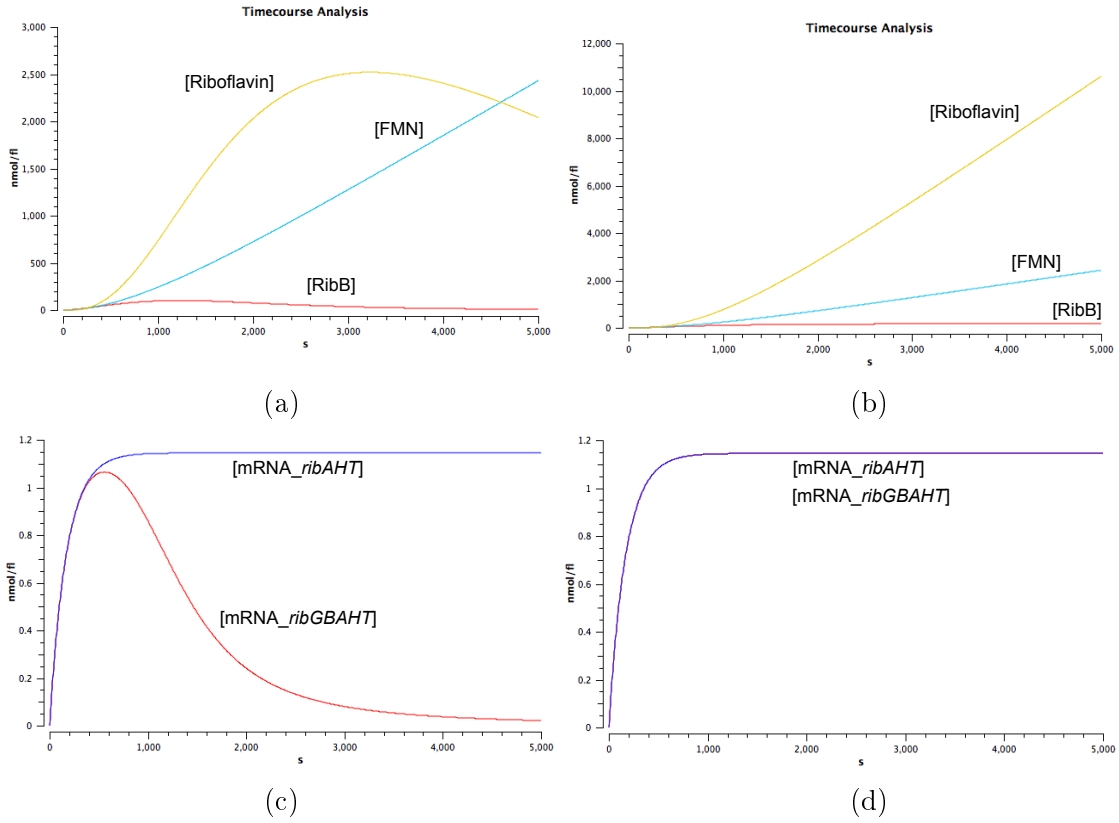


Figure 6.5: SBML models examining flavin concentrations over time

Temporal changes in flavin concentration were examined with or without FMN inhibition on the *rib* operon in *B. subtilis*: **(a)** Timecourse analysis with FMN regulatory feedback enabled, **(b)** Timecourse analysis with FMN regulatory feedback disabled, **(c)** FMN inhibiting the transcription of the *rib* operon corresponding to the result in (a), and **(d)** Uninhibited transcription of the *rib* operon corresponding to the result in (b).

simulation exhibited approximately about a four-fold difference in the riboflavin biosynthesis depending on whether the inhibitory regulation of FMN on the *rib* operon is available or not. The difference in the expression level of of the *rib* operon corresponding to the FMN concentration further explains the effect of such an inhibitory feedback control. Evidently, the use of dynamic modelling discussed here allows detailed investigation of molecular dynamics, otherwise unfathomable. Nevertheless, the computational complexity of dynamic models restricts the number of reactions to be considered together. This makes it extremely challenging to apply dynamic modelling at the genome-scale level.

6.3.3 Constructing a genome-scale static model for metabolic pathways

Static models can offer an alternative perspective to investigating metabolic pathways. Static modelling of metabolic pathways is epitomised by flux balance analysis (FBA). In comparison to dynamic modelling, FBA can tremendously reduce the computational complexity involved in analysing the biochemical system of metabolic networks. The reduction in complexity is attributed to the steady-state conditions, resulting in the mathematical simplification, from considering the biochemical system in equilibrium. This makes it reasonably possible, as opposed to dynamic modelling, to apply FBA for modelling metabolic pathways in the genome-scale level.

FBA defines a metabolic network using the following mathematical formalism:

$$S \cdot v = 0 \tag{6.14}$$

, where S denotes a stoichiometric matrix, v a vector of fluxes through all chemical reactions, each of which fluxes is bound by $v_i^{min} <= v_i <= v_i^{max}$. The value 0 on the right hand side of the equation means that the metabolic network is in steady state. A flux vector v is said to be in the *null space* of S, if the vector provides a solution to satisfy this equation [133].

The biggest advantage of using FBA in modelling cellular systems is probably the fact that the technique relies solely on stoichiometric characteristics without the need of involving difficult-to-obtain kinetic parameters. The stoichiometry of metabolic reactions can even be inferred bioinformatically by sequencing organisms, blasting the sequence to identify proteins, and looking up the proteins on databases such as KEGG, Uniprot, Reactome, or MetaCyc among others. The downside of using FBA is that fluxes in FBA cannot be uniquely specified as a function of regulatory mechanisms, due to the lack of kinetic parameters necessary in determining the concentration of metabolites involved in regulation.

A metabolic network would typically contain more reactions than metabolites. With respect to FBA, this means that there are more variables (i.e. fluxes) to be

Table 6.1: Genome-scale reconstruction of metabolic network in *Bacillus subtilis*

PMID	Strain	Genes	Metabolites	Rxns	Reference	Year
17573341	unspecified	844	988	1020	[130]	2007
19555510	168	1103	1138	1437	[83]	2009

determined than there are equations (whose cardinality corresponds to the number of metabolites), making the problem underdetermined. Underdetermined problems can be analysed by a technique called linear programming which can find optimal solutions that satisfy certain objective functions. FBA's default choice of objective function is the maximisation of biomass. Given such an objective function, a linear solver can find a set of fluxes v leading to the maximisation of biomass, representing the steady-state metabolic conditions of a cell undergoing growth. FBA models should have a virtual reaction specifying the stoichiometry of metabolites (i.e. raw materials) needed in producing biomass. Alternative objective functions can also be used including, for example, minimisation or maximisation of ATP production, energy currency production, or redox potential per unit of glucose [102]. The energy currency involves metabolite species such as ATP and NADPH, while redox potential involves NADH, NADPH, and FADH2. Often, the values for v_i^{min} and v_i^{max} are unknown and set at arbitrary ranges. The lack of precise ranges could affect the predictability of FBA, and it is sometimes needed to fit models against the flux ranges of experimentally measurable reactions, such as those involved in nutrient uptake and excretion.

6.3.3.1 Modelling riboflavin biosynthesis and metabolic pathways in bacteria

Genome-scale metabolic models such as shown in Figure 6.6 enable two possibilities. One is that they open the doors to using an uniform interface in taking a holistic view on the entire metabolic reaction network that covers the biochemistry of both extracellular and cytosolic spaces. The other is that the uniform interface can be manipulated programmatically. This would mean that extensive numerical surveys can be conducted to elucidate the relationships between genes and metabolic pathways, with respect to varying metabolic and nutrient condi-

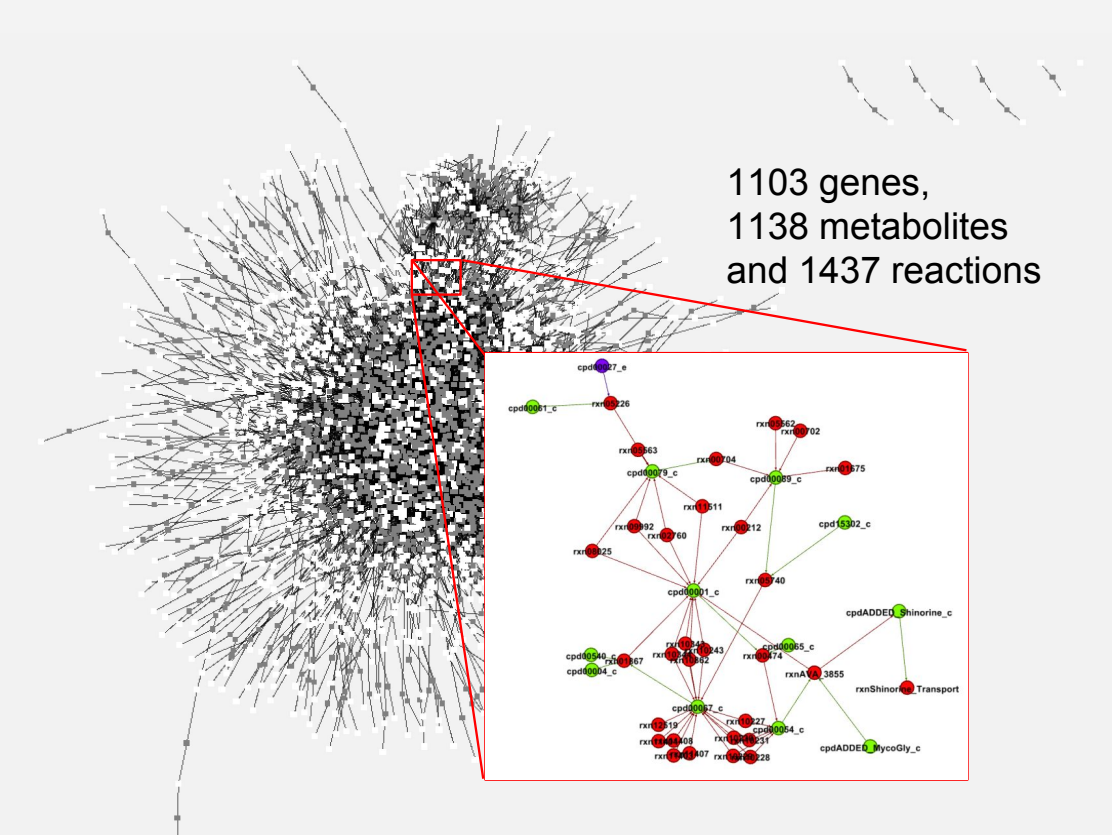


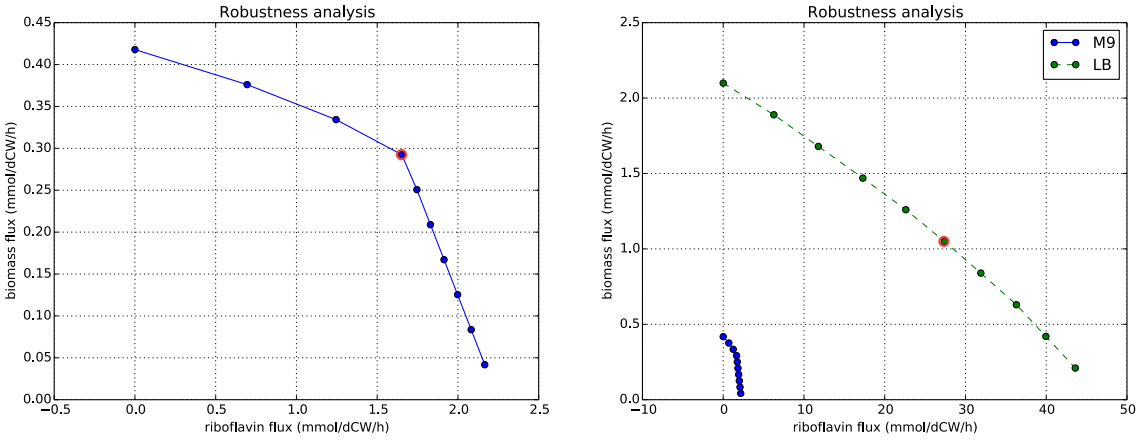
Figure 6.6: Graph-based visualisation of the metabolic network of *B. subtilis* 168
A reconstructed model of the genome-scale *B. subtilis* metabolic pathways from the second entry in Table 6.1 was used in the visualisation. The green and red nodes in the zoomed-in pane represent metabolites and reactions respectively, and the edges represent the participation of connected metabolites in reactions.

tions. As such, FBA together with genome-scale metabolic models offers to be a good framework for systematic investigation of the theoretical limits of metabolic reactions, in the context of DEA.

6.3.3.2 Metabolic pathway model simulation

A proof-of-concept FBA was performed to compare the theoretical limits of riboflavin biosynthesis rates in *B. subtilis* under different nutrient conditions[1]. Fig-

[1]Please refer to Appendix A.1 for details on the experimental setup, code bases, and access to other archived resources.



(a) (b)

Figure 6.7: Applying FBA on the genome-scale metabolic pathway model of *B. subtilis* Flux balance analysis (FBA) was performed on the genome-scale metabolic pathway model of *B. subtilis* 168 (Table 6.1), with minimal media (M9) and rich media (LB), given biomass maximisation as objectives: **(a)** Shown in blue solid line is the result of robustness analysis on the biomass vs riboflavin fluxes in M9. The red-circled dot shows a point where riboflavin production is optimal, with approximate biomass and riboflavin fluxes of about 0.29 and 1.65 $\frac{mmol}{gDW \cdot h}$ respectively. **(b)** Shown in green broken line is the result of robustness analysis on the biomass vs riboflavin fluxes in LB. The red-circled dot shows a point where riboflavin production is optimal, with approximate biomass and riboflavin fluxes of about 1.05 and 27.39 $\frac{mmol}{gDW \cdot h}$ respectively. The M9 result shown in solid blue is the same plot as in (a) at a different scale to match that of LB.

ure 6.7 summarises the results of robustness analysis of biomass versus riboflavin fluxes under minimal (M9) and rich (LB) nutrient conditions. As expected, it was shown that the rich media dwarfs the minimal media in the resulting riboflavin biosynthesis rates as well as the growth rates (Figure 6.7b).

It is noteworthy that the units used here for both the biomass and riboflavin fluxes are in $\frac{mmol}{gDW \cdot h}$, where gDW denotes *grams of dry cell weight*. In general, biomass fluxes are bluntly declared to be growth rates in units of h^{-1} [1] without clear explanations as to how that unit is determined to be so. While, it is true that the flux unit $(\frac{mmol}{gDW} \cdot \frac{1}{h})$ is a rate of a kind, hence the dimension h^{-1}, the fractional term $\frac{mmol}{gDW}$ should not be neglected as it bears a special meaning in interpreting any flux values. This fractional term indicates the amount of metabolite molecules ($mmol$)

partaking in the corresponding metabolic reaction per unit biomass produced in gram weights (gDW). Given the *relative molar mass*[1] of metabolites of concern, mole units can be converted into weights. Such conversion cancels the fractional term as part of the unit and reduces it to a scalar constant adjunct to a rate unit (h^{-1}).

In the example case of riboflavin biosynthesis catalysed by RibB in *B. subtilis*, the metabolic reaction is defined as:

$$2 \cdot \text{DMRL} \xrightarrow{\text{RibB}} \text{RIBF} + \text{ArU} \tag{6.15}$$

, where **DMRL**: 6,7-dimethyl-8-ribityl-lumazine, **RIBF**: riboflavin, and **ArU**: 5-amino-6-ribityl-aminouracil.

When it is claimed that the unit of biomass fluxes is h^{-1}, it would need to be assumed that $\frac{mmol}{gDW} = 1$. What this assumption entails is that the virtual metabolic product of *biomass*, as part of the biomass reaction in a genome-scale metabolic model, needs to be defined with the relative molar mass of $1000\frac{g}{mol}$. This way, the unit conversion of $1mmol = 1g$ holds true for the biomass reaction, to end up reducing the flux unit neatly into h^{-1}.

In order to verify if this was the case with the genome-scale metabolic model discussed here, the mass balance of the biomass reactions defined in that model was assessed in full[2]. The biomass reactions were scrutinised in light of the relative molar mass of all reactants and products defined. Table 6.2 enlists the information on compounds that are part of the biomass reactions defined by the genome-scale metabolic model. The estimated relative molar mass (RMM) for DNA, mRNA, proteins, cell wall, lipid, and lipoteichoic acid compositions (LAC) were calculated based on respective biosynthesis reactions (See Appendix A.6). According to the calculation shown here, the biomass reaction of the genome-scale metabolic pathway model was defined with a RMM value of $1100.78\,\mathrm{g\,mol^{-1}}$, instead of the

[1] The technically more proper term of *relative molar mass* was used in lieu of the more popular term of *molecular weight*, as the old term is in the process of getting "deprecated."

[2] Please refer to the following for the meanings of abbreviated terms used in Table 6.2. **ACP**: Acyl-carrier protein, **RMM**: relative molar mass, **NMW**: normalised molecular weight. Normalised molecular weight is the molecular weight of compounds required in producing $1mol$ of the main product as part of the metabolic reaction, in this case, to produce *biomass*.

Table 6.2: Composition of biomass reaction of *B. subtilis* genome-scale metabolic model: information leading to estimation of biomass' relative molar mass.

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd00001	105	reactant	metabolite	H2O	18.02	1892.1
cpd00002	105.003371	reactant	metabolite	C10H13N5O13P3	504.16	52938.49952
cpd00003	0.01822	reactant	metabolite	C21H26N7O14P2	662.42	12.0692924
cpd00005	0.0002367	reactant	metabolite	C21H27N7O17P3	742.40	0.17572608
cpd00006	0.001053	reactant	metabolite	C21H26N7O17P3	741.39	0.78068367
cpd00010	0.000127618	reactant	metabolite	C21H33N7O16P3S	764.51	0.097565237
cpd00012	0.0008548	reactant	metabolite	H2O7P2	175.96	0.150410608
cpd00018	0.005253	reactant	metabolite	C10H13N5O7P	346.21	1.81864113
cpd00031	0.0002215	reactant	metabolite	C10H13N5O11P2	441.18	0.09772137
cpd00038	0.0004883	reactant	metabolite	C10H13N5O14P3	520.16	0.253994128
cpd00046	0.001153	reactant	metabolite	C9H13N3O8P	322.19	0.37148507
cpd00052	0.0006105	reactant	metabolite	C9H13N3O14P3	480.13	0.293119365
cpd00096	0.0002924	reactant	metabolite	C9H13N3O11P2	401.16	0.117299184
cpd00126	0.0005939	reactant	metabolite	C10H13N5O8P	362.21	0.215116519
cpd00201	0.000206831	reactant	metabolite	C20H21N7O7	471.42	0.09750427
cpd11451	0.00014977	reactant	metabolite	C46H66O2	651.01	0.097501768
cpd00063	0.002983	reactant	metal ion	Ca	40.08	0.11955864
cpd00205	0.6576	reactant	metal ion	K	39.10	25.71216
cpd00254	0.09474	reactant	metal ion	Mg	24.31	2.3031294
cpd10516	0.003209	reactant	metal ion	Fe	55.85	0.17922265
cpd11493	0.000273109	reactant	ACP	C11H21N2O7PS-R	356.33	0.09731693
cpd11461	0.026	reactant	DNA	N/A	955.24	24.83623204
cpd11462	0.0655	reactant	mRNA	N/A	950.99	62.28989327
cpd11463	0.5284	reactant	Protein	N/A	998.67	527.6992058
cpd15664	0.2242	reactant	Cell wall	N/A	997.32	223.5995288
cpd15670	0.0304	reactant	LAC	N/A	996.94	30.3068812
cpd15800	0.076	reactant	Lipid	N/A	995.92	75.69027691
cpd00008	104.9971	by-product	metabolite	C10H13N5O10P2	425.18	44642.66698
cpd00009	104.9866	by-product	metabolite	HO4P	95.98	10076.61387
cpd12370	0.000273109	by-product	Apo-[ACP]	HO-R	17.01	0.004645584
cpd11416	1	product	Biomass	N/A	1100.78	1100.783498

commonly assumed value of $1000\,\mathrm{g\,mol^{-1}}$. It is noteworthy that it would be technically correct to have this kind of aberration accounted for, for example via using normalisation, when interpreting FBA results and their flux values.

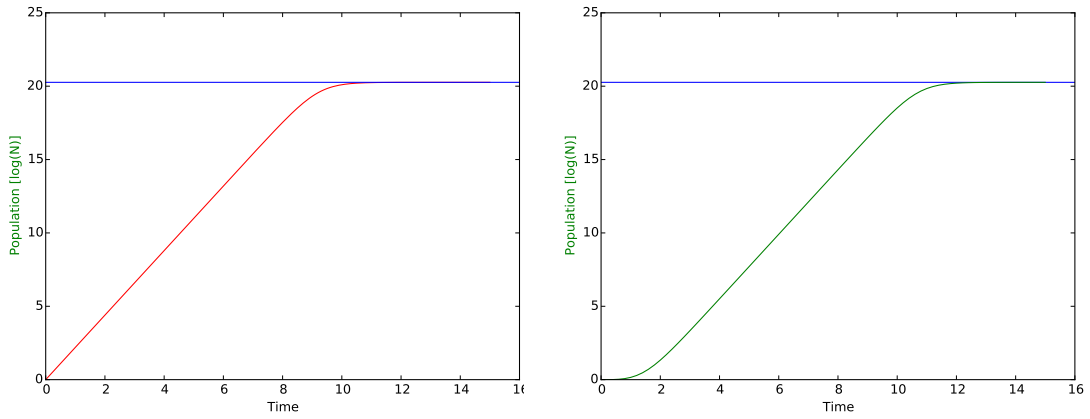
6.3.4 Modelling the dynamics of mutagenesis

Mutagenesis cannot be considered in full by means of simple rate values, as the stochastic process gives rise to complex time-dependent population dynamics among different mutants. In order to gain further insight into the time-dependent dynamics of mutant populations, a realistic bacterial growth model was needed. Generally, the *de facto* mathematical model often discussed for bacterial growth

takes the form of logistic curves, as given by Equation 6.16. N_{max} is the limit of the maximum population, referred to as the carrying capacity of the environment representing an asymptote (See Figure 6.8a).

$$dN/dt = rN(1 - N/N_{max}) \tag{6.16}$$

The pitfall in using the conventional logistic growth model (Eq. 6.16) is that it



(a) (b)

Figure 6.8: Comparison of growth curves by two logistic models

(a) A growth curve (shown in red) exhibited by the logistic model in Eq. 6.16, where $r = 2.2$, and $N_{max} = 10^{8.8}$. The blue horizontal asymptote represents $log(N_{max})$. The logistic model lacks the lag phase of growth typical in bacterial growth. **(b)** A growth curve (shown in green) exhibited by the Fujikawa model (Eq. 6.17), where $r = 2.2$, $N_{max} = 10^{8.8}$, $N_{min} = 0.99999$, and $c = 0.74$. There is a lag phase to the curve before entering the logarithmic phase of growth.

lacks the lag phase of bacterial growth, rendering the model unrealistic at best. Fujikawa and his colleagues introduced a modification to the logistic model to incorporate the lag phase into the growth curve [68] (See Equation 6.17 and Figure 6.8b):

$$dN/dt = rN(1 - N/N_{max})(1 - N_{min}/N)^c \tag{6.17}$$

, where $c >= 0$ is an adjustment factor. Fujikawa model as shown in Equation 6.17 can provide accurate growth patterns of bacterial culture as a whole, and be used to predict the total population level after a certain time has elapsed. However, it cannot be used to predict the population levels of individual mutants sprouting up as a result of growth. A new growth model was needed to accommodate the notion of mutagenesis and to model the dynamics of mutant populations as a function of time. As opposed to Fujikawa model's use of N_{max} as a constant, the new model, dubbed the name *Enveloped growth model*, defines the maximum term as a variable, denoted N'_{max} (See Equation 6.18). N'_{max} is a variable defining the limit (or envelop) of growth of an individual mutant as a result of other competing cells in the culture. The time-dependent nature of N'_{max} is due to the term N_{env}, an integral of the ordinary differential equation of Fujikawa model (Equation 6.17) with respect to time. N_{env} defines the total population of the culture at any given time t_c.

$$\begin{aligned} N'_{max} &= N_{max} - \left(N_{env} - N_m \left(1 - \left(\frac{N_m}{1 + N_{env}} \right)^{(1-P_f)} \right) \right) \\ N_{env} &= \int_0^{t_c} rN(1 - N/N_{max})(1 - N_{min}/N)^c dt \end{aligned} \tag{6.18}$$

, where N_{env} is the integral of Equation 6.17 with respect to time up to the current time point t_c, and P_f is a *congestion factor* used for adjusting the magnitude of dampened growth of a mutant population N_m, and $0 \leqslant P_f < 1.0$. Given N'_{max}, the ordinary differential equation of *Enveloped growth model* for a mutant population N_m is defined as follows (Equation 6.19).

$$\frac{dN_m}{dt} = rN_m \left(1 - \frac{N_m}{N'_{max}} \right) \left(1 - \frac{N_{min}}{N_m} \right)^c \tag{6.19}$$

, where r is a growth rate, $N_{min} > 0$, and c is an adjustment factor proportional to the duration of the lag phase of growth.

Figure 6.10 shows how *Enveloped growth model* (Equation 6.19) improves upon Fujikawa model (Equation 6.17) in that the predicted mutant population does not end up representing 100 % of the total cell population, which would be un-natural if so happened. This so-called mutant population saturation issue associated to

Fujikawa model is evident in Figure 6.9.

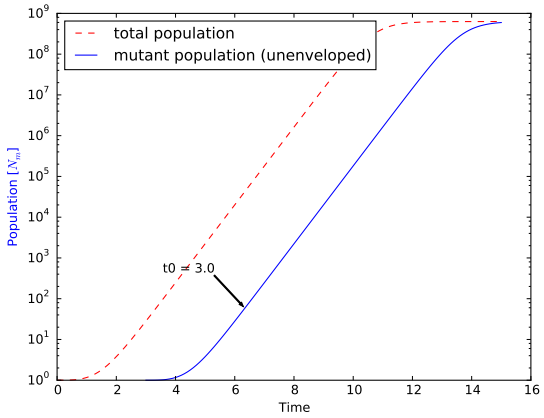


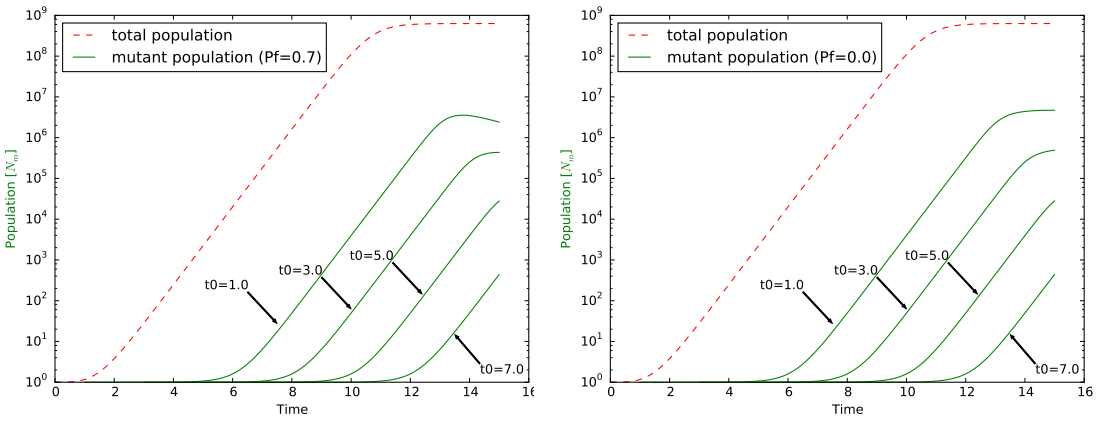
Figure 6.9: Mutant population prediction by applying Fujikawa model as is
The growth curve shown in broken red shows the total population as determined by Fujikawa model in Eq.6.17, where $t0 = 0$, $r = 2.2$, $N_{min} = 0.99999$, $N_{max} = 10^{8.8}$, and $c = 0.74$. The blue growth curve represents a mutant population shown in logarithmic scale, $log(N_m)$, where $t0 = 3$, and was calculated using Fujikawa model (i.e. unenveloped). Due to the lack of enveloping to dampen the mutant growth as its curve gets close to the total population, the mutant population ends up saturating the entirety of the total population towards the end ($t = 15.0$). This saturation problem makes the unenveloped growth model invalid for predicting mutant populations in a realistic manner.

As per Section 5.1.2, the mutagenesis rate (θ) of bacterial cells can, for instance, be defined as 10^{-3} mutation per replication, or a SNP per 1000 cells replicated. Given θ, the number of mutation events N_θ at time t, denoted $N_\theta(t)$, can be predicted by tracking the temporal change in the total cell population:

$$\begin{aligned} N_\theta(t) &= \Delta N \cdot \theta \\ \Delta N &= N(t) - N(t-1) \end{aligned} \tag{6.20}$$

, where N is the total cell population calculable by integrating Equation 6.17 with respect to time. Figure 6.11 shows a plot of $N_\theta(t)$ based on the same Fujikawa growth curve used in deducing the total cell population in Figure 6.10.

Now, $N_\theta(t)$ provides a ground for modelling the population dynamics of individual mutants, by suggesting a firm numerical reference to how many SNP



(a) (b)

Figure 6.10: Dampened mutant populations in *Enveloped growth model*
(a) The growth curves at varying starting time points (shown in green) were calculated using the enveloped growth model (Eq. 6.18), where $r = 2.2$, $N_{max} = 10^{8.8}$, $N_{min} = 0.99999$, $c = 1.0$, and $P_f = 0.7$. The larger value for c results in longer lag phases. Mutant growth is further retarded as it reaches close to the total population, while the degree of retardation is less when the growth level is farther away from the total population level. For example, the curve with $t0 = 7$ at $t = 15$ is dampened less in comparison to the curve with $t0 = 1$ at $t = 15$. **(b)** The growth curves with identical parameter values except the congestion factor, $P_f = 0.0$. The dampening effect due to congestion is milder in comparison to the curves with $P_f = 0.7$.

events may occur at any given time point. Combining the information from $N_\theta(t)$ and the temporal estimation of mutant population from $N_m(t)$ (Equation 6.19), we have all the modelling elements necessary for materialising individual mutants and calculating their contributions towards the total population at any time point.

Let's say a set of $N_\theta(t)$ mutants arising at time t is denoted $S_\theta(t) \in S_m$, where S_m is a set of all mutant sets $S_\theta(t)$ throughout a span of time from t_0 upto the current time t_c. Any individual mutant $m_\omega \in \bigcup S_m$ has the time of inception, t_α, associated with it. At any moment in time $t > t_\alpha$, the mutant population continues to change due to growth. The estimated mutant population N_m at a current time point t_c of an individual mutant that was given rise to at t_α is an

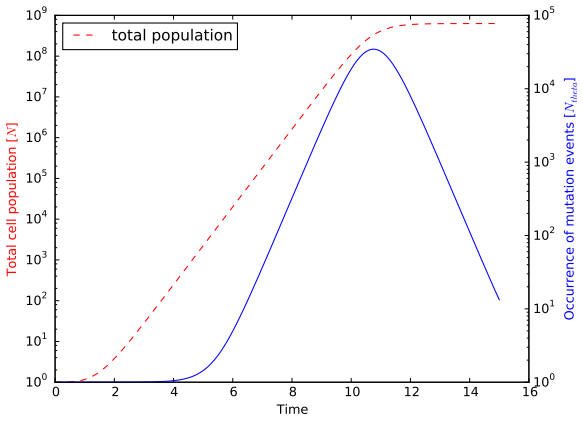


Figure 6.11: Predicting the occurrence of mutation events.
The growth curve shown in broken red shows the total population as determined by Fujikawa model in Eq.6.17, where $t0 = 0$, $r = 2.2$, $N_{min} = 0.99999$, $N_{max} = 10^{8.8}$, and $c = 0.74$. This growth curve serves as the basis for calculating ΔN as shown in Equation 6.20. The blue curve represents the number of occurrences of mutation events as a function of time ($N_\theta(t)$).

integral of Equation 6.19 with respect to a time duration $t_\alpha \leqslant t \leqslant t_c$:

$$N_m(t_\alpha, t_c) = \int_{t_\alpha}^{t_c} rN_m \left(1 - \frac{N_m}{N'_{max}}\right)\left(1 - \frac{N_{min}}{N_m}\right)^c dt \tag{6.21}$$

Any individual mutant $m_\omega \in \bigcup S_m$ has its identity specified by the cardinal variable denoted by $\omega \in \{0, 1, 2, \cdots, N_\Omega\}$, where $N_\Omega = |\bigcup S_m| - 1$. In fact, the mutant with $\omega = 0$ represents a mother cell, or a wildtype cell with zero mutation, hence $m_0 = \varnothing$. The mutant set $\bigcup_{t=0}^{t_c} S_m$ represents all the cell genotypes available in the culture upto the time point t_c, and any cell $m_\omega \in \bigcup_{t=0}^{t_c} S_m$ may serve as a template for a new mutant genotype with the probability defined as:

$$P(X = \omega)(t) = \frac{N_m(t_\alpha, t)}{N_{env}(t)} \tag{6.22}$$

, where X is a random variable such that $0 \leqslant X \in \mathbb{Z} \leqslant N_\Omega$; ω is a positive integer

representing the cardinal variable to specify an individual mutant m_ω, given rise to at t_α; $N_{env}(t)$ is the total cell population envelop from Equation 6.18; and $N_m(t_\alpha, t)$ is the population of mutant m_ω from Equation 6.21. $P(X = 0)$, the probability of a new mutant arising from the mother cell, is defined as:

$$P(X = 0)(t) = 1 - \sum_{\omega=1}^{N_\Omega} P(X = \omega)(t) \tag{6.23}$$

In this mutant population model, a mutant m_ω, once assigned a membership to a cardinal value ω or a genotype, does not change its membership. Such a membership arrangement was based on the assumptions that mutations only happen during replication, and only affect daughter cells. These somewhat naive assumptions, discounting mutagenesis factors other than replication errors, were enough for a proof-of-concept modelling to study a simple mutagenesis dynamics.

The bar graph in Figure 6.12 shows the mean population levels versus SNP counts per cell as a result of running simulations ($n = 4$) according to the novel mutagenesis model discussed so far. Figure 6.10b shows the corresponding mutant growth curves used in the simulations. The simulations were based on a single mother cell grown on 1 mL media for 15 hours under conditions allowing a maximum growth upto about $10^{8.8}$ cells. This cell density is a little less than an OD600 equivalent value of 1.0 which is about $8 \cdot 10^8$ *E. coli* cells/mL according to literature [153]. The results showed that mutants comprise only a small proportion of the entire cell population, and that double SNPs with respect to the mother cell's genotype seem to be the practical limit to the mutagenesis achievable in a single mutant, within the given growth timeframe. The unique mutant count results revealed that the mutagenesis covered on average 4261 different kinds of unique SNPs for the single-SNP mutants, and 444 unique double SNPs for the double-SNP mutants. The number 4261 corresponds to the total number of CDSs available in the *B. subtilis* 168 genome sequence (Genbank accession ID AL009126.3 [105]) used as part of the simulations here. This means that the given mutagenesis timeframe was enough to explore all of the coding sequence basis in the organism of interest, as far as single-SNP mutations are concerned. A spatio-temporal analysis

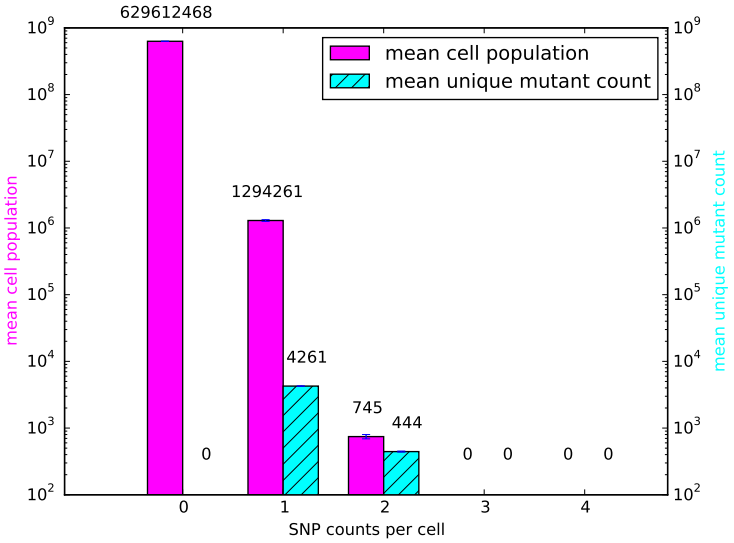


Figure 6.12: Mean populations and unique mutant counts for cells with varying SNP counts

of mutagenesis as a nested function of changing population such as shown here is a novel attempt. This analysis method has a potential to be applied in estimating optimal culture volumes and time lengths required in the evolutionary rounds of the *in vivo* domain, for achieving a targeted SNP count. This estimation can provide the ground for determining an efficient load share between the *in vivo* and *in silico* evolutionary domains.

6.4 Mitigating the curse of combinatorial explosion

6.4.1 The grand challenge

One of the primary challenges in genome-scale design is the lack of effective ways to mitigate the immensity of its combinatorial search space. The size of search space can be estimated by looking at the number of SNPs (μ) out of the total number of genes of interest (Ω). The μ combination of Ω genes, denoted $C(\Omega, \mu)$, is written as:

$$C(\Omega, \mu) = \frac{\Omega!}{(\Omega - \mu)!\mu!} \tag{6.24}$$

Each SNP, for the example case of riboflavin enzymatic pathways, may induce different flux changes in the relevant metabolic pathway step. Let's say a SNP may result in three different effects in the relevant enzymatic flux: flux reduction, flux increase, or no change. The set of possible phenotypic changes in the flux is denoted Φ, and its cardinality $|\Phi|$. Hence, $|\Phi| = 3$ in this example case. For each combination defined as part of $C(\Omega, \mu)$, there are $|\Phi|^{\mu}$ possible permutations with repetition to be considered in going over phenotypic changes in metabolic fluxes. From these numbers, a simple metric to estimate the size of search space (N_X) was given by the following equation:

$$N_X = |\Phi|^{\mu} \cdot C(\Omega, \mu) \tag{6.25}$$

The problem of dealing with immense search space being addressed here is two-fold. Firstly, the size of search space (estimated by N_X) increases exponentially as design complexity (i.e. the length of SNP combinations, μ, under consideration) increases linearly (See Figure 6.13). Secondly, it becomes increasingly more difficult to introduce additional SNPs into a single cell, as evident in Figure 6.12. The former problem cannot be bypassed as there is no way to change the physical nature of combinatorial explosion. The latter is a throughput issue which can be

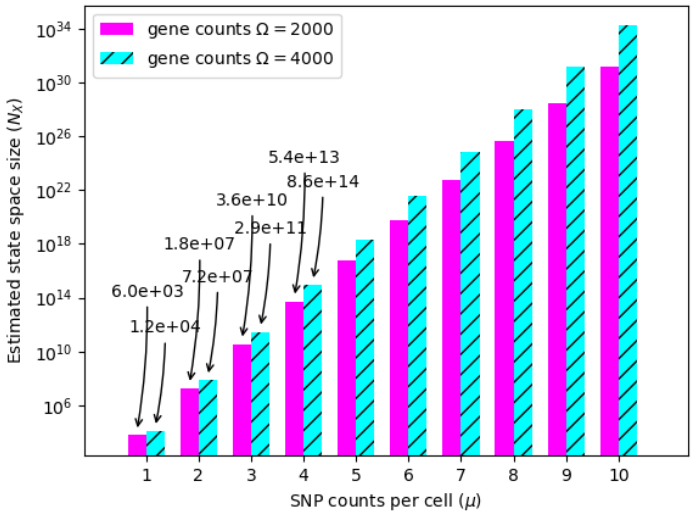


Figure 6.13: Estimation of state space size with respect to SNP counts

The size of state space was estimated in terms of N_X for varying SNP counts (μ) and gene counts (Ω) per cell. The phenotype count of $|\Phi| = 3$ was used in all calculations as per Equation 6.25.

overcome, to a limited extent, by increasing the mutation rate (See θ in Equation 6.20). Nevertheless, such an *in vivo* level countermeasure alone is still far from ameliorating the curse of combinatorial explosion.

6.4.2 Data exchange strategies for the cross-domain interface

The premise of the efficiency gain from harnessing the dual-evolutionary approach requires that the cross-domain gap be bridged. The primary data medium, used by the evolutionary process in the *in vivo* domain, is defined at the genotypic level. This medium, once sequenced and analysed, can be transferred to the *in silico* domain, in the form of SNP data, as explained in Chapter 5. On the contrary, the primary data medium in the *in silico* domain is defined at the phenotypic level, as a function of *in silico* fitness evaluation. The conceptual differences between genotypic and phenotypic elements attribute to the cross-domain gap in merging their

data. The advantage of adopting a genome-scale FBA model, like that detailed in this study, is that there is a model-level support to link phenotypes to genes. The SNP data obtained from the *in vivo* domain can feed directly into the *in silico* fitness evaluation model defined in terms of FBA. The *in silico* domain can stipulate certain constraints on the flux of relevant enzyme(s) (i.e. phenotypes) in the FBA data model to reflect potential implications of having a certain SNP on a gene. Any downstream simulations of the *in silico* model would then be affected by the SNP data from the *in vivo* domain.

Data can flow in the reverse direction as well, going from the *in silico* domain to the *in vivo* domain. A combinatorial exploration, by the *in silico* EA, of applying different flux constraints in the FBA model would result in optimal solutions defined in sets of key enzymatic constraints. Such *in silico* findings can be translated to influence *in vivo* genetic manipulation decisions such as a gene knock-out, or a random mutagenesis scheme targeted to a particular gene of interest. The smorgasbord of genotypic manipulations on key target genes can be included in the construction of the next mother mutant to start the subsequent round of iterations in the dual evolutionary cycle. The suggested cross-domain interface, by means of enabling two-way data exchange, can help close the cross-domain gap.

6.4.3 Bridging the *in vivo-in silico* gap

As demonstrated in Figure 6.13, a higher number of SNP counts can harbour exponentially larger solution space. At the SNP count of 9, for instance, the number of different solutions in the search space already outnumbers the total number of stars in the observable universe [88] which is roughly estimated to be around 10^{29}. At the SNP count of 23, a figure less than treble the former SNP count, the search space can hold as many solutions as there are atoms in the universe [183]. These are numbers rather too large for any search operations to finish going through within a reasonable time span. The *in vivo* search operation, especially considering the throughput limiting nature of the mutation rate θ, a finite amount of culture volume and time given, can only explore so far in the solution space.

Figure 6.12 illustrates this point further by showing that only a little number

of unique double-SNP solutions (444) had been encountered within 15 hours of evolution in 1 mL culture volume. That is to say, only a tiny fraction ($\frac{444}{1.8e7}$) of the available search space was explored. By increasing the volume, let's say by about three orders of magnitude higher, to 1 L culture, the *in vivo* search operation would still have only scratched the surface by covering about 2.5 % of the solution space. Any numbers regarding triple-SNP solutions have not even been part of the discussion yet. Though, it seems probable that double-SNP solutions are the practical limit to the *in vivo* operation, searchable within reasonable amounts of time and culture volume. The obvious question to be asked here, as far as DEA is concerned, is whether it is possible for the *in silico* operation to help speed up the solutions search. If so, how?

The variant analysis pipeline explained as part of the *in vivo* domain can inform the *in silico* EA domain about any non-silent mutations that may provide significant contribution towards meeting the design goal of interest. For the reasons that have just been discussed, this information would be limited to double- or single-SNP solutions. In order to find out if such information can be exploited to reduce the search space of higher order solutions having SNP counts larger than two, a hypothetical solution was considered, out of a total of 2000 hypothetical genes in a cell (i.e. $\Omega = 2000$), based on the following solution criteria.

Table 6.3: The criteria for a hypothetical solution, based on $\Omega = 2000$ and $\mu = 9$

Name	solution SNP count (μ)	hint pool size	unit hint size limit	upfront knowledge (μ_0)
0to9	9	6	2	0

The numbers in Table 6.3 was designed to indicate, to a limited extend, the search complexity of a target solution. The *solution SNP count* (μ) shows how many SNPs any hypothetical solution befitting the criteria may consists of.

The *hint pool size* shows an aggregate total number of SNPs that can be possibly elucidated by the *in vivo* search domain, with respect to a given phenotype of interest. It is a sum of of SNP set sizes, limited by *unit hint size limit*, hypothetically made available by processes, such as variant analysis discussed in Chapter 5. The size limit of 2, here, corresponds to the information limit offered by suboptimal solutions comprising single- or double-SNP(s). The evolutionary domains can potentially use these intermediary, local solutions, to help find higher order so-

lutions, consisting of $\mu >> 2$ SNPs. The *upfront knowledge*, μ_0, shows the amount of information at hand, before the solution search is initiated by either domain. The distance to solution, $\mu - \mu_0$, is a simple factor indicating the likely search complexity.

Table 6.4: The hypothetical solution (iv) used in the study and its intermediary hint pool (i,ii,iii)

Type	μ	solution id:{hypothetical solution}:fitness score
0to9	9	i:{(900,2),(995,1)}:0.2, ii:{(1954,0),(940,0)}:0.2, iii:{(1,1),(100,0)}:0.2, iv:{(900,2),(995,1),(1954,0),(940,0),(998,1),(994,0),(1,1),(100,0),(200,1)}:1.0

Arbitrary choices were made, with respect to satisfying the above criteria, namely '0to9', to come up with a hypothetical solution shown in Table 6.4. The hypothetical solution featured sub-optimal solutions (i, ii, iii) embedded as part of the global solution (iv), each of which was associated to a phenotype from the set $\Phi = \{0, 1, 2\}$ explained earlier, and a fitness score. For example, the sub-optimal solution i, had a double SNP, at genes $\{900, 995\} \subset \{1, 2, \cdots, \Omega\}$, harbouring their respective phenotypes (2 and 1). Each solution's fitness score was given by the values following the second colons in the table. The partial solutions i, ii, iii were given a fitness score of 0.2 each, and the global solution iv was given 1.0. These scores were evaluated in an additive manner, so the maximum fitness scorable by the solution iv would be 1.6, due to the partial solutions embedded within it.

A set of *in silico* experiments were devised to study the conceptual differences between DEA and a single domain framework such as directed evolution, based on a GA developed using Python and DEAP [51] (See Figure 6.14). The following assumptions were made. That characteristic model bacterial organisms such as *E. coli* or *B. subtilis*, when applying directed evolution without intermittent human interventions and the conjugation factor, would be taking an evolutionary strategy, functionally equivalent to a mathematical optimisation method known as Hill Climbing (HC) [95]. Therefore, HC was used as a simulated proxy for directed evolution in this experiments, constituting the *in vivo* half of the DEA framework.

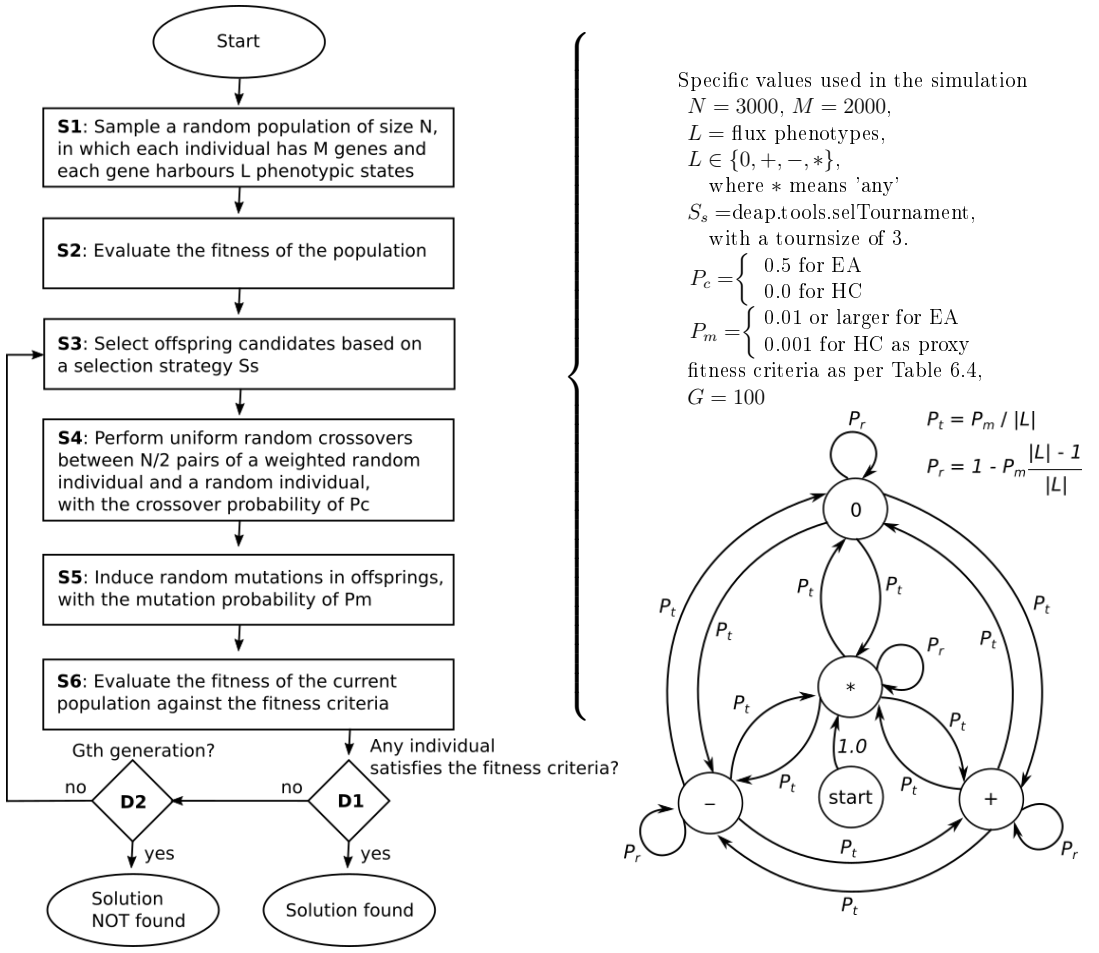


Figure 6.14: Overview of GA steps, decisions, and models used in this study

The left half of the figure is a flowchart showing the steps and decisions taken by the GA used as part of simulating the proof-of-concept test scenario. The specific parameters used in the simulation are shown on the right half. The Markov chain, shown in the lower right corner, provided the mutagenesis policy for phenotypic as well as genotypic state transitions in each gene as a function of the mutation probability P_m. Four phenotypes were used in the study, namely $0, +, -$, and $*$, respectively representing enzymatic flux constraints resulting from mutations: zero flux, positive flux, negative flux, and 'any' (to mean unconstrained). In this Markov chain, two types of transition probabilities, P_t and P_r were defined. P_t was used to specify the probability of any given state to change to a different state, and P_r was used to specify the probability of any given state to stay the same. The GA setup shown here was used for simulating the concept of both Evolutionary Algorithm (EA) and Hill Climbing (HC). The EA simulations had the step **S4** in the flowchart enabled with $P_c = 0.5$, and the HC skipped **S4** by setting the value of P_c, the crossover probability, to zero.

The experimental definition of HC:

```
exp_def_list = {
'HC': {
    'tot_iteration': 1,
    'exp_cycle':[{'seqid': 1, 'domain':'invivo', 'ngen': 100, 'pheno_threshold'
        :1.0, 'geno_threshold':0.9, 'CXPB':0.0, 'MUTPB':0.001, 'domain_error'
        :0.0},]
} }
```

The experimental setup of HC shown here defined the notion of directed evolution, upon using the GA framework depicted in Figure 6.14 as part of simulating the DEA concept entirely *in silico*. One noteworthy parameter setting, in light of directed evolution, was `CXPB`, the crossover probability. This field was set to zero in all HC experiments, by definition of single-cellular systems taking evolutionary strategies in solitude. Likewise, the `domain_error` field was as important a concept to HC as it was critical to have the field switched off in the simulated *in vivo* domain. The `domain_error` parameter was used to account for any uncertainties associated to fitness evaluation *in silico* due to potential knowledge errors in the evaluation system. It would be a reasonable assumption to make that *in vivo* systems do not suffer from errors in their fitness evaluation as a result of poor understanding of cellular dynamics. By virtue of HC being used here as a proxy for an *in vivo* system, the `domain_error` field was set to zero for all HC experiments.

Other parameter settings, including `tot_iteration, seqid, domain, ngen, pheno_threshold`, and `geno_threshold`, bore commonly needed functions and usages, along with the experimental definitions of DEA. For example, `ngen` was used to set the number of generations for which any mutant population was subject to evolutionary changes. The respective parameters of `pheno_threshold` and `geno_threshold` were used to set acceptable fitness levels in the phenotype and the genotype of a mutant.

In light of the experimental setups introduced so far and in Figure 6.14, a data model was designed to support the *in silico* state space representation of a single mutant cell (See Figure 6.15). It was non-trivial, with respect to the research questions being answered, to design an efficient data model supporting the feature completeness of the simulated DEA.

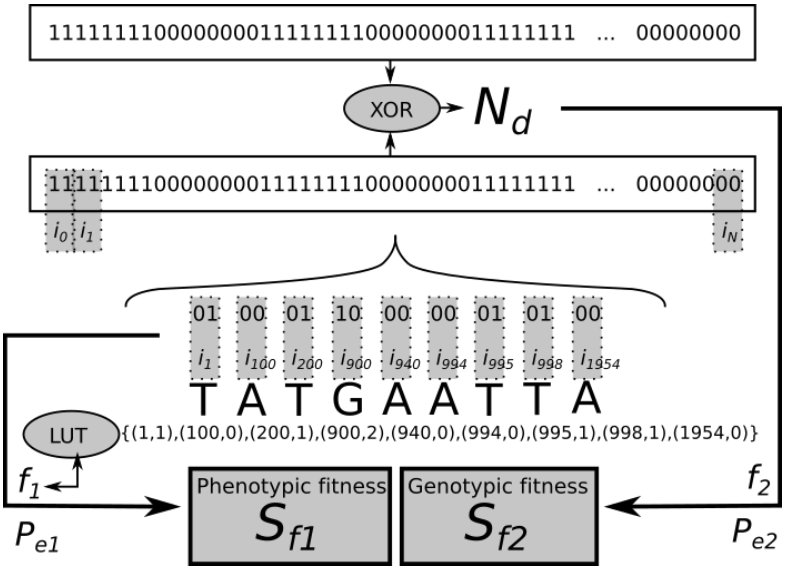


Figure 6.15: The data model used in simulated DEA to represent the state space of a single mutant cell supporting efficient *in silico* fitness evaluation

A 4000-bit-long sequence of alternating eight 1's and eight 0's was arbitrarily chosen as a genome sequence template. N_d denotes the number of mismatching bits against the genome sequence template, obtained from taking XOR between two bitmaps. In this hypothetical genome, a single gene was made of a single nucleotide, represented by using 2 bits. The 4000-bit-long sequence bitmap, therefore, accommodated 2000 2-bit-long genes indexed by $i_0, i_1, ..., i_N$, where $N = 1999$. The look-up-table (LUT) was a list of tuples consisting of a gene index and an integer specifying the phenotypic state of the gene. For simplicity, the bit patterns ($\{00, 01, 10, 11\}$) used for respectively representing four phenotypic states ($\{0, +, -, *\}$) were directly converted to corresponding genotypes and their bit patterns (A = 00, T = 01, G = 10, C = 11). For example, the nine genes making up the pre-agreed solution in LUT, with the gene index values of 1, 100, 200, 900, 940, 994, 995, 998, and 1954, were respectively associated to T, A, T, G, A, A, T, T, and A, and were respectively mapped to the phenotypic state combination $[+, 0, +, -, 0, 0, -, -, 0]$. The bitmap of a mutant can be compared to LUT and the genome sequence template to determine the mutant's phenotypic fitness (S_{f1}) and genotypic fitness (S_{f2}). The respective fitness evaluation functions, denoted f_1 and f_2 can accommodate random errors specified by P_{e1} and P_{e2}.

As depicted in Figure 6.15, fitness evaluation was performed with respect to the two functions f_1 and f_2, each of which respectively calculates the phenotypic (S_{f1}) and the genotypic (S_{f2}) fitnesses. The fitness calculations can accommodate random errors occurring with probabilities P_{e1} and P_{e2}, or P_e, where $P_{e1} = P_{e2}$.

Each mutant was represented with a bitmap data structure that can minimally encode the state space, using two bits per gene or phenotype (cf. $|L| = 4$, where $L \in \{0, +, -, *\}$). For $\Omega = 2000$, it required 4000 bits to encode 2000 genes (cf. $i_N = 1999$ in the bitmap) per mutant. Performing the bitwise XOR operation on the bitmap data structure representing each mutant against two answer templates enabled the fitness evaluation for both S_{f1} and S_{f2}. The genome template, serving the purpose of a hypothetical organism's genome sequence, was arbitrarily chosen to be an alternating sequence of eight 1's and eight 0's, as depicted above. The genotype to phenotype conversion, with respect to having the hypothetical solution (LUT) encoded into this genome sequence, was done by assuming a direct one-to-one match between the genomic and the phenotypic state spaces. For example, each gene out of 2000 was assumed to consist of a single nucleotide (i.e. A, T, G, or C). As far as the current study was concerned, the arbitrary genes 1, 100, 200, 900, 940, 994, 995, 998, and 1954 were the only ones with any significance in their specific phenotypes, as per LUT constructed from the pre-agreed answersheet in Table 6.4. For the sake of simplicity, the four single nucleotides of their genes were directly mapped to their four phenotypic states respectively representing $L \in \{0, +, -, *\}$. The LUT and the total gene count, given by Ω, formed the basis of calculating S_{f1} and S_{f2}. S_{f1} was a simple look-up-table (LUT) operation against the answersheet values, and $S_{f2} = \frac{2\Omega - N_d}{2\Omega}$, where N_d denotes the number of mismatching bits against the genome sequence template.

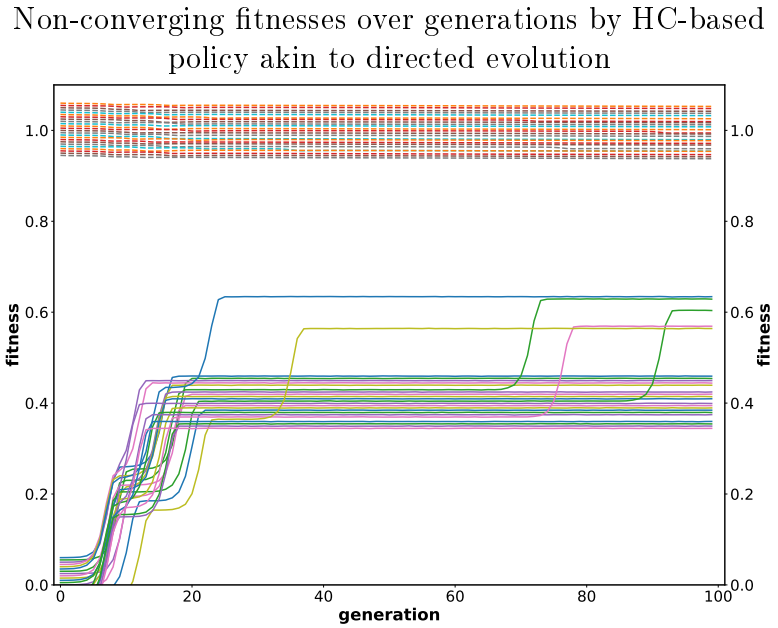


Figure 6.16: 24 out of 24 mutant colonies failed to achieve the phenotypic objective, by relying on an evolutionary strategy based on HC

The mean fitness scores of 24 mutant colonies were plotted over 100 generations. Each mutant colony consisted of 3000 mutants that evolved based on the HC strategy. The solid lines on the bottom half of the plot represent phenotypic fitness scores S_{f1}, and the broken lines on the top part of the plot represent genotypic fitness scores. The mutant populations' genotypic integrity was maintained to a high standard, but their phenotypic fitnesses failed to reach the evolutionary objective. The `pheno_threshold` was set at 1.0, and the `geno_threshold` at 0.9.

Figure 6.16 shows the result of HC experiments conducted on 24 mutant colonies, each consisting of 3000 mutants evolved for 100 generations. These were proxy experiments in lieu of directed evolution *in vivo*. All the mutant colonies taking the evolutionary strategy provided by HC failed to achieve the phenotypic objective (i.e. the phenotypic fitness level above `pheno_threshold`). They were all dwelling on the local optima of high genotypic fitness levels, driven by `geno_threshold`.

The experimental definition of error-free DEA:

```
exp_def_list = {
'DEA': {
    'tot_iteration': 1,
    'exp_cycle':[
        {'seqid': 1, 'domain':'invivo', 'ngen': 30, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.0, 'MUTPB':0.001, 'domain_error':0.0},
        {'seqid': 2, 'domain':'insilico', 'ngen': 40, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.5, 'MUTPB':0.01, 'domain_error':0.0},
        {'seqid': 3, 'domain':'invivo', 'ngen': 30, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.0, 'MUTPB':0.001, 'domain_error':0.0},
    ]
} }
```

The experimental definition of DEA assuming a probability of 0.3 for random *in silico* fitness evaluation error:

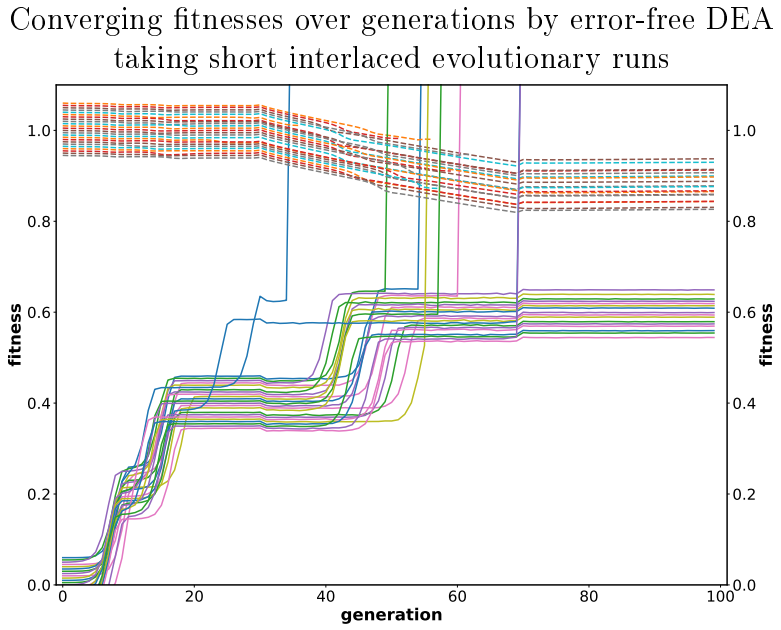
```
exp_def_list = {
'DEA_err': {
        'tot_iteration': 1,
        'exp_cycle':[
        {'seqid': 1, 'domain':'invivo', 'ngen': 30, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.0, 'MUTPB':0.001, 'domain_error':0.0},
        {'seqid': 2, 'domain':'insilico', 'ngen': 40, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.5, 'MUTPB':0.01, 'domain_error':0.3},
        {'seqid': 3, 'domain':'invivo', 'ngen': 30, 'pheno_threshold':1.0, '
            geno_threshold':0.9, 'CXPB':0.0, 'MUTPB':0.001, 'domain_error':0.0},
        ]
} }
```

Different to the HC experimental definition discussed earlier, the two simulated DEA experiments had the experimental cycle (`exp_cycle`) defined in multiple steps. Each step was given a `seqid` value for noting the sequential order of its execution. In the three steps defined for both DEA categories, the `domain` field was used to define alternating evolutionary strategies, switching from taking the HC, to taking the EA, and back to taking the HC again. This alternating `domain` parameter setup was for simulating the iterative design cycle in the dual domains of the DEA framework. Also, the `MUTPB` field was set to alternate between 0.001

and 0.01. This was done to reflect in the simulations the flexibility of the *in silico* domain in setting the mutation rate or the rate at which different solutions are explored. A higher mutation rate would enable a faster exploration of diverse mutants in a population at the expense of the stability of desirable mutants. Biological systems in nature have evolved to put a hard limit on increasing this value, as high mutation rates can pose an immediate threat to the functional integrity of their genome base, limiting their survival chances. The primary objective of biological systems is to maximise the chance of survival. The primary objective of the *in silico* domain, in the context of DEA, is to find solution(s) conferring desired phenotypes as quickly as possible. The survival of an organism of interest, as far as the *in silico* domain is concerned, would only be secondary an objective.

The capability of efficiently making crossover mutations is another useful feature offered by the *in silico* domain. Crossover mutations could unlock hidden solutions via a mechanism known in the mathematical optimisation community as *implicit parallelism*. The value of 0.5 was used for `CXPB` on all *in silico* simulations. The error-prone DEA was defined with a probability of random errors in the *in silico* system's assessment of fitness scores. This experimental setup was introduced to serve not only as a control, but also as a main comparison point to assess the potential feasibility of a real-world DEA system. This experimental setup along with the corresponding results presented in Figure 6.18 formed the basis of discussing how errors in *in silico* fitness evaluation may affect the integrity of the DEA framework.

Figure 6.17 shows the result of simulated DEA experiments conducted on 24 mutant colonies, each with 3000 mutants evolved for 100 generations. In the DEA and the HC experiments, the duality of two fitness evaluation categories, phenotypic and genotypic fitnesses introduced in Figure 6.15, was acting as a push-pull driving force for directing mutations. The mutually conflicting dual objectives biased selection events for directing mutants to acquire mutations that would provide an optimal balance between the two objectives. The two types of evolutionary pressure acting in opposite directions had more pronounced effects in DEA experiments than they did in HC experiments. It can be seen in Figure 6.17 that the phenotypic objective affected mutants to take upward trajectories in phenotypic fitnesses at the expense of decreasing genotypic fitnesses, while being



(a)

Figure 6.17: 7 out of 24 mutant colonies succeeded in achieving the phenotypic objective, based on error-free DEA

The mean fitness scores of 24 mutant colonies, each consisting of 3000 mutants, evolved based on the error-free DEA strategy, were plotted over 100 generations. The mutant populations' genotypic integrity was somewhat challenged but was successfully maintained, while resulting in 7 phenotypically successful colonies.
The solid lines on the bottom half of the plot represent mean phenotypic fitness scores S_{f1}, and the broken lines on the top part of the plot represent mean genotypic fitness scores S_{f2}. The threshold values of 1.0 and 0.9 were respectively set for `pheno_threshold` and `geno_threshold`.

resisted by the genotypic objective.

Also evident was the different mode of evolutionary trajectories exhibited by DEA in comparison to the results from HC. The modal change occurred when mutant colonies switched from the simulated *in vivo* domain to the *in silico* evolutionary domain in generation 30 (See Figure 6.17). The tug of war between the two fitness criteria started tilting towards favouring phenotypic changes over genomic stability. This was indicated by the fast paced decline in the genotypic fitnesses and by the sudden appearances of fast converging mutant colonies making steep turns upwards in the fitness landscape, between the generations of 30

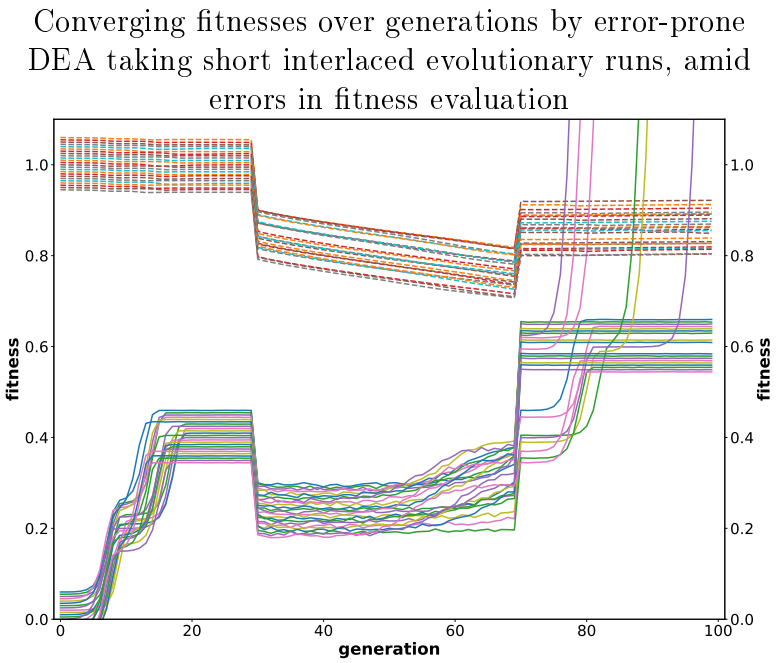


Figure 6.18: 6 out of 24 mutant colonies succeeded in achieving the phenotypic objective, based on error-prone DEA

The mean fitness scores of 24 mutant colonies, each consisting of 3000 mutants, evolved based on the error-prone DEA strategy, were plotted over 100 generations. Fitness evaluation errors, with a random occurrence probability of 0.3, were plaguing the *in silico* evolutionary decisions. The mutant populations' genotypic integrity was severely challenged but was impressively defended and maintained, while also resulting in 6 phenotypically successful colonies. The solid lines on the bottom half of the plot represent mean phenotypic fitness scores S_{f1}, and the broken lines on the top part of the plot represent mean genotypic fitness scores S_{f2}. The threshold values of 1.0 and 0.9 were respectively set for `pheno_threshold` and `geno_threshold`.

and 70. This period coincided with the generations governed by the evolutionary strategies of the *in silico* domain, as specified to be so in the experimental definition of the simulated DEA. It appears that the *in silico* domain, in contributing to find converging mutants, took a full advantage of the higher mutation rate as well as crossover mutations. The accelerated mutant maneuver went back to the initial conservative characteristics of HC at generation 70, but only after optimal converging solutions had been found by multiple mutant colonies.

Figure 6.18 shows the result of experiments on simulated DEA with the pres-

ence of random errors in *in silico* fitness evaluation. The two mutually conflicting fitness criteria were acting against each other just like in the previous two cases. The increase in the mutation rate upon entering the *in silico* domain was reflected on the accelerated decline in the genomic fitness by all mutant colonies. The *in silico* fitness evaluation error, due to the domain switch at generation 30, had severely affected the fitness scores, causing huge drops among all mutant colonies between generations 30 and 70. The sudden drops in mean fitnesses happened in both genotypic and phenotypic categories. At first, it seemed as though the fitness evaluation error only had adverse effects on both fitness categories in the short run. However, the error must have also induced some positive effects on some mutant colonies, perhaps by nudging them out of stagnation, or by forcing them to take uncharted evolutionary trajectories. After all, the *in silico* domain seemed to have primed mutants in such a way that some colonies started achieving the phenotypic objective shortly after switching back to the *in vivo* domain in generation 70. The accelerated mutant maneuver went back to the initial conservative characteristics of HC, as expected. However, the error-prone DEA exhibited remarkable resilience not only in terms of error recovery but also in its success in driving mutant colonies to achieve the dual objectives in the hands of the *in vivo* step supposedly taking a conservative evolutionary stance. Given the fact that a real-world DEA would most likely suffer from errors in *in silico* fitness evaluation, the results shown here further corroborated the idea that DEA can be useful in facilitating *in vivo* solution search.

6.5 Discussion

This chapter covered a wide range of ideas with respect to the focus on their modelling aspects. Be it concepts from *in silico* or *in vivo* domains, the emphasis was given to their modelling aspects.

A comprehensive model was developed to convey many of the concepts comprising the DEA framework. The plausibility of the DEA framework was investigated using the simulation of the model. The simulated DEA resulted in converging mutant colonies. This can be interpreted as the framework's capability of finding solutions within the limited timeframe. This result was contrasted by the simu-

lated directed evolution not beig able to solely resolve any solutions given the same timeframe. These results were in agreement with the idea that hiring the *in silico* domain can increase the search efficiency of the *in vivo* domain, or vice versa, like that envisioned by DEA. Given the assumption that time to solution is positively correlated to design complexity, this study successfully corroborated the idea that the DEA framework offers to be an effective means to mitigate design complexity.

Throughout the study involving the *in silico* domain, many novel methods were developed. These include the enveloped growth model (Equation 6.19) and the novel way of analysing the spatio-temporal population dynamics of mutants. These methods can potentially be used in the estimation of parameters such as optimal culture volume and evolution time, crucial to *in vivo* experiments in minimising evolutionary overheads. Minimising evolutionary overheads as such is directly relevant to reducing the time to solution, to closing the cross-domain gap, and to the capability of dealing with complex designs.

With hindsight, the gap between the dichotomous evolutionary domains was bridged in various angles. The use of variant analysis to make the information exchange compatible between the two domains was one such angle. The decision to adopt the genome-scale FBA, as a suggested means for *in silico* fitness evaluation, was for addressing the primary concern of using a model as closely compatible to SNP data as possible. This suggestion was driven by the thought that the adoption of *in silico* models closely resembling the real world would be the only effective way to minimise errors that may potentially plague the cross-domain interface to end up exacerbating the gap. Without doubt, the point being made was rational. However, the error-prone DEA experiments cast an interesting view point with respect to maintaining the cross-domain integrity, amid errors in *in silico* fitness evaluation. It is a view point suggesting that *in silico* errors can be effectively mitigated, in a more systematic manner, via *in situ* corrections provided by the *in vivo* evolutionary domain. The results presented in Figure 6.18 hinted at this possibility, where the error-prone DEA was able to cope with *in silico* fitness evaluation error, upon switching the domain to the *in vivo* counterpart. This is suggestive of the idea that the resilience of the *in vivo* domain may render the cross-domain gap a non-issue, at least to a certain extent.

Now, these discussions about seemingly eclectic subject matters involving evo-

lution and design appear excessively intertwined. Yet, they also converge at the focal point of achieving design automation. One principle adamantly held up throughout this study was concerning automation, such that methodological choices deemed unfit for automation were simply ruled out. Automation is perhaps the single most important element in increasing the data exchange throughput between the two domains. The throughput increase would in turn contribute towards bridging the gap from whole another angle, let alone the contribution it would have on reducing the time to solution. In Chapter 7, the discussion will shift gears more towards bolstering the automation aspect of realising the DEA framework, via introducing a series of ideas underpinning single-cell-level microfluidics. These microfluidic techniques certainly attributed towards closing the gap from a perspective different to what had been discussed so far.

Chapter 7

Developing microfluidics-based platforms for automating single-cell-level phenotypic measurements

7.1 Introduction

Synthetic biology is aimed at developing organisms with novel functionalities. As part of realising this aim, DEA was suggested [80], and its proof-of-concept implementation ideas were tested in Chapter 5 and 6. In the DEA framework concerning phenotypic characterisation, however, there still exists a disconnect between the two domains in the properties perceivable by measurement. Such discrepancy arises, in part, from the incompatibility of phenotypic data obtainable in the two domains. For instance, measurable properties in the *in silico* domain are often modelled on a single-cell basis as shown in Chapter 6, whereas actual measurements in the *in vivo* counterpart are usually taken on populations of cells due to limitations in conventional lab techniques. Colony- or population-level measurements may end up obscuring the identification and characterisation of rare, stochastic cellular events occurring at single-cell levels [163]. Single-cell based *in vivo* measurement and analysis techniques are needed to close this gap.

While flow cytometry can offer single-cell-level measurements, it is limited to measurements at a single time point, and therefore is incapable of measuring single-cell-level events in a time series. Studies claiming time-series measurement of single cells using flow cytometry [162] are, in fact, based on time-series measurements of populations of cells and reporting single-cell-level statistics by calculating the population mean values.

Synthetic biology and systems biology have constantly increasing needs for true single-cell-level time-lapse measurements, as temporal variation is integral to cellular properties. Microfluidics is promising in terms of developing novel measurement systems that can measure the temporal dynamics of single-cell-level cellular events, prone to cancelling out in population based measurement systems. Photolithography techniques [191] as well as programmatic image analysis [81, 149] underpin the development of sub-micron scale microfluidic systems for carrying out time-lapse measurements at fine granularities. Synthetic biology, especially with respect to DEA, is an area of research that can greatly benefit from this kind of novel measurement systems.

However, the current practice of designing single-cell-level microfluidic devices relying on designers' ad-hoc intuition is subject to expensive trial-and-error cycles. It is not unusual for human designers to mistakenly apply their understanding of the macroscopic world to the microfluidic world where liquid behaves rather strangely due to the physics of low Reynold number conditions [24]. A model-based approach [146] to designing microfluidics devices would help reduce such human errors. Working with models before finalising device designs not only improves the validity of designs but also reduces the overall production cost.

Discussed in the following are a number of microfluidic designs that support single-cell-level measurements. The purpose of presenting these design examples are mainly two fold. Firstly, they exemplify the usecases of single-cell-level microfluidics to assist with design automation in light of DEA in synthetic biology. Secondly, they show the successful application of model-based engineering of microfluidic devices.

7.2 Fabrication of sub-micron features

One of the biggest challenges in this work in producing microfluidic designs capable of single-cell level analysis was achieving sub-micron sized features as part of the fabrication process. The fabrication of submicron-level, high-resolution features can be directly achieved by using electron-beam lithography (e-beam). However, applying e-beam is not cost-effective for lengthy and repetitive feature types spanning a large surface area, due to low-throughput sequential e-beam head movement [178]. A photolithography method along with etching was successfully applied in building lengthy and repetitive sub-micron features, in the fabrication of the single-cell chemostat by Cluzel's group [123]. While the use of photolithography was more suitable for the feature types this work involved, the minimum feature width reliably attainable by photolithography in the multiple foundries consulted for the work was only about 2.0 µm, due to the diffraction limited nature of their photolithography setup. Apart from the need for achieving submicron feature widths, the microfluidics design introduced here also required a uniform feature depth of 1.5 µm across all small features. Etching is a common technical choice in the fabrication process to attain these features [89, 123]. However, it was not possible to reliably control etching to terminate at a depth of 1.5 µm in practice without rounds of costly optimisation runs. The etching rate can vary depending on how exposed the surface area of a feature is due to etching anisotropy [89], leading to variable etching depths - the wider the feature width, the higher the etching rate.

The photolithography method used in fabricating the single-cell chemostat [123] was adapted to develop a new method as illustrated in Figure 7.1. The use of Deep Reactive Ion Etching (DRIE) along with SiO_2 was desirable for meeting the design requirements in this study. Using DRIE, the etching process can be controlled to be substrate specific. For example, the etching can be selective only towards SiO_2, or towards Si, depending on the choice of a gas mixture. Given such control and the use of SiO_2, instead of Si, as the substrate of choice for microfluidic structures, it was immediately possible to reliably etch the depth of 1.5 µm without optimisation, regardless of feature widths. The etching depth control was achieved by depositing a 1.5 µm silicon oxide layer on silicon wafers (Figure 7.1a **step 1**).

The SiO_2 deposition is much easier a task than directly controlling the etching depth of Si.

Oxide backfill was used as a measure for improving upon the 2.0 µm resolution limit arising from the use of contact photolithography (Figure 7.1a **steps 11 and 12**). The combination of these methodological choices, as described further in the following, enabled the fabrication of lengthy submicron features in a manner that is not only cost-effective, but also reliable. The microfluidics fabrication process consisted of three main parts, and was performed by Lionix BV, The Netherlands. The first part was the lithography of small features having a minimum feature width of 2.0 µm, and a uniform feature height of 1.5 µm. The second part was the lithography of large features having feature widths between 50 µm and 100 µm, and a feature height of 40 µm. Lastly, the third part was the backfill steps for reducing the 2.0 µm minimum feature width to a submicron level.

7.2.1 Lithography of small features

Thermal oxidation was used to introduce a 1.5 µm layer of SiO_2 on 4-inch silicon wafers (See Figure 7.1a **step 1**). The wafers were spin coated with photoresist Fujifilm Oir 906/12, at a thickness of 1.2 µm and were soft baked (**step 2**). The photoresist layer acts as a protective layer from DRIE. For each wafer, a small-feature photo mask was overlaid on the photoresist layer using a vacuum to tighten the contact (**step 3**). The wafer was then exposed to UV light (**step 4**), and developed (**step 5**). A post-exposure hard baking was performed in order to densify the developed photoresist on the wafers (**step 6**). DRIE was performed to etch small features using a CHF_3 and O_2 gas mixture, (**step 7**). After exhausting the gas chamber (**step 8**), O_2 plasma followed by HNO_3 gas was used for stripping the photoresist layer (**step 9**). The gas was exhausted afterwards. (**step 10**).

7.2.2 Lithography of large features

The lithography of large features was done by repeating steps **2** through **10** (Figure 7.1a) as in the lithography of small features, using a variation of the photoresist and gas mixtures. In **step 2**, a Fujifilm Oir 908/35 layer was spin-coated at the thickness of 3.5 µm. In **step 3**, a separate photo mask for large features was used.

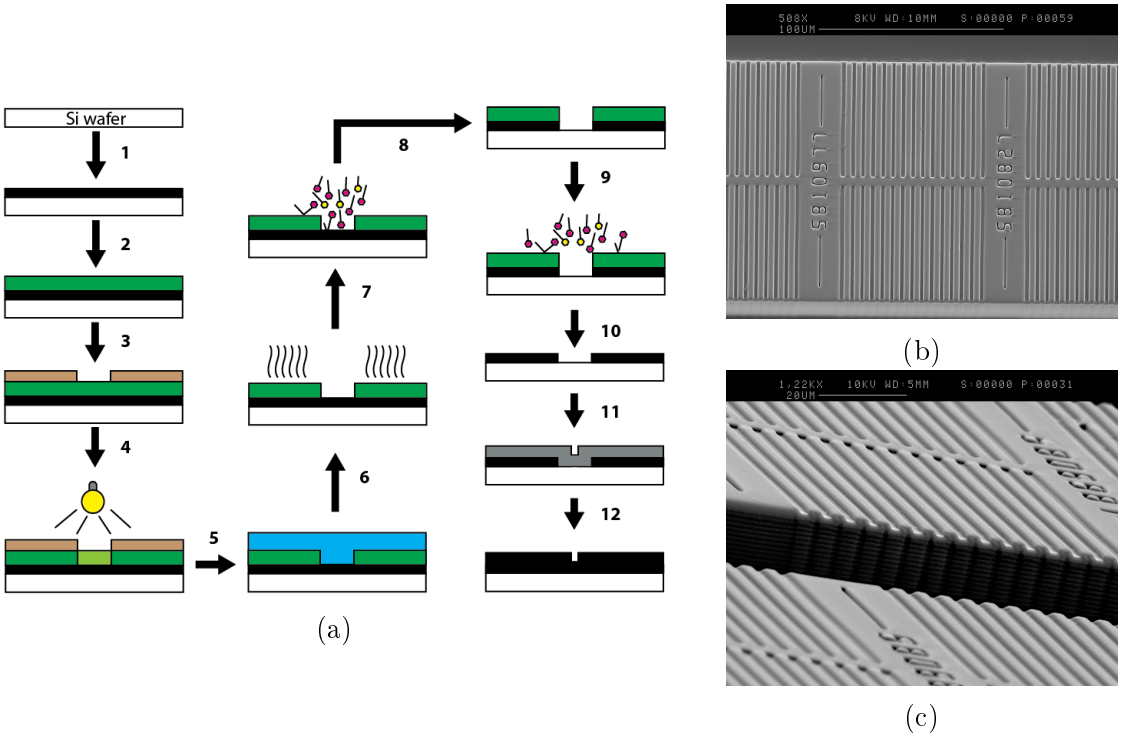


Figure 7.1: Fabrication protocol of submicron-scale microfluidic structures
(a) Fabrication processes: **1**. Thermal oxidation for introducing 1.5 µm of SiO_2 **2**. Spin coating of photoresist, followed by soft baking. **3**. Contact overlay of the photo mask (shown in brown) using vacuum. **4**. Exposure to UV light. **5**. Development **6**. Hard baking to densify the developed photoresist **7**. DRIE of the oxide layer. **8**. Exhaust **9**. O_2 plasma followed by 100% HNO_3 gas for stripping the photoresist layer. **10**. Exhaust **11**. Deposition of TEOS for 0.8 µm backfill. **12**. Thermal treatment for conversion to SiO_2 (b) A low magnification SEM image from the top of a chip fabricated using this protocol. (c) A high magnification SEM image of the chip at a tilted angle. The dark cavity feature is 40 µm deep, and the small trench features are 1.5 µm deep.

In **step 7**, a gas mixture of C_4F_8 and CH_4 was applied for stripping the 1.5 µm layer of SiO_2, and a mixture of C_4F_8 and SF_6 for etching 40 µm deep cavities into the silicon material (See Figure 7.1b,7.1c).

7.2.3 Oxide backfill for reaching submicron resolutions

Tetra-ethyl orthosilicate (TEOS) was deposited on the wafers at a thickness of 0.8 µm, reducing the overall feature widths by a total of 1.6 µm (**step 11**). For

example, a 2.0 µm wide feature would become 0.4 µm wide. The TEOS deposition does not affect the depths of features. The deposited TEOS was then thermally treated to be converted into SiO_2 (**step 12**).

7.3 Microfluidics for single-cell-level measurement and analysis

Discussed in the following are designs for microfluidics to support the measurement of cellular properties at the single-cell level. These designs exemplify microfluidic elements to form the basis of building automated measurement and analysis systems. Automation at this level will play vital roles in closing the gap between the *in vivo* and *in silico* domains discussed as part of the DEA framework, and in achieving the ultimate goal of design automation in synthetic biology.

7.3.1 Growth rate vs marker fluorescence chip

The microfluidic designs shown in Figure 7.2b and 7.2c have structural support for promoting bacterial growth in the direction of channels. Together with a software component to automate the microscopy and image analysis, they can be used in measuring single-cell-level growth rate and fluorescence as part of the DEA framework.

A silicon wafer fabricated using the method in Figure 7.1 was cast with Polydimethylsiloxane (PDMS) to make intermediate moulds. The intermediate moulds were used, in turn, to cast agarose gel slabs replicating the imprints of the submicron channels of the initial wafer. The gel slabs and their channel imprints were used to trap and grow bacterial cells. The gel slabs were encased in a PDMS chamber to which defined growth media were programmatically delivered via a pneumatic pump control unit. The agarose-based microfluidic chip can accommodate a large number of uniquely labelled locations, each of which hosts the growth of a small number of cell clusters. Figure 7.11e shows one of many such locations in a single microfluidic chip at two different time points t_0 and t_1. A motorized microscope stage (Nikon Ti-E) and motorized shutters (Sutter) were programmatically controlled by using the Micro-Manager core API [58], in order

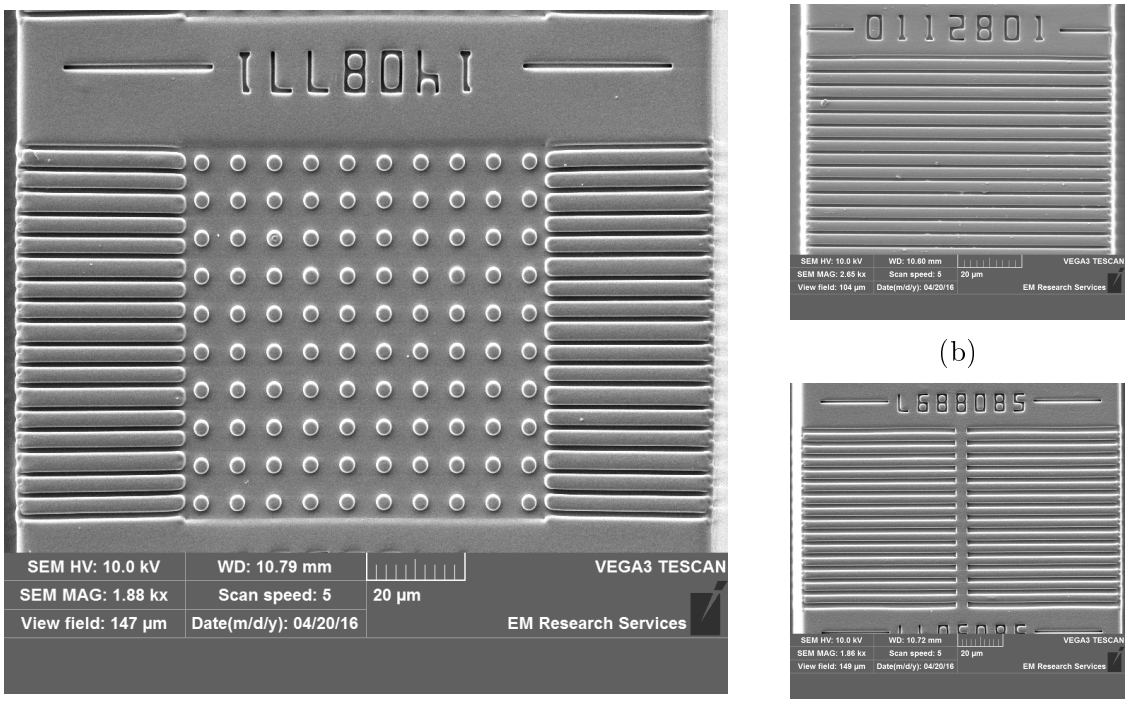


Figure 7.2: SEM images of agarose-based microfluidics chip designs
Agarose-based microfluidics chips were designed for **(a)** investigating biofilm formation, and for **(b,c)** growth rate vs marker fluorescence measurement.

to acquire multi-dimensional (bright field and fluorescent field) images in multiple locations and time points. The images were processed via a sequence of steps to determine cell boundaries, as shown in Figure 7.11f. In this case the processing steps involved the Fiji commands, `sharpen`, `contrast`, `log`, `contrast`, `make binary`, `erode`, `dilate`, `watershed`, and `analyze particles` (for finding regions of interest or ROIs). However, the processing steps may vary depending on cell types and imaging conditions.

For each segmented cell in each image, a cell node was created in a local graph-based database (Rexster Tinkergraph). Cells that form clusters were grouped together into a cluster node. Each cell node stored properties such as a bounding polygon, a cluster membership and references to the bitmap data of relevant phase contrast and fluorescence images. Pairs of temporally corresponding cluster nodes were connected by edges indicating relationships. Finding temporal correspon-

dences between images required that the images be spatially aligned across time. This image-to-image alignment was implemented by including a temporally invariant landmark as a cue to map local coordinate space to global coordinate space. Temporal correspondence between single cells was established in consecutive time frames according to the algorithm described in Section 7.4.1. The same steps can be repeated for an arbitrarily long temporal image series. Given a time series of length N, for example, information about the temporal correspondence of single cells can be obtained by repeating the calculation of pair-wise connections at time points $[t_1, t_2]$, $[t_2, t_3]$ $\ldots$ $[t_{N-1}, t_N]$.

Single-cell-level time-series data were extracted (as detailed in Section 7.4.1) and stored in the database. Important statistics regarding the analysis of cellular events over time can then be gathered by querying the database (See Figure 7.13). The software to perform this analysis was customised for the characterisation of fluorescent markers in bacterial cells, including *Escherichia coli* and *Bacillus subtilis*, but can be applied to the analysis of any other cell types, once the appropriate image processing steps for segmentation are identified.

7.3.2 Model-driven design of microfluidics

The versatile lens of model-driven engineering (MDE) also proved its worth in designing microfluidics chips in this study. The microfluidics designs featured in Section 7.3.3, 7.3.4, and 7.3.5 were subject to model checks before finalising their designs for fabrication. This design practice was cost-effective in minimising the fabrication cost. The modelling was done using a multi-physics simulation software package (COMSOL v4.2), based on the parameter settings in Table 7.1.

7.3.3 Chemical gradient generator

The microfluidic device shown in Figure 7.3 was designed to generate a chemical gradient out of two input liquids in laminar flow. Liquids exhibiting laminar flow do not mix well, as modelled[1] in the blue and red liquids flowing in parallel without lateral mixing between the y-axis segment of 310 µm and 380 µm in Figure 7.3a.

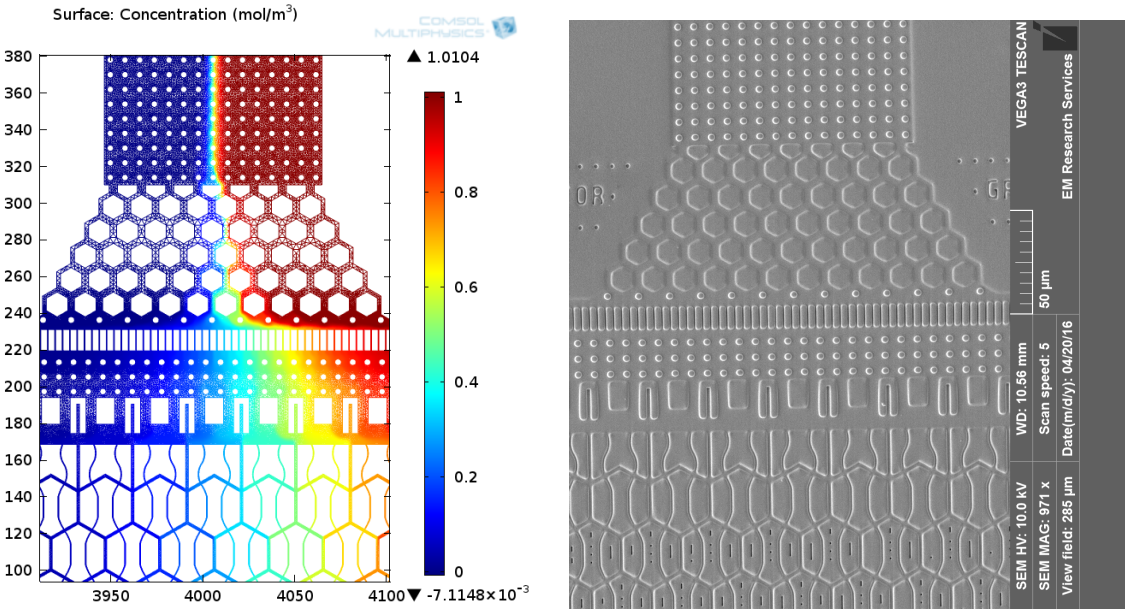
[1]The liquid-structure dynamics was modelled using COMSOL

Table 7.1: COMSOL parameters used in this study:

Category	Property	Settings
Material definition	Water	Standard library definition (Entire domain)
	Silicone [solid]	Standard library definition (Boundaries except inlet and outlet)
Physics packages	Laminar Flow	Temperature (T); 293.15[K] Fluid properties: Water Compressible flow (Ma<0.3)
	Transport of Dilute Species	Velocity field $u = spf/fp1$ Diffusion coefficient$[m^2/s]$ $D_c = 1e-10$
Global definitions	inlet pressures two inlet concentrations	$p1 = 70000$[Pa] pure water and 1mM chemical solution
Mesh definition	size type	Extra fine free triangular
Linear solver	Solver type Preordering algorithm Scheduling method Row preordering Bunch-Kaufman Multithreaded forward and backward solve Pivoting perturbation Nonlinear method Initial damping factor Minimum damping factor	PARDISO Nested dissection multithreaded Auto Enabled Disabled Disabled $1.00E-08$ Automatic (Newton) 0.01 $1.00E-06$

Shown in the y-axis segment of 240 µm and 310 µm of Figure 7.3a is a gradient generator model designed to disrupt the laminar flow and expedite the mixing of the two liquids. The liquid-structure dynamics due to the gradient generator model successfully formed a chemical gradient, as evidenced by the spread of rainbow colours in Figure 7.3a towards the bottom of the y-axis. After the model-based verification of expected behaviours of the design, an actual chip was fabricated using the method from Section 7.2. In order to verify the quality of fabrication results, Scanning Electron Microscopy (SEM) (Tescan Vega 3LMU) was performed at Electron Microscopy Research Services, Newcastle University.

It was qualitatively shown in the SEM image in comparison to the CAD image of the corresponding design in Figure 7.3b that the sub-micron features of the chip design resolved as expected. This SEM result warranted that the fabrication of



(a) (b)

Figure 7.3: Gradient Generator model vs end-product

the chip as far as structural elements are concerned was successful. The successful fabrication of sub-micron features was confirmed quantitatively via measurements of control features (Appendix C.2).

A gradient generator chip was made by curing PDMS directly on the silicon wafer, followed by plasma bonding the PDMS layer onto a glass coverslip (#1 thickness) for use in microscopy. The chip was tested for fluid dynamics to see if the structural design could function as a gradient generator as predicted in the model. 0.1 % (w/v) solution of fluorescent brightener 28, also known as Calcofluor White M2R, was made by dissolving 0.01 g of Calcofluor White in 10 mL of MilliQ water. The solution was titrated using NaOH till the solution turned from opaque white to clear champagne colour. The final pH of the solution reached approximately about 11. Excitation light in the wavelength range of 365 nm and 395 nm incident on Calcofluor White produces emission light at the wavelength of 420 nm. The chip was staged on an epi-fluorescence microscopy (Nikon Ti-E) set up with a DAPI filter. DAPI can pick up fluorescence at the emission wavelength of the Calcofluor White solution. The excitation UV light was generated by using Nikon

Intensilight C-HGFI. MilliQ water was injected into the left inlet of the chip at 1 psi of pressure, and the 0.1 % (w/v) Calcofluor White solution into the right inlet at 1 psi of pressure. The chip was left running for 30 min at room temperature in order to give it some time to form a stable laminar flow. Images were acquired in the bright-field and fluorescent channels, using an image acquisition software (MicroManager), a 40x objective (Nikon) and a CCD camera (QImaging Retiga 2000R).

Figure 7.4 summarises the characterisation results of the gradient generator chip from analysing the fluorescence of the Calcofluor White solution. The PDMS pillar structures visible throughout the bright field image (a) were used as references to make a mask for filtering out background fluorescence in (c) due to PDMS structures. The (b) was the resulting mask showing pillars with elevated levels of background fluorescence. The overall background fluorescence in (c) was estimated by fitting the entire field of view against the background fluorescence from (b). The fitted result was shown in (d). The raw fluorescence image (c) was normalised against the estimated background fluorescence (d), to result in (f). Normalised fluorescence levels across the x-axis in (f) were sampled on four y-axis locations (the red horizontal lines numbered 0, 1, 2, and 3). These fluorescence levels were plotted in (e) for all labelled y-axis locations. The y-axis of the plot shows fluorescence in arbitrary unit (AU) and the x-axis shows horizontal locations matching those of the image field in micrometre unit. Going from right to left along the x-axis of the plot in (e), the fluorescence level at the y-axis location labelled 0 showed a sharp drop, because the fluorescent solution injected from the right would exhibit a laminar flow against the water injected from the left. Upon nearing the bottom (y-axis locations 2 and 3), the fluorescence gradient became much less steep, indicating that the laminar flow started mixing well to form a continuous gradient field. The fluctuation of fluorescence levels in y-axis location labelled 3 was caused by the honeycomb structures in the gradient generator. Overall, the gradient generator chip seemed to function as intended, according to this simple experiment. More extensive characterisation was left for future studies, due to time constraints in the project.

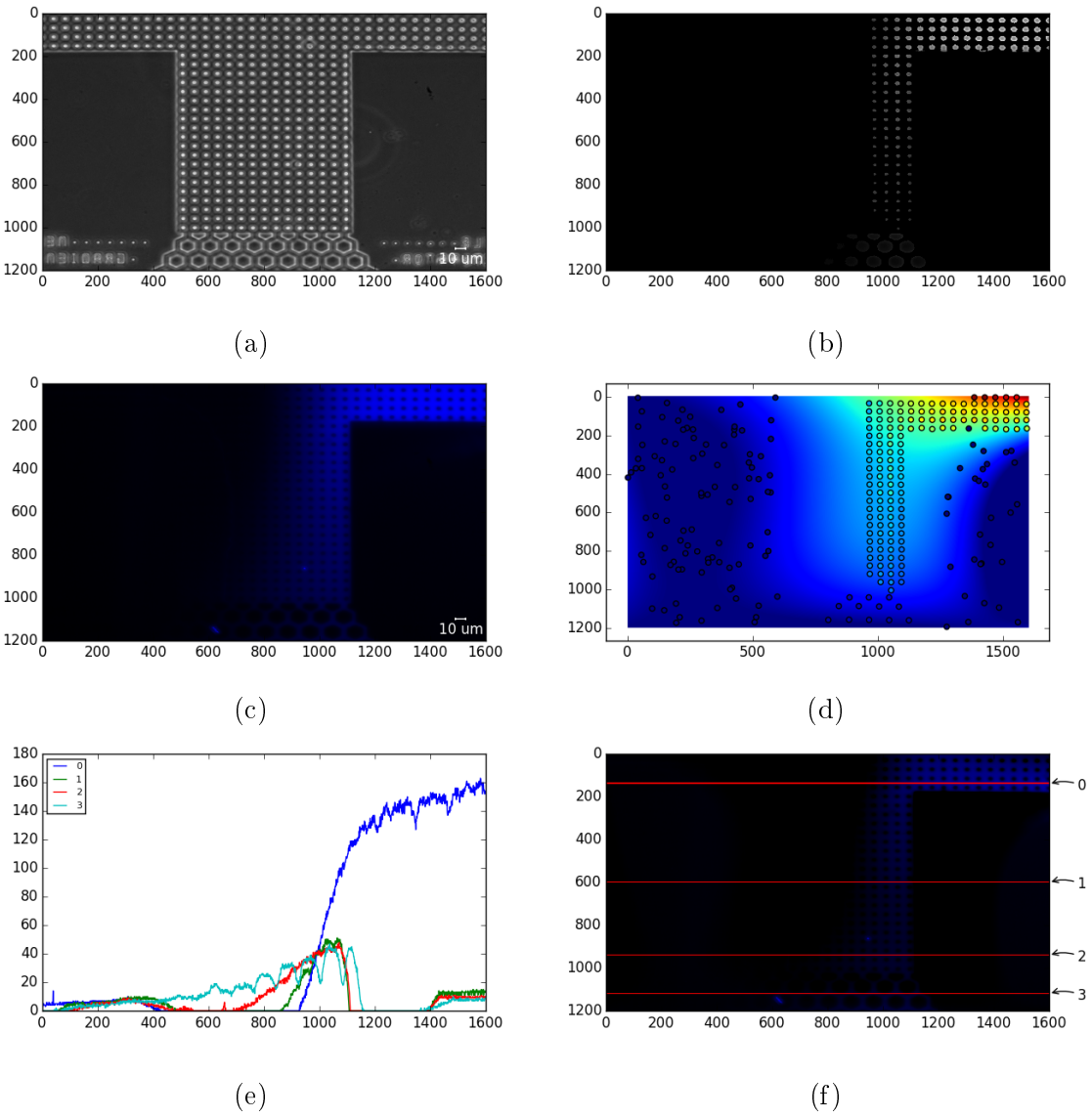


Figure 7.4: Gradient generator characterisation

(a) Bright field, **(b)** PDMS channel support pillars with elevated background fluorescence, **(c)** Fluorescence field (DAPI), **(d)** A heat map showing background fluorescence across the chip fitted from the background fluorescence of the support pillars, **(e)** Normalised fluorescence (AU) profiles across x-axis (µm) of four y-axis locations, **(f)** Normalised fluorescence field.

7.3.4 Chemical-gradient-passthrough decision tree

Microfluidic channels resembling a decision tree as shown in Figure 7.5 was designed for measuring bacterial chemotaxis-like behaviours. The design can be used for testing if cells would prefer growing in the direction of increasing or decreasing chemical concentrations. One of the obstacles in the design of the microfluidic decision tree was that the structure resembles that of the chemical gradient generator. This means that the decision tree structure would disrupt the chemical gradient of the fluid flowed into it as the fluid travels further down the tree.

As a remedy to this issue, channels in the shape of waves running down vertically were introduced into the decision tree structure (See the SEM image in Figure 7.5). The decision tree structure featuring these wavy passthrough channels was dubbed the name *chemical gradient passthrough decision tree*.

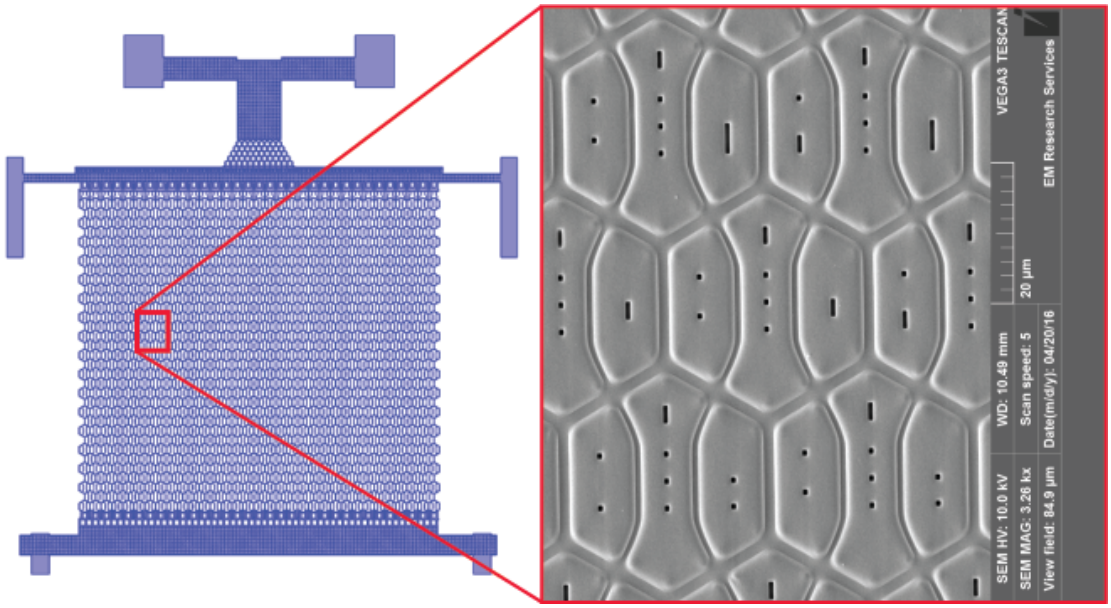


Figure 7.5: Passthrough decision tree

In fact, this chemical gradient passthrough channel design was finalised by modelling numerable structural variations. The COMSOL simulation result shown in Figure 7.6 revealed that this passthrough channel design is effective in neutralising the adverse effect of the decision tree structure on chemical gradient.

With respect to automation, there was another critical design aspect to be

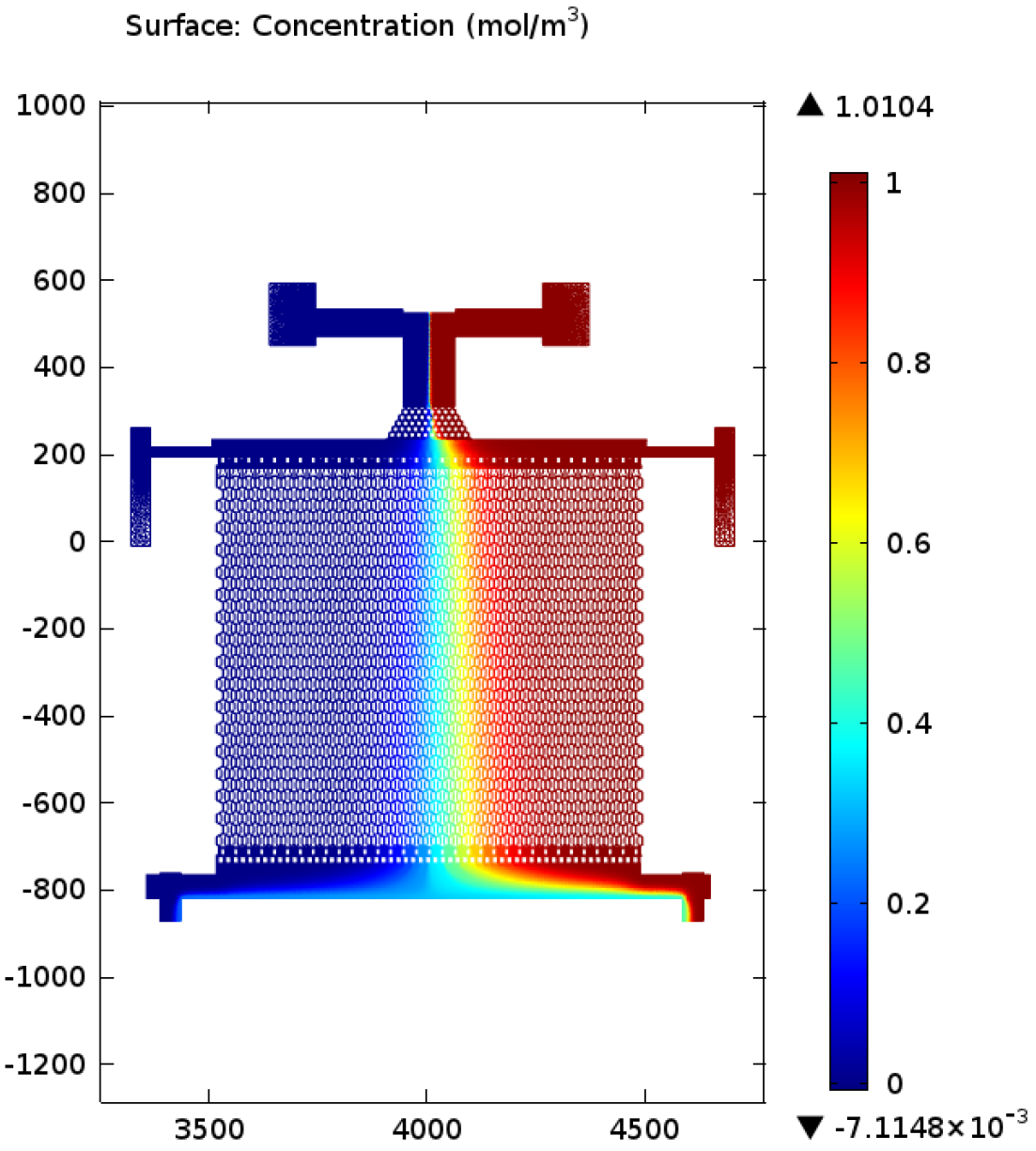


Figure 7.6: Modelling of the passthrough decision tree

considered in the making of the decision tree. The dimensions of the area occupied by the decision tree channels were approximately about 1000 µm by 1000 µm. An area this size is rather too large to be covered as a whole by microscopy using high magnification objectives (e.g. 100x or 40x). At the same time, using high

magnification objectives is necessary in acquiring microscopy data suitable for image analysis. Microscopic barcodes were designed as shown in Figure 7.7 for labelling multiple microfluidic locations. Each labelled location was designed to be small enough to fit into the field of view of high magnification microscopy. This would allow for the use of high power microscopy along with this chip design.

The area of the structural feature representing one bit was designed to be $1\,\mu m^2$. Given that the chip design was already crowded with small features, fabricating the wafer to cleanly resolve all those tiny details was non-trivial. This kind of tiny features in microfluidics design was putting the fabrication method discussed in Section 7.2 to a truly stringent test. The success in fabricating those features using this method, as can be seen in the SEM image of Figure 7.5, was a significant achievement. A microscopic barcode design such as this would become one of the essential features in the design of sub-micron-scale microfluidic devices for supporting lab automation.

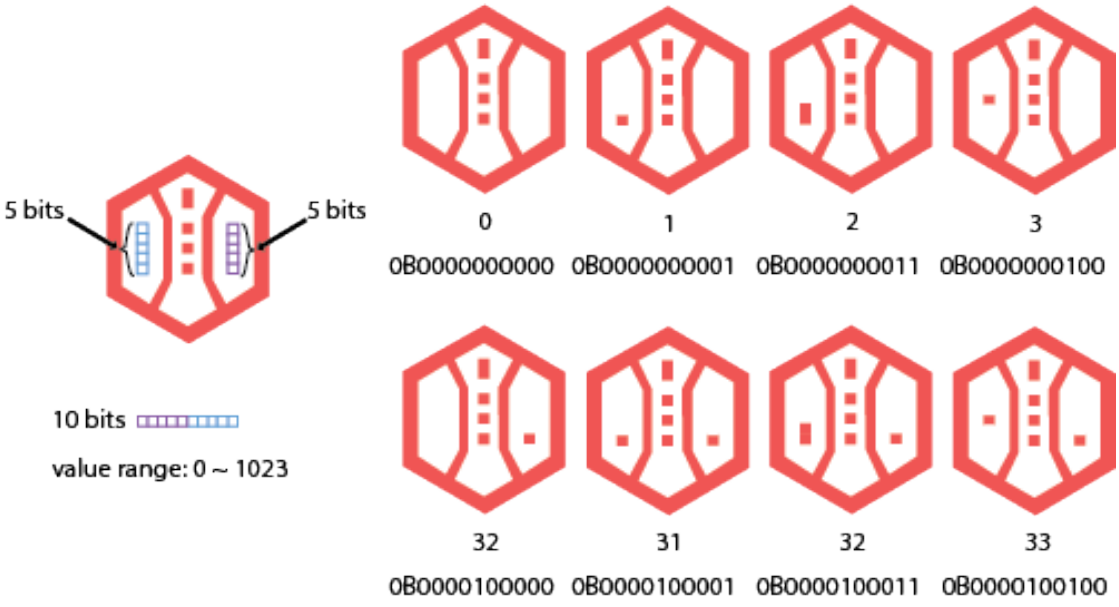
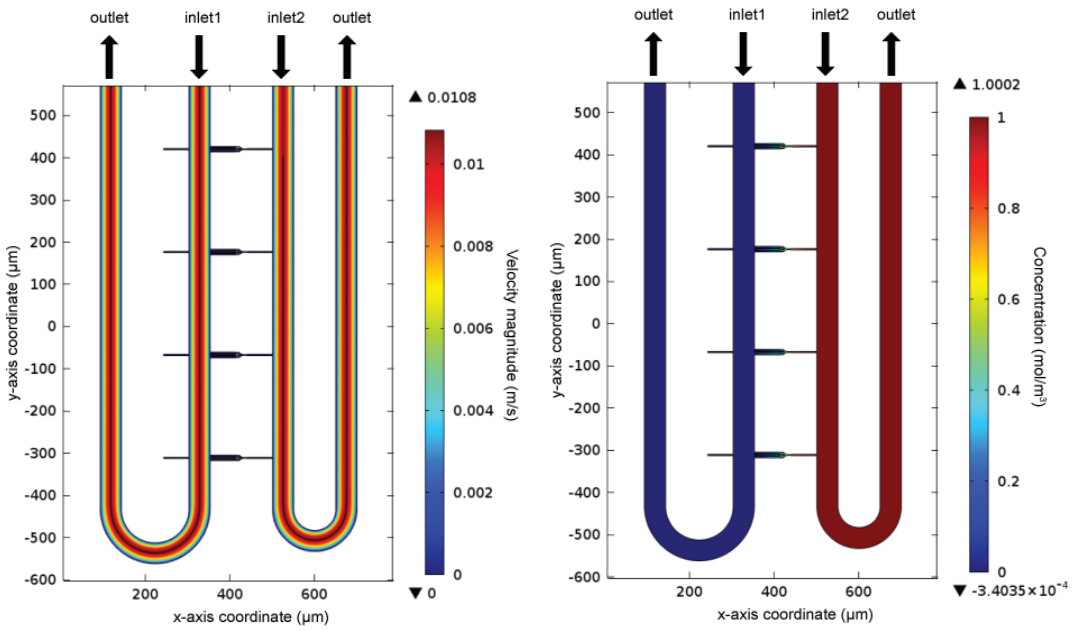


Figure 7.7: Microscopic barcodes to label locations in microfluidics

7.3.5 Quorum communication tester



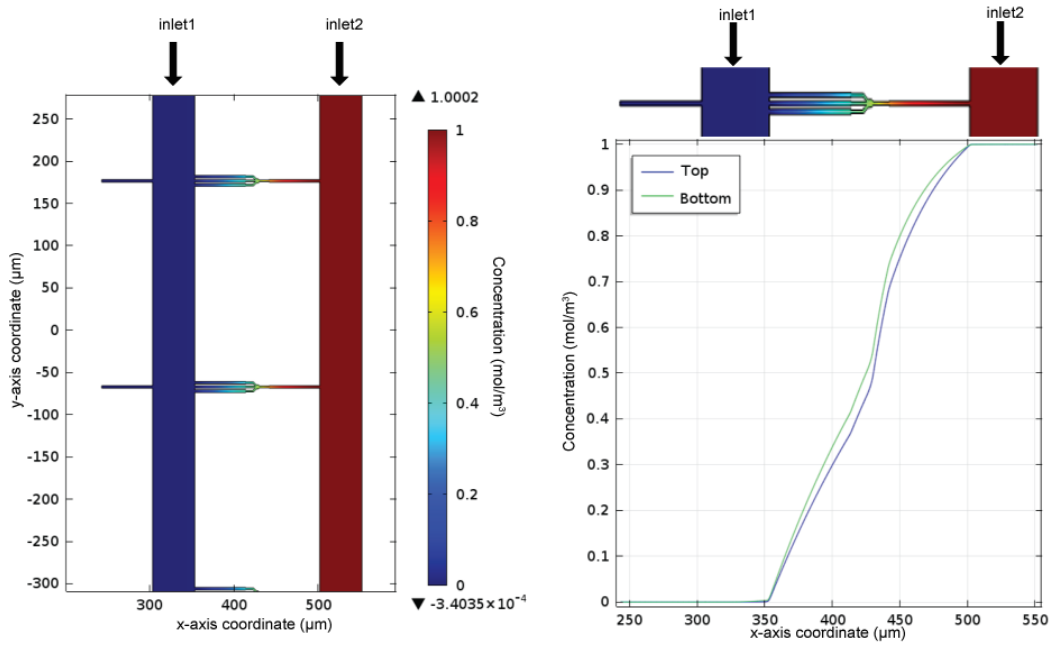
(a) (b)

Figure 7.8: COMSOL simulation of fluid dynamics in the quorum sensing assay chip

A PDMS-based microfluidics assay chip was designed to be used in experiments for investigating microbial quorum sensing. The COMSOL simulation showed liquid velocity and concentration gradient profiles with respect to the chip's liquid-structure interaction dynamics. **(a)** The liquid velocity profile of the fluidic channels, given 70,000 Pa of pressure on the two inlets. **(b)** The concentration gradient of the fluidic channels when water was injected through inlet1 and 1 mM concentration of inducer molecule was injected through inlet2.

Yet another class of microfluidics-based measurement device was designed to allow for experiments involving quorum communication behaviours in cellular systems. As were the cases in previous microfluidics designs, finalising this chip design was a result of subjecting different structural variations to extensive modelling and simulation. Figure 7.8 shows a fluid dynamics simulation result of the quorum communication tester chip. The chip was designed to accept two input fluid flows with different chemical compositions, and to form chemical interfaces between the two

fluids at the perpendicular channels connecting the two main inlets. The perpendicular channels were designed to allow the exchange of chemicals via diffusion. The result shown in Figure 7.8b assured that the chemical concentration injected into the right hand side channel (inlet2) does not affect the left hand side channel (inlet1). The only places in the chip where the two fluid types exchanged chemical compositions were the perpendicular channels.



(a) (b)

Figure 7.9: COMSOL simulation of concentration gradients in the quorum sensing assay chip

A comparison of the COMSOL simulation of concentration gradients was performed to see if there are location specific differences in the gradient profile of the microfluidics assay chip. The given design was confirmed to support the desirable property of exhibiting a consistent concentration gradient profile invariant to the difference in the y-axis coordinate locations. **(a)** A close-up view of the concentration gradient field shown in Figure 7.8b. **(b)** Comparison of the concentration gradient across the x-axis between two locations (Top & Bottom) separated by about 6000µm in the y-axis direction.

Figure 7.9 shows close-up views of the perpendicular channels exhibiting diffusive chemical exchange between the two inlets. In the experimental setup of this simulation, a glucose solution in 1 mM concentration was injected into inlet2, and

Figure 7.10: An SEM image showing the fabrication result of the quorum communication tester chip

water was injected into inlet1. Figure 7.9b shows a graph comparing the glucose concentration profile across the x-axis of perpendicular channels in two different y-axis locations separated by about 6000 µm (Top vs Bottom). The graph confirmed three desired properties of the microfluidic design: that the inlet2 glucose concentration did not affect inlet1, that the perpendicular channels provided an interface for diffusive chemical exchange, and that the y-axis location of perpendicular channels did not affect the chemical diffusion characteristics.

The quorum communication tester chip was fabricated using the methods in Section 7.2. Figure 7.10 shows an SEM image of the fabrication result. Disregarding the minor blemishes visible on the top part of the image, likely to be dirt particles inadvertently introduced in preparing for the SEM samples, the submicron features of the chip design were successfully fabricated, from inspecting the SEM image qualitatively. The *in vitro* characterisation of the chip was not attempted due to time constraints in the project. However, the same experimental

setup used in the *in vitro* characterisation of the gradient generator chip (Section 7.3.3) can be used for the characterisation of this chip as well.

7.4 Programmatic image processing and analysis

Using a microfluidic platform for observing bacteria at single-cell and single-lineage levels typically involves time-lapse microscopy, which can generate a large amount of image data. Image data in their raw form require much pre-processing before any quantifiable data can be extracted for analysis. Typical image processing steps involve, for instance, the identification of cell boundaries (segmentation), gathering statistics on cell sizes and fluorescence levels, and tracking temporally corresponding cells in time series. Not only are these tasks computationally expensive, they also generate large amounts of single-cell data, making manual data curation cumbersome and error-prone. Dealing with large volumes of data as such necessitates the use of integrated and automated image analysis pipelines that can process image data as they are generated with accuracy and speed.

There are many tools available for image analysis in the biological sciences, including MicrobeTracker [70], TLMTracker [100], CellProfiler [28], and ImageJ [151]. While these tools are widely used, they require significant manual intervention in the analysis process, and they are not designed to process large numbers of images in a systematic, parallel and automated fashion. MicrobeTracker is popular among biologists for image analysis. However, the software is GUI-based and primarily intended for manual use. Furthermore, MicrobeTracker is written in Matlab, requiring a commercial license to be able to use its code base. CellProfiler is probably one of the strongest contenders for providing flexible, parallelisable pipelines. However, this tool is still primarily intended for manual use, with a GUI-based interface. Although CellProfiler provides a headless operation mode without the GUI interface, the mode is intended, not for being integrated as part of automated microscopy, but for a postexperimental batch image processing. There is no API-level support that would allow other software to easily invoke CellProfiler and retrieve results in a modular fashion. The architecture of ImageJ and its sibling Fiji [150] is noteworthy as it allows the software's core library full of useful image processing algorithms to be readily reused.

An image analysis pipeline, employing Fiji's core library for much of the image processing needs, was developed with the goal of having the pipeline integrated as part of the microfluidic platform for automated time-lapse microscopy. One of the key advantages of using an integrated system such as this is that a full automation, from image acquisition to analysis, can be achieved. This also means that the results of image analysis can be used to determine the way in which image acquisition is done, as part of an active feedback loop. This capability is especially useful in dealing with large sample sizes. If need be, a rule-based filtering algorithm can be implemented depending upon analysis results, to reduce the number of samples being observed to those exhibiting phenotypes of interest.

7.4.1 Correspondence algorithm: an overview

Dealing with time-series cellular microscopy data via using a computerised logic is non-trivial, largely due to the difficulty in establishing the spatio-temporal correspondence between cellular data sets. A pattern recognition algorithm was needed in order to extract cellular image features suitable for uniquely identifying cells undergoing growth in space and time. Introduced in the following is a novel correspondence algorithm that can process time-series images featuring bacterial growths. The algorithm can extract single cell data, and track each cell spatio-temporally. The algorithm consists of two parts: establishing cluster-level correspondence followed by establishing single-cell level correspondence.

7.4.2 Establishing cluster correspondence

Consider a set B consisting of single cells from multiple time points, and a set P consisting of subsets of single cells from B satisfying the condition L. The condition L, used to qualify cells for belonging to the same subset, is a predicate for testing cells to see if they coincide in the same spatio-temporally isolated cluster. These

sets can be defined[1] as:

$$
\begin{aligned}
&B = \{x \mid x \text{ is a single cell at time points } t \in \{1, 2, 3 \dots\}\} \\
&P = \{x \subseteq B \mid x \text{ satisfies } L\} \text{ is a partition set of } B, \text{ fulfilling} \\
&\quad \forall x_1, x_2 \in P, x_1 \neq x_2 \neq \emptyset, x_1 \cap x_2 = \emptyset, \text{ and } \bigcup P = B \\
&\quad L : \forall y \in x \text{ is in the same spatio-temporally isolated cluster}
\end{aligned}
\tag{7.1}
$$

$P(t)$ is a subset of P, defining a set of cell clusters co-occurring at time t, hence corresponding to the collective information of single cells captured by microscopy image(s) at time t. In addition, $E(t)$ is a set of ordered pairs of clusters having temporal correspondence from time t to $t+1$, defined as:

$$
\begin{aligned}
&P(t) = \{x \in P \mid x \text{ is a cluster of cells at time } t\} \\
&E(t) = \{(x, y) \in P(t) \times P(t+1) \mid x \text{ temporally corresponds to } y\}
\end{aligned}
\tag{7.2}
$$

, where the cluster temporal correspondence is defined by clusters x and y having spatial overlap with respect to the global coordinate space.

The transitive closure of $E(t)$, denoted $E_T(t)$ can be defined as,

$$
\begin{aligned}
E_T(t) = &\{(x, z) \in P_p(t) \times P_p(t) \mid \exists Y \subseteq P_p(t) \text{ satisfying } W\}, \\
&\text{where } (x, z) = (z, x) \text{ and } P_p(t) = P(t) \cup P(t+1) \\
&W : \ \forall y, y^{'} \in \{Y \cup \{x, z\}\}, y \neq y^{'} \neq \emptyset \rightarrow f_d(y, y^{'}) > 0
\end{aligned}
\tag{7.3}
$$

Here, $f_d(y, y^{'})$ denotes the degree of relatedness in cluster correspondence between clusters y and $y^{'}$, or the minimum number of corresponding pairs $p \in E(t)$ required

[1] The convention for mathematical notations used here adheres to that of Set Theory and Logic. $\bigcup P$, for example, refers to the union set of all members of set P. The notation $\emptyset$ refers to an empty set, unless noted otherwise.

to associate cluster y to cluster $y^{'}$, as defined in Eq.7.4.

$$\begin{aligned} f_d(y, y^{'}) = min\left(\left|\{(x, x^{'}) \in E(t) \mid \exists R \subseteq P_p(t), \text{ satisfying } K\}\right|\right) \\ K: \ \forall x \in \{R \cup \{y, y^{'}\}\}, \exists x^{'} \in \{R \cup \{y, y^{'}\}\}, \text{such that} \\ x \neq x^{'} \wedge (x, x^{'}) = (x^{'}, x) \wedge (x, x^{'}) \in E(t) \end{aligned} \tag{7.4}$$

$$f_d(y, y^{'}) = \begin{cases} 0, & \text{if } R = \emptyset \wedge (y, y^{'}) \ni E(t) \\ 1, & \text{if } (y, y^{'}) \in E(t) \\ 2 \text{ or greater}, & \text{otherwise} \end{cases}$$

$f_d(y, y^{'}) = 0$ means clusters y and $y^{'}$ are not associated in terms of cluster correspondence. $f_d(y, y^{'}) = 1$ means a single cluster, y, at time t, is directly associated with a single cluster at time t+1, namely $y^{'}$ (Figure 7.11a). $f_d(y, y^{'}) = n$, where $n > 1$, means cluster y at time t is associated with cluster $y^{'}$ at time $t + 1$, via a minimum of n number of cluster pairs $p \in E(t)$. Having the number $n > 1$ signifies one of the following events: multiple clusters at time t merged to a single cluster at time $t + 1$ (Figure 7.11c), a single cluster at time t split into multiple clusters at time $t + 1$ (Figure 7.11b), or both (Figure 7.11d).

It can be said that clusters $y \in P(t)$ and $y^{'} \in P(t + 1)$ have cluster correspondence if the binary relation, $y \ E_T(t) \ y^{'}$ exists, by evaluating the predicate Q (See Eq.7.5). In the events of clusters merging or splitting, $E_T(t)$ can also be used to decide, for instance, if clusters y and $y^{'}$ at the same time frame t have cluster correspondence, by asserting the predicate Q.

$$Q\left[y \ E_T(t) \ y^{'}\right] = \begin{cases} True, & \text{if } \exists (y, y^{'}) \in E_T(t) \\ False, & \text{otherwise} \end{cases} \tag{7.5}$$

$C(t)$ is a partition set of $P(t)$ consisting of a set of non-empty, mutually exclu-

sive subsets of $P(t)$:

$$\begin{aligned} C(t) =& \{x \subseteq P(t) \mid x \text{ satisfies } J\}, \text{ fulfilling} \\ & \forall x_1, x_2 \in C(t), x_1 \neq x_2 \neq \emptyset, x_1 \cap x_2 \neq \emptyset, \bigcup C(t) = P(t) \\ & J : (\forall x_i, x_j \in x, Q[x_i E_T(t) x_j] \text{ is } True) \vee (\mid x \mid = 1) \end{aligned} \tag{7.6}$$

Establishing one-to-one temporal correspondences between members of $C(t)$ and $C(t+1)$ can be provided by the injective function $f_c : x \in C(t) \mapsto y \in C(t+1)$„ defined in terms of a set of mapping pairs,

$$\begin{aligned} & f_c : x \in C(t) \mapsto y \in C(t+1) = \\ & \quad \{(x, y) \mid \forall x_i \in x \ \& \ \forall y_j \in y, Q[x_i E_T(t) y_j] \text{ is } True\} \end{aligned} \tag{7.7}$$

7.4.3 Establishing single-cell-level correspondence

Provided that $C_1 \in C(t)$ and $C_2 \in C(t+1)$ have a temporal correspondence via $f_c : C_1 \mapsto C_2$, it becomes relatively trivial to establish single-cell-level correspondences between $c_i \in \bigcup C_1$ and $c_j \in \bigcup C_2$, as follows:

Let a single cell c (e.g. c_i or c_j) have a geometrical representation of a bounding polygon c_P given by a set of N_V vertices,

$$c_P = \{v_0, v_1, \ldots, v_{N_V - 1}\} \tag{7.8}$$

The minimum bounding box of c_P, denoted $B(c_P)$, is a set of four vertices that forms a rectangle to enclose the polygon in two-dimensional coordinate space:

$$\begin{aligned} B(c_P) =& \{\hat{v}_0, \hat{v}_1, \hat{v}_2, \hat{v}_3\}, \text{ where} \\ & \hat{v}_0 = [x_{min} \ y_{min}], \hat{v}_1 = [x_{max} \ y_{min}] \\ & \hat{v}_2 = [x_{min} \ y_{max}], \hat{v}_3 = [x_{max} \ y_{max}] \end{aligned} \tag{7.9}$$

Given the minimum bounding boxes of $c_P(i)$ and $c_P(j)$, $c_P(j)$ is scaled to match the dimensions of $c_P(i)$ by finding the dot products of the individual vertices of

$c_P(j)$ and the transformation matrix A:

$$\begin{aligned} c_P^{'}(j) = c_P(j) \cdot A, \quad A = \begin{bmatrix} \frac{w(i)}{w(j)} & 0 \\ 0 & \frac{h(i)}{h(j)} \end{bmatrix} \\ \text{, where } w = (\hat{v}_1 - \hat{v}_0) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad h = (\hat{v}_2 - \hat{v}_0) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{7.10}$$

Now that $B(c_P(i))$ and $B(c_P^{'}(j))$ have the same dimensions sizewise, $c_P(i)$ and $c_P^{'}(j)$ can be directly compared to evaluate the geometrical differences between the two polygons (See Figure 7.11g).

Sweeping a horizontal line across the y-axis to scan and sample N_L number of lines that fill a polygon can be done for both $c_P(i)$ and $c_P^{'}(j)$. The sweeping line $\overline{S_0 S_{1k}}$ at position k is defined by two vertices S_0 and S_1 with respect to k and the polygon's minimum bounding box:

$$\begin{aligned} S_0 = \frac{(N_L - 1 - k)\hat{v}_0 + k\hat{v}_2}{N_L - 1} \quad S_1 = \frac{(N_L - 1 - k)\hat{v}_1 + k\hat{v}_3}{N_L - 1} \\ \text{, where } 0 \leq k \in \mathbb{Z} \leq N_L - 1 \end{aligned} \tag{7.11}$$

The line segment $\overline{S_0 S_{1k}}$ intersects a polygon c_P at two vertices P_0 and P_1, where P_0 may or may not coincide with P_1. Given two polygons $c_P(i)$ and $c_P^{'}(j)$, there exists a sequence of four intersecting vertices, $(P_0(i), P_1(i), P_0(j), P_1(j))$, at $y = k$. When sorted in the descending order of the vertices' x-axis coordinate component, the ordered sequence can be re-labelled and defined as $(v_k(0), v_k(1), v_k(2), v_k(3))$. Then, the MSE of geometrical differences between the polygons $c_P(i)$ and $c_P^{'}(j)$ can be calculated by:

$$\begin{aligned} & f_{MSE}(c_P(i), c_P^{'}(j)) = \\ & \frac{1}{N_L} \sum_{k=0}^{N_L - 1} \left((v_k(0) + v_k(2) - v_k(1) - v_k(3)) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)^2 \end{aligned} \tag{7.12}$$

Given Eq.7.12 as a cost function, the problem of finding temporal correspon-

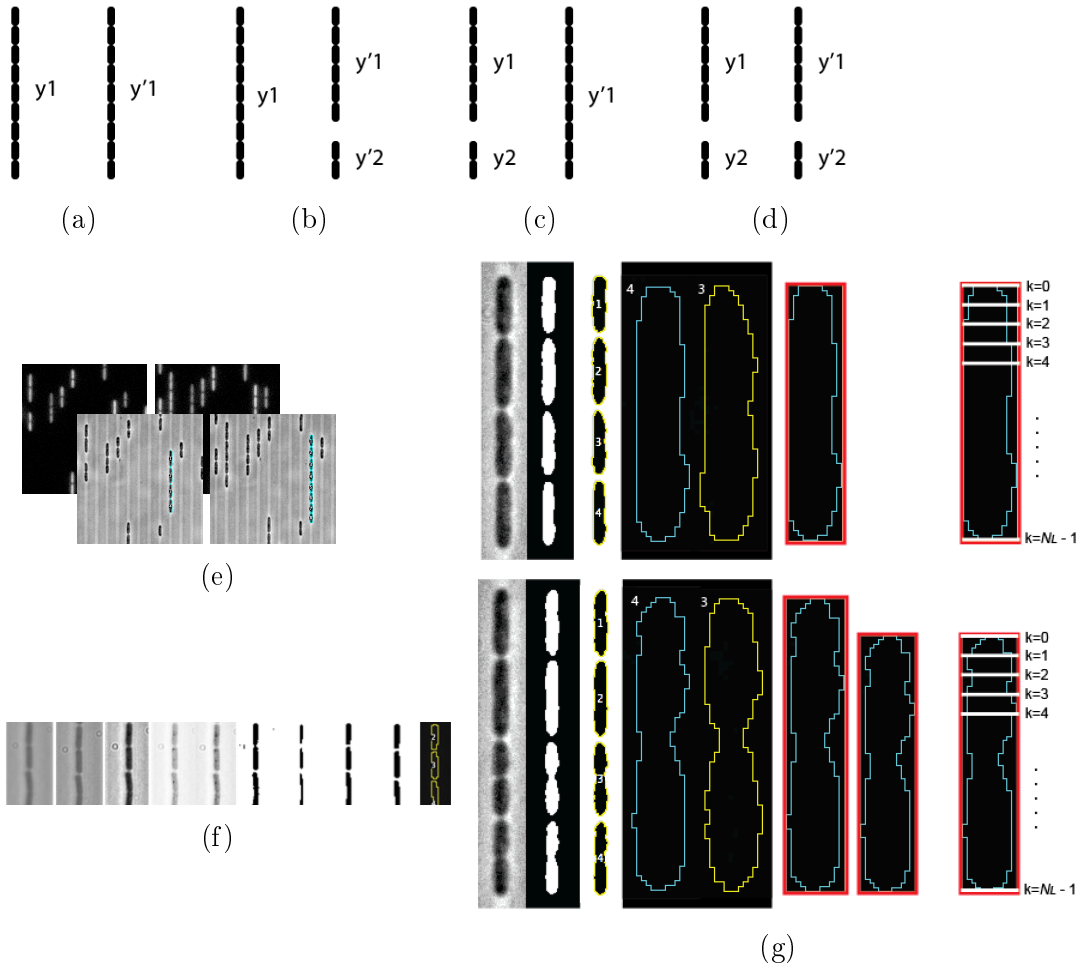


Figure 7.11: Overview of the image processing pipeline to compare cells for temporal correspondence

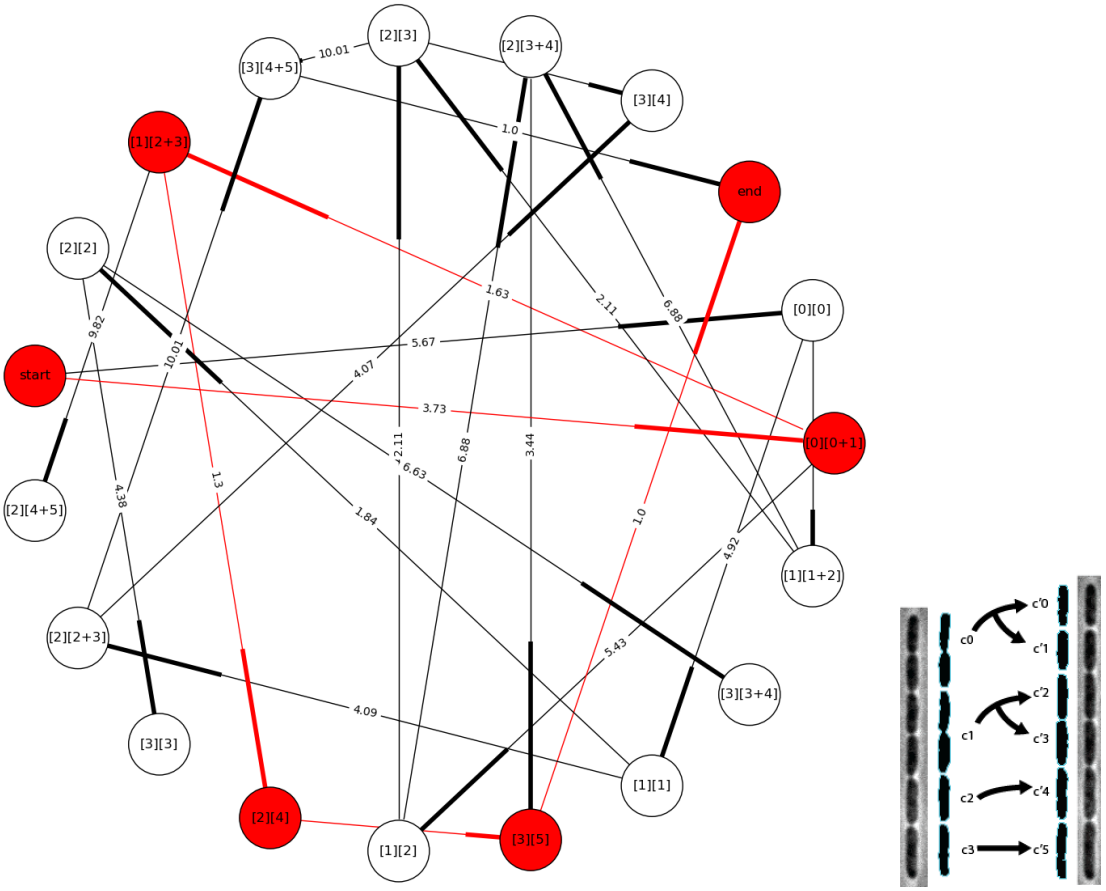
(a,b,c,d) Clusters at two timepoints having different geometrical patterns. (e) Two pairs of phase contrast and fluorescence images, at time points t_0 (left) and t_1 (right), of *E. coli* constitutively expressing GFP. The cell cluster highlighted in cyan at t_0 corresponds to the cluster in cyan at t_1. (f) From left to right: a series of Fiji's commands `sharpen`, `contrast`, `log`, `contrast`, `make binary`, `erode`, `dilate`, `watershed`, and `analyze particles`. (g) Two cell clusters from successive time points are shown. The upper half shows the image processing steps involved in sampling polygonal data representative of a single cell at time t_0, and so does the bottom half for a single cell at time t_1. The single cell being sampled at time t_1 is subject to an extra step of scaling its dimension to match that of the cell at time t_0 before data sampling and comparison can take place.

dence can effectively be considered in light of the general assignment problem. Assignment problems, in general, can be solved by the algorithm known as the Hungarian method [104].

In order to minimise the calculation of paring costs, the potential assignment of a single cell $c_i \in \bigcup C_1$ at time t can be limited to a subset of $\bigcup C_2$, where the two cluster sets $C_1 \in C(t)$ and $C_2 \in C(t+1)$ have an one-to-one temporal correspondence according to Eq.7.7. For example, if $\bigcup C_1 = \{c_i \mid i \in \{0,1,2,3\}\}$ and $\bigcup C_2 = \{c_j \mid j \in \{0,1,2,3,4,5\}\}$ are sets of corresponding cells in time t and $t+1$ respectively, four cells in $\bigcup C_1$ need to be matched to six cells in $\bigcup C_2$ (See Figure 7.12b). Provided that images are taken at intervals small enough to ensure the cell divisions for cells in $\bigcup C_1$ at time t to occur at most once at time $t+1$, the following cost matrix can be constructed for solving the assignment problem via the Hungarian algorithm, where $c_{[i][j]} = f_{MSE}(c_P(i), c_P^{'}(j))$ and $c_P^{'}(j+j+1) = c_P^{'}(j) \cup c_P^{'}(j+1)$.

$$\begin{bmatrix}
c_{[0][0]} & \infty & \infty & \infty \\
c_{[0][0+1]} & \infty & \infty & \infty \\
\infty & c_{[1][1]} & \infty & \infty \\
\infty & c_{[1][1+2]} & \infty & \infty \\
\infty & c_{[1][2]} & c_{[2][2]} & \infty \\
\infty & c_{[1][2+3]} & c_{[2][2+3]} & \infty \\
\infty & \infty & c_{[2][3]} & c_{[3][3]} \\
\infty & \infty & c_{[2][3+4]} & c_{[3][3+4]} \\
\infty & \infty & c_{[2][4]} & c_{[3][4]} \\
\infty & \infty & c_{[2][4+5]} & c_{[3][4+5]} \\
\infty & \infty & \infty & c_{[3][5]}
\end{bmatrix}$$

Cell pairs with the infinity sign (∞) are invalid matches that would leave unmatchable cells in the clusters, and hence are given the maximum cost value without any calculation. The cost functions highlighted in red denote cell pair matches that would leave unmatchable cells in the clusters due to boundary conditions, and therefore can also be given the maximum cost value without any calculation. The cost functions shown in black are the possible matches that would result in successful correspondence matches satisfying all cells in the clusters, and therefore need to be calculated. Unfortunately, the Hungarian algorithm does not scale well. Its most efficient implementation known to date has a time complexity of $O(n^3)$ [94].



(a) (b)

Figure 7.12: A graph-encoded assignment problem, and its Dijkstra path as a solution

(a) The assignment problem of corresponding cells can be encoded into a weighted, directed graph. Currently shown is an example of a cluster of four cells at time t_0 being matched to a cluster of six cells at time t_1. Nodes represent possible correspondence matches, and weighted edges the costs of respective matches. For instance, the node labelled $[0][0+1]$ represents the assignment of the cell $c0$ at time t_0 to the cells $c^{'}0$ and $c^{'}1$ at time t_1. Each edge connecting two nodes carries a weight, whose value indicates the cost of the next matching pair of cells between the two clusters. The nodes and edges highlighted in red represent the best matches, or the matches with the smallest total MSE, searchable by finding the Dijkstra's shortest weighted path of the graph. **(b)** The phase contrast images of *E. coli* cell clusters at time t_0 and t_1 used for building the weighted directed graph. The black arrows connecting the segmented cells between the two clusters are deduced from the graph path highlighted in red.

In the cell to cell matching case, the cardinal number n in discussing the time complexity can be determined by adding the cell count at time t ($|\bigcup C_1|$), and the counts of individual cells ($|\bigcup C_2|$) and merged cells ($|\bigcup C_2|-1$) at time $t+1$, hence $n = |\bigcup C_1| + 2 \cdot |\bigcup C_2| - 1$ [140].

In fact, the assignment problem can also be encoded into a weighted graph, where the cost function (Eq.7.12) provides the weights. Having the problem encoded in a graph allows graph search algorithms such as Dijkstra's algorithm to be used to find solutions for the assignment problem, at a more scalable time complexity of $O(|E|+|V|log|V|)$. This approach is different from applying the shortest augmenting paths based approach [94] in solving an assignment problem, in that the algorithm shown here only needs a single calculation of Dijkstra's shortest path per assignment problem as opposed to many iterations of path calculations required by the augmenting paths method.

For example, in the case of four cells being matched to six cells (Figure 7.12b), a weighted graph representing all permutations of possible cell matches (nodes) along with the costs of the matches (edges) can be constructed (Figure 7.12a). Using Dijkstra's algorithm [54], the graph can then be solved for the shortest path, or the path with the minimum total cost (highlighted in red), from which single-cell-level temporal correspondences can easily be deduced. Interestingly, Dijkstra's explanation of using graph structures in shortest path problems [54] is in agreement with Richard Bellman's idea of Principle of Optimality [17], a necessary condition for dynamic programming. Bellman's concept of dynamic programming is a mathematical optimisation achieved by breaking down a complex problem into simpler sub-problems, via offering a simple incremental handle to deal with combinatorial complexity. Having such an element of algorithmic compatibility would mean that the graph search problem can be transformed to an optimisation problem defined in terms of dynamic programming, or vice versa. For problems requiring small input sizes, such as that demonstrated here, adopting dynamic programming in lieu of Dijkstra's graph-based method would have no dire consequences. However, doing so in a general sense, shall need to be approached with caution. Dynamic programming may end up with performance penalties, as it is known to exhibit a pseudo-polynomial time complexity with respect to the input bit length, unlike Dijkstra's graph-based approach regardlessly offering a polynomial time complex-

ity.

7.4.4 Application of the correspondence algorithm in synthetic biology

A set of time-lapse images were acquired using the microfluidic system as described in Section 7.3.1, and were analysed via the image processing pipelines and algorithms as detailed in Section 7.4.2 and 7.4.3. *Bacillus subtilis* 168, chromosomally modified to constitutively express GFP, was used to test the system. Figure 7.13a shows an example of the generations of a cell lineage, with respect to a mother cell, being tracked using the system developed in this study. It was possible for the system to gather statistics much like those offered by flow cytometry (Figure 7.13b). Statistics available from flow cytometry lack temporal dimensions at the single-cell level. The only possible way to incorporate a temporal dimension into the measurements in flow cytometry is to use biological replicates from different static time points. In doing so, however, flow cytometry loses single-cell-level details in the temporal dimension, and is restricted to gathering population-level statistics. This microfluidics-based system was capable of tracking single cells over time, including cellular events such as the growth and splitting of cells. The stochastic nature of single cells may result in inter-cellular variabilities in the time of occurrence of certain cellular events. For example, the cellular event of reaching the fifth generation from the mother cell took place at different time points for different cells (See the cells highlighted in red in Figure 7.13a). Therefore, for single-cell-level analyses involving temporal dimensions, it is important to differentiate the kinds of statistics based on static time points (i.e. when measurements were taken) from those based on cellular events (i.e. when something important happened). The measurements shown in Figure 7.13c were from the cells five generations away from those of Figure 7.13b. The collection of these generation-based statistics demonstrate that this novel system was successful not only in incorporating temporal dimensions into single-cell-level statistics, but also in capturing statistics based on cellular events rather than on static time points.

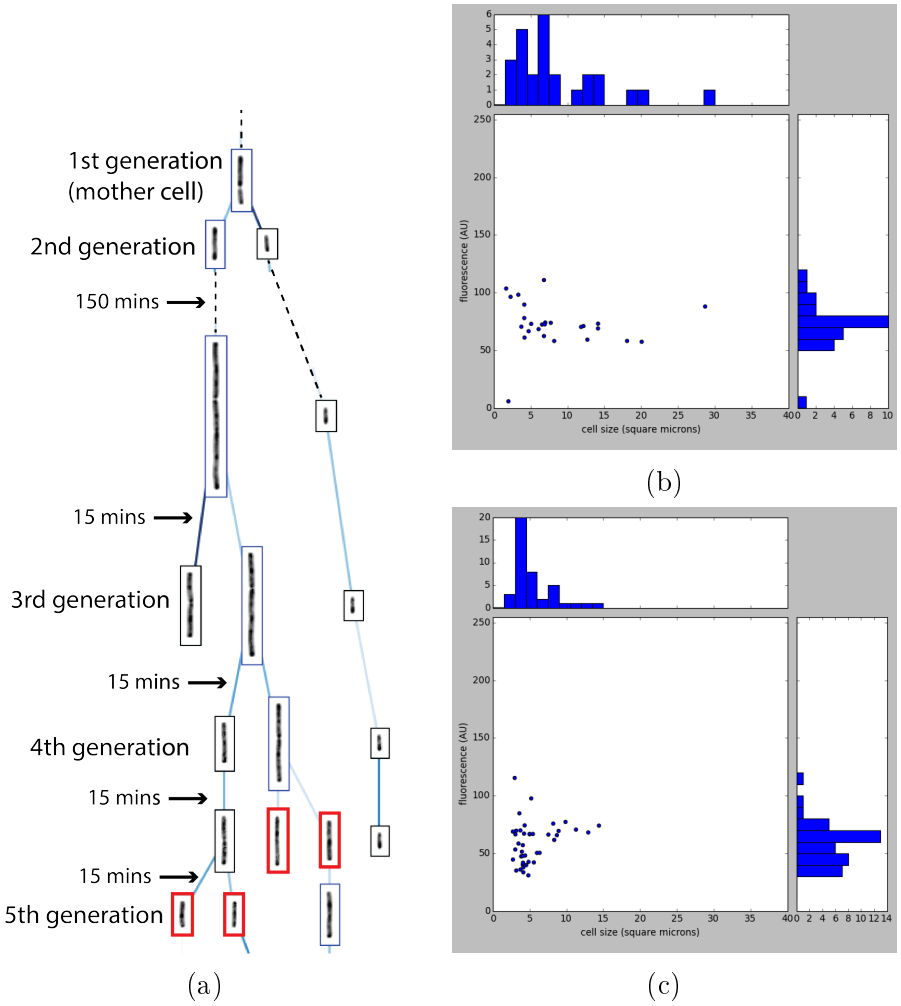


Figure 7.13: Single-cell-level statistics based on cell lineages and generations **(a)** An example lineage of *B. subtilis* cells. The cells highlighted in red are in their fifth generation. **(b)** An example analytic data showing the scatter plot of fluorescence vs size of single cells (*B. subtilis*) at their first generation. **(c)** An example analytic data showing the scatter plot of fluorescence vs size of single cells (*B. subtilis*) at their fifth generation.

7.5 Discussion

One of the primary achievements worthy of mentioning as part of this research work in microfluidics was the development of a fabrication protocol that can produce submicron features without relying on costly etching optimisation runs. This fabrication protocol was a product of extreme scrutiny of various intermediate pro-

tocols given the constrains of the budget and time in the project. The protocol was what made it possible, in the first place, to successfully fabricate all the chip designs featured in this study. The innovations shown here would not have been possible without this fabrication protocol.

As a rewarding consequence, the single-cell-level microfluidic measurement devices shown in this work opened up many possibilities in terms of achieving lab automation. The possibilities were especially promising as the programmatic image analysis was demonstrated to work successfully in conjunction with single-cell level measurements. As much as the single-cell level measurements are useful, they can generate copious amounts of data, rendering manual analysis infeasible. To this end, the programmatic analysis of measurement data using the software developed in this research work played a crucial role. In the real-world automation scenario, there would be a need for the mass acquisition of single-cell-level measurement data from many different coordinate points on a microfluidics chip. In order to be braced for such highly parallel experimental cases, a one of a kind microfluidic barcode system was invented to help label and identify the coordinates of locations on chips.

Another major achievement in this work was the development of a microfluidic measurement system capable of performing true time-lapse measurements of the fluorescence of individual cells. The benefit of having such a measurement system is quite a significant one in that the system allows the investigation of rare cellular events at the single-cell-level with which direct comparisons to the findings from single-cell-level *in silico* models can be made. This is yet another realisation of an idea that can close the gap in the dual evolutionary approach.

The novelty in introducing such a single-cell-level measurement system in conjunction with the programmatic data analysis necessitated the development of a novel algorithm to calculate spatio-temporal correspondences at the single-cell-level. The development of the correspondence algorithm was based on the clever concept of transforming an assignment problem into a graph search problem. This was a noteworthy achievement in such a way that the general principles of the algorithm could potentially find other uses in different fields of science or engineering. Another achievement as part of this work in microfluidics was that it served to be a good example of MDE in designing microfluidic systems, forming

an uncanny counterpoint to the maxim of DEA: "using models to help better the reality."

As it stands at the current moment, however, it takes a considerable amount of effort just to have a microfluidics chip made to be ready for a time-lapse microscopy run. What this entails is a call for future studies centered towards an integration effort to develop, based on the working pieces shown here, a measurement apparatus that is readily deployable with ease.

Chapter 8

General Discussion

Proof-of-concept implementations of various research ideas shown throughout Chapter 5, 6, and 7 were a concerted effort in an attempt to accomplish enabling technologies for the dual evolutionary approach. This work resulted in a well-defined methodological framework that can be extended for exploring the two solution domains to find converging solutions in a format the two domains can readily share. Implementations involving molecular devices and programmatic sequence analysis pipelines, shown in Chapter 5, addressed many important aspects with respect to realising the *in vivo* half of the framework. The work detailed in Chapter 6 provided comprehensive modelling methods that can be used in bridging the gap between the two domains. Microfluidics techniques detailed in Chapter 7 explored areas in which process automation can be achieved to gain automated control of experimental processes at the single cell level. The collective research work detailed in these chapters would serve to be an example to suggest how DEA can be realised in the future not only for an efficient exploration of the *in vivo* solution space, but also for facilitating cross-domain data transfer in a manner highly amenable to automation. While there were drawbacks in some implementations, other more successful aspects of the work showed promising results, enough to demonstrate the plausibility of the suggested methodological framework.

I believe the research work presented here established a good starting point for adopting the DEA framework to address the ultimate concern of dealing with design complexity in synthetic biology as well as of achieving automation. One of the key aspects explored in this study, was to define the search space of design.

The two domains of DEA rely on different conceptual perspectives to see design, hence their search spaces had to be defined from different conceptual elements. For instance, the *in vivo* domain would use genetic sequences as a primary handle to explore different genotypic possibilities (See Note **17** in Figure 3.1). So the unknown here are the phenotypes to be unveiled after applying changes to genetic sequences. On the contrary, the *in silico* domain would take the opposite direction, by working on model parameters that can change phenotypes (See Note **16** in Figure 3.1) before reasoning about genetic constructs that conform to the phenotypic changes. So the unknown are the sequences. These conceptual differences were reflected in the adoption of the molecular device for controlling random mutagenesis (i.e. the solution generator) as a primary means to explore the *in vivo* genotypic solution space (Section 5.2.3), and that of the genome-scale FBA as a primary means to explore the *in silico* phenotypic solution space (Section 6.3.3).

These differences attributed to the gap in interpreting and consolidating their search results. The cross-domain gap can be bridged by building compatible data interfaces between the two domains to make the task of resolving the differences in their search results straightforward. I investigated measurement technologies combining molecular biology, software and microfluidics to provide a way to merge these differences. This resulted in the development of variant analysis pipelines (Section 5.3.2.2) for measuring genotypes, and the microfluidic (Chapter 7) or the molecular (Section 5.2.2) means for measuring phenotypes. The consolidation of genotypic measurements (i.e. SNPs) and phenotypic measurements was briefly discussed in Section 6.4.2. However, a further investigation is still needed to suggest implementation-level details, as to how conclusions can be drawn on potential correlations between genes and phenotypes. This was left for future studies.

The molecular phenotypic measurement device was designed to enable a high-throughput assessment of fitness criteria in random *in vivo* solutions (Section 5.2). The molecular device was used to incorporate the function of evaluating environmental incentives or fitness landscape (Note **3, 11** in Figure 3.1) into a single mutant cell. Such a molecular technique can enable the screening of putative *in vivo* solutions in a massively parallel manner. Sequencing the genomes of putative solutions, in turn, can transfer the solutions resident *in vivo* to the *in silico* design domain (Note **21** in Figure 3.1). To help streamline this important, yet time-

consuming process, software pipelines were developed to automate the genome sequence assembly (Section 5.3.2.1).

In fact, the design of the molecular phenotypic measurement device suggested in this study has a huge drawback, in that its design is highly prone to false-positive measurements due to accumulating mutations on the measurement device. While the automated assembly and variant analysis pipelines can, in principle, be used to weed out false positives, having to deal with many false positives would be severely burdensome in practice. This shortcoming could potentially be overcome if a plasmid-based device was used instead of the chromosomally integrated device featured in this study. The stability of plasmid-based devices, unlike that of chromosomally integrated devices, is short-lived allowing mutant cells to lose the molecular device over a number of generations. Gram positive cells such as *B. subtilis* are known to exhibit low plasmid retention rates, and are naturally competent. This means that the cells can easily be made to lose or to take up plasmids by adjusting growth conditions. Such bacterial properties can be exploited to replenish mutant cells with fresh plasmids free of undesirable false-positive mutations during the course of evolutionary cycles. The investigation of this promising idea was left for future studies.

It was shown that bridging the gap between the two domains can increase the search efficiency of design problems in synthetic biology (Section 6.4.3). The increase in search efficiency would, in effect, amount to the reduction of search space or time to design. The DEA framework, as a consequence of such gain, would have a higher capacity for dealing with design complexity and provide a ground for mitigating the search space explosion issue discussed earlier. Having said this, one of the main argument points in this study was that partial solutions made available via *in vivo* evolutionary processes could be combined *in silico*, and vice versa, to achieve design optimisation requiring more complex solutions (Figure 6.18). It might be argued, though, that DEA is too cumbersome without offering any gains over directed evolution, given the discussion in Chapter 5 which presented directed evolution as an efficient means to facilitate the *in vivo* solution search. However, employing such an *in vivo* strategy alone is conceptually equivalent to the mathematical optimisation strategy called Stochastic Hill Climbing [95]. Such a strategy, according to the simulations presented in Chapter 6, would be a poor choice for

complex time-constrained problems that have combinatorially challenging search spaces.

In a similar vein, it is worth noting that the advantage of using EA over Hill Climbing (HC) was discussed in the mathematical optimisation context by Mitchell, Holland and Forrest [122]. Holland postulated [77, 85] that the power of EA lies at implicit parallelism, whereby making crossovers in the population can reduce the time to search for optimal solutions compared to only using random mutations. While the initial discussion of Holland was confined to the theory of mathematical optimisation in the context of computing science, the idea, in my opinion, can be transposed to synthetic biology without loss of generality. My research, at least in part, demonstrated that the general principle of Holland's argument about EA over HC still holds to be true in the context of adopting DEA over directed evolution in finding synthetic biological design solutions (Section 6.4.3).

In light of this argument, the DEA framework is a real-world case of EA, in which the concept of implicit parallelism was incorporated into the *in vivo* domain via the capability of recombining solutions in the *in silico* domain. This theorisation is also directly in line with the reasoning behind the idea that DEA would offer to be a more efficient platform than employing directed evolution in the *in vivo* domain alone. As such, this study established a groundwork for bolstering the premise that DEA is a powerful framework for design automation in synthetic biology.

Accomplishing the integration of different working concepts shown in this study to function together is significantly challenging a project in and of itself. Due to time constraints, much of the integration effort was reserved for future studies. For example, the fitness evaluation to guide the *in silico* domain, in the real-world scenario, would need to take the form of executing FBA on the genome-scale metabolic pathway model, as shown in Section 6.3.3.2. In lieu of FBA, the fitness evaluation in the proof-of-concept experiments was done by using a look-up table (LUT) out of the pre-agreed answersheet from Table 6.4. Nevertheless, this test case was enough for evaluating the validity of the DEA framework. It was evident from the simulation results (Figure 6.17) that the *in silico* domain could offer an efficient means to conduct population mating, which the *in vivo*

domain lacked. The *in silico* search could effectively recombine partial solutions to successfully explore the search space, leading to a high SNP count solution, otherwise unachievable. The *in silico* domain was able to reach SNP counts that the *in vivo* domain alone could not reach given the limited timeframe (Figure 6.18 vs 6.16). The results shown here are strongly suggestive of the possibility that the dual-evolutionary strategy, if its two domains were bridged to exchange SNP-level information in short iterative cycles, can mitigate the combinatorial explosion of solution space to a significant degree.

Single-cell-level spatio-temporal measurements demonstrated in this study would play an important role in evaluating the exit condition in the iterative design cycle of the DEA framework. The caveat is that single-cell-level measurements can produce high volumes of data, too cumbersome for manual curation and analysis. Motivated by this, the microfluidic measurement and analysis platform developed in this study was designed with an end-to-end lab automation in mind. To this end, I believe lab automation will serve to be one of the most important enabling technologies for design automation in synthetic biology in the future soon to come. Lab automation would be accomplished not by building microfluidic devices alone, but by incorporating software intelligence for the programmatic execution of device control and data analysis. This led to the development of microfluidic devices with structural features, such as microscopic barcode (Figure 7.7), specially designed to facilitate the development of software intelligence for use in highly parallel experimental automation. The microfluidics research done as part of this study was a clear demonstration that software research is indispensable to achieving lab automation. Hence, software should be an integral part of microfluidics research aimed at design automation in synthetic biology.

It was argued throughout this study that we need to shift gears in synthetic biology towards pursuing design automation. This study successfully highlighted some of the key technologies with which to envisage an automatic platform for designing complex synthetic biological systems. Driven by strong commercial interests, synthetic biology has been gravitating towards applications in metabolic engineering. Such a trend has made the line between metabolic engineering and synthetic biology quite blurry. My investigation too looked at an example of metabolic engineering in discussing design automation in synthetic biology. It

needs to be clarified though, that metabolic engineering is not synthetic biology but a subset of which. Synthetic biology, as a ground for a greater set of biological design problems, should be bracing for a higher level of complexity than that suited for metabolic engineering. To this end of supporting yet higher levels of design complexity, a fully automated synthetic biology design cycle is not an option but a necessity. This requisite calls for future studies on a plethora of subjects that have not been addressed in this study.

Some of the future works needed regarding this include, but are not limited to, the following. DNA sequence data, at the whole-genome scale, can be overwhelming, and the analysis of which requires significant amounts of computing power. Dealing with complex designs will necessarily result in the inundation of sequence data, pouring out of the *in vivo* design domain. It will be critically important to be able to support parallelism in the data analysis pipelines in order to help increase the data exchange throughput between the two evolutionary design domains. Also, a microfluidic platform supporting the programmatic execution of general-purpose lab protocols such as Gibson assembly would be highly desirable. As part of such a platform, it would greatly help to have a software control stack providing an API-level access to the platform. Building a microfluidics platform that supports automated execution of a suite of experiments required by the DEA framework would be considered a significant milestone to be achieved in the future. Provided with such a feature-complete lab automation platform together with a well-defined API, the design practice in synthetic biology will be completely redefined as envisioned in my research work.

In closing, the significance of this research work lies in developing and demonstrating key technologies that can, when extended and put together, embody what's ultimately envisioned by the DEA framework for accomplishing design automation in synthetic biology.

Appendices

A Appendix A

A.1 The genome-scale FBA in *B. subtilis*

The source code used in the study of genome-scale FBA was made available via the git repository: `https://bitbucket.org/sungshic/mpa.git`.

Downloading the source code:

```
$ git clone https://bitbucket.org/sungshic/mpa.git
```

Installing library dependencies on Ubuntu 14.04:

```
$ sudo apt-get update
$ sudo sh -c "echo 'deb http://download.opensuse.org/repositories/home:/fbergman:/libsbml/xUbuntu_14.04/ /'
     >> /etc/apt/sources.list.d/python-libsbml.list"
$ sudo apt-get update
$ sudo apt-get install python-libsbml
$ sudo apt-get install libglpk-dev
$ sudo apt-get install python-glpk
$ sudo apt-get install python-matplotlib
$ sudo apt-get install python-numpy
$ sudo apt-get install python-scipy
$ sudo pip install cobra

$ sudo sh -c "echo 'deb http://downloads.skewed.de/apt/trusty trusty universe
deb-src http://downloads.skewed.de/apt/trusty trusty universe' >> /etc/apt/sources.list"
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install python-graph-tool

$ sudo apt-get install python-reportlab
$ sudo pip install PyPDF2
$ sudo pip install pdfrw
$ sudo pip install rdflib

$ sudo apt-get install openjdk-7-jre-headless # need this for installing CPLEX
#install CPLEX package v12.6.3 # it is available from IBM, free for academic use.
$ cd /opt/ibm/ILOG/CPLEX_Studio1263/cplex/python/2.6/x86-64_linux/
$ sudo python setup.py install
```

A.2 NGS assembly and variant analysis pipelines

The source code used in the study of NGS assembly and variant analysis was made available via the following git repository:
`https://bitbucket.org/sungshic/dea_hybrid_assembler.git`.

Downloading the source code:

```
$ git clone https://bitbucket.org/sungshic/dea_hybrid_assembler.git
```

A Docker image (`sungshic/dea_hybrid_assembler:1.1`) was created to demonstrate how the assembly and variant analysis pipelines shown in Section 5.3.2 can be used for verifying the genomic correctness of the EVOt2 clone (Section 5.2.5). Executing the following in `bash` shell would pull the Docker image from Docker Hub to a local host, and make a folder (`assembly`) under the home directory. Docker version 1.12.3 (build 6b644ec) was used for the following tutorial.

Invoking a docker container for NGS assembly and variant analysis pipelines:

```
$ sudo docker pull sungshic/dea_hybrid_assembler:1.1
$ cd             # go to the home directory
$ mkdir assembly # and make a folder to store assembly results
```

The following command will run the `bash` command-line prompt on a docker container from the base image `sungshic/dea_hybrid_assembler:1.1` located at Docker Hub. This command, with the `-v` option, will also mount the local host folder `~/assembly` to the working directory in the container.

```
$ sudo docker run -it -v ~/assembly:/root/workspace/dea_hybrid_assembler/genomeassemblypipeline/data/gandalf/
     basespace/bsb1_evot2/assembly sungshic/dea_hybrid_assembler:1.1 /bin/bash
```

A.3 Processing cytometry data

The source code used in the study of programmatic processing of cytometry data was made available via the following git repository:

https://bitbucket.org/sungshic/fcs_analysis.git.

Downloading the source code:

```
$ git clone https://bitbucket.org/sungshic/fcs_analysis.git
```

A.4 Evolutionary algorithm *in silico*

The source code used in the study of applying evolutionary algorithm in the *in silico* design domain was made available via the following git repository:
https://bitbucket.org/sungshic/idea_silico.git.

Downloading the source code:

```
$ git clone https://bitbucket.org/sungshic/idea_silico.git
```

A.5 Graph-based image analysis for single-cell-level microfluidics

The source code used in the study of applying graph-based image analysis techniques in single-cell-level microfluidics was made available via the following git repository:
https://bitbucket.org/sungshic/gbimageanalyzer.git.

Downloading the source code:

```
$ git clone https://bitbucket.org/sungshic/gbimageanalyzer.git
```

A.6 Estimating relative molar mass of macro-molecules in biomass reaction

Table A.1: Composition of peptidoglycan polymer reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd03495	1	reactant	met polymer	C95H152N8O28P2	1916.20	1916.2
cpd15666	1	reactant	met polymer	C40H63N8O21R	991.97	991.97
cpd02229	1	by-product	met polymer	C55H90O7P2	925.24	925.24
cpd15665	1	product	met polymer	C80H125N16O42R	1982.93	1982.93

Table A.2: Composition of cell wall synthesis reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd11459	0.0145	reactant	met polymer	C420H692N30O391P30	13347.14	193.53353
cpd15665	0.453	reactant	met polymer	C80H125N16O42R	1982.93	898.26729
cpd15667	0.016	reactant	met polymer	C191H359N10O259P46R	8364.59	133.83344
cpd15668	0.0112	reactant	met polymer	C326H584N55O304P46R	11563.08	129.506496
cpd15669	0.00808	reactant	met polymer	C461H809N10O484P46R	15660.90	126.540072
cpd15666	0.48828	by-product	met polymer	C40H63N8O21R	991.97	484.3591116
cpd15664	1	product	Cell wall	N/A	997.32	997.3217164

Table A.3: Composition of protein synthesis reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd00023	0.4928	reactant	L-glu	C5H8NO4	146.12	72.007936
cpd00033	0.7723	reactant	L-gly	C2H5NO2	75.07	57.976561
cpd00035	0.5051	reactant	L-ala	C3H7NO2	89.09	44.999359
cpd00039	0.6114	reactant	L-lys	C6H15N2O2	147.19	89.991966
cpd00041	0.2801	reactant	L-asp	C4H6NO4	132.09	36.998409
cpd00051	0.3653	reactant	L-arg	C6H15N4O2	175.21	64.004213
cpd00053	0.4928	reactant	L-gln	C5H10N2O3	146.14	72.017792
cpd00054	0.4091	reactant	L-ser	C3H7NO3	105.09	42.992319
cpd00060	0.2145	reactant	L-met	C5H11NO2S	149.21	32.005545
cpd00065	0.1028	reactant	L-trp	C11H12N2O2	204.22	20.993816
cpd00066	0.3329	reactant	L-phe	C9H11NO2	165.19	54.991751
cpd00069	0.2097	reactant	L-tyr	C9H11NO3	181.19	37.995543
cpd00084	0.1073	reactant	L-cys	C3H7NO2S	121.16	13.000468
cpd00107	0.6555	reactant	L-leu	C6H13NO2	131.17	85.981935
cpd00119	0.1546	reactant	L-his	C6H9N3O2	155.15	23.98619
cpd00129	0.3041	reactant	L-pro	C5H8NO2	114.12	34.703892
cpd00132	0.2801	reactant	L-asn	C4H8N2O3	132.12	37.006812
cpd00156	0.5807	reactant	L-val	C5H11NO2	117.15	68.029005
cpd00161	0.3526	reactant	L-thr	C4H9NO3	119.12	42.001712
cpd00322	0.5107	reactant	L-ile	C6H13NO2	131.17	66.988519
cpd11463	1	product	Protein	N/A	998.67	998.673743

Table A.4: Composition of lipid synthesis reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd15529	0.01816	reactant	metabolite	C33H66NO8P	635.85	11.547036
cpd15531	0.07327	reactant	metabolite	C37H74NO8P	691.96	50.6999092
cpd15533	0.02483	reactant	metabolite	C41H82NO8P	748.06	18.5743298
cpd15536	0.005699	reactant	metabolite	C34H66O10P	665.85	3.79467915
cpd15538	0.02308	reactant	metabolite	C38H74O10P	721.96	16.6628368
cpd15540	0.007846	reactant	metabolite	C42H82O10P	778.07	6.10473722
cpd15695	0.05369	reactant	metabolite	C39H78NO8P	720.01	38.6573369
cpd15696	0.1262	reactant	metabolite	C39H78NO8P	720.01	90.865262
cpd15697	0.008684	reactant	metabolite	C33H66NO8P	635.85	5.5217214
cpd15698	0.1459	reactant	metabolite	C35H70NO8P	663.90	96.86301
cpd15699	0.2473	reactant	metabolite	C35H70NO8P	663.90	164.18247
cpd15700	0.0341	reactant	metabolite	C37H74NO8P	691.96	23.595836
cpd15707	0.008709	reactant	metabolite	C53H98O20	1055.33	9.19086897
cpd15708	0.002095	reactant	metabolite	C49H90O20	999.22	2.0933659
cpd15709	0.003029	reactant	metabolite	C57H106O20	1111.44	3.36655176
cpd15710	0.006468	reactant	metabolite	C55H102O20	1083.38	7.00730184
cpd15711	0.0152	reactant	metabolite	C55H102O20	1083.38	16.467376
cpd15712	0.001002	reactant	metabolite	C49H90O20	999.22	1.00121844
cpd15713	0.0171	reactant	metabolite	C51H94O20	1027.28	17.566488
cpd15714	0.02897	reactant	metabolite	C51H94O20	1027.28	29.7603016
cpd15715	0.004053	reactant	metabolite	C53H98O20	1055.33	4.27725249
cpd15722	0.01694	reactant	metabolite	C40H78O10P	750.01	12.7051694
cpd15723	0.03982	reactant	metabolite	C40H78O10P	750.01	29.8653982
cpd15724	0.002726	reactant	metabolite	C34H66O10P	665.85	1.8151071
cpd15725	0.04589	reactant	metabolite	C36H70O10P	693.91	31.8435299
cpd15726	0.07776	reactant	metabolite	C36H70O10P	693.91	53.9584416
cpd15727	0.01074	reactant	metabolite	C38H74O10P	721.96	7.7538504
cpd15728	0.01447	reactant	metabolite	C47H88O15	893.19	12.9244593
cpd15729	0.003517	reactant	metabolite	C43H80O15	837.08	2.94401036
cpd15730	0.004989	reactant	metabolite	C51H96O15	949.30	4.7360577
cpd15731	0.0107	reactant	metabolite	C49H92O15	921.24	9.857268
cpd15732	0.02515	reactant	metabolite	C49H92O15	921.24	23.169186
cpd15733	0.001682	reactant	metabolite	C43H80O15	837.08	1.40796856
cpd15734	0.02855	reactant	metabolite	C45H84O15	865.14	24.699747
cpd15735	0.04838	reactant	metabolite	C45H84O15	865.14	41.8554732
cpd15736	0.006735	reactant	metabolite	C47H88O15	893.19	6.01563465
cpd15737	0.01119	reactant	metabolite	C41H78O10	731.05	8.1804495
cpd15738	0.00276	reactant	metabolite	C37H70O10	674.94	1.8628344
cpd15739	0.003807	reactant	metabolite	C45H86O10	787.15	2.99668005
cpd15740	0.008216	reactant	metabolite	C43H82O10	759.10	6.2367656
cpd15741	0.01931	reactant	metabolite	C43H82O10	759.10	14.658221
cpd15742	0.00132	reactant	metabolite	C37H70O10	674.94	0.8909208
cpd15743	0.02224	reactant	metabolite	C39H74O10	703.00	15.63472
cpd15744	0.03768	reactant	metabolite	C39H74O10	703.00	26.48904
cpd15745	0.005208	reactant	metabolite	C41H78O10	731.05	3.8073084
cpd15782	0.002845	reactant	metabolite	C44H88N2O11P	852.15	2.42436675
cpd15783	0.0006934	reactant	metabolite	C40H80N2O11P	796.04	0.551974136
cpd15784	0.0009777	reactant	metabolite	C48H96N2O11P	908.25	0.887996025
cpd15785	0.0021	reactant	metabolite	C46H92N2O11P	880.20	1.84842
cpd15786	0.004935	reactant	metabolite	C46H92N2O11P	880.20	4.343787
cpd15787	0.0003316	reactant	metabolite	C40H80N2O11P	796.04	0.263966864
cpd15788	0.005621	reactant	metabolite	C42H84N2O11P	824.09	4.63220989
cpd15789	0.009523	reactant	metabolite	C42H84N2O11P	824.09	7.84780907
cpd15790	0.001324	reactant	metabolite	C44H88N2O11P	852.15	1.1282466
cpd15791	0.0005977	reactant	metabolite	C73H140O17P2	1351.82	0.807982814
cpd15792	0.0001484	reactant	metabolite	C65H124O17P2	1239.61	0.183958124
cpd15793	0.0002022	reactant	metabolite	C81H156O17P2	1464.04	0.296028888
cpd15794	0.0004375	reactant	metabolite	C77H148O17P2	1407.93	0.615969375
cpd15795	0.001028	reactant	metabolite	C77H148O17P2	1407.93	1.44735204
cpd15797	0.001192	reactant	metabolite	C69H132O17P2	1295.72	1.54449824
cpd15798	0.002019	reactant	metabolite	C69H132O17P2	1295.72	2.61605868
cpd15799	0.0002781	reactant	metabolite	C73H140O17P2	1351.82	0.375941142
cpd15800	1	product	Lipid	N/A	995.92	995.9246962

Table A.5: Composition of LAC synthesis reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd15746	0.004866	reactant	metabolite	C119H232O135P24	4566.40	22.2201024
cpd15747	0.001122	reactant	metabolite	C115H224O135P24	4510.29	5.06054538
cpd15748	0.001761	reactant	metabolite	C123H240O135P24	4622.50	8.1402225
cpd15749	0.003687	reactant	metabolite	C121H236O135P24	4594.45	16.93973715
cpd15750	0.008667	reactant	metabolite	C121H236O135P24	4594.45	39.82009815
cpd15751	0.0005365	reactant	metabolite	C115H224O135P24	4510.29	2.419770585
cpd15752	0.009356	reactant	metabolite	C117H228O135P24	4538.34	42.46070904
cpd15753	0.01585	reactant	metabolite	C117H228O135P24	4538.34	71.932689
cpd15754	0.002264	reactant	metabolite	C119H232O135P24	4566.40	10.3383296
cpd15755	0.002269	reactant	metabolite	C263H472O255P24	8457.76	19.19065744
cpd15756	0.0005201	reactant	metabolite	C259H464O255P24	8401.66	4.369703366
cpd15757	0.0008257	reactant	metabolite	C267H480O255P24	8513.87	7.029902459
cpd15758	0.001724	reactant	metabolite	C265H476O255P24	8485.81	14.62953644
cpd15759	0.004053	reactant	metabolite	C265H476O255P24	8485.81	34.39298793
cpd15760	0.0002488	reactant	metabolite	C259H464O255P24	8401.66	2.090333008
cpd15761	0.00435	reactant	metabolite	C261H468O255P24	8429.71	36.6692385
cpd15762	0.00737	reactant	metabolite	C261H468O255P24	8429.71	62.1269627
cpd15763	0.001056	reactant	metabolite	C263H472O255P24	8457.76	8.93139456
cpd15764	0.002032	reactant	metabolite	C311H544N24O255P24	9443.00	19.188176
cpd15765	0.0004655	reactant	metabolite	C307H536N24O255P24	9386.90	4.36960195
cpd15766	0.0007401	reactant	metabolite	C315H552N24O255P24	9499.11	7.030291311
cpd15767	0.001545	reactant	metabolite	C313H548N24O255P24	9471.06	14.6327877
cpd15768	0.003631	reactant	metabolite	C313H548N24O255P24	9471.06	34.38941886
cpd15769	0.0002227	reactant	metabolite	C307H536N24O255P24	9386.90	2.09046263
cpd15770	0.003895	reactant	metabolite	C309H540N24O255P24	9414.95	36.67123025
cpd15771	0.006599	reactant	metabolite	C309H540N24O255P24	9414.95	62.12925505
cpd15772	0.0009457	reactant	metabolite	C311H544N24O255P24	9443.00	8.9302451
cpd15773	0.006441	reactant	metabolite	C191H352N24O159P24	6272.26	40.39962666
cpd15774	0.00148	reactant	metabolite	C187H342N24O159P24	6214.14	9.1969272
cpd15775	0.002339	reactant	metabolite	C195H360N24O159P24	6328.37	14.80205743
cpd15776	0.004889	reactant	metabolite	C193H356N24O159P24	6300.31	30.80221559
cpd15777	0.01149	reactant	metabolite	C193H356N24O159P24	6300.31	72.3905619
cpd15778	0.0007078	reactant	metabolite	C187H344N24O159P24	6216.16	4.399798048
cpd15779	0.01236	reactant	metabolite	C189H348N24O159P24	6244.21	77.1784356
cpd15780	0.02094	reactant	metabolite	C189H349N24O159P24	6245.22	130.7749068
cpd15781	0.002997	reactant	metabolite	C191H352N24O159P24	6272.26	18.79796322
cpd15670	1	product	LAC	N/A	996.94	996.9368815

Table A.6: Composition of DNA synthesis reaction of *B. subtilis* genome-scale metabolic model

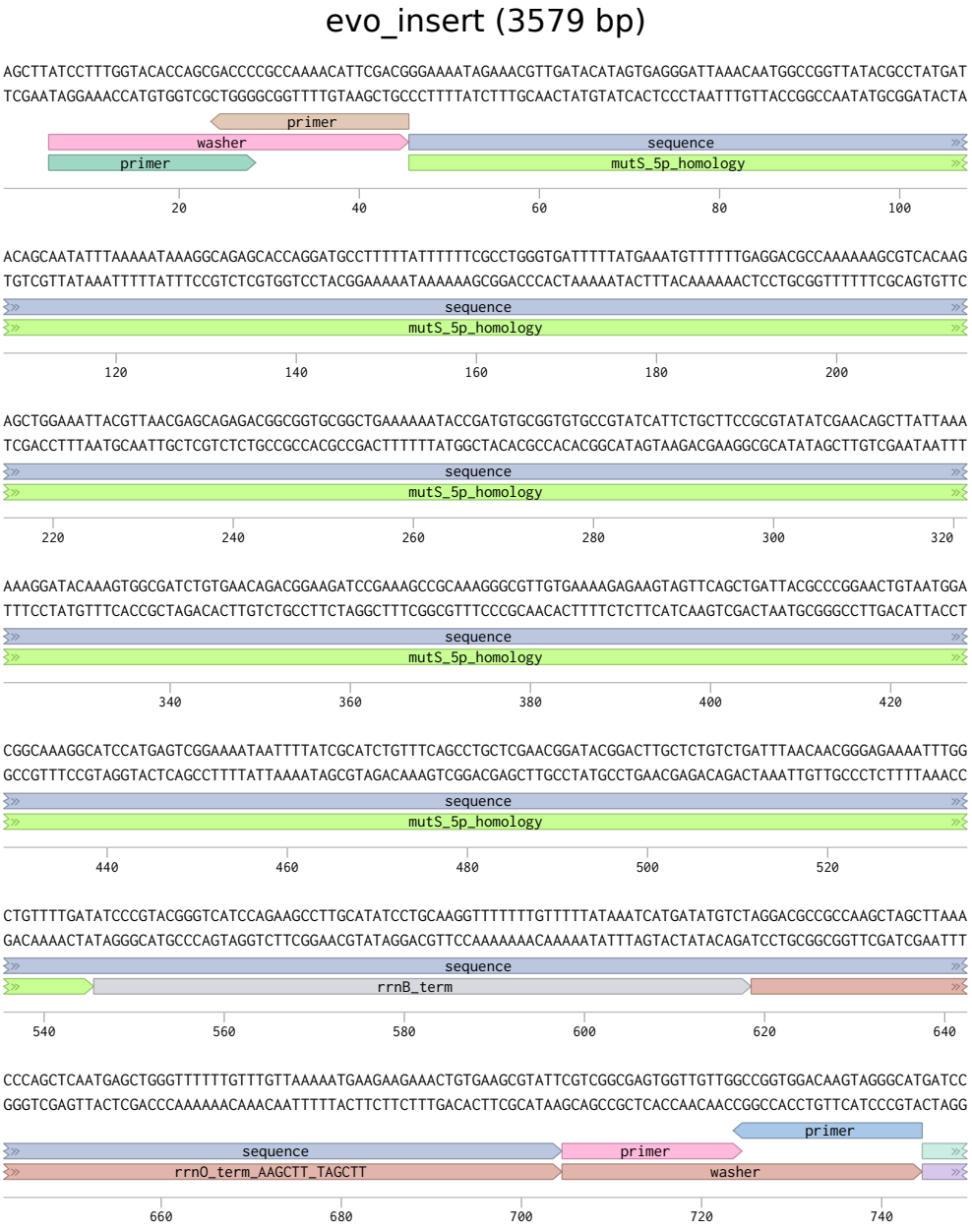
Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd00115	0.884	reactant	dATP	C10H13N5O12P3	488.16	431.53344
cpd00241	0.6692	reactant	dGTP	C10H13N5O13P3	504.16	337.383872
cpd00356	0.6684	reactant	dCTP	C9H13N3O13P3	464.13	310.224492
cpd00357	0.8807	reactant	dTTP	C10H14N2O14P3	479.14	421.978598
cpd00012	3.1023	by-product	PPi	H2O7P2	175.96	545.880708
cpd11461	1	product	DNA	N/A	955.24	955.239694

Table A.7: Composition of mRNA synthesis reaction of *B. subtilis* genome-scale metabolic model

Cmp ID	Stoichiometry	Rxn role	Kind	Molecular formula	RMMg/mol	NMWg/mol_{prod}
cpd00002	0.7706	reactant	ATP	C10H13N5O13P3	504.16	388.505696
cpd00038	0.9496	reactant	GTP	C10H13N5O14P3	520.16	493.943936
cpd00052	0.5853	reactant	CTP	C9H13N3O14P3	480.13	281.020089
cpd00062	0.6331	reactant	UTP	C9H12N2O15P3	481.12	304.597072
cpd00012	2.9386	by-product	PPi	H2O7P2	175.96	517.076056
cpd11462	1	product	mRNA	N/A	950.99	950.990737

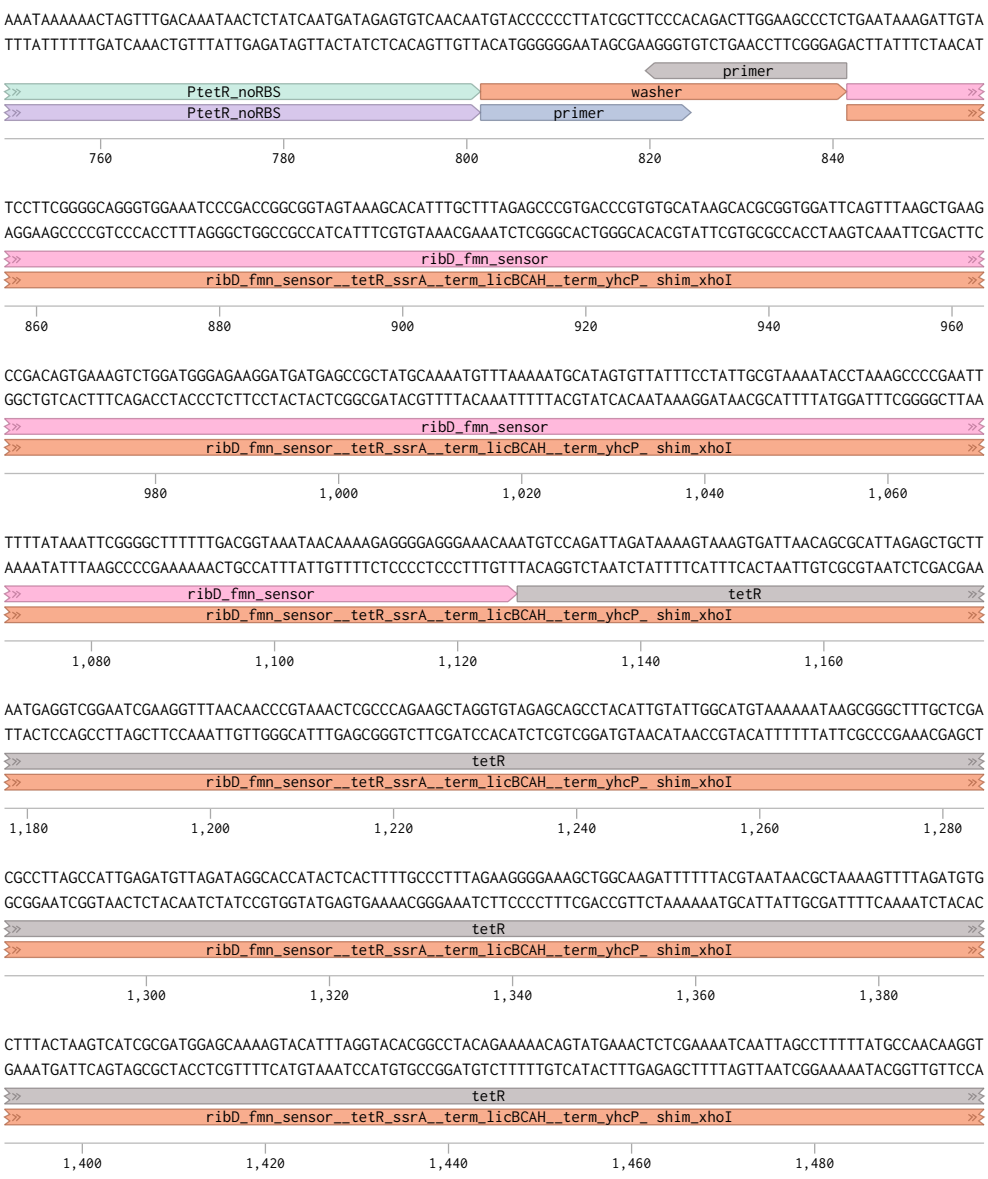
B Appendix B

B.1 The evo_insert construct

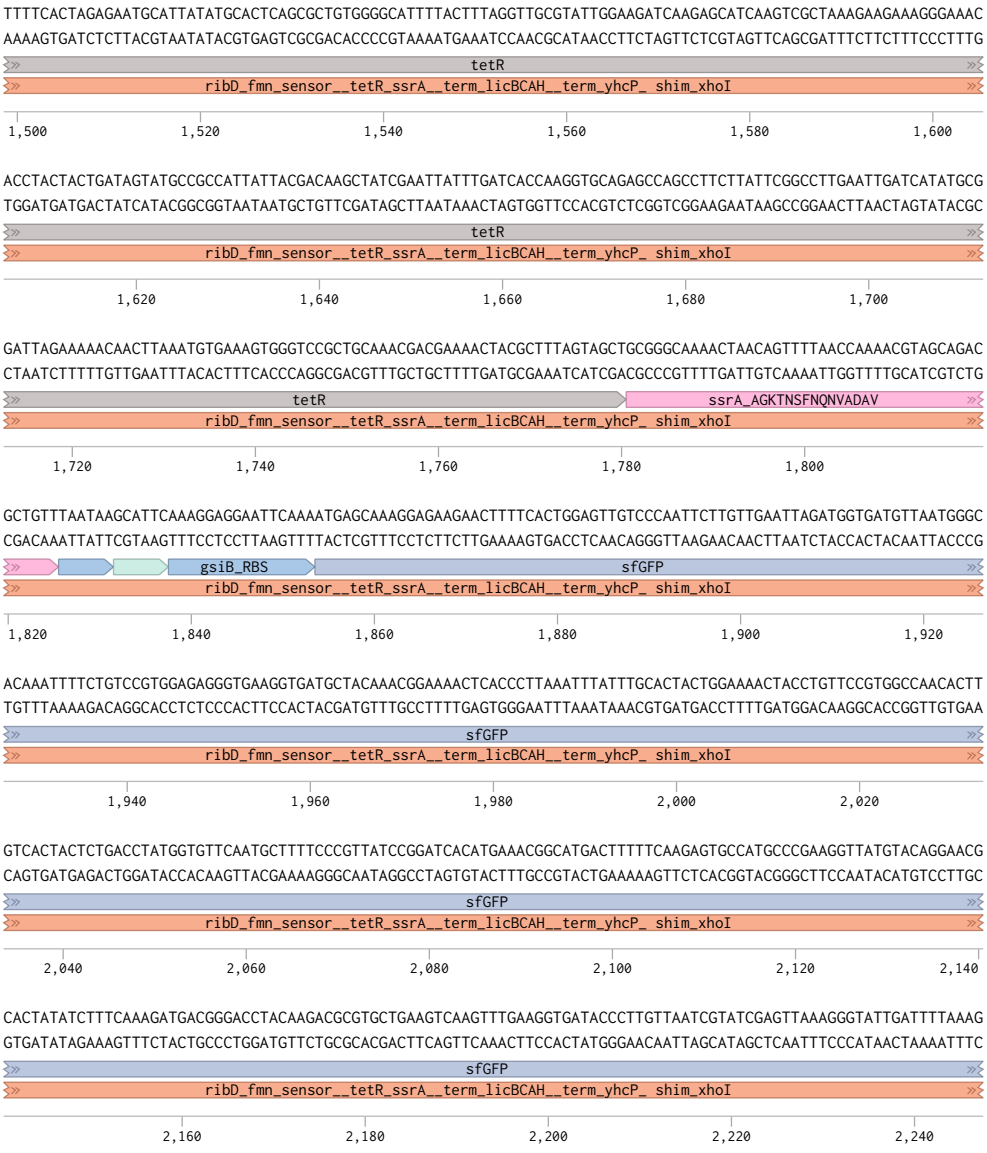


1/5

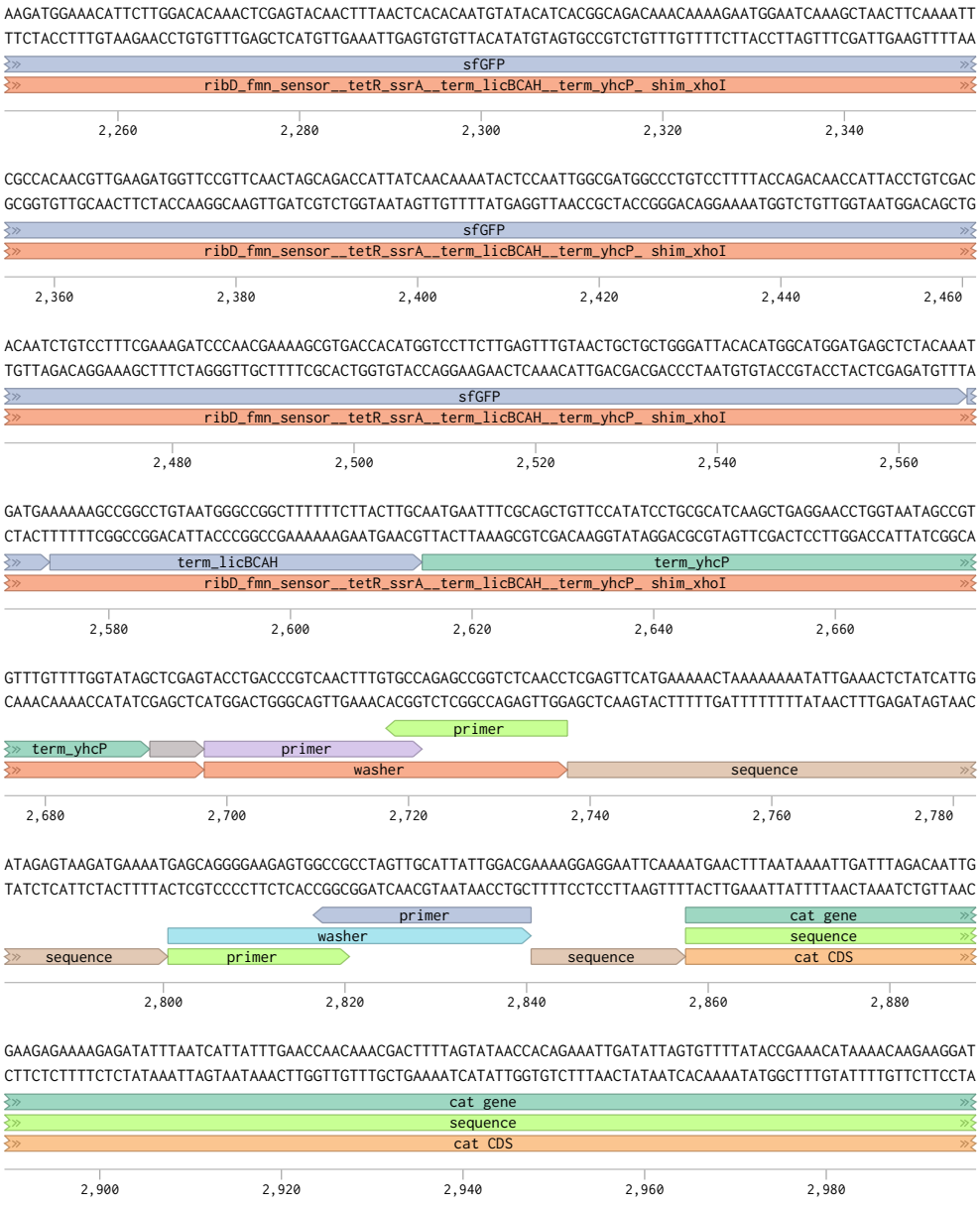
evo_insert (3579 bp) (from 750-1498 bp)



evo_insert (3579 bp) (from 1499-2247 bp)

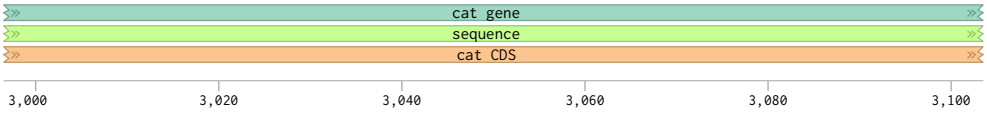


evo_insert (3579 bp) (from 2248-2996 bp)

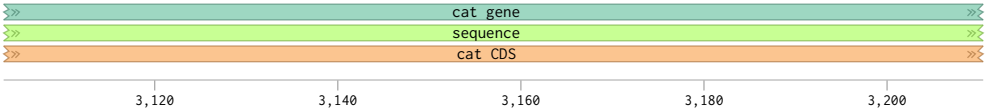

AAGATGGAAACATTCTTGGACACAAACTCGAGTACAACTTTAACTCACACAATGTATACATCACGGCAGACAAACAAAAGAATGGAATCAAAGCTAACTTCAAAATT
TTCTACCTTTGTAAGAACCTGTGTTTGAGCTCATGTTGAAATTGAGTGTGTTACATATGTAGTGCCGTCTGTTTGTTTTCTTACCTTAGTTTCGATTGAAGTTTTAA
sfGFP
ribD_fmn_sensor__tetR_ssrA__term_licBCAH__term_yhcP_ shim_xhoI
2,260
2,280
2,300
2,320
2,340
CGCCACAACGTTGAAGATGGTTCCGTTCAACTAGCAGACCATTATCAACAAAATACTCCAATTGGCGATGGCCCTGTCCTTTTACCAGACAACCATTACCTGTCGAC
GCGGTGTTGCAACTTCTACCAAGGCAAGTTGATCGTCTGGTAATAGTTGTTTTATGAGGTTAACCGCTACCGGGACAGGAAAATGGTCTGTTGGTAATGGACAGCTG
sfGFP
ribD_fmn_sensor__tetR_ssrA__term_licBCAH__term_yhcP_ shim_xhoI
2,360
2,380
2,400
2,420
2,440
2,460
ACAATCTGTCCTTTCGAAAGATCCCAACGAAAAGCGTGACCACATGGTCCTTCTTGAGTTTGTAACTGCTGCTGGGATTACACATGGCATGGATGAGCTCTACAAAT
TGTTAGACAGGAAAGCTTTCTAGGGTTGCTTTTCGCACTGGTGTACCAGGAAGAACTCAAACATTGACGACGACCCTAATGTGTACCGTACCTACTCGAGATGTTTA
sfGFP
ribD_fmn_sensor__tetR_ssrA__term_licBCAH__term_yhcP_ shim_xhoI
2,480
2,500
2,520
2,540
2,560
GATGAAAAAAGCCGGCCTGTAATGGGCCGGCTTTTTTCTTACTTGCAATGAATTTCGCAGCTGTTCCATATCCTGCGCATCAAGCTGAGGAACCTGGTAATAGCCGT
CTACTTTTTTCGGCCGGACATTACCCGGCCGAAAAAAGAATGAACGTTACTTAAAGCGTCGACAAGGTATAGGACGCGTAGTTCGACTCCTTGGACCATTATCGGCA
term_licBCAH
term_yhcP
ribD_fmn_sensor__tetR_ssrA__term_licBCAH__term_yhcP_ shim_xhoI
2,580
2,600
2,620
2,640
2,660
GTTTGTTTTGGTATAGCTCGAGTACCTGACCCGTCAACTTTGTGCCAGAGCCGGTCTCAACCTCGAGTTCATGAAAAACTAAAAAAAATATTGAAACTCTATCATTG
CAAACAAAACCATATCGAGCTCATGGACTGGGCAGTTGAAACACGGTCTCGGCCAGAGTTGGAGCTCAAGTACTTTTTGATTTTTTTTATAACTTTGAGATAGTAAC
primer
term_yhcP
primer
washer
sequence
2,680
2,700
2,720
2,740
2,760
2,780
ATAGAGTAAGATGAAAATGAGCAGGGGAAGAGTGGCCGCCTAGTTGCATTATTGGACGAAAAGGAGGAATTCAAAATGAACTTTAATAAAATTGATTTAGACAATTG
TATCTCATTCTACTTTTACTCGTCCCCTTCTCACCGGCGGATCAACGTAATAACCTGCTTTTCCTCCTTAAGTTTTACTTGAAATTATTTTAACTAAATCTGTTAAC
primer
cat gene
washer
sequence
sequence
primer
sequence
cat CDS
2,800
2,820
2,840
2,860
2,880
GAAGAGAAAAGAGATATTTAATCATTATTTGAACCAACAAACGACTTTTAGTATAACCACAGAAATTGATATTAGTGTTTTATACCGAAACATAAAACAAGAAGGAT
CTTCTCTTTTCTCTATAAATTAGTAATAAACTTGGTTGTTTGCTGAAAATCATATTGGTGTCTTTAACTATAATCACAAAATATGGCTTTGTATTTTGTTCTTCCTA
cat gene
sequence
cat CDS
2,900
2,920
2,940
2,960
2,980

evo_insert (3579 bp) (from 2997-3579 bp)

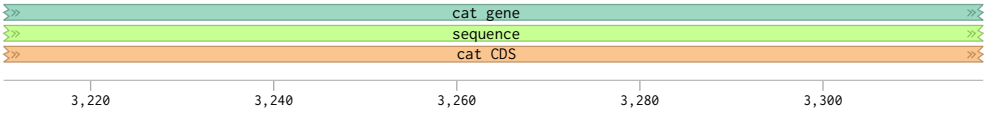
ATAAATTTTACCCTGCATTTATTTTCTTAGTGACAAGGGTGATAAACTCAAATACAGCTTTTAGAACTGGTTACAATAGCGACGGAGAGTTAGGTTATTGGGATAAG
TATTTAAAATGGGACGTAAATAAAAGAATCACTGTTCCCACTATTTGAGTTTATGTCGAAAATCTTGACCAATGTTATCGCTGCCTCTCAATCCAATAACCCTATTC



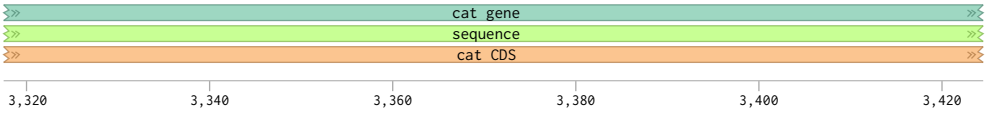
TTAGAGCCACTTTATACAATTTTTGATGGTGTATCTAAAACATTCTCTGGTATTTGGACTCCTGTAAAGAATGACTTCAAAGAGTTTTATGATTTATACCTTTCTGA
AATCTCGGTGAAATATGTTAAAAACTACCACATAGATTTTGTAAGAGACCATAAACCTGAGGACATTTCTTACTGAAGTTTCTCAAAATACTAAATATGGAAAGACT



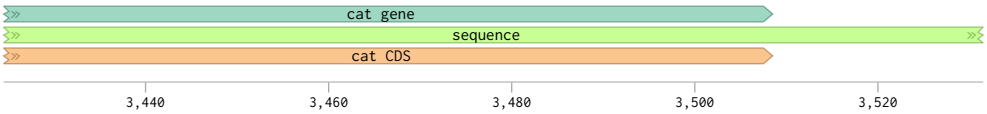
TGTAGAGAAATATAATGGTTCGGGGAAATTGTTTCCCAAAACACCTATACCTGAAAATGCTTTTTCTCTTTCTATTATTCCATGGACTTCATTTACTGGGTTTAACT
ACATCTCTTTATATTACCAAGCCCCTTTAACAAAGGGTTTTGTGGATATGGACTTTTACGAAAAAGAGAAAGATAATAAGGTACCTGAAGTAAATGACCCAAATTGA



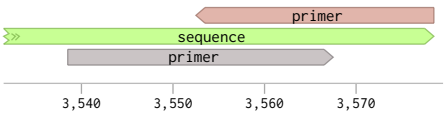
TAAATATCAATAATAATAGTAATTACCTTCTACCCATTATTACAGCAGGAAAATTCATTAATAAAGGTAATTCAATATATTTACCGCTATCTTTACAGGTACATCAT
ATTTATAGTTATTATTATCATTAATGGAAGATGGGTAATAATGTCGTCCTTTTAAGTAATTATTTCCATTAAGTTATATAAATGGCGATAGAAATGTCCATGTAGTA



TCTGTTTGTGATGGTTATCATGCAGGATTGTTTATGAACTCTATTCAGGAATTGTCAGATAGGCCTAATGACTGGCTTTTATAATATGAGATAATGCCGACTGTACT
AGACAAACACTACCAATAGTACGTCCTAACAAATACTTGAGATAAGTCCTTAACAGTCTATCCGGATTACTGACCGAAAATATTATACTCTATTACGGCTGACATGA



TTTTACAGTCGGTTTTCTAATGTCACTAACCTGCCCCGTTAGTTGAAG
AAAATGTCAGCCAAAAGATTACAGTGATTGGACGGGGCAATCAACTTC



5/5

B.2 The pMUTIN4 plasmid vector

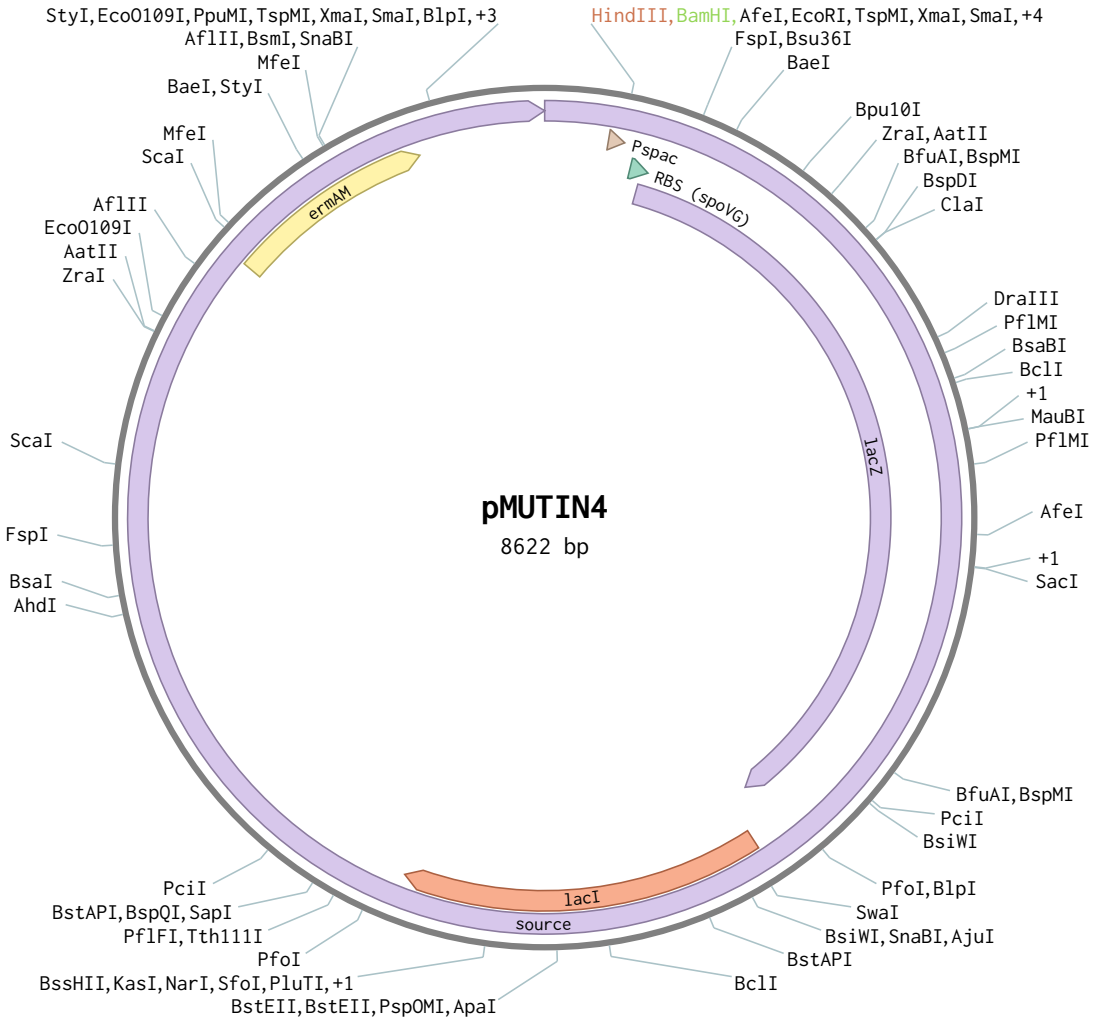


Figure B.1: The pMUTIN4 plasmid map

B.3 The pSG1729 plasmid vector

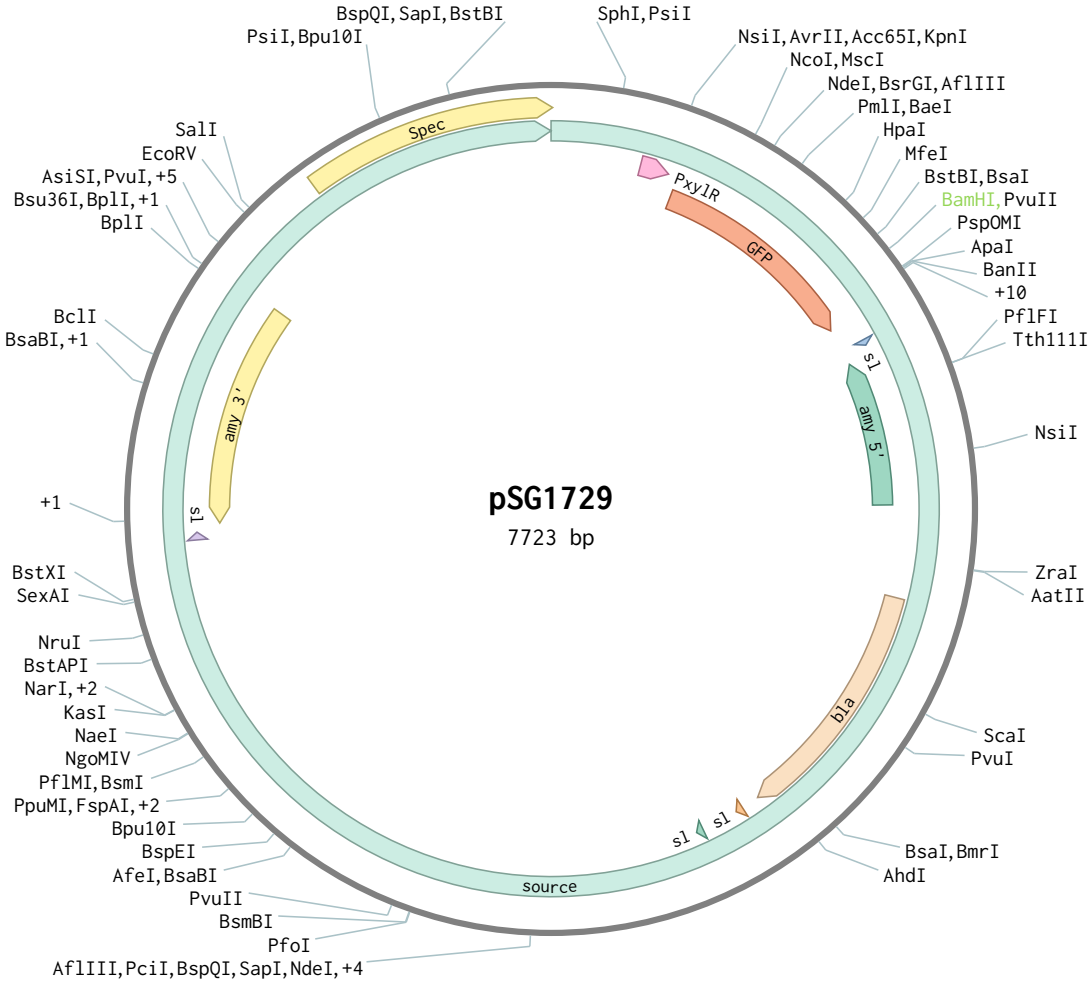


Figure B.2: The pSG1729 plasmid map

B.4 Primers used for colony PCR

Table B.1: Primers used for colony PCR of *B. subtilis* BSB1 with pMUTIN4_evo

Primer ID	Oligo sequence	Length	GC(%)	T_m(°C)
evo_insert_fwd1	AGCGTTACATGGAGCTGGTGCG	22	59	73
evo_insert_rev1	CGTCCAATAATGCAACTAGGCGGC	24	54	71

Table B.2: Primers used for colony PCR of *B. subtilis* BSB1 with pSG1729_EVOt2

Primer ID	Oligo sequence	Length	GC(%)	T_m(°C)
pEVOt2_colpcr_fwd	AGAGTGTGATGATAAGTGG	19	42	56
pEVOt2_colpcr_rev	TTGAATTCCTCCTTTTCGTC	20	40	58
colpcr_pSG_spec_cds_fwd	CTGAACCAAATAGATATACTCC	22	36	56
colpcr_amyE_3p_end_rev	AAGTATTTCACATTTATATTGTGC	24	25	55

B.5 Media recipes

B.5.1 Spizizen Minimal Media (SMM) - per 1 L solution

- 2.0 g ammonium sulphate ($(NH_4)_2SO_4$)
- 14.0 g dipotassium hydrogen phosphate (K_2HPO_4)
- 6.0 g potassium dihydrogen phosphate (KH_2PO_4)
- 1.0 g sodium citrate dehydrate (trisodium citrate), (Na_3.citrate.$2H_2O$)
- 0.2 g magnesium sulphate, ($MgSO_4.7H_2O$)
- Make up to 1 L using MilliQ H_2O, and use a hotplate-stirrer to mix and dissolve the ingredients.
- Split the solution into 5X 200 mL bottles, and autoclave them using a sensitive cycle.

B.5.2 MM competence media - per 5 mL

This media needs to be prepared just prior to use.

- 5 mL SMM media (see B.5.1)
- 62.5 µL Solution E (40 % glucose)
- 50 µL tryptophan solution[1]
- 30 µL Solution F (1 M $MgSO_4$ or $MnSO_4$)
- 5 µL casamino acid
- 2.5 µL $Fe-NH_4-$citrate

B.5.3 Starvation media - per 5 mL

This media needs to be prepared just prior to use.

- 5 mL SMM media (see B.5.1)
- 62.5 µL Solution E (40 % glucose)
- 30 µL Solution F (1 M $MgSO_4$ or $MnSO_4$)

[1] Necessary for 168, a tryptophan auxotroph strain

C Appendix C

C.1 Analysis of sub-micron microfluidics wafer design and production

The production of microfluidics chips requires a complex design and manufacture process. This process is increasingly more complex when sub-micron features are required within the chip. Designs for microfluidics chips were produced using L-edit software, with a constant channel depth across the designs of 1.4 µm. The designs were used to manufacture a wafer mask and subsequently produce a silicon wafer with the design etched upon it with lithography techniques.

In order to achieve sub-micron scale features two masks were prepared: one for shallow features (i.e. 1.4 µm) and one for deep features (i.e. 40 µm). The wafers were then prepared using the process described in Figure 7.1 and Section 7.2.

This novel production process was meticulously designed to minimise the costly optimisation need at the foundry. One of the fabricated silicon wafers was characterised using electron microscopy, in order to determine whether this novel process can successfully produce the intended feature dimensions of my sub-micron scale microfluidics design.

The silicon wafer was cleaned by soaking in the order of isopropanol, acetone, and isopropanol. An intermediate mould was made using a thin layer of hard-PDMS for the microfluidic features and a thick layer of soft-PDMS as a support. Please refer to Section C.3 for the details on the wafer cleaning and the mould making protocols. Then, the intermediate mould was surface salinized, and cast with soft-PDMS to make a reverse mould. This reverse mould was prepared for analysis on the scanning electron microscope (SEM). Microfluidics designs to be analysed were selected from across the wafer as shown in Figure C.1.

The reverse mould was diced up using a scalpel and samples with a maximum size of 15 mm by 15 mm were cut out. Chip samples were cut diagonally across the design to expose a cross section, for depth measurements. Samples were gold plated after being mounted onto circular SEM stubs using carbon tapes (Electron Microscopy Research services, Newcastle University). The samples were mounted

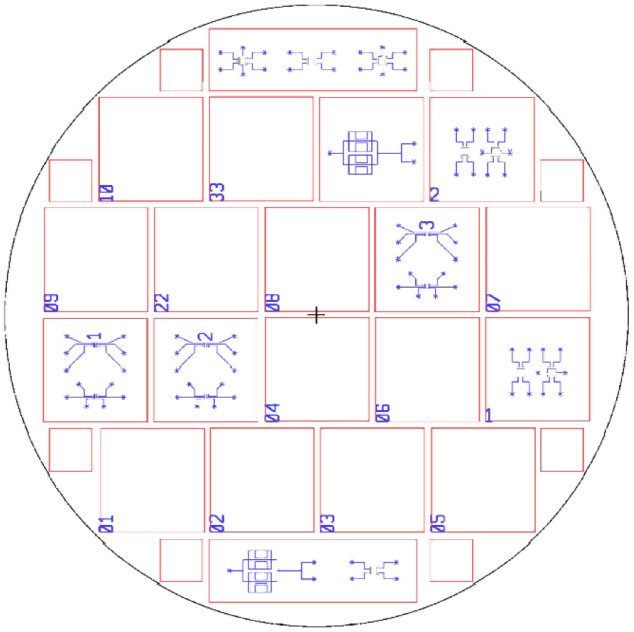
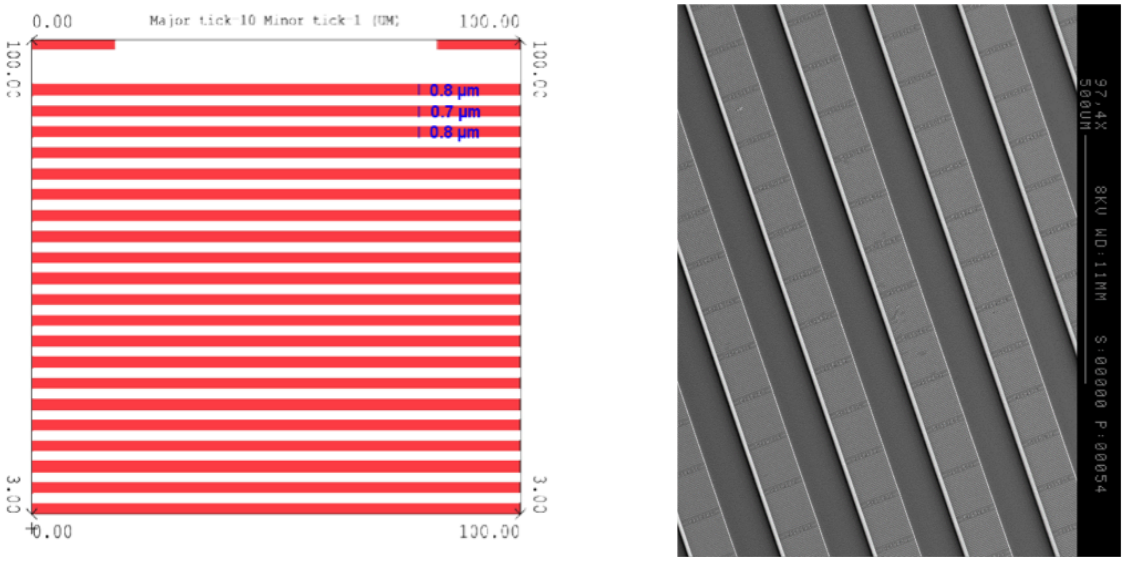


Figure C.1: The SEM sample locations on the wafer: a bird's eye view

into the SEM (Cambridge Stereoscan 240) and images taken of the top surface of the samples. SEM samples were then tilted to allow images showing the channel depths to be taken. Sample images were then measured using ImageJ software. The scale bar was measured first and the scale applied to the rest of the image. Features of interest were then measured by applying lines to the image at high magnification and using the measurement function.

For depth measurements, measured lines from the corresponding flat image were transposed onto the same feature in the tilted image to compare the distortion caused by perspective when looking at a three dimensional image (the scale bar at the top of the tilted images is only accurate at the point of focus, due to the effect of perspective). A series of lines were then drawn to assist with the measurement and the measurement function used for final measurements. All lines drawn and measurements taken appear on the images in the following section.



(a) (b)

Figure C.2: Chip 1: design view vs product SEM view

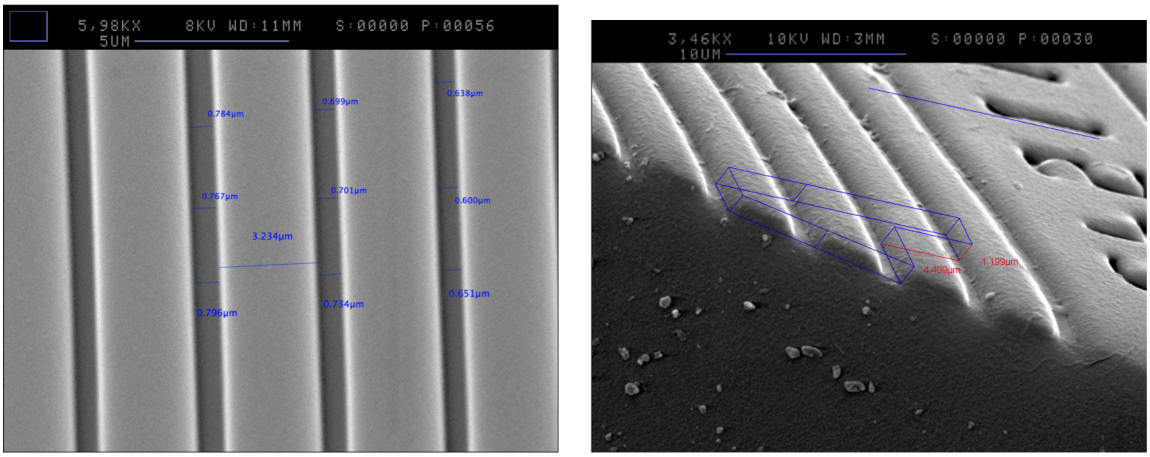
(a) Design of one square repeat panel within Chip 1, including measurements of expected final channel widths. **(b)** Overview SEM image showing the square panels repeating across Chip 1, with deep gutters between rows of square panels.

C.2 Measurements of wafer mould samples

C.2.1 Chip 1

The width of the channels was designed to be 2.2 µm, 2.3 µm and 2.4 µm in a repeating pattern. The backfill depth was 0.8 µm of oxide from each surface. This gives final channel widths of 0.6 µm, 0.7 µm and 0.8 µm (Figure C.2). The depth of the shallow channels was designed to be 1.4 µm.

The SEM images in Figure C.3 shows the channel width and depth measurements. The depth of channels within Chip 1 was measured at 1.2 µm. The width measurements were summarised in Figure C.4 to have respective mean channel widths of 0.63 µm, 0.71 µm and 0.78 µm.



(a) (b)

Figure C.3: Chip 1: SEM images with line overlays

The lines were overlaid in ImageJ to show measurements of channel widths **(a)** and depths **(b)** using a tilted view.

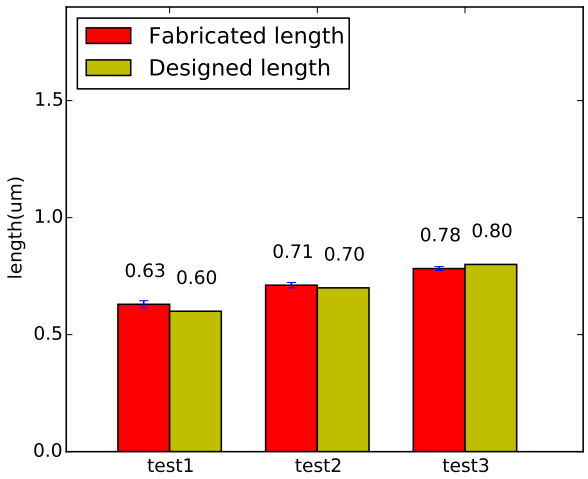
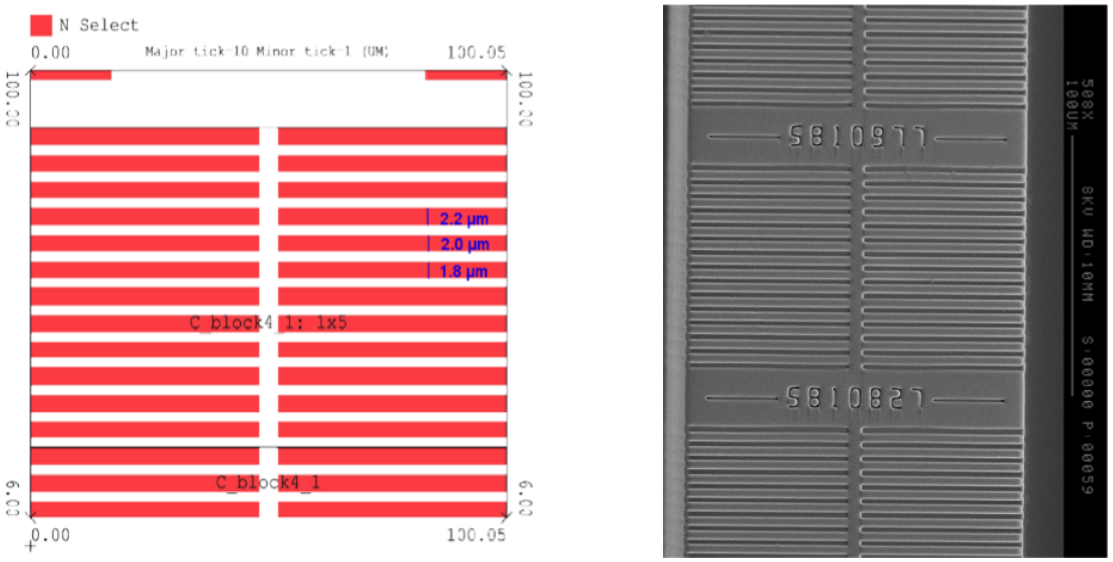


Figure C.4: Chip 1: comparison of fabricated vs designed channel width lengths

C.2.2 Chip 7

The width of the channels was designed to be 3.4 µm, 3.6 µm and 3.8 µm in a repeating pattern, with one closed end and the other opening onto the channel.

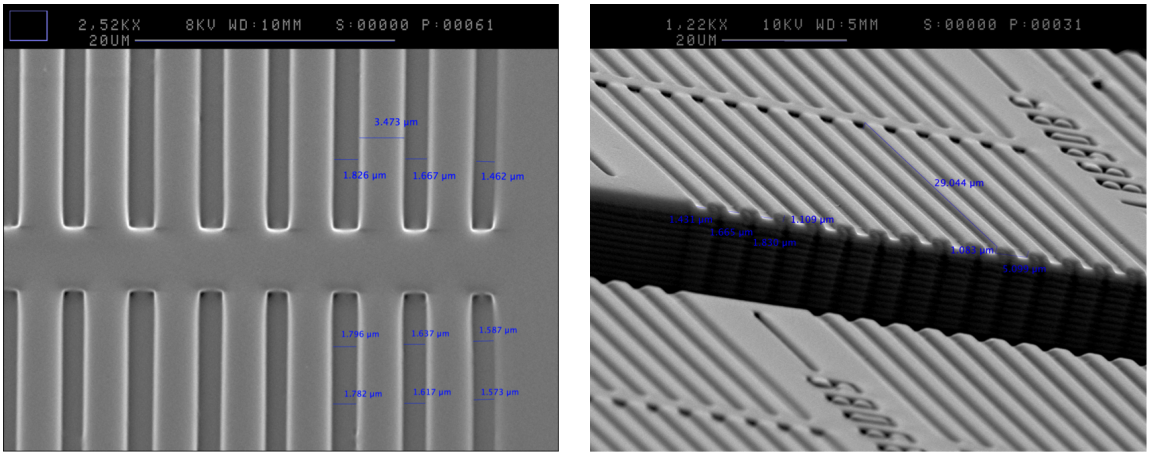


(a) (b)

Figure C.5: Chip 7: design view vs product SEM view

(a) Design of one square repeat panel within Chip 7, including measurements of expected final channel widths. **(b)** Overview SEM image showing the square panels repeating across Chip 7, with deep gutters between rows of square panels.

The backfill depth was 0.8 µm of oxide from each surface. This gives final channel widths of 1.8 µm, 2.0 µm and 2.2 µm (Figure C.5). The depth of the channel was designed to be 1.4 µm. The large fluid flow gutter depth is designed to be 40 µm. The SEM image in Figure C.6 shows the measurements of channel width and depth. The depth of channels within Chip07 was measured at 1.1 µm. Figure C.7 summarises the respective mean channel widths to be 1.54 µm, 1.64 µm and 1.80 µm. The channel depths and widths in Chip 7 resolved to be less than the intended lengths.



(a) (b)

Figure C.6: Chip 7: SEM images with line overlays

The lines were overlaid in ImageJ to show measurements of channel widths ((**a**)) and depth ((**b**)) using a tilted view.

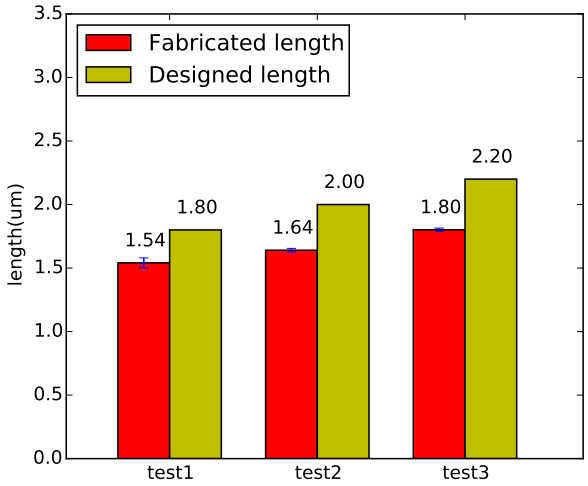
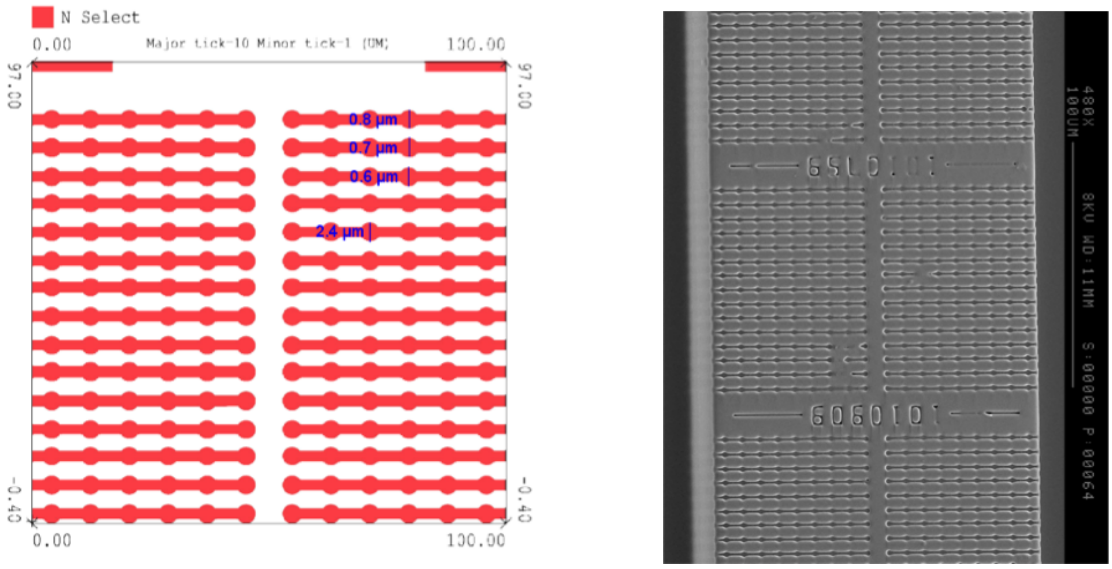


Figure C.7: Chip 7: comparison of fabricated vs designed channel width lengths

C.2.3 Chip 8

The width of the channels was designed to be 2.2 µm, 2.3 µm and 2.4 µm in a repeating pattern, with one closed end and the other opening onto the channel.

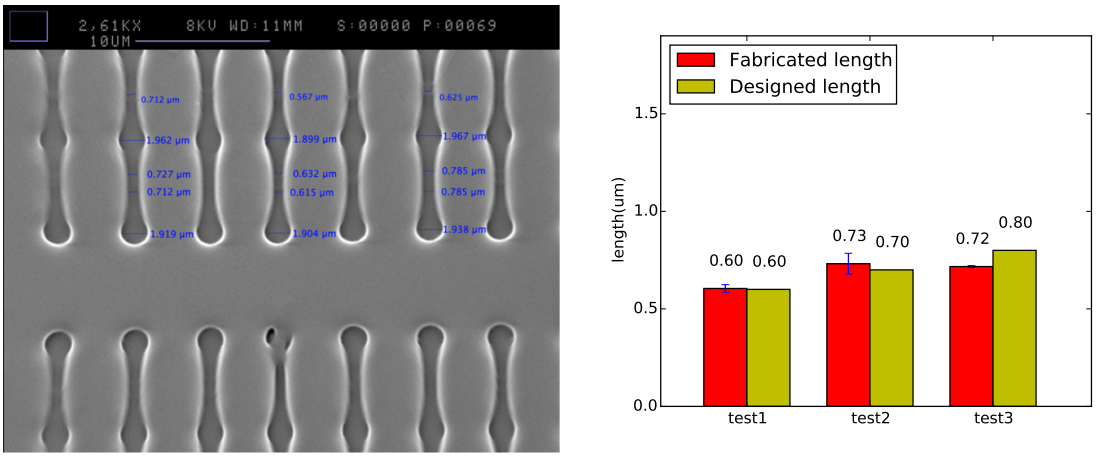


(a) (b)

Figure C.8: Chip 8: design view vs product SEM view

(a) Design of one square repeat panel within Chip 8, including measurements of expected final channel widths. **(b)** Overview SEM image showing the square panels repeating across Chip 8, with deep gutters between rows of square panels.

Circular features are present at the closed end of the channels and at intervals along the channel. The diameter of the circle was designed to be 4.0 µm. The backfill depth was 0.8 µm of oxide from each surface. This gives final channel widths of 0.6 µm, 0.7 µm and 0.8 µm, and a circle diameter of 2.4 µm (Figure C.8). The depth of the channel was designed to be 1.4 µm. The large fluid flow gutter depth is designed to be 40 µm. The SEM image in Figure C.9 shows the measurements of channel width. The circle diameter measured between 1.9 µm and 1.962 µm. Channel widths in this case measured as expected, though the circular features diameters measured less than expected. Some damage can be seen to the circular features in Figure C.9a It is not clear whether this is as a result of a blemish in the original wafer or if it occurred during the production of the intermediate wafer. The depth of channels within Chip08 could not be measured due to the lack of appropriate SEM images. Figure C.9b summarises the respective mean channel widths to be 0.60 µm, 0.73 µm and 0.72 µm.



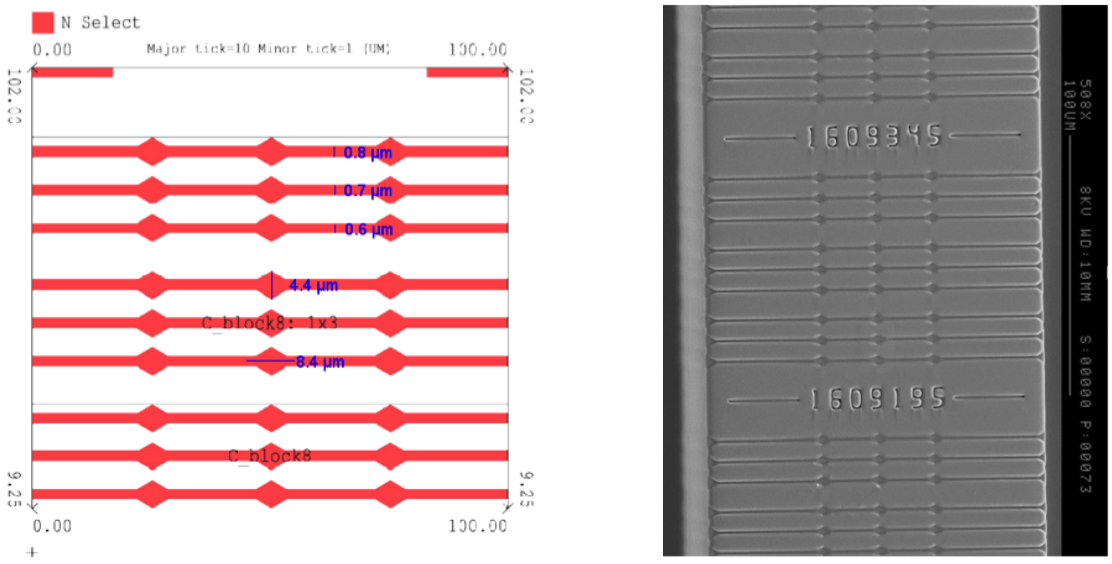
(a) (b)

Figure C.9: Chip 8: SEM images with line overlays

(a) SEM images with line overlays from ImageJ to show measurements of channel widths. **(b)** Comparison of fabricated vs designed channel width lengths.

C.2.4 Chip 10

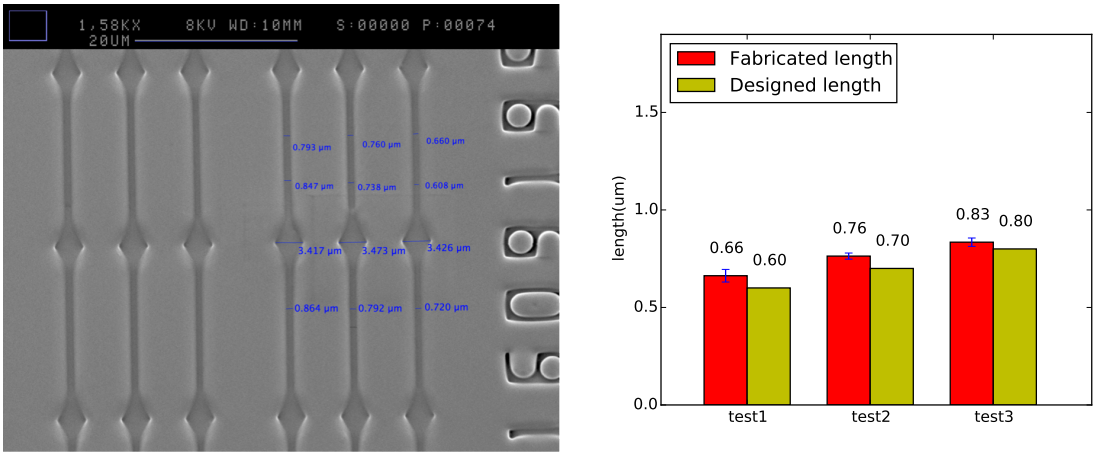
The width of the channels was designed to be 2.2 µm, 2.3 µm and 2.4 µm in a repeating pattern, with one closed end and the other opening onto the channel. The dimensions of the diamond were designed to be 6.0 µm by 10.0 µm. The backfill depth was 0.8 µm of oxide from each surface. This gives final channel widths of 0.6 µm, 0.7 µm and 0.8 µm, and diamond dimensions of 4.4 µm by 8.4 µm (Figure C.10). The depth of the channel was designed to be 1.4 µm. The large fluid flow gutter depth is designed to be 40 µm. The SEM image in Figure C.11 shows the measurements of channel width. The shorter of the two diamond dimensions measured between 3.417 µm and 3.473 µm. The depth of channels within Chip 10 could not be measured due to the lack of appropriate SEM images. Figure C.11b summarises the respective mean channel widths to be 0.66 µm, 0.76 µm and 0.83 µm.



(a) (b)

Figure C.10: Chip 10: design view vs product SEM view

(a) Design of one square repeat panel within Chip 10, including measurements of expected final channel widths. **(b)** Overview SEM image showing the square panels repeating across Chip 10, with deep gutters between rows of square panels.



(a) (b)

Figure C.11: Chip 10: SEM images with line overlays

(a) SEM images with line overlays from ImageJ to show measurements of channel widths. **(b)** Comparison of fabricated vs designed channel width lengths.

C.2.5 Wafer fabrication control features

Test features with a range of sizes and shapes were included in four corner positions on the wafer. Two of these test features were assessed using SEM.

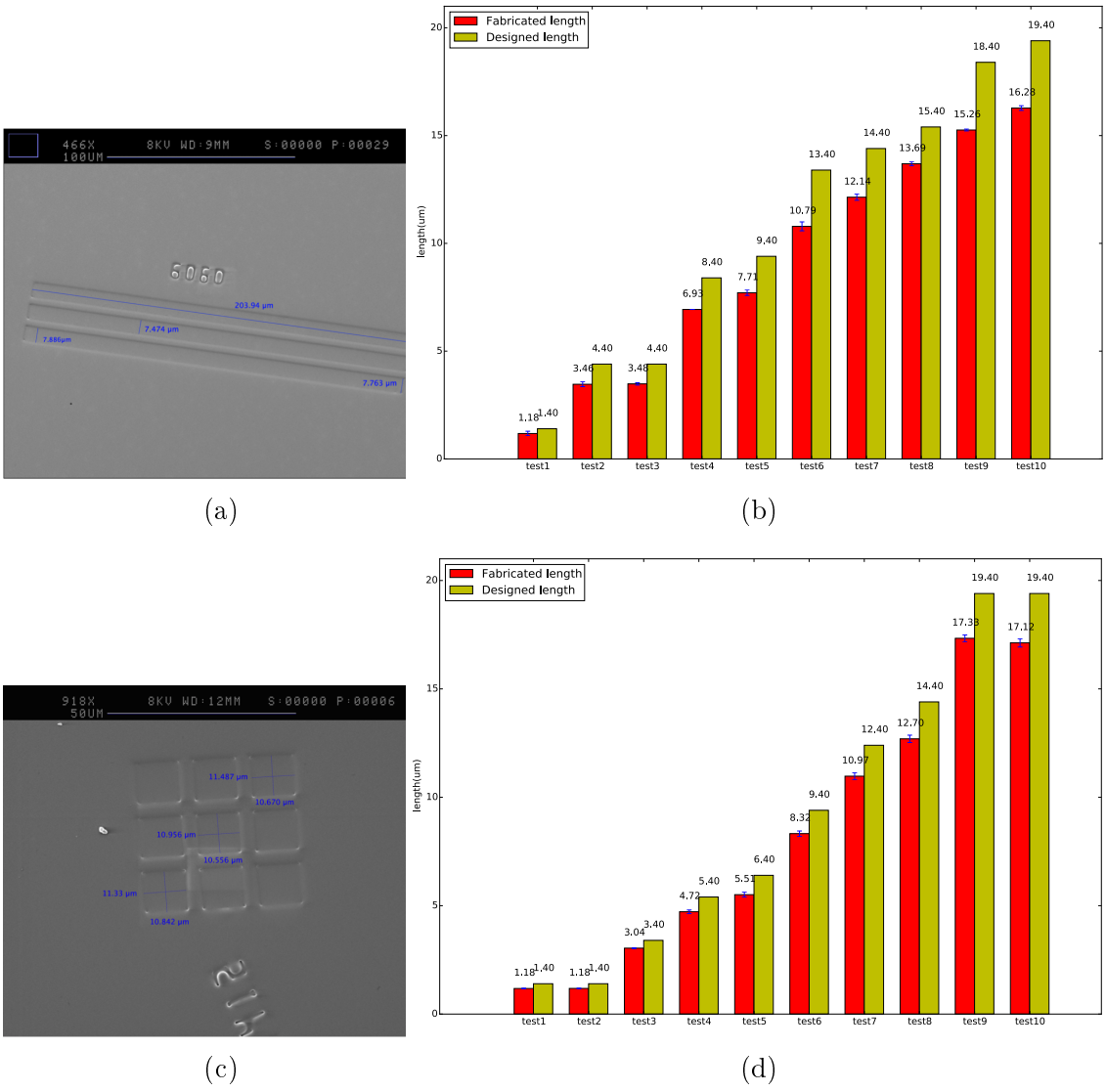


Figure C.12: SEM images of control features for wafer fabrication and their measurements

(a) SEM images with line overlays from ImageJ to show measurements of the rectangular test features. **(b)** Comparison of fabricated vs designed control feature lengths. **(c)** SEM images with line overlays from ImageJ to show measurements of the square test features. **(d)** Comparison of fabricated vs designed control feature lengths.

C.3 Wafer cleaning and microfluidics mould production protocols

The following protocols were adapted from that of Cluzel's lab [123].
Reagents and items:

- Copolymer 1: (7-8% Vinylmethylsiloxane)-(dimethylsiloxane) copolymer; Gelest; VDT-731
  or Vinylmethylsiloxane-dimethylsiloxane copolymer, trimethysiloxy terminated cSt 800-1200; Fluorochem Ltd.; VDT-731
- Copolymer 2: (25-30% methylhydrosiloxane) - dimethylsiloxane copolymer, hydride terminated, 30-50 cst; Gelest; HMS-H271
  or Methylhydrosiloxane- dimethylsiloxane copolymer, hydride terminated cSt 30-50; Fluorochem Ltd.; HMS-HM271
- Modulator: 1,3,5,7-Tetramethyltetravinylcyclotetrasiloxane; Sigma; 396281
  or 1,3,5,7-Tetravinyl tertamethylcyclotetrasiloxane; Fluorochem Ltd; S16500
- Catalyst: Platinum divinyl-tetramethyl-disiloxane; Gelest; SIP6839.3
  or Platinum-divinyltetramethyldisiloxane complex in vinyl terminated polydimethylsiloxane; Fluorochem Ltd; SIP6830.3
- Solvent: Hexane
- Sylgard 184 PDMS; Ellsworth
- Tridecafluoro-1,1,2,2-tetrahydrooctyl-1-trichlorosilane; United Chemical Technology;T2492
  or 1H,1H,2H,2H-perfluorooctyltricholosilane; Fluorochem Ltd; S13125
- Isopropanol; VWR; BDH1133-4LP
- Acetone; VWR; BDH1101-4LP
- 200 mL glass Borosilicate bottle
- 200 µL Pipette-man and tips

- Disposable, plastic pipettes
- Electronic pipette controller
- Vacuum chamber
- Digital timer
- Hot plate
- Printed silicon wafers
- Large glass petri dish to hold wafer
- Large glass petri dishes for chemical baths
- Kim wipes or lens cleaning tissue
- Plastic petri dish
- Dremmel drill with coffee frother adapter
- Scalpel
- Aluminum foil
- 70 % Ethanol spray
- Blue tack

C.3.1 Preparing the wafer

Protocol:

- Rinse wafer in order with isopropanol, acetone, and then isopropanol in large glass petri dishes. Take care not to let the residue from the gloves deposit on the wafer.
- Let the wafer sit in room temperature to dry (optionally using an N_2 air blow gun).

- Stick the wafer onto the inverted cap of a 14 mL culture tube using blue tack and then onto the floor of the vacuum chamber in the fume hood. By balancing the wafer on the cap, both sides of the wafer will be silanized. This step prevents PDMS from sticking to the bottom of the wafer.
- Place 40 µL of Tridecafluoro-1,1,2,2-tetrahydrooctyl-1-trichlorosilane on a kim wipe placed in the bottom of a plastic petri dish. Place within the vacuum chamber. This compound is corrosive, so take appropriate precautions. Only silanize within the dedicated vacuum chamber in the hood.
- Apply vacuum using a vacuum pump, taking extra caution not to knock off the wafer. If wafer tilts, stop and secure it.
- Allow the wafer to incubate for 30 minutes.
- Turn off house vacuum and slowly reintroduce air to the chamber. If air is introduced too quickly, the wafer can be thrown against the chamber and be damaged.

C.3.2 Preparing the aluminium wafer holder

Protocol:

- Clean any dust or residual PDMS from the glass dish to be used as a holder.
- Place a layer of aluminum foil (one piece) dull side up, on the bottom and up the side walls of the holder. The layer of aluminum foil allows the PDMS coated wafer to be easily removed from the holder, reducing the risk of cracking the wafer in later steps.

C.3.3 Preparing the h-PDMS

Quantities are for a 6 inch wafer, though making this amount for a 4 inch wafer allows for any leftover in the bottle to be spared in making sure the coverage is even.

Protocol:

- Add the following to a 200 mL Borosilicate bottle and mix well using the Dremmel and coffee frother adapter at the lowest speed setting (5000 RPM).
  - 13.6 g copolymer 1 (pour directly from bottle).
  - 72 µL catalyst (pipette slowly from bottle with 200 µL tip; watch the meniscus to make sure that the tip is filled.
- Add the following to the above mixture and mix well using the Dremmel and coffee frother adapter at the lowest speed setting (5000 RPM).
  - 0.4 g modulator (use a disposable pipette to transfer this from the bottle).
  - 4 g copolymer 2 (use a disposable pipette).
  - 2 g hexane (use a disposable pipette).
  - Degas this mixture in the vacuum chamber for 5 minutes, increasing and decreasing the pressure to pop the bubbles.
  - A thin layer of bubbles may remain on the surface, as they will eventually disappear.

C.3.4 Creating the h-PDMS layer

Protocol:

- Heat hot plate to 65 °C (outside the vacuum chamber).
- While the h-PDMS is degassing, mount the wafer on the spin coater.
  - Use the largest spin chuck.
  - The aluminum foil from previous users may need to be removed to fit the wafer in place.
- Following the instructions on the spin coater, create the following program:
  - Speed: 100 RPM; Duration: 1 minute (Named program T)
- When the h-PDMS is done degassing, remove it from the vacuum chamber.

- Start the spin coater, and gently pour the h-PDMS onto the spinning wafer.
  - Start in the center and move outwards until the entire wafer is coated.
  - Stop the spin coater when the wafer is completely coated.
- Gently remove the wafer and place it on a circular piece of paper towel.
- Place the wafer and towel in the vacuum chamber and degas for 3 minutes.
  - After 1 minute bubbles should form on the surface of the wafer and migrate to the surface.
  - Take care not to spread PDMS on the bottom of the wafer as this will interfere with the vacuum seal needed by the spin coater. PDMS can be removed from the bottom of the wafer with a kim wipe.
- While the wafer is degassing, reprogram the spin coater with the following program:
  - 500 RPM for 5 s followed by 1000 RPM for 40 s (Named program S)
- When the wafer is done degassing (3 minutes), place it on the spin coater and run the above program, with the lid on.
- During the 3 minute degas step, the h-PDMS may have slowly withdrawn from some of the external chips. Make a note of these chips on the blemish chart, and at the end of the process discard the PDMS intermediates created from them.
- Place the wafer in the glass petri dish and place the holder on the 65 °C hot plate.
- Bake the wafer for 45 minutes.
- Begin the next step whilst baking is underway.

C.3.5 Creating the s-PDMS support layer

Quantities are for a 6-inch wafer, if making a 4-inch wafer use the quantities in brackets.

Protocol:

- Mix 70 g (50 g) of Sylgard 184 base with 7 g (5 g) of the Sylgard 184 curing agent in a 200 mL Borosilicate bottle using the Dremmel and coffee frother attachment.
- Degas for 30 minutes in a vacuum chamber.
- While the s-PDMS is degassing, place the wafer, in the aluminum wafer holder, on a hot plate set at 65 °C.
- After the s-PDMS has degassed, remove the wafer holder from the hot plate, and gently pour the s-PDMS onto the wafer.
- Allow the s-PDMS to settle and coat the wafer (5 minutes), and then place the wafer holder back on the hot plate.
- Cover with the Pyrex cover and bake overnight. A more solid mould can be obtained by extending the bake step to 24 hours.

C.3.6 Removing the PDMS intermediate mould from the wafer

Protocol:

- When done baking, remove the wafer holder from the hot plate and allow to cool to room temperature.
- In the following steps be very careful to never bend or apply pressure to the silicon wafer. It will crack under gentle pressure.
- Gently remove the side wall of the wafer holder. The PDMS intermediates, the wafer, and the bottom layer of aluminum foil will come off with the side wall.

- Gently flip over the side wall/wafer/PDMS intermediate and slowly remove the aluminum foil.
- Gently insert a scalpel between the PDMS intermediate and the aluminum side wall. Run the scalpel around the edge of the intermediate to dislodge the intermediate from the side wall.
- Remove any PDMS that has leaked under the wafer by running the scalpel along the edge of the wafer.
- Flip the intermediate and the wafer over so that the wafer is lying flat on the table.
- Gently peel the PDMS intermediate from the wafer.
- Move very slowly. The entire process should take at least 5 minutes. Removing the intermediate too quickly will damage the intermediate and will leave residual PDMS on the surface of the wafer.
- Once the intermediate has been removed, wash the wafer with isopropanol, acetone, and isopropanol, as above, and dry with N_2. Return the wafer to its plastic storage disk.
- Dice the PDMS intermediates and place in a labeled container.

Glossary (Abbreviations)

CAD Computer Aided Design. 81, 85, 133

NGS Next Generation Sequencing. 69–72

SBO Systems Biology Ontology. 93

Glossary (Nomenclature)

DEA Dual-evolutionary algorithm or dual-evolutionary approach. 7–10, 17, 19–22, 66, 67, 69–71, 78, 97, 111, 112, 114, 115, 118–126, 130, 156–162

DNA Deoxyribonucleic acid. 19, 53, 56, 69, 79, 162, 170

DRIE Deep Reactive Ion Etching. 127–129

EA Evolutionary algorithm. 20, 71, 110, 111, 118, 160

EtBr Ethidium bromide. 25, 48

FAD Flavin adenine dinucleotide. 59

FBA Flux Balance Analysis. 95–98, 110, 123, 160

FMN Flavin mononucleotide. 59, 61, 62, 64, 68, 69, 72

GA Genetic algorithm. 112

LAC Lipoteichoic acid. 170

MDE Model-driven engineering. 6, 132, 155

mRNA Ribonucleic acid. 170

MSE Mean Squared Error. 148, 151

ODE Ordinary Differential Equation. 93

PDMS Polydimethylsiloxane. 130, 180, 193–197

RBS Ribosome binding site or Shine-Dalgarno sequence. 61

ROI Region of interest. 131

SEM Scanning Electron Microscopy. 129, 133, 137, 139, 142

SNP Single-nucleotide polymorphism. 56, 71, 72, 103, 106–112, 123, 161

TEOS Tetra-ethyl orthosilicate. 129, 130

References

[1] Roi Adadi, Benjamin Volkmer, Ron Milo, Matthias Heinemann, and Tomer Shlomi. Prediction of microbial growth rate versus biomass yield by a metabolic network with kinetic parameters. *PLoS Computational Biology*, **8**[7]:e1002575–e1002575, January 2012. 98

[2] Rudolf Adlung and Hamid Mamdouh. How to Design Trade Agreements in Services: Top Down or Bottom Up? *SSRN Electronic Journal*, jun 2013. 4

[3] Alessandra M. Albertini and Alessandro Galizzi. The sequence of the trp operon of Bacillus subtilis 168 (trpC2) revisited. *Microbiology*, **145**[12]:3319–3320, dec 1999. 75

[4] G Alterovitz, T Muso, and M F Ramoni. The challenges of informatics in synthetic biology: from biomolecular networks to artificial organisms. *Briefings in bioinformatics*, **11**[1]:80–95, January 2010. 13

[5] Ernesto Andrianantoandro, Subhayu Basu, David K Karig, and Ron Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular systems biology*, **2**:2006–0028, January 2006. 2, 11

[6] Narayana Annaluru, Héloïse Muller, Leslie A Mitchell, Sivaprakash Ramalingam, Giovanni Stracquadanio, Sarah M Richardson, Jessica S Dymond, Zheng Kuang, Lisa Z Scheifele, Eric M Cooper, Yizhi Cai, Karen Zeller, Neta Agmon, Jeffrey S Han, Michalis Hadjithomas, Jennifer Tullman, Katrina

Caravelli, Kimberly Cirelli, Zheyuan Guo, Viktoriya London, Apurva Yeluru, Sindurathy Murugan, Karthikeyan Kandavelou, Nicolas Agier, Gilles Fischer, Kun Yang, J Andrew Martin, Murat Bilgel, Pavlo Bohutski, Kristin M Boulier, Brian J Capaldo, Joy Chang, Kristie Charoen, Woo Jin Choi, Peter Deng, James E DiCarlo, Judy Doong, Jessilyn Dunn, Jason I Feinberg, Christopher Fernandez, Charlotte E Floria, David Gladowski, Pasha Hadidi, Isabel Ishizuka, Javaneh Jabbari, Calvin Y L Lau, Pablo A Lee, Sean Li, Denise Lin, Matthias E Linder, Jonathan Ling, Jaime Liu, Jonathan Liu, Mariya London, Henry Ma, Jessica Mao, Jessica E McDade, Alexandra McMillan, Aaron M Moore, Won Chan Oh, Yu Ouyang, Ruchi Patel, Marina Paul, Laura C Paulsen, Judy Qiu, Alex Rhee, Matthew G Rubashkin, Ina Y Soh, Nathaniel E Sotuyo, Venkatesh Srinivas, Allison Suarez, Andy Wong, Remus Wong, Wei Rose Xie, Yijie Xu, Allen T Yu, Romain Koszul, Joel S Bader, Jef D Boeke, and Srinivasan Chandrasegaran. Total synthesis of a functional designer eukaryotic chromosome. *Science (New York, NY)*, **344**[6179]:55–58, April 2014. 16

[7] Rutherford Aris. *Mathematical modelling techniques*. Courier Corporation, 2012. 83

[8] Timothy Arndt. Visual software tools for bioinformatics. *Journal of Visual Languages & Computing*, **19**[2]:291–301, 2008. 67

[9] FH Arnold. Design by directed evolution. In *FASEB JOURNAL*, **11**, pages A872–A872. FEDERATION AMER SOC EXP BIOL 9650 ROCKVILLE PIKE, BETHESDA, MD 20814-3998 USA, 1997. 58

[10] A Bacher, S Eberhardt, M Fischer, and K Kis. Biosynthesis of vitamin B2 (riboflavin). *Annual Review of Nutrition*, 2000. 87, 89

[11] Manuel F Balandrin, James A Klocke, Eve Syrkin Wurtele, and William Hugh Bollinger. Natural plant chemicals: sources of industrial and medicinal materials. *Science*, **228**[4704]:1154–1160, 1985. 58

[12] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology*, **19**[5]:455–477, 2012. 67

[13] Christopher D Bayliss and E Richard Moxon. Hypermutation and bacterial adaptation. *ASM news*, **68**[11], 2002. 56

[14] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mallor, Ken Shwaber, and Jeff Sutherland. The agile manifesto. Technical report, The Agile Alliance, 2001. 5

[15] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change; 2nd ed.* Addison-Wesley, Boston, MA, 2005. 5

[16] Richard Bellman. COMBINATORIAL PROCESSES AND DYNAMIC PROGRAMMING. pages 19–23, February 1958. 54

[17] Richard Bellman. *Dynamic programming*. Courier Corporation, 2013. 152

[18] Cristina Bicchieri. Rationality and game theory. *The Oxford handbook of rationality*, pages 182–205, 2004. 53

[19] L Bilitchenko, A Liu, S Cheung, E Weeding, and B Xia. Eugene–a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PloS one*, 2011. 14, 81

[20] Ivana Bjedov, Olivier Tenaillon, Benedicte Gerard, Valeria Souza, Erick Denamur, Miroslav Radman, François Taddei, and Ivan Matic. Stress-induced mutagenesis in bacteria. *Science*, **300**[5624]:1404–1409, 2003. 55

[21] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, **30**[15]:2114–2120, 2014. 67

[22] Monica Bordegoni and Caterina Rizzi. *Innovation in product design: from CAD to virtual prototyping*. Springer, 2011. 81

[23] Eckard Bringmann and Andreas Kr. Model-Based Testing of Automotive Systems. In *2008 International Conference on Software Testing, Verification, and Validation*, pages 485–493. IEEE, apr 2008. 6

[24] James P Brody, Paul Yager, Raymond E Goldstein, and Robert H Austin. Biotechnology at low reynolds numbers. *Biophysical journal*, **71**[6]:3430–3441, 1996. 126

[25] Polly Brown. Cad: Do computers aid the design process after all? *Intersect: The Stanford Journal of Science, Technology and Society*, **2**[1]:52–66, 2009. 85

[26] John Cairns, Julie Overbaugh, and Stephan Miller. The origin of mutants. *Nature*, **335**[6186]:142–145, 1988. 55

[27] James M Carothers, Jonathan A Goler, Darmawi Juminaga, and Jay D Keasling. Model-Driven Engineering of RNA Devices to Quantitatively Program Gene Expression. *Science*, **334**[6063]:1716–1719, 2011. 6

[28] Anne Carpenter, Thouis Jones, Michael Lamprecht, Colin Clarke, In Kang, Ola Friman, David Guertin, Joo Chang, Robert Lindquist, Jason Moffat, Polina Golland, and David Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, **7**[10]:R100, 2006. 143

[29] Sean B Carroll. Genetics and the making of homo sapiens. *Nature*, **422**[6934]:849–857, 2003. 82

[30] Malcolm J. Casadaban and Stanley N. Cohen. Analysis of gene control signals by DNA fusion and cloning in Escherichia coli. *Journal of Molecular Biology*, **138**[2]:179–207, apr 1980. 23

[31] Jeronimo Cello, Aniko V Paul, and Eckard Wimmer. Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. *Science (New York, NY)*, **297**[5583]:1016–1018, August 2002. 16

[32] L Y Chan, S Kosuri, and D Endy. Refactoring bacteriophage T7 - Chan - 2005 - Molecular Systems Biology - Wiley Online Library. *Molecular systems biology*, 2005. 16

[33] Deepak Chandran, Frank T Bergmann, and Herbert M Sauro. Computer-aided design of biological circuits using tinkercell. *Bioengineered Bugs*, **1**[4]:276–283, October 2014. 14, 81

[34] Antje Chang, Ida Schomburg, Sandra Placzek, Lisa Jeske, Marcus Ulbrich, Mei Xiao, Christoph W Sensen, and Dietmar Schomburg. BRENDA in 2015: exciting developments in its 25th year of existence. *Nucleic Acids Research*, **43**[Database issue]:D439–46, January 2015. 93

[35] M. M. Y. Chen, C. D. Snow, C. L. Vizcarra, S. L. Mayo, and F. H. Arnold. Comparison of random mutagenesis and semi-rational designed libraries for improved cytochrome P450 BM3-catalyzed hydroxylation of small alkanes. *Protein Engineering Design and Selection*, **25**[4]:171–178, apr 2012. 57

[36] G Chotani, T Dodge, A Hsu, M Kumar, R LaDuca, D Trimbur, W Weyler, and K Sanford. The commercial production of chemicals using pathway engineering. *Biochimica et biophysica acta*, **1543**[2]:434–455, dec 2000. 15

[37] Edmund M. Clarke, Orna Grumberg, and David E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, **16**[5]:1512–1542, sep 1994. 6

[38] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**[11]:1422–1423, 2009. 67

[39] M T Cooling, V Rouilly, G Misirli, J Lawson, T Yu, J Hallinan, and A Wipat. Standard virtual biological parts: a repository of modular modeling components for synthetic biology. *Bioinformatics*, **26**[7]:925–931, April 2010. 83

[40] D Coquard, M Huecas, M Ott, J M Van Dijl, A P van Loon, and H P Hohmann. Molecular cloning and characterisation of the ribC gene from Bacillus subtilis: a point mutation in ribC results in riboflavin overproduction. *Molecular and General Genetics MGG*, **254**[1]:81–84, March 1997. 59, 89

[41] Athel Cornish-Bowden. One hundred years of Michaelis–Menten kinetics. *Perspectives in Science*, **4**:3–9, March 2015. 90

[42] Mélanie Courtot, Nick Juty, Christian Knüpfer, Dagmar Waltemath, Anna Zhukova, Andreas Dräger, Michel Dumontier, Andrew Finney, Martin Golebiewski, Janna Hastings, Stefan Hoops, Sarah Keating, Douglas B Kell, Samuel Kerrien, James Lawson, Allyson Lister, James Lu, Rainer Machne, Pedro Mendes, Matthew Pocock, Nicolas Rodriguez, Alice Villeger, Darren J Wilkinson, Sarala Wimalaratne, Camille Laibe, Michael Hucka, and Nicolas Le Novère. Controlled vocabularies and semantics in systems biology. *Molecular systems biology*, **7**[1]:543, January 2011. 93

[43] Brian Cox and J. R. (Jeffrey Robert) Forshaw. *The quantum universe : (and why anything that can happen, does)*. Da Capo Press, 2012. 1

[44] Valentino Crespi, Aram Galstyan, and Kristina Lerman. Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, **24**[3]:303–313, apr 2008. 4

[45] Francis Crick. Central dogma of molecular biology. *Nature*, **227**[5258]:561–563, 1970. 55

[46] P F Culverhouse. Constraining designers and their CAD tools. *Design Studies*, **16**[1]:81–101, January 1995. 81, 86

[47] Kathleen A Curran and Hal S Alper. Expanding the chemical palate of cells by combining systems biology and metabolic engineering. *Metabolic engineering*, **14**[4]:289–297, July 2012. 15

[48] M J Czar, Y Cai, and J Peccoud. Writing DNA with GenoCADTM. *Nucleic Acids Research*, **37**[Web Server]:W40–W47, June 2009. 14, 81

[49] Charles Darwin. *On the Origin of Species by Means of Natural Selection.* Murray, London, 1859. or the Preservation of Favored Races in the Struggle for Life. 54, 78

[50] Benedetto De Martino, Dharshan Kumaran, Ben Seymour, and Raymond J Dolan. Frames, biases, and rational decision-making in the human brain. *Science (New York, NY)*, **313**[5787]:684–687, August 2006. 3

[51] François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP. In *the fourteenth international conference*, page 85, New York, New York, USA, 2012. ACM Press. 112

[52] A L Demain. Riboflavin oversynthesis. *Annual Review of Microbiology*, **26**:369–388, 1972. 88

[53] Henry Winram. Dickinson. *A Short History of the Steam Engine*. Cambridge University Press, 2011. 12

[54] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, **1**[1]:269–271, 1959. 152

[55] Steve Dorus, Eric J Vallender, Patrick D Evans, Jeffrey R Anderson, Sandra L Gilbert, Michael Mahowald, Gerald J Wyckoff, Christine M Malcom, and Bruce T Lahn. Accelerated evolution of nervous system genes in the origin of homo sapiens. *Cell*, **119**[7]:1027–1040, 2004. 82

[56] John W Drake, Brian Charlesworth, Deborah Charlesworth, and James F Crow. Rates of spontaneous mutation. *Genetics*, **148**[4]:1667–1686, 1998. 55

[57] C L Dym. Engineering Design: A Synthesis of Views. page 1. Cambridge University Press, New York, 1994. 2

[58] Arthur Edelstein, Nenad Amodaj, Karl Hoover, Ron Vale, and Nico Stuurman. *Computer Control of Microscopes Using MicroManager*. John Wiley & Sons, Inc., Hoboken, NJ, USA, May 2001. 130

[59] W Ernst Eder and Stanislav Hosnedl. *Design engineering: a manual for enhanced creativity*. CRC Press, 2007. 1, 54

[60] W Ernst Eder and Stanislav Hosnedl. *Introduction to design engineering: systematic creativity and management*. CRC Press, 2010. 54

[61] Lucy Eland, Anil Wipat, S Lee, Sungshic Park, and Ling Wu. *Microfluidics for bacterial imaging*, pages 69–111. Elsevier, Oct 2016. 9

[62] T Ellis, X Wang, and J Collins. Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nature biotechnology*, 2009. 83

[63] R Ian Faulconbridge and Michael J Ryan. *Managing complex technical projects: A systems engineering approach*. Artech House, 2003. 83

[64] Robert France and Bernhard Rumpe. Model-driven Development of Complex Software: A Research Roadmap. In *Future of Software Engineering (FOSE '07)*, pages 37–54. IEEE, may 2007. 6

[65] Peter Freeman and David Hart. A science of design for software-intensive systems. *Communications of the ACM*, **47**[8]:19–21, August 2004. 2

[66] Jacob Frelinger, Thomas B Kepler, and Cliburn Chan. Flow: Statistics, visualization and informatics for flow cytometry. *Source Code for Biology and Medicine*, **3**[1]:10, 2008. 67

[67] Errol C Friedberg, Graham C Walker, Wolfram Siede, and Richard D Wood. *DNA repair and mutagenesis*. American Society for Microbiology Press, 2005. 55

[68] Hiroshi Fujikawa, Akemi Kai, and Satoshi Morozumi. A new logistic model for Escherichia coli growth at constant and dynamic temperatures. *Food Microbiology*, **21**[5]:501–509, October 2004. 101

[69] M Galdzicki, K P Clancy, E Oberortner, and M Pocock. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology : Nature Biotechnology : Nature Research. *Nature*, 2014. 3

[70] Ethan C Garner. MicrobeTracker: quantitative image analysis designed for the smallest organisms. *Molecular microbiology*, **80**[3]:577–579, 2011. 143

[71] D Gärtner, M Geissendörfer, and W Hillen. Expression of the Bacillus subtilis xyl operon is repressed at the level of transcription and is induced by xylose. *Journal of Bacteriology*, 1988. 62

[72] M Geissendörfer and W Hillen. Regulated expression of heterologous genes inBacillus subtilis using the Tn10 encodedtet regulatory elements. *Appl Microbiol Biotechnol*, 1990. 62

[73] John Burdon Sanderson Haldane George Edward Briggs. A Note on the Kinetics of Enzyme Action. *Biochemical Journal*, **19**[2]:338, 1925. 90

[74] D G Gibson, J I Glass, C Lartigue, V N Noskov, R Y Chuang, M A Algire, G A Benders, M G Montague, L Ma, M M Moodie, C Merryman, S Vashee, R Krishnakumar, N Assad-Garcia, C Andrews-Pfannkoch, E A Denisova, L Young, Z Q Qi, T H Segall-Shapiro, C H Calvey, P P Parmar, C A Hutchison, H O Smith, and J C Venter. Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome. *Science (New York, NY)*, **329**[5987]:52–56, July 2010. 16

[75] Thomas Gilovich, Dale Griffin, and Daniel Kahneman. *Heuristics and Biases*. The Psychology of Intuitive Judgment. Cambridge University Press, July 2002. 3

[76] F Ginetti, M Perego, A M Albertini, and A Galizzi. Bacillus subtilis mutS mutL operon: identification, nucleotide sequence and mutagenesis. *Microbiology (Reading, England)*, **142 (Pt 8)**:2021–2029, August 1996. 63

[77] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, **3**[2]:95–99, 1988. 160

[78] Cato M Guldberg and Peter Waage. Studies concerning affinity. *CM Forhandlinger: Videnskabs-Selskabet i Christiana*, **35**[1864]:1864, 1864. 90, 91

[79] Armando Martin Haeberer and TSE Maibaum. Scientific rigour, an answer to a pragmatic question: a linguistic framework for software engineering. In *Proceedings of the 23rd International Conference on Software Engineering*, pages 463–472. IEEE Computer Society, 2001. 84

[80] J Hallinan, S Park, and A Wipat. Bridging the gap between design and reality: A dual evolutionary strategy for the design of synthetic genetic circuits. In *BIOINFORMATICS 2012*, pages 263–268, 2012. 7, 9, 19, 54, 82, 83, 125

[81] Kyle IS Harrington, Timothy S Stiles, Lakshmi Venkatraman, Claudia Prahst, and Katie Bentley. Functional image processing with imagej/fiji. In *BioImage Informatics Conference*, 2015. 126

[82] Matthias Heinemann and Sven Panke. Synthetic biology—putting engineering into biology. *Bioinformatics*, **22**[22]:2790–2799, 2006. 2, 11

[83] Christopher S Henry, Jenifer F Zinner, Matthew P Cohoon, and Rick L Stevens. iBsu1103: a new genome-scale metabolic model of Bacillus subtilis based on SEED annotations. *Genome biology*, **10**[6]:R69, 2009. 96

[84] T A Hince and S Neale. A comparison of the mutagenic action of the methyl and ethyl derivatives of nitrosamides and nitrosamidines on Escherichia coli. *Mutation research*, **24**[3]:383–7, sep 1974. 55

[85] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992. 160

[86] Gerard J Holzmann. Conquering Complexity. *Computer*, **40**[12]:111–113, 2007. 83

[87] S Hoops, S Sahle, R Gauges, C Lee, J Pahle, N Simus, M Singhal, L Xu, P Mendes, and U Kummer. COPASI–a COmplex PAthway SImulator. *Bioinformatics*, **22**[24]:3067–3074, December 2006. 93

[88] Elizabeth Howell and SPACE com Contributor. How many stars are in the universe? *Space. com, May*, **31**, 2014. 110

[89] Zhipeng Huang, Nadine Geyer, Peter Werner, Johannes De Boor, and Ulrich Gösele. Metal-assisted chemical etching of silicon: a review. *Advanced materials*, **23**[2]:285–308, 2011. 127

[90] Vladimir. Hubka and W. Ernst. Eder. *Theory of Technical Systems : a Total Concept Theory for Engineering Design*. Springer Berlin Heidelberg, 1988. 1

[91] M Hucka, A Finney, H M Sauro, H Bolouri, J C Doyle, H Kitano, , the rest of the SBML Forum, A P Arkin, B J Bornstein, D Bray, A Cornish-Bowden, A A Cuellar, S Dronov, E D Gilles,

M Ginkel, V Gor, I I Goryanin, W J Hedley, T C Hodgman, J H Hofmeyr, P J Hunter, N S Juty, J L Kasberger, A Kremling, U Kummer, N Le Novere, L M Loew, D Lucio, P Mendes, E Minch, E D Mjolsness, Y Nakayama, M R Nelson, P F Nielsen, T Sakurada, J C Schaff, B E Shapiro, T S Shimizu, H D Spence, J Stelling, K Takahashi, M Tomita, J Wagner, and J Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**[4]:524–531, March 2003. 93

[92] Holger Jenke-Kodama, Rolf Müller, and Elke Dittmann. Evolutionary mechanisms underlying secondary metabolite diversity. In *Natural Compounds as Drugs Volume I*, pages 119–140. Birkhäuser Basel, Basel, 2008. 15, 58

[93] Paul R. Jensen, Tracy J. Mincer, Philip G. Williams, and William Fenical. Marine actinomycete diversity and natural product discovery. *Antonie van Leeuwenhoek*, **87**[1]:43–48, jan 2005. 15

[94] R Jonker and A Volgenant. Ein Algorithmus mit kürzesten alternierenden Wegen für dichte und dünne Zuordnungsprobleme. *Computing*, **38**[4]:325–340, December 1987. 150, 152

[95] Ari Juels and Martin Wattenberg. Stochastic hillclimbing as a baseline method for evaluating genetic algorithms. In *Advances in Neural Information Processing Systems*, pages 430–436, 1996. 112, 159

[96] David J Kasik, William Buxton, and David R Ferguson. Ten cad challenges. *IEEE Computer Graphics and Applications*, **25**[2]:81–92, 2005. 81, 85

[97] Jay D Keasling. Manufacturing molecules through metabolic engineering. *Science (New York, N.Y.)*, **330**[6009]:1355–8, dec 2010. 6

[98] K. Keutzer, A.R. Newton, J.M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: orthogonalization of concerns and

platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **19**[12]:1523–1543, 2000. 4

[99] Ahmad S Khalil and James J Collins. Synthetic biology: applications come of age. *Nature Reviews Genetics*, **11**[5]:367–379, May 2010. 2, 11

[100] Johannes Klein, Stefan Leupold, Ilona Biegler, Rebekka Biedendieck, Richard Munch, and Dieter Jahn. Tlm-tracker: software for cell segmentation, tracking and lineage analysis in time-lapse microscopy movies. *Bioinformatics*, **28**[17]:2276–2277, 2012. 143

[101] Edward H Knight. *Knights American Mechanical Dictionary*. 1876. xv, 12

[102] A L Knorr, R Jain, and R Srivastava. Bayesian-based selection of metabolic objective functions. *Bioinformatics*, **23**[3]:351–357, February 2007. 96

[103] Peter Kroes. Engineering design. pages 127–161. Springer Netherlands, Dordrecht, April 2012. 3

[104] H W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics*, **52**[1]:7–21, February 2005. 150

[105] F Kunst, N Ogasawara, I Moszer, A M Albertini, G Alloni, V Azevedo, M G Bertero, P Bessieres, A Bolotin, S Borchert, R Borriss, L Boursier, A Brans, M Braun, S C Brignell, S Bron, S Brouillet, C V Bruschi, B Caldwell, V Capuano, N M Carter, S K Choi, J J Cordani, I F Connerton, N J Cummings, R A Daniel, F Denziot, K M Devine, A Düsterhöft, S D Ehrlich, P T Emmerson, K D Entian, J Errington, C Fabret, E Ferrari, D Foulger, C Fritz, M Fujita, Y Fujita, S Fuma, A Galizzi, N Galleron, S Y Ghim, P Glaser, A Goffeau, E J Golightly, G Grandi, G Guiseppi, B J Guy, K Haga, J Haiech, C R Harwood, A Hènaut, H Hilbert, S Holsappel, S Hosono, M F Hullo,

M Itaya, L Jones, B Joris, D Karamata, Y Kasahara, M Klaerr-Blanchard, C Klein, Y Kobayashi, P Koetter, G Koningstein, S Krogh, M Kumano, K Kurita, A Lapidus, S Lardinois, J Lauber, V Lazarevic, S M Lee, A Levine, H Liu, S Masuda, C Mauël, C Médigue, N Medina, R P Mellado, M Mizuno, D Moestl, S Nakai, M Noback, D Noone, M O'Reilly, K Ogawa, A Ogiwara, B Oudega, S H Park, V Parro, T M Pohl, D Portelle, S Porwollik, A M Prescott, E Presecan, P Pujic, B Purnelle, G Rapoport, M Rey, S Reynolds, M Rieger, C Rivolta, E Rocha, B Roche, M Rose, Y Sadaie, T Sato, E Scanlan, S Schleich, R Schroeter, F Scoffone, J Sekiguchi, A Sekowska, S J Seror, P Serror, B S Shin, B Soldo, A Sorokin, E Tacconi, T Takagi, H Takahashi, K Takemaru, M Takeuchi, A Tamakoshi, T Tanaka, P Terpstra, A Togoni, V Tosato, S Uchiyama, M Vandebol, F Vannier, A Vassarotti, A Viari, R Wambutt, H Wedler, T Weitzenegger, P Winters, A Wipat, H Yamamoto, K Yamane, K Yasumoto, K Yata, K Yoshida, H F Yoshikawa, E Zumstein, H Yoshikawa, and A Danchin. The complete genome sequence of the gram-positive bacterium Bacillus subtilis. *Nature*, **390**[6657]:249–256, November 1997. 61, 106

[106] John Laird. The law of parsimony. *The Monist*, pages 321–344, 1919. 85

[107] Russell Lande. Natural selection and random genetic drift in phenotypic evolution. *Evolution*, **30**[2]:314–334, 1976. 82

[108] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, **9**[4]:357–359, 2012. 67

[109] Yoann Le Breton, Nrusingh Prasad Mohapatra, and WG Haldenwang. In vivo random mutagenesis of bacillus subtilis by use of tnylb-1, a mariner-based transposon. *Applied and environmental microbiology*, **72**[1]:327–333, 2006. 55

[110] Elaine R Lee, Kenneth F Blount, and Ronald R Breaker. Roseoflavin is a natural antibacterial compound that binds to FMN riboswitches and regulates gene expression. *RNA biology*, **6**[2]:187–194, 2009. 61

[111] David Lewis. How to define theoretical terms. *The Journal of Philosophy*, **67**[13]:427–446, 1970. 84

[112] P J Lewis and A L Marston. GFP vectors for controlled expression and dual labelling of protein fusions in Bacillus subtilis. *Gene*, **227**[1]:101–10, feb 1999. 24

[113] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, **25**[16]:2078–2079, 2009. 67

[114] Seong Han Lim, Jong Soo Choi, and Enoch Y Park. Microbial production of riboflavin using riboflavin overproducers,Ashbya gossypii, Bacillus subtilis, andCandida famate: An overview. *Biotechnology and Bioprocess Engineering*, **6**[2]:75–88, April 2001. 59, 87

[115] Matthew W Lux, Brian W Bramlett, David A Ball, and Jean Peccoud. Genetic design automation: engineering fantasy or scientific renewal? *TRENDS in Biotechnology*, **30**[2]:120–126, February 2012. 2, 8, 11, 14, 15

[116] M Mack, A Van Loon, and H Hohmann. Regulation of Riboflavin Biosynthesis inBacillus subtilis Is Affected by the Activity of the Flavokinase/Flavin Adenine Dinucleotide Synthetase Encoded byribC. *Journal of Bacteriology*, 1998. 89

[117] P C Mangelsdorf and R G Reeves. The origin of maize. *Proceedings of the National Academy of Sciences*, 1938. 11

[118] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al.

The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, **20**[9]:1297–1303, 2010. 67

[119] D Mérino, H Réglier Poupet, and P Berche. A hypermutator phenotype attenuates the virulence of Listeria monocytogenes in a mouse model - Mérino - 2002 - Molecular Microbiology - Wiley Online Library. *Molecular Microbiology*, 2002. 56

[120] L Michaelis and M L Menten. Die kinetik der invertinwirkung (The Kinetics of Invertase Action). *Biochem z*, 1913. 90

[121] Alexander S Mironov, Ivan Gusarov, Ruslan Rafikov, Lubov Errais Lopez, Konstantin Shatalin, Rimma A Kreneva, Daniel A Perumov, and Evgeny Nudler. Sensing Small Molecules by Nascent RNA. *Cell*, **111**[5]:747–756, November 2002. 87

[122] Melanie Mitchell, John H Holland, and Stephanie Forrest. When will a genetic algorithm outperform hill climbing. In *Advances in neural information processing systems*, pages 51–58, 1994. 160

[123] J Moffitt, J Lee, and P Cluzel. The single-cell chemostat: an agarose-based, microfluidic device for high-throughput, single-cell studies of bacteria and bacterial communities. *Lab on a Chip*, 2012. 127, 191

[124] Tae Seok Moon, Chunbo Lou, Alvin Tamsir, Brynne C Stanton, and Christopher A Voigt. Genetic programs constructed from layered logic gates in single cells. *Nature*, **491**[7423]:249–253, November 2012. 13

[125] A.E. MUSSON. Industrial Motive Power in the United Kingdom, 180070. *The Economic History Review*, **29**[3]:415–439, aug 1976. 11

[126] Pierre Nicolas, Ulrike Mäder, Etienne Dervyn, Tatiana Rochat, Aurélie Leduc, Nathalie Pigeonneau, Elena Bidnenko, Elodie Marchadier, Mark Hoebeke, Stéphane Aymerich, Dörte Becher, Paola Bisicchia, Eric Botella, Olivier Delumeau, Geoff Doherty, Emma L Denham, Mark J Fogg, Vincent Fromion,

Anne Goelzer, Annette Hansen, Elisabeth Härtig, Colin R Harwood, Georg Homuth, Hanne Jarmer, Matthieu Jules, Edda Klipp, Ludovic Le Chat, François Lecointe, Peter Lewis, Wolfram Liebermeister, Anika March, Ruben A T Mars, Priyanka Nannapaneni, David Noone, Susanne Pohl, Bernd Rinn, Frank Rügheimer, Praveen K Sappa, Franck Samson, Marc Schaffer, Benno Schwikowski, Leif Steil, Jörg Stülke, Thomas Wiegert, Kevin M Devine, Anthony J Wilkinson, Jan Maarten van Dijl, Michael Hecker, Uwe Völker, Philippe Bessières, and Philippe Noirot. Condition-dependent transcriptome reveals high-level regulatory architecture in Bacillus subtilis. *Science (New York, N.Y.)*, **335**[6072]:1103–6, mar 2012. 23

[127] Jens Nielsen and Jay D Keasling. Synergies between synthetic biology and metabolic engineering. *Nature biotechnology*, **29**[8]:693–695, August 2011. 8, 15

[128] E Nudler. The riboswitch control of bacterial metabolism. *Trends in Biochemical Sciences*, **29**[1]:11–17, January 2004. 89

[129] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46. ACM, 2000. 84

[130] You-Kwan Oh, Bernhard Ø Palsson, Sung M Park, Christophe H Schilling, and Radhakrishnan Mahadevan. Genome-scale reconstruction of metabolic network in Bacillus subtilis based on high-throughput phenotyping and gene essentiality data. *The Journal of biological chemistry*, **282**[39]:28791–28799, September 2007. 96

[131] Konstantin Okonechnikov, Olga Golosova, Mikhail Fursov, and UGENE team. Unipro ugene: a unified bioinformatics toolkit. *Bioinformatics*, **28**[8]:1166–1167, 2012. 75

[132] Antonio Oliver. Hypermutation in natural bacterial populations: consequences for medical microbiology. *Reviews in Medical Microbiology*, **16**[1]:25–32, 2005. 56

[133] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, **28**[3]:245–248, March 2010. 95

[134] David Lorge Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, **15**[12]:1053–1058, 1972. 83

[135] Michael Pedersen and Andrew Phillips. Towards programming languages for genetic engineering of living cells. *Journal of The Royal Society Interface*, **6**[Suppl 4]:S437–S450, August 2009. 14, 81

[136] J Perkins, A Sloma, and T Hermann. Genetic engineering of Bacillus subtilis for the commercial production of riboflavin. *Journal of Industrial Microbiology and Biotechnology*, 1999. 89

[137] M. Pizka and A. Bauer. A brief top-down and bottom-up philosophy on software evolution. In *Proceedings. 7th International Workshop on Principles of Software Evolution, 2004.*, pages 131–136. IEEE. 4

[138] György Pósfai, Guy Plunkett, Tamás Fehér, David Frisch, Günther M Keil, Kinga Umenhoffer, Vitaliy Kolisnychenko, Buffy Stahl, Shamik S Sharma, Monika de Arruda, Valerie Burland, Sarah W Harcum, and Frederick R Blattner. Emergent properties of reduced-genome Escherichia coli. *Science (New York, NY)*, **312**[5776]:1044–1046, May 2006. 16

[139] Nathan D Price, Jason A Papin, Christophe H Schilling, and Bernhard O Palsson. Genome-scale microbial in silico models: the constraints-based approach. *Trends in Biotechnology*, **21**[4]:162–169, apr 2003. 7

[140] Fatemeh Rajabi-Alni, Alireza Bagheri, and Behrouz Minaei-Bidgoli. An o (n^ 3) time algorithm for the maximum weight b-matching problem on bipartite graphs. *arXiv preprint arXiv:1410.3408*, 2014. 152

[141] Paul Ralph and Yair Wand. A Proposal for a Formal Definition of the Design Concept. In *Design requirements engineering: A ten-year perspective*, pages 103–136. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 2

[142] Sergi Regot, Javier Macia, Núria Conde, Kentaro Furukawa, Jimmy Kjellén, Tom Peeters, Stefan Hohmann, Eulàlia de Nadal, Francesc Posas, and Ricard Solé. Distributed biological computation with multicellular engineered networks. *Nature*, **469**[7329]:207–211, December 2010. 13

[143] Elijah Roberts, Andrew Magis, Julio O. Ortiz, Wolfgang Baumeister, and Zaida Luthey-Schulten. Noise Contributions in an Inducible Genetic Switch: A Whole-Cell Simulation Study. *PLoS Computational Biology*, **7**[3]:e1002010, mar 2011. 7

[144] BF Robertson and DF Radcliffe. Impact of cad tools on creative problem solving in engineering design. *Computer-Aided Design*, **41**[3]:136–146, 2009. 81

[145] M L Ryder. Medieval Sheep and Wool Types. *The Agricultural History Review*, 1984. 11

[146] Laure Saias, Julien Autebert, Laurent Malaquin, and Jean-Louis Viovy. Design, modeling and characterization of microfluidic architectures for high flow rate, small footprint microfluidic systems. *Lab on a Chip*, **11**[5]:822–832, 2011. 126

[147] Jayodita C Sanghvi, Sergi Regot, Silvia Carrasco, Jonathan R Karr, Miriam V Gutschow, Benjamin Bolival, Markus W Covert, and Markus W. Covert. Accelerated discovery via a whole-cell model. *Nature methods*, **10**[12]:1192–5, dec 2013. 7

[148] Claudia Schillings, Mikael Sunnåker, Jörg Stelling, Christoph Schwab, and Costas D Maranas. Efficient Characterization of Parametric Uncertainty of Complex (Bio)chemical Networks. *PLoS Comput Biol*, **11**[8], 2015. 7

[149] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, **9**[7]:676–682, 2012. 126

[150] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature methods*, **9**[7]:676–682, June 2012. 143

[151] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, **9**[7]:671–675, June 2012. 143

[152] Markus Schwehm. Parallel stochastic simulation of whole-cell models. *Proceedings of 2nd International Conference of Systems Biology*, pages 333 –341, 2001. 7

[153] G Sezonov, D Joseleau-Petit, and R D'Ari. Escherichia coli Physiology in Luria-Bertani Broth. *Journal of Bacteriology*, **189**[23]:8746–8749, November 2007. 56, 106

[154] Herbert A Simon. The architecture of complexity. In *Facets of systems science*, pages 457–476. Springer, 1991. 83

[155] Herbert A. Simon. Decision Making: Rational, Nonrational, and Irrational. *Educational Administration Quarterly*, **29**[3]:392–411, aug 1993. 53

[156] DJ Skelton, JS Hallinan, S Park, and Anil Wipat. Computational intelligence for metabolic pathway design: Application to the pentose phosphate pathway. In *2016 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–6. IEEE, 2016. 9

[157] Leyla Slamti and Mathieu Picardeau. Construction of a Library of Random Mutants in the Spirochete Leptospira Biflexa Using a Mariner Transposon. In *Methods in molecular biology (Clifton, N.J.)*, **859**, pages 169–176. 2012. 55

[158] B Smith, A Grossman, and G Walker. Visualization of mismatch repair in bacterial cells. *Molecular cell*, 2001. 63

[159] Hamilton O Smith, Clyde A Hutchison, Cynthia Pfannkoch, and J Craig Venter. Generating a synthetic genome by whole genome assembly: phiX174 bacteriophage from synthetic oligonucleotides. *Proceedings of the National Academy of Sciences of the United States of America*, **100**[26]:15440–15445, December 2003. 16

[160] Elliott Sober. The principle of parsimony. *The British Journal for the Philosophy of Science*, **32**[2]:145–156, 1981. 85

[161] Hyun-Seob Song and Chongxuan Liu. Dynamic Metabolic Modeling of Denitrifying Bacterial Growth: The Cybernetic Approach. *Industrial & Engineering Chemistry Research*, **54**[42]:10221–10227, oct 2015. 7

[162] Heidi M Sosik, Robert J Olson, Michael G Neubert, Alexi Shalapyonok, and Andrew R Solow. Growth rates of coastal phytoplankton from time-series measurements with a submersible flow cytometer. *Limnology and Oceanography*, **48**[5]:1756–1765, 2003. 126

[163] David G Spiller, Christopher D Wood, David A Rand, and Michael RH White. Measurement of single-cell dynamics. *Nature*, **465**[7299]:736–745, 2010. 125

[164] K-P Stahmann, JL Revuelta, and H Seulberger. Three biotechnical processes using ashbya gossypii, candida famata, or bacillus subtilis compete with chemical riboflavin production. *Applied Microbiology and Biotechnology*, **53**[5]:509–516, 2000. 59

[165] Brynne C Stanton, Alec A K Nielsen, Alvin Tamsir, Kevin Clancy, Todd Peterson, and Christopher A Voigt. Genomic min-

ing of prokaryotic repressors for orthogonal logic gates. *Nature chemical biology*, **10**[2]:99–105, December 2013. 13

[166] C R Stewart. Mutagenesis by acridine yellow in Bacillus subtilis. *Genetics*, **59**[1]:23–31, may 1968. 55

[167] Jeffrey V. Straight and Doraiswami Ramkrishna. Cybernetic Modeling and Regulation of Metabolic Pathways. Growth on Complementary Nutrients. *Biotechnology Progress*, **10**[6]:574–587, nov 1994. 7

[168] Ivan E Sutherland. Sketchpad a man-machine graphical communication system. *Transactions of the Society for Computer Simulation*, **2**[5]:R–3, 1964. 85

[169] Yu-ichiro Tago, Masaru Imai, Makoto Ihara, Hironari Atofuji, Yuki Nagata, and Kazuo Yamamoto. Escherichia coli mutator δpola is defective in base mismatch correction: the nature of in vivo dna replication errors. *Journal of molecular biology*, **351**[2]:299–308, 2005. 55

[170] Alvin Tamsir, Jeffrey J Tabor, and Christopher A Voigt. Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature*, **469**[7329]:212–215, January 2011. 13

[171] Brandon K Tan, Mikhail Bogdanov, Jinshi Zhao, William Dowhan, Christian R H Raetz, and Ziqiang Guan. Discovery of a cardiolipin synthase utilizing phosphatidylethanolamine and phosphatidylglycerol as substrates. *Proceedings of the National Academy of Sciences of the United States of America*, **109**[41]:16504–9, oct 2012. 23

[172] Kosei Tanaka, Christopher S Henry, Jenifer F Zinner, Edmond Jolivet, Matthew P Cohoon, Fangfang Xia, Vladimir Bidnenko, S Dusko Ehrlich, Rick L Stevens, and Philippe Noirot. Building the repertoire of dispensable chromosome regions in Bacillus subtilis entails major refinement of cognate large-scale metabolic model. *Nucleic Acids Research*, **41**[1]:687–699, January 2013. 16

[173] Karsten Temme, Dehua Zhao, and Christopher A Voigt. Refactoring the nitrogen fixation gene cluster from Klebsiella oxytoca. *Proceedings of the National Sciences*, 2012. 16

[174] Kostas Terzidis. The Etymology of Design: Pre-Socratic Perspective. *Design Issues*, **23**[4]:69–78, October 2007. 2

[175] M. Thomas and F. McGarry. Top-down vs. bottom-up process improvement. *IEEE Software*, **11**[4]:12–13, jul 1994. 4

[176] M Tomita. Whole-cell simulation: a grand challenge of the 21st century. *Trends in biotechnology*, **19**[6]:205–10, jun 2001. 7

[177] M Tomita, K Hashimoto, K Takahashi, T. Shimizu, Y Matsuzaki, F Miyoshi, K Saito, S Tanida, K Yugi, J. Venter, and C. Hutchison. E-CELL: software environment for whole-cell simulation. *Bioinformatics*, **15**[1]:72–84, jan 1999. 7

[178] Ampere A Tseng, Kuan Chen, Chii D Chen, and Kung J Ma. Electron beam lithography in nanoscale fabrication: recent development. *IEEE Transactions on Electronics Packaging Manufacturing*, **26**[2]:141–149, 2003. 127

[179] María-Carmen Turrientes, Fernando Baquero, Bruce R Levin, José-Luis Martínez, Aida Ripoll, José-María González-Alba, Raquel Tobes, Marina Manrique, Maria-Rosario Baquero, Mario-José Rodríguez-Domínguez, Rafael Cantón, and Juan-Carlos Galán. Normal Mutation Rate Variants Arise in a Mutator (Mut S) Escherichia coli Population. *PloS one*, **8**[9]:e72963, September 2013. 55, 56

[180] V Vagner, E Dervyn, and S Ehrlich. A vector for systematic gene inactivation in Bacillus subtilis. *Microbiology*, 1998. 24, 25, 65

[181] Marc H V Van Regenmortel. The rational design of biological complexity: A deceptive metaphor. *Proteomics*, **7**[6]:965–975, 2007. 53

[182] Marc HV Van Regenmortel. Reductionism and complexity in molecular biology. *EMBO reports*, **5**[11]:1016–1020, 2004. 8, 82

[183] John Carl Villanueva. How many atoms are there in the universe. *Universe Today*, **30**, 2009. 110

[184] Alexey G Vitreschak, Dmitry A Rodionov, Andrey A Mironov, and Mikhail S Gelfand. Regulation of riboflavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Research*, 2002. 87, 88

[185] Adam C. Wilson and Hendrik Szurmant. Transposon-Mediated Random Mutagenesis of Bacillus subtilis. In *Methods in molecular biology (Clifton, N.J.)*, **765**, pages 359–371. 2011. 55

[186] W Winkler, R Breaker, and D Crothers. The speed of RNA transcription and metabolite binding kinetics operate an FMN riboswitch. *Molecular cell*, 2005. 61

[187] Sydnor T. Withers and Jay D. Keasling. Biosynthesis and engineering of isoprenoid small molecules. *Applied Microbiology and Biotechnology*, **73**[5]:980–990, dec 2006. 15

[188] Tuck Seng Wong, Danilo Roccatano, Martin Zacharias, and Ulrich Schwaneberg. A statistical analysis of random mutagenesis methods used for directed protein evolution. *Journal of molecular biology*, **355**[4]:858–871, 2006. 55

[189] Niel Woodford and Matthew J Ellington. The emergence of antibiotic resistance by mutation. *Clinical Microbiology and Infection*, **13**[1]:5–18, 2007. 56

[190] A Wysoker, K Tibbetts, and T Fennell. Picard tools version 1.90, 2013. 67

[191] Younan Xia and George M Whitesides. Soft lithography. *Annual review of materials science*, **28**[1]:153–184, 1998. 126

[192] Tim Wing Yam and Joseph Arditti. History of orchid propagation: a mirror of the history of biotechnology. *Plant Biotechnology Reports*, **3**[1]:1–56, January 2009. 11

[193] G Yang, J Rich, M Gilbert, and W Wakarchuk. Fluorescence activated cell sorting as a general ultra-high-throughput screening method for directed evolution of glycosyltransferases. *Journal of the American Chemical Society*, 2010. 57, 58

[194] Daniel R Zeigler, Zoltán Prágai, Sabrina Rodriguez, Bastien Chevreux, Andrea Muffler, Thomas Albert, Renyuan Bai, Markus Wyss, and John B Perkins. The origins of 168, W23, and other Bacillus subtilis legacy strains. *Journal of bacteriology*, **190**[21]:6983–95, nov 2008. 23