Genetic Stability and Costs of Engineered Biological Constructs



Axel Nyström

Corpus Christi College

University of Oxford

A thesis presented for the degree of

Doctor of Philosophy

Trinity 2017

To Tuss

Acknowledgements

First and foremost I thank my two fantastic supervisors Prof. Antonis Papachristodoulou and Dr. Andrew Angel. I thank you for helping me to identify and unravel the most interesting research questions from what was at first a maze of options and wide landscape of criss-crossing paths. I am equally grateful for your guidance for finding and selecting the procedures and methods for finding answers to my questions. In addition I sincerely thank you for your great personal support and warm encouragement during these years. This thesis would not exist without your insightful guidance and encouraging support.

I thank my thesis committee and transfer assessors Prof. Wei Huang, Dr. Alessandro Abate and Dr. George Wadhams for important comments, interesting discussions and valuable feedback. I thank Prof. Judith Armitage for valuable research feedback, and also for her kind support and guidance.

During the course of producing this thesis, and in addition to structured help and guidance, I have also had the fortune to meet and interact with many talented and interesting people. These interactions have greatly expanded my knowledge, while they have also generated many fun and interesting conversations. In particular I wish to thank: Andreas Harris, James Scott Brown, Nicolas Delalez, Dhruva Raman, James Anderson, Thomas Prescott, Yang Zheng, Aivar Sootla, Reza Ahmadi, Thomas Folliard, Alberto Merchante Gonzalez, Thomas Brown, Roderick Lubbock, Nicholas Bartos, Tyler Ambrose, James Johnson, Robert Main and Ciaran Kelly.

Finally I thank my wonderful family who with their love, support, and great patience have let me dedicate so many of my days and weekends for producing this thesis. To Vito, Alma, and particularly to Milena, I know you have been waiting a long time for this work to finish. I will always be grateful for that. Here it is.

Abstract

Synthetic Biology applies engineering principles to biology in order to formalise and facilitate the design of biological systems. Current applications of Synthetic Biology include agriculture, tissue engineering, chemical sensors, drug discovery and manufacturing. One of the field's primary foci is the construction of complex biological entities with predictable behaviours from sets of standardised components. However, the insertion of engineered biological entities causes fitness costs to their host which engenders a disadvantageous selection pressure and favours loss of function mutations. This is a key challenge to the robust engineering of biology as it reduces the construct's genetic stability and functional lifetime. Moreover, many end goals of Synthetic Biology are applications which involve close contact with humans, other living organisms and the environment as a whole. As such, system designs need to be trusted as operating faithfully to specification. Previous work has focussed on enhancing construct stability and lifetime by reducing the mutational frequency of engineered DNA sequences. This thesis focuses on burden and costs in order to improve the stability of engineered functions. To this aim a variety of methods are investigated, ranging from deterministic models of cellular resource allocation to stochastic evolutionary models for engineered constructs. A design model is presented for predicting and identifying burden due to unfavourable design ratios between transcription initiation rates, translation initiation and elongation efficiency. This was tested against high-throughput data from an automated experimental protocol, which identified a need for improved quantification methods of burden. A dynamic resource allocation model was formulated to investigate how burden manifests during growth-phase transitions. When tested against growth data this model identified three new parameters for burden from engineered constructs: delayed signal response to a new environment; peak ribosome expression levels during growth-phase transitions; and the rate of ribosome loss during adverse growth conditions. Lastly, a stochastic evolutionary model was developed for how hosts can adapt to accommodate engineered constructs, *i.e.* to reduce the fitness cost while preventing function loss. The adaptive model was used to inform time cycles, population sizes and colony picking protocols that enhanced such accommodation. Favourable conditions were found to be dependent on the culture environment and engineered construct parameters.

Contents

0 Introduction **8**

0.1 Synthetic Biology 8

0.2 Genetic Stability and Burden of Engineered Biology 10

0.2.1 Burden and Fitness Costs 12

0.2.2 Mutation Rate, Fitness Cost and Measurement Volumes 15

0.3 Thesis Outline 22

1 Chapter 1 **24**

1.1 Introduction 24

1.1.1 Genetic Stability in the Design-Build-Test Cycle 24

1.1.2 Automation of Experimental Data Interpretations 27

1.2 Experiment Automation 28

1.2.1 Experiment Automation in Practice - Robotic Implementation . . . 30

1.3 Experimental Methods and Design Conditions 35

1.3.1 For all the experiments 35

1.3.2 For all plasmid transformations 35

1.3.3 For all experiments for burden predictions and model developments (Chapters 2 and 3) 36

1.3.4 Experiments for web-interface interaction for engineering biology (Chapter 1, Section 1.4) 37

1.4 Online Automation and Digital Synthetic Biology 38

1.4.1 Laboratory Automation Cloud Software Tools for Synthetic Biology 42

1.4.2 The Nuts and Bolts of the Digital Synthetic Biology Platform 43

1.4.3 Examples and Results of the Digital Synthetic Biology Platform . . 45

1.5 Discussion and Contributions . . . 57
1.5.1 Discussion . . . 57
1.5.2 Contributions . . . 59

2 Chapter 2 **60**
2.1 Introduction . . . 60
2.2 Design Models for Burden . . . 63
2.2.1 RBS versus Promoter Strength . . . 64
2.3 Biology Design Parameters for Experiments . . . 75
2.3.1 Promoters for Experiments . . . 75
2.3.2 Ribosome-Binding-Sites for Experiments . . . 77
2.3.3 Plasmid Replication Systems for Experiments . . . 79
2.4 Library Expression Levels . . . 83
2.4.1 Experimental Procedures . . . 83
2.4.2 Expression Levels . . . 84
2.5 Library Growth Levels . . . 90
2.5.1 Experimental Procedures . . . 90
2.5.2 Growth Levels . . . 92
2.6 Relating Model to Data . . . 97
2.6.1 Relative Growth Level and Expression . . . 97
2.6.2 Growth Rates and Fraction Sizes . . . 97
2.6.3 Engineered Fraction Sizes: Φ_E . . . 100
2.6.4 Engineered Fraction Sizes: R_E . . . 103
2.6.5 Model versus Data . . . 105
2.7 Conclusions and Contributions . . . 111

3 Chapter 3 **112**
3.1 Introduction . . . 114
3.1.1 Global and Coarse-grained Resource Allocation Models . . . 114
3.1.2 Model Development . . . 117
3.2 Methods . . . 119

3.2.1 Data Collection and Measurement Methods . . . 119
3.2.2 Data Fitting . . . 119
3.3 Fitting of Data to an Existing Growth Model . . . 120
3.3.1 Growth Curves and Models . . . 120
3.3.2 Data fitting to the Gompertz function . . . 127
3.4 Derivation of the Dynamic Resource Allocation Model (DRAM) . . . 130
3.4.1 Fitting of a Dynamic Resource Allocation Model to Gompertz growth curves . . . 133
3.4.2 Fitting of DRAM directly to cell growth measurements . . . 141
3.4.3 Static R_R DRAM fitting from μ_L data . . . 142
3.4.4 R_R delayed response for DRAM fitting from μ_L data . . . 143
3.4.5 Initial Φ_R levels and correlations from ENV_1 to ENV_2 . . . 156
3.4.6 Parameterisation of the decline of the Φ_R fraction . . . 156
3.5 Conclusions and Contributions . . . 162
3.5.1 Conclusions . . . 164
3.5.2 Contributions . . . 166

4 Chapter 4 **168**

4.1 Introduction . . . 168
4.1.1 Fitness Costs of Engineered Constructs . . . 171
4.1.2 The LTEE data source . . . 173
4.1.3 Distribution of Fitness Effects . . . 175
4.2 Approach . . . 182
4.2.1 Model Step 1 - DFE fitting to LTEE data . . . 182
4.2.2 Model Step 2 - Including Engineered Constructs . . . 204
4.3 Engineered Construct Accommodation and Results . . . 215
4.3.1 Simulation Results . . . 217
4.4 Discussion . . . 230
4.4.1 Practical Use of the Model . . . 231
4.4.2 An Initial Proposal of Experimental Setup . . . 231

4.4.3 Model Validation and Improved Parameterisation 236
4.4.4 Tracking Time or Loss of Construct 238
4.4.5 Different Constructs 238
4.5 Conclusions and Contributions 240

5 Chapter 5 **241**
5.1 Thesis Summary 241
5.2 Summary of Key Limitations 243
5.3 Future Work 244

0. Introduction

This thesis investigates genetic stability and burden in Synthetic Biology. This chapter serves as a brief introduction to the field of Synthetic Biology in general, and the investigation of genetic stability in Synthetic Biology in particular. The key concepts of engineered costs and loss frequency are introduced, as well as the measurable metrics used for genetic stability in Synthetic Biology. This chapter concludes with a concise outline of the thesis.

0.1 Synthetic Biology

Synthetic Biology is the design and engineering of biologically based parts, novel devices and systems as well as the redesign of existing, natural biological systems [44]. It builds on foundational research from a range of fields (including systems engineering, biochemistry, computer science, plant sciences, chemical engineering, informatics and molecular biology), and integrates previous findings with the help of engineering design principles [44].

The first proof-of-concepts for Synthetic Biology emerged in the year 2000 with the successful constructions of a bacterial toggle-switch [71] and a bacterial oscillator [63]. Synthetic Biology has since emerged as an independent research field with a rapidly expanding literature and the creation of an immense number of synthetic devices. The variety of functions that have been designed and built include counters [69], switches [7, 104], logic formulas [152], timers [50, 62, 70], time delay circuits [193], biocomputers [10] and more oscillators [178, 50, 7]. The success of Synthetic Biology in rationally engineering regulatory circuits has enabled new practical advances in areas ranging from pharmaceutical manufacturing [190], tissue engineering [159], prototype gene therapies [97] and drug discovery [191, 192]. Synthetic Biology has recently expanded from using single cell engineered func-

tions into interconnected systems distributed between consortia of engineered cells, and connected via modular devices for intracellular communication [11]. Engineered systems dispersed between different cells provide many clear advantages for future Synthetic Biology applications. More complex systems can be built when partial functions can be split up between a larger number of hosts, spatial directions and separations can be created for functional interactions, while it further allows for an increasing plug-and-play approach where subsystems can be tested and replaced more rapidly [11, 53, 172]. However, the development of robustly functioning multicellular systems also requires higher and increasingly well-defined engineering standards to enable predictable design.

Synthetic Biology relies on three foundational tenets: modularisation, characterisation and standardisation. Modularisation consists of decomposing a system into a series of well-defined parts or components (*e.g.* open reading frames of proteins, transcriptional terminators, promoters or other regulatory components). It has been argued that the application of engineering principles to biology is possible because biology is inherently modular; for example biological functions are expressed through proteins and RNA which is primarily encoded in DNA [44]. However, and from a very different perspective, it has also been argued that biology is inherently non-modular as it is currently difficult to predict the function of a larger biological system from the sum of its genetic components [105, 33]. An objective of Synthetic Biology is to use the diversity of parts and subcomponents already present in nature, and combine these in novel configurations to build new devices and systems [44, 45]. To enable the application of engineering principles to biology it is first necessary that the biological entities plus their assemblies can be mathematically modelled and quantitatively understood [128, 32]. Characterisation is the process of defining the behaviour and function of biological components. Mature engineering fields have made significant advances through the standardisation of components and processes [44, 32]. An ongoing challenge for Synthetic Biology is to create and develop equally useful standards for biological systems. A key aspect which separates Synthetic Biology from other molecular biotechnology approaches is the ambition to formalise and scale the design complexity for making new bespoke functions [4]. Design complexity in biology will be enabled by well-characterised and modular components which are defined according to clearly specified

standards.

Biology, biotechnology and genetic engineering have traditionally been discovery-based research fields, moving forward with empirical tests. The workflow in biology is in stark contrast to more mature engineering fields; *e.g.* the development of a new software is not dependent on new empirical discoveries. Engineering fields can follow roadmaps, whereas empirical science fields are constrained and limited to scientific discoveries. The purpose of Synthetic Biology research is to transport biology from a world of empirical scientific discovery into a world of predictable engineering.

0.2 Genetic Stability and Burden of Engineered Biology

Synthetic Biology applies mechanistic understanding of molecular biology in order to engineer new biological functions [41]. One of its primary foci has been the construction of complex biological entities with predictable behaviours from sets of standardised genetic sequences [32, 98]. However, living systems also introduce unprecedented engineering challenges. Copying errors from mutations degrade the DNA blueprints for biological functions and stop them from operating according to specification [149], in other words, these machines evolve [5, 64].

According to the current convention, the reliability of an engineered construct is quantified using its half-life ($E\frac{1}{2}$) defined as the number of generations over which 50% of constructs persist [32]. A typical engineered function comprised of only a few genes may be reliable for $\sim$ 100 cell divisions in an *E. coli* host [171, 149], equivalent to 2-3 days during moderate growth. This limit can be acceptable for simple constructs intended for a single day experiment in laboratory conditions [149]. Nevertheless, genetic instability can cause problems also at laboratory scale. Two notable examples include: (i) mutational breakdown within 3-6 days for circuits aimed at limiting cell growth [209], and (ii) an engineered system of "predator" and "prey" *E. coli* having to be studied in a micro-chemostat due to loss of function by the number of generations required to reach a normal culture size [14].

Synthetic Biology has an ambitious aim to serve practical applications as well as ex-

tending design methods to large scale gene networks [41]. Unfortunately, the current level of genetic design breakdown seen for single gene constructs will be amplified as the number of nodes susceptible for failure increase. Put simply, time before failure will be much shorter. In addition, beneficial applications might require much longer, instead of shorter, functional lifetimes.

A synthetic biologist faced with the prospect of genetic breakdown can adopt two different approaches for addressing genetic instability:

- Reduce the rate of function loss for designed genetic constructs [92].
- Accept that engineering biology includes malfunctioning of designed constructs over time [149, 64, 5].

This project favours the former approach. However, even if selecting the latter method it remains important to quantify, measure and predict the level of construct breakdown to comprehend design and application limits. The unpredictability of engineered biology is one of the primary reasons why it is difficult to assess the risk of engineered constructs. Targeted quantification of construct trajectory will enable enhanced applicability by reducing the risk of potential bio-contamination [149, 124].

For a large number of previously implemented genetic designs, the most direct method to ameliorate function breakdown is to reduce the frequency at which mutations are introduced [92, 171]. The basal mutation rates for *E. coli* DNA is 1-2 $\times 10^{-10}$ per basepair per cell division [106, 198]. However, large numbers of currently designed and used constructs have mutation rates many orders of magnitude above this basal rate, primarily due to repeats of identical DNA sequences causing deletions from both homologous recombination and elevated replication slippage [92]. An online tool has recently been made available ("the EFM Calculator") for the detection of: (i) homologous recombination and (ii) indels caused by simple sequence repeats, which are two of the most common mutational hotspots that drastically increase the frequency of construct destructions [92]. Experimental studies have also demonstrated that protein expression levels in the form of a "cost" parameter can have a profound impact on constructs' half-lives [171].

In order to turn Synthetic Biology into an engineering discipline of broader practical

application, allowing larger and more complex genetic designs, it is essential to increase attention to understand, quantify but also be able to predict rates of construct breakdown in conjunction with their inherent links to construct design. It is the aim of the work presented in this thesis to provide a stepping stone and pathway for the future design of robust genetic systems, small or large.

0.2.1 Burden and Fitness Costs

In its simplest form, genetic stability of engineered biology can be described as a function of two parameters:

1. The frequency of mutations destructive to the function (μ_d).

2. The fitness cost that a functional construct is causing to its host (S).

The rate of destructive mutations is a function of (i) the basal mutation rate of the host [106, 198], (ii) the length of DNA used to encode the construct (*i.e.* the 'size' of the construct) and (iii) the presence of mutational hotspots within the DNA sequence of an engineered construct [92].

Fitness is a term used to capture the reproductive success of a given genotype [73, 136], and 'construct fitness cost' refers to the reduction in reproductive success of a host organism caused by the construct. The fitness cost of a construct is difficult to assess during the engineering design phase, but it can be tested experimentally. Measurable units are required in order to quantify engineered burden and fitness costs.

The relative fitness, denoted w, quantifies the difference in fitness between two genotypes [42, 73]. The relative fitness of a query strain (w_q) versus a reference strain can be described as follows

$$\frac{R_q(\tau)}{R_r(\tau)} = w_q^{\tau}, \tag{0.1}$$

where $R(\tau)$ is the fold change in cell population (N) after τ generations, such that $R(\tau) = \frac{N_{x+\tau}}{N_x}$; τ is measured with respect to generations of reference strain r; R_r: fold change for parent reference strain; R_q: fold change for query strain. The reference fitness value is

equal to one, such that a relative fitness value larger than one ($w > 1$) signifies an increase in fitness, and a value below one ($w < 1$) means a reduction in fitness.

Similarly, the selection coefficient (denoted s) is used to describe the reduction in fitness between a mutated genotype *versus* its parent [73]:

$$s = 1 - w_q, \tag{0.2}$$

where w_q is the relative fitness described in Equation 0.1. A positive value for s means that the fitness has been reduced, and a negative value for s that the fitness has been improved. If $s = 1$ then the fitness has been reduced to zero and the genotype is not viable, *e.g.* due to a lethal mutation [73, 136].

A common growth rate measure which can be related to the selection coefficient is the specific growth rate (λ):

$$N_{x+t} = N_x e^{\lambda t} \tag{0.3}$$

where N is the population size, t is time and λ is the specific growth rate.

From Equations 0.1 and 0.2 then the relative fitness and the growth rate are linked as follows,

$$\frac{e^{\lambda_q t}}{e^{\lambda_r t}} = w_q^{(t/T_r)},$$

where λ_r is the growth rate of the parent strain, λ_q is the growth rate of the mutated strain, and T_r is the generation time for the referens strain. Also,

$$(\lambda_q - \lambda_r)t = \ln(w_q)(t/T_r),$$

$$(\lambda_q - \lambda_r)T_r = \ln(w_q).$$

For populations where assuming a doubling of the population size at generation cycle, then $2 = e^{\lambda_r T_r}$, and $T_r = \ln(2)/\lambda_r$. From substituting the generation time (T_r),

$$(\lambda_q - \lambda_r)\frac{\ln(2)}{\lambda_r} = \ln(w_q),$$

$$\ln(2)\Big(\frac{\lambda_q}{\lambda_r} - 1\Big) = \ln(w_q),$$

and

$$w_q = 2^{\frac{\lambda_q}{\lambda_r}-1}, \tag{0.4}$$

or equally (from Equation 0.2)

$$s = 1 - 2^{\frac{\lambda_q}{\lambda_r}-1}. \tag{0.5}$$

The referens generation time (T_r) is equal to the time required for one cell divison of the parent strain. Note, here the growth rate is given the symbol λ. However, in Chapter 3 the symbol μ is used for growth rate in order to separate it from the growth lag-phase, which traditionally is also annotated with λ.

Figure 0.1 illustrates the relationship between the growth rate (λ), destructive mutations (μ_d) and subpopulation (N) growth. Losing an engineered construct from mutation leads to an increase in fitness as the growth rate of cells move towards the original wild type growth rate. In this thesis, a capital S is used for the specific case of referring to a fitness disadvantage and cost of hosting an engineered construct, and is the fitness coefficient where the wild type is the reference strain (*i.e.* $w_r = 1$), and its engineered version (which carries an intact construct) is the query strain, and has fitness w_q. In other words, the capital S simply means the fitness reduction inherent to hosting a specific engineered construct, and emphasises this origin of fitness change versus other fitness changes (including those caused by random mutations). Further, a distinction is made between burden and fitness costs. Burden refers to a fitness cost from competition for resources within a cell, *i.e.* the competition in the cellular economy between resources needed for cell division and resources being consumed by engineered expression. A broader definition is attributed to the fitness cost, or *cost*, of a construct. The fitness cost (S) is a measurable output of all negative effects given to a host from an engineered construct. In addition to burden from resource competition, engineered constructs can reduce the overall fitness of a host for a variety of reasons, including toxicity [202] or pathway imbalances [120, 100].

One approach to increase the genetic stability of systems suffering from high burden could be to repartition the gene expressing machinery into different compartments. Some

progress has been made with controlling and dividing resource allocation in microbial hosts [52]. In theory, if resources in a cell are already pre-divided between separate compartments of growth and engineered functions, then loss of function from engineered systems will not increase resources available for growth. Therefore, the evolutionary pressure of function loss could be alleviated, because losing an engineered function does not increase the pool of resources available for growth. However, in practice it could be technically difficult to implement a robust and evolutionary resistant compartmentalisation system.

0.2.2 Mutation Rate, Fitness Cost and Measurement Volumes

The current method and unit for measuring genetic stability of engineered constructs was provided in the half-life metric ($E^{\frac{1}{2}}$), defined as the number of generations it takes for a designed function to be lost from half the population [32]. The half-life is obtained experimentally through subsequent redilutions of a culture, until only half of the individuals in the population maintain the initially provided design. The half-life is measured in the unit of generations (τ), or cell division (not in time, or the number or frequencies of medium dilutions, both of which would obscure comparisons between organisms or media with different growth rates). Figure 0.1 illustrates the origins of the key factors affecting genetic stability (destructive mutation rates and growth rates), and Figure 0.2 shows how these two factors affect the half-life of a construct. Figure 0.1 illustrates a growth rate impact (due to a mutation) as a change in cell divison time ($\ln 2/\lambda$), which in turn can be converted to a relative fitness value or selection coefficient via Equation (0.4) or (0.5).

The values in Figure 0.2 are obtained from a simple deterministic simulation of the rate of destruction and the resulting growth rate increase. The deterministic calculation used in Figure 0.2 simply means that the population size is assumed to be infinite. In Figure 0.2 the half-life is given on the vertical axis as a function of the destruction rate (μ_d) and the fitness cost of the construct (S). When either the mutational likelihood of losing a construct (μ_d) or the relative fitness cost of keeping a construct (S) increases, this results into a reduced half-life ($E^{\frac{1}{2}}$). Figure 0.2 also highlights the construct T9002 (A), which was the first construct measured for half-life genetic stability when the unit was developed in 2008 [32]. Mutational loss of the construct was shown to decrease the

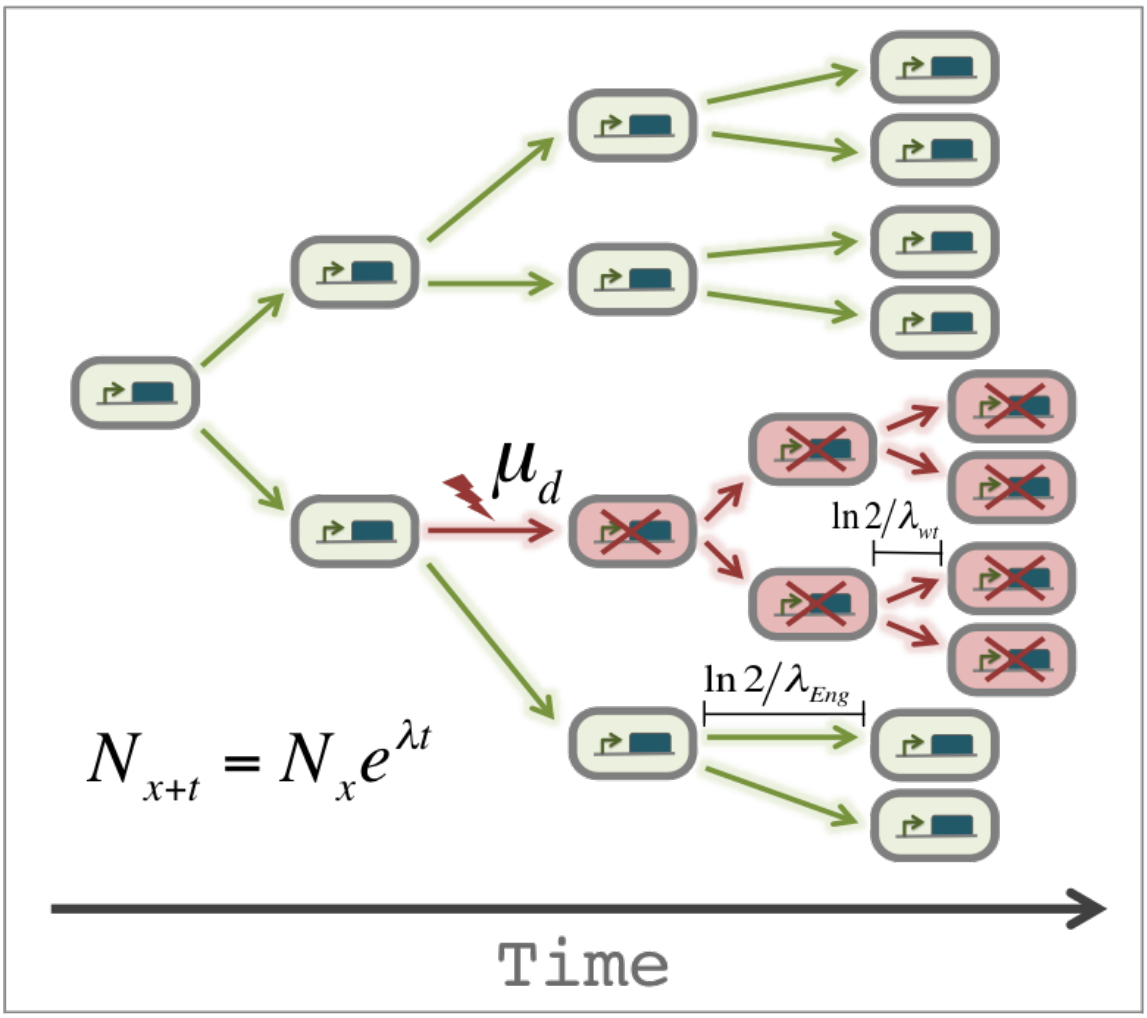


Figure 0.1: The loss of engineered constructs over time. Natural mutations destroy the engineered function with a frequency of μ_d. Destruction of the engineered function from mutation fully or partly restores the original wild type fitness (λ_{wt}) with an increased replication rate. The subpopulations (N) increase with a growth rate of λ and have a cell divison generation time of $\ln(2)/\lambda$. The growth rate is linked to relative fitnesses via Equation (0.4).

cell division time ($\ln(2)/\lambda$) from 59 minutes to 54 minutes. The genetic stability half-life ($E_{1/2}$) was measured to 56 generations. Later studies by a different group improved the genetic stability of this construct by removing homology between two (previously identical) 183-basepair transcriptional terminators. The improved construct is annotated with B in Figure 0.2.

In order to introduce some of the concepts and notations in this thesis, we will here look at what happens to construct stability at general measurement conditions. The general concepts are revisited in Chapter 4, but here we will look at the half-life measurement unit. The purpose of this is to see whether the half-life metric is a consistent and reliable metric for genetic stability, or if other factors, such as growth volume, also have an impact. The unit of half-life with a fixed value for genetic stability might not be the best method for understanding and parameterising genetic stability observed in experiments. Figure 0.3 shows the same genetic stability input parameters (μ_d and S) as Figure 0.2 but simulates them stochastically for a range of experimental working volumes (*i.e.* the volume within a container which is being used for growth, *e.g.* the volume of growth medium). At different working volumes of experimental measurements the population sizes in each volume will be different, even if all other parameters stay constant (*e.g.* growth medium, temperature, etc). Figures 0.3 and 0.4 are produced with Monte Carlo simulations implemented in Python. The calculations make the following basic assumptions:

- Each bacterium replicates into two daughter cells at cell divison.
- One of the daughter cells receives the original genome and the other daughter cell receives a copy of the genome.
- Mutations disrupting the output of engineered constructs arise during genome replication at likelihood μ_d and locate at the newly produced genome copy.
- Backward mutations are nonexistent (*i.e.* mutations cannot restore the exact original function in an already mutated construct).
- The rate of replication of a bacterium, compared to other bacteria, depends on its relative fitness.

- Relative fitness depends on the cost of expressing a specified engineered construct.
- The populations are growing at a steady-state growth phase [60] by maintaining an optical cell density (600nm) below 0.3 in Luria-Bertani broth, equivalent to 5x10^7 cells/mL [167].
- Cells are diluted ten-fold into fresh medium upon reaching growth-phase limiting cell concentrations.

In Figure 0.4 the example construct (denoted E in Figure 0.2) is simulated with the same stochastic method as used for Figure 0.3 and the distribution and spread of results are compared between different working volumes. The example construct produces a fitness cost of $S = 0.2$, and has a sequence length of 500 - 2,000 bp ($\mu_d =\sim 10^{-7}$ for a host with a mutational frequency of $1 - 2 \times 10^{-10}$ [106, 198]). At small population sizes the half-life is increased, while the probability distribution between samples is wide.

Figures 0.4 and 0.3 show how the previously provided half-life [32] unit for genetic stability is intrinsically dependent on the working volume and population size used during experimental tests. For each combination of μ_d and S, there will be a threshold for the working volume by which a measured half-life begins to approximate the half-life value for an infinite cell population (see comparison between Figures 0.2 and 0.3). Nevertheless, the variance and sample distribution can still be considerable (see Figure 0.4). Instead of using the current genetic stability unit of half-life ($E\frac{1}{2}$), an improved and more accurate method of representing genetic stability would be to simply specify the stability as the two combined values of μ_d and S, knowing that stability (as well as spread of readings) will differ depending on test conditions.

In order to improve the genetic stability of engineered biology, most previous work has investigated mutation frequencies. Improving the genetic stability through reducing the rate of mutation is an iterative and empirical process. In practice the process consists of: (i) identifying the most common mutation leading to functional loss (often a sequence repeat leading either to deletion through homologous recombination or indels caused by replication slippage [92, 171, 149]), (ii) recode the identified sequence to a more stable alternative, (iii) assess what improvement is made and (iv) restart the process with the

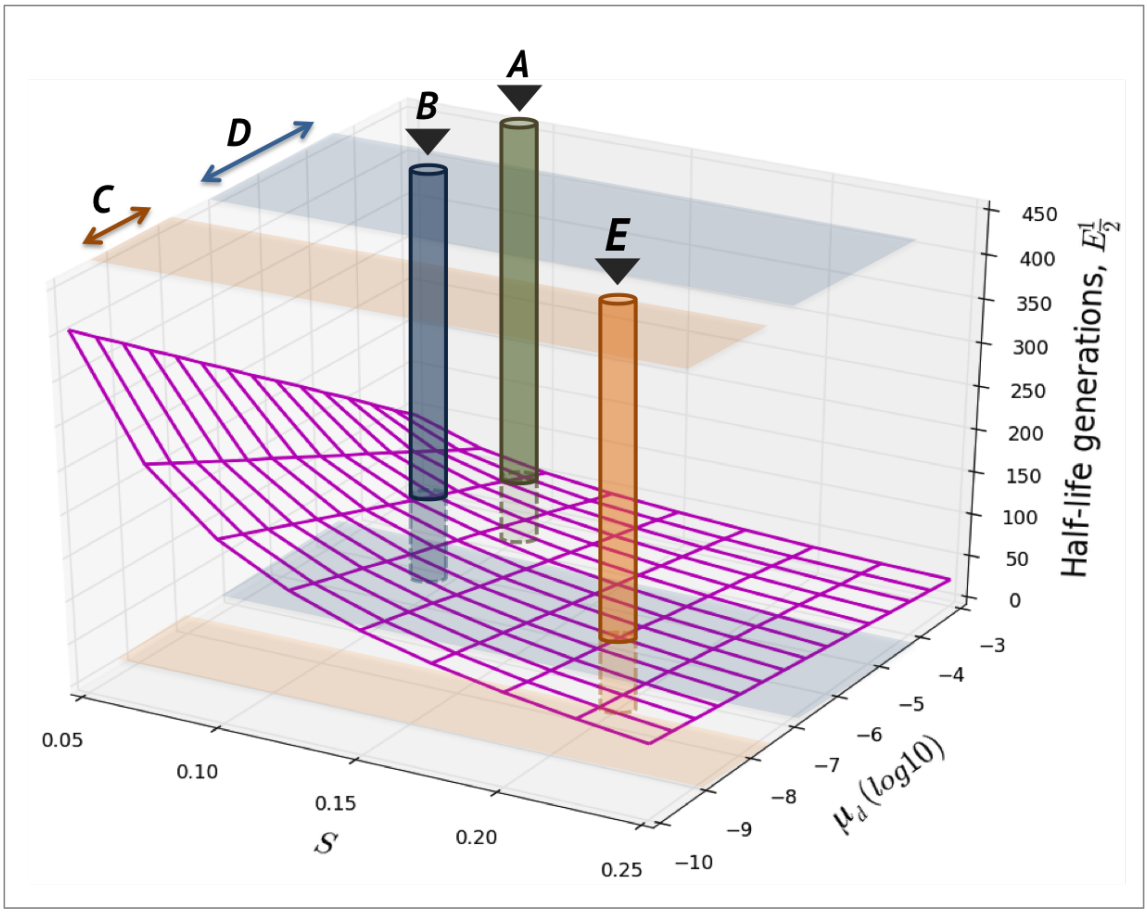


Figure 0.2: The effect on half-life ($E_{\frac{1}{2}}^{1}$) from the destructive mutation frequency (μ_d) and fitness cost (S) of a construct. **A:** The first half-life tested construct, in 2008 [32]. **B:** An improved version of A, made by a 17-fold reduction of the destructive mutation rate, in 2010 [171]. **C:** The expected range of destructive mutation frequencies for a construct with a size of 100-5,000bp. **D:** The estimated range of destructive mutation frequencies for the majority of constructs in the Registry of Standard Genetic Parts [92]. **E:** An example construct without excessive DNA sequence repeats and with a high protein expression level.

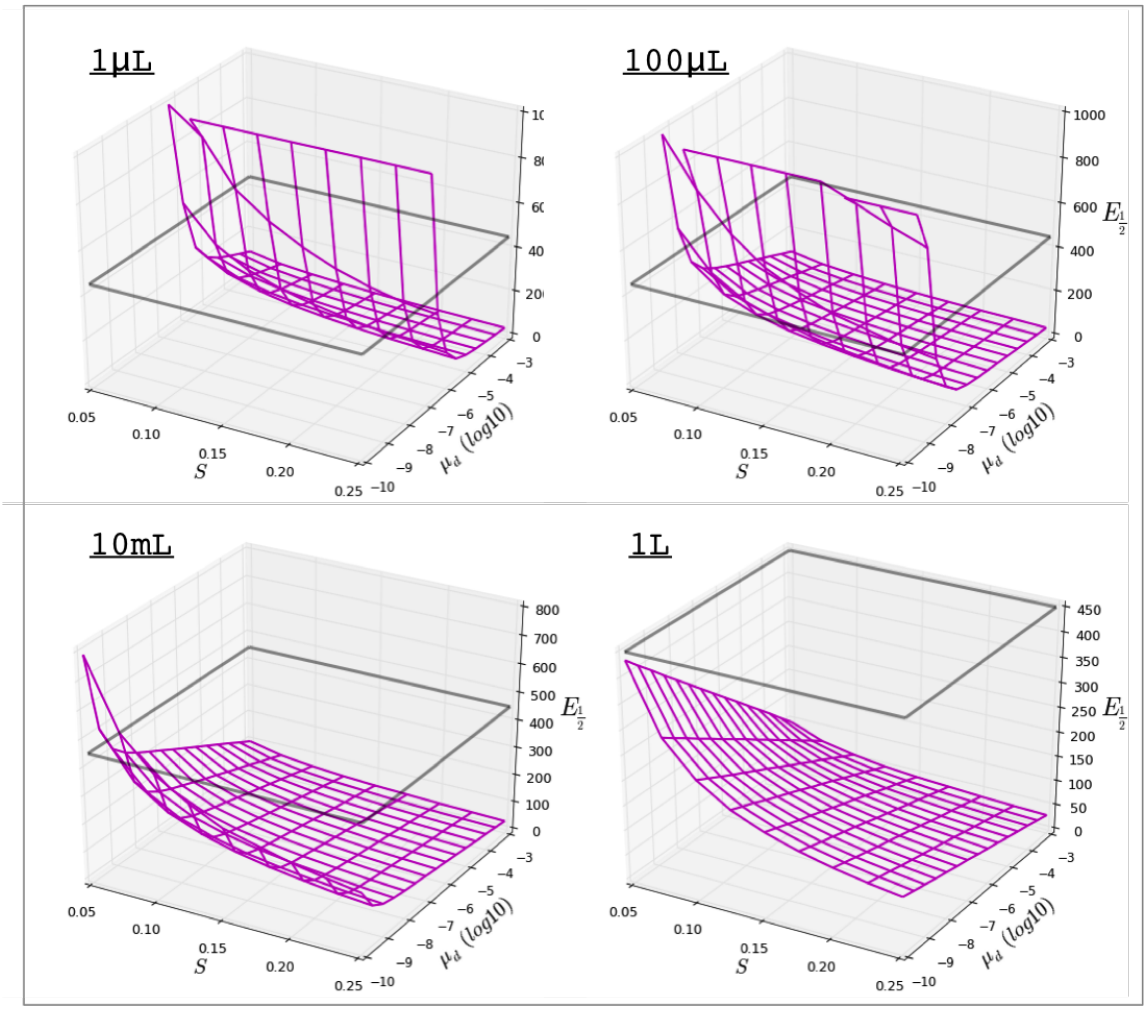


Figure 0.3: Stochastic simulation results of genetic stability of *E. coli* constructs in different volumes for growth. The simulations show that the half-life from an engineered construct is strongly dependent on the container sizes and cell densities used for measuring. At very large measurement cultures, construct sizes or mutational frequencies, the half-lives approximate the deterministic half-lives that are shown in Figure 0.2. The simulation assumes a ten fold dilution into fresh growth medium each time the population reaches an optical density (OD_{600}) of 0.3, which is equal to 5×10^7 cells per mL [167]. μ_d is the frequency of destructive mutations. S is the fitness reduction from hosting the construct relative to the wild type. $E_{1/2}$ is the evolutionary half-life as defined previously [32], *i.e.* when half the population has lost the engineered construct. The data shown on the plots are capped at a half-life of 1000 generations. The gray square is a visual cue at 450 generations. The plotted values are the mean result after 100,000 model realisations of each parameter combination.

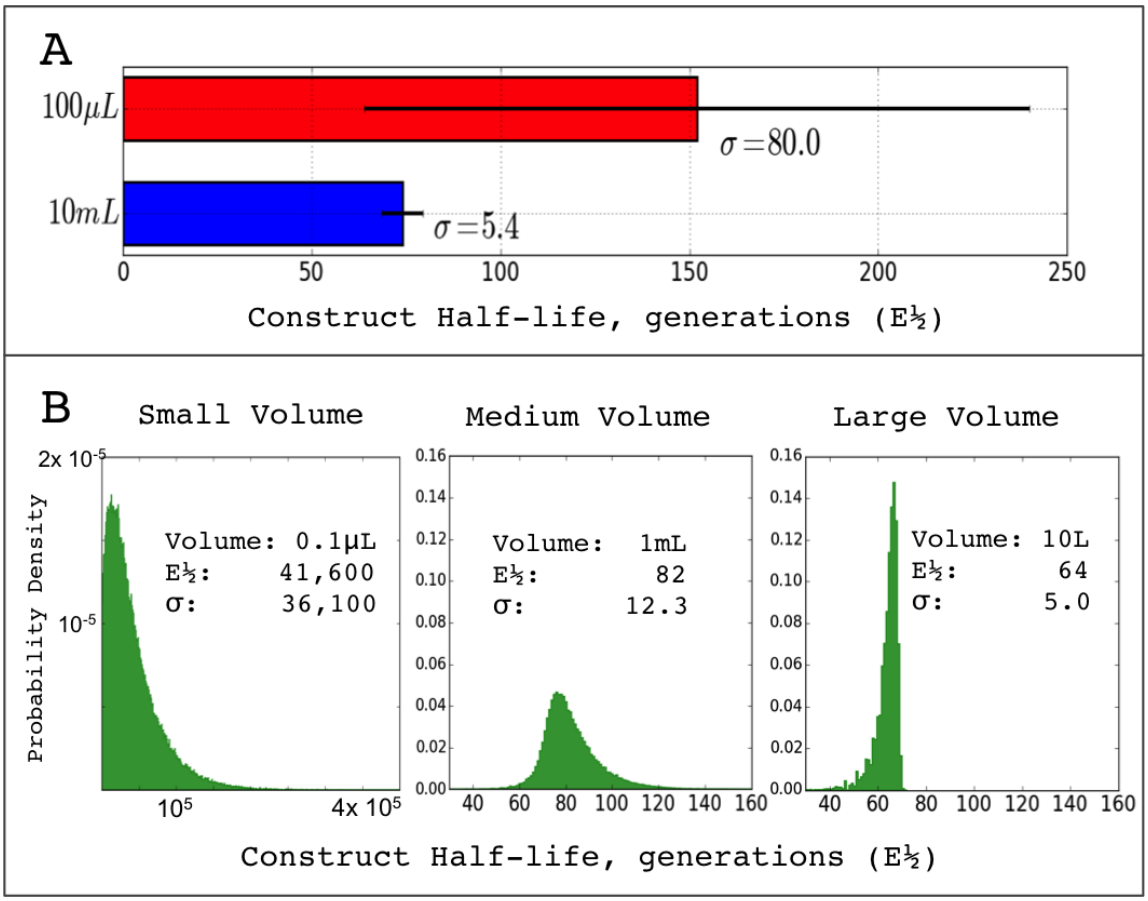


Figure 0.4: The half-life of the example construct C in Figure 0.2, simulated at different growth volumes and population sizes. **A:** The mean half-life and standard deviation (σ) simulated at experiment container sizes of $100\mu L$ and $10mL$. **B:** The probability distributions of half-lives for three different container volumes and population sizes. Each volume is simulated with 100,000 model realisations.

second most common source of mutations.

Instead of mutational frequency, this thesis focuses on the instability contributed by burden. In particular, the work investigates (i) how to predict burden upfront, (ii) methods to measure burden and (iii) how to reduce burden; all in order to successfully predict and increase the functional life time of engineered constructs.

0.3 Thesis Outline

The structure of the thesis is as follows. Chapter 1 develops automated experiment protocols and robotic methods for the investigation of burden over long time-periods and with high repeatability. The methods developed in Chapter 1 were used for generating experimental data in Chapters 2 and 3. Chapter 1 also develops new digital abstraction layers for engineering biology which facilitate general Synthetic Biology research. This work was made into an online tool which can be accessed by investigators remotely.

Chapter 2 produces models to detect, measure and reduce burden from engineered biology during the design phase. In particular, a model was developed to (i) guide in what ratio to combine strengths from RBS to promoters and (ii) detect from growth data when the RBS strength should be reduced. The produced model is an extension of previously developed 'growth law' models that partition the proteome of cells into different functional fractions. A library of constructs with different burden levels was built for collecting data and testing the model predictions.

Chapter 3 investigates transitionary cost of engineered constructs during changes in growth-phase. A model was developed to specify and quantify the burden during growth-phase transitions. After testing the model with experiments, two new metrics are proposed for quantifying and defining burden: (i) response time delay and (ii) the reduction in peak ribosome expression (measured as the second derivate of growth rate during lag-phase). In addition, the decline rate of the ribosome concentration was indicated as an additional parameter informing the level of burden.

Chapter 4 develops models and methods for how to reduce the burden in engineered biology with evolutionary methods. Specifically, it identifies criteria for enabling host or-

ganisms of engineered constructs to evolve to reduce costs, whilst not losing the construct during the process. The underlying model is developed by assuming a similar process to how multi-resistant 'superbugs' arise. Chapter 4 formalises and reconstructs this process whilst using it for obtaining longer lasting engineered constructs. Finally Chapter 5 concludes the thesis and discusses future work.

1. Chapter 1

Chapter 1 develops repeatable, robotic and automated experimental methods to test the impact of burden on engineered biology. The robotic experimental methods developed in Chapter 1 are built upon in Chapters 2 and 3 for the development and testing of novel burden models for Synthetic Biology. An automated and online accessible Synthetic Biology platform was built which enables the design and testing of biology circuits remotely from any location. The online platform provides elementary access, faster execution and automated data interpretation in an interactive format. It facilitates collaboration on synthetic biology projects and speeds up research progress with enhanced reliability. Among others, Dr. Andreas Harris, Dr. Thomas Folliard and Dr. Dhruva Raman contributed by testing the developed platform. An example of them applying the platform for genetic circuit design is shown at the end of the Chapter.

1.1 Introduction

1.1.1 Genetic Stability in the Design-Build-Test Cycle

The issues arising from genetic stability, or instability, are usually not considered upfront during the engineering process of biology. Instead, genetic instability is one of the factors that is examined once a design or implementation has already failed and an explanation is being sought. There are many problems that can arise during design implementation and testing in Synthetic Biology, many of which are likely not even known today. Genetic instability at least belongs to the group of 'known' potential problems. However, finding the correct reason (out of known reasons) for a failure of a circuit can still be a long process, and one without a clearly defined starting point.

An obvious improvement to the general process of designing, implementing, testing and then fine-tuning or troubleshooting would be to move as many as possible of the known potential design pitfalls to the first stage of the design-build-test cycle. In other words, an enhanced method would be to model and predict the likelihood of failure for as many design parameters as possible during the design phase, before building and testing. Designing is fast. Building, testing and in particular troubleshooting is expensive and slow. A good example of a design phase prediction tool is the EFM calculator [92] which identifies short regions within DNA sequences with elevated mutation rates. As highlighted in the introduction of this thesis, there are two sides to the coin of genetic stability of engineered biology: (i) the frequency of mutations within an engineered construct and (ii) the fitness costs that the constructs exercise on their host. An initial tool has already been provided for informing mutation frequency [92], whereas this thesis focuses more on the second genetic stability aspect which comprises fitness cost and burden. In cases when this approach is difficult or fails, a second useful approach is to automate the interpretation of results such that the correct reason for failure can be identified, *i.e.* prior to further experimental troubleshooting, which can be significantly slower as well as more expensive. Chapter 2 in this thesis provides a method for informing burden during the design phase and Chapter 3 develops a method for the automation of burden interpretation from experimental data.

Convincing designers of biology that it is crucial to consider the genetic stability or burden level of their constructs prior to implementation of designs could be a hard task, and depending on the circumstances it might not even be the best advice. Due to the complexity of engineering biology, as well as the large number of factors that "we don't know that we don't know" (especially when trying to push the boundaries of a field) means that close attention is needed for a multitude of engineering and design considerations. Many of these lack any models or prediction tools today, and instead rely on best guesses. Maybe more importantly, the models used to predict or design biology today can be at best used as a guide. Even the most frequently used tools, such as the RBS calculator [155], many times gives an incorrect prediction (see Chapter 2 of this thesis). In the light of this landscape, it can be challenging to fit further models and methods into the design process that needs simulation as well as learning from the user in order to provide any

useful design guidance. Engineers might have more pressing considerations to make, such as (i) what model to trust or not trust, (ii) whether the genetic sequence is at all feasible to make and if so how difficult will it be or (iii) how many variations of different dials (RBS, promoters, terminators etc) need to be made to secure an acceptable likelihood of design success during subsequent experiments (iv) how to design a feasible experiment etc.

Instead, in order to be most helpful, efforts should be narrowed on how to (i) produce easy to use tools that can guide the design phase or to (ii) automate the interpretation of data from experiments, such that additional aspects can be extracted. For the first point, it becomes more and more important that each tool produced is easy to use in practice, due to the multitude of ancillary engineering considerations that have to be taken into account when implementing biology successfully. This is because a large number of different tools are needed to cover each different aspect, and each of them needs time investment in order to use. In other words, in order to focus the engineering on the key specifications of the engineered circuit, the ancillary design aspects need to be abstracted through approachable tools. The genetic stability of a circuit is a prime example of an ancillary consideration, a consideration which generally does not emerge when designs are successful, but generates failures when designs are suboptimal. For the second point, making inroads to automation of experimental data interpretation would enable engineers to leverage their individual and specific skills further in order to build new functions, instead of going back to find and learn new ways for troubleshooting, thus enhancing the general understanding of designs as well as freeing up time for investigators to focus on their core investigation targets. This will allow for a higher level of skills specialisation of biology engineering. Due to the complexity of biological systems, a high degree of specialisation is strongly needed to enable faster progress. Automation of data interpretation would specifically: (i) reduce the time needed to find and interpret data patterns providing clues about design mechanics and malfunctions, (ii) reduce the time needed to learn how to extract data patterns informing design behaviours, (iii) provide insight from data for engineers that do not know upfront what exact data metrics to look for and (iv) allow for skills specialisation.

1.1.2 Automation of Experimental Data Interpretations

Enabling the automation of data analysis for understanding designs is not a trivial task. There are many aspects that can change between one experiment to the next, including lab equipment or personal preferences of individual investigators. Those are only a few examples of parameters which are difficult to account for upfront when producing data analysis and design interpretation tools. The differences between experimental datasets are not only issues for automated data interpretation, but they are also important issues in biological science, which make experiments in general difficult to repeat and confirm [13, 36], and further hinders investigators to use and build on each others' contributions.

In order to provide a solution to the problem of experimental variation and incompatibilities between datasets, one option is to provide a standardised framework for how experiments should be conducted. One example would be a standard framework that is precise and narrow enough such that it fits to a broad range of potential data interpretation tools for deciphering biological designs. However, there are also many obvious and well known problems with this approach of designing and imposing a particular standard. First of all, it is difficult to convince users to use a particular standard, and frequently rightly so.

Standards are often designed and proposed with an emphasis on reading and interpretation of data and are then imposed on the producer of data, by adding work or limits in production or display methods. The hypothetical standard discussed above is no exception to this rule; the key purpose is to facilitate and empower data interpretation not to liberate or facilitate data production. This fallacy arises because data production and subsequent reading of shared results is done by different agents, causing a conflict of interest. In order to enable adoption of any type of standard, the key point should be to make a procedure easier for those who are producing data, not simply more convenient for those who subsequently wish to use the data.

To target the core issues of ease-of-use and ease-of-adoption as highlighted above, a novel approach to data production has been built and is investigated in this Chapter. The intended outcome of ease-of-use and ease-of-adoption is realised through an abstraction

layer to the physical laboratory equipment and processes. Instead of handwritten laboratory protocols and manual processes the entire data production is accessed online through a browser interface (see Section 1.4: **Online Automation and Digital Synthetic Biology**). Further to enabling standards for data pattern interpretation of Synthetic Biology designs, one side benefit of this approach is that it opens up data production to more participants, by eliminating a number of key barriers to entry. The work presented here focuses on informing design fitness costs and burdens through automated data production, but the vision is that the format can be directly expanded to a wide range of design enhancing tools. Thus, the system has the power to serve as a platform between methods/tools producers and their users in the future of Synthetic Biology. Meanwhile, biology has the potential to become even more interconnected to the digital world, primarily because of the benefits that digitalisation can lend to repeatability, sharing and collaboration.

1.2 Experiment Automation

All experiments described in the different chapters of this thesis are performed using a robotic platform. The traditional methods for measuring burden rely on the ability to accurately record subtle differences in growth. Recording subtle differences means that data collection becomes even more sensitive to small changes in experimental procedures. The experimental procedures for recording small variations over repeated experiments require a high degree of robustness and repeatability. Robots repeating a set of computer-written instructions are regarded as providing less variation than humans who are reading a set of instructions written on a piece of paper. To collect sufficiently long time periods of data samples these experiments are often long and repetitive. An experiment implemented with the help of the robotic platform allows for longer experiments with regular experimental operations, something that a human operations can not do. The experimental methods shown in this Chapter use robots to provide multiple redilutions (sub-culturing events) of samples at regular intervals. A key benefit of the robotic continuous sub-culturing is that it provides the foundation to test engineered constructs for prolonged periods of time at a high throughput. For example, a chemostat or turbidostat provides the ability to test

constructs for longer time periods, but it does so at a very low throughput. In the future of engineered biology, control of signals over time is set to become more important, in the same way as time is a crucial parameter for electrical switches in order to build functional electrical circuits. The robotic sub-culturing experiment developed here enables the design and testing of circuits with changing and fine tuned dynamics over prolonged time series. As listed above, repeatability and robustness in execution over long periods of time are the key benefits of using a robotic implementation.

However, there is also a broader benefit with the particular method selected for how the robots were used during the experiments. Instead of using disconnected robots in an isolated location, it was decided to use robots that can be made accessible to others via the internet. This provides easier access to others to repeat (or improve upon) the methods developed in this thesis by harnessing 'cloud' equipment accessible from any location in the world. One key difficulty for the reliable repetition of experiments is that physical conditions as well as the type of equipments available are different between each independent laboratory [96]. Hardware and equipment for experiments are in general very costly, which means that a laboratory with one type of equipment is unlikely to purchase a different, but similar, machine simply for repeating the exact experiments described from a previous group working with a different set of equipment. By making the experiments in this project accessible in the cloud, with the exact same equipment and conditions, repeatability by others is substantially enhanced. The enhanced repeatability is further expected to facilitate subsequent rounds of improvements, and therefore also the speed of progress in the research field. There is one more important benefit with the experimental setup implemented and demonstrated in this Chapter. By enabling the robotic experiments to be implemented in the cloud, not only is repeatability made easier for those with other laboratories, but elementary access is also provided to those without the same laboratory equipment. It becomes possible for those without any laboratory at all to directly repeat and build upon the research and laboratory results presented in this thesis. Further, it becomes accessible to those that have neither the laboratory nor the skilled manpower to operate the laboratory to re-run and improve upon the models (by producing complementary data) as well as modifying the experiments. Thus, the approach opens up the field

to researchers from other backgrounds. Specifically, it opens up Synthetic Biology data collection and design comparisons to researchers from a wider spectrum of quantitative backgrounds, groups that frequently do not have direct access to the laboratory equipment that is being used. It is the aim that this specific cloud approach will benefit the larger community of Synthetic Biology, by enabling repeatability while also providing access to intelligent investigators from adjacent disciplines.

In summary there are three specific experimental issues identified and addressed with the help of the current setup: (i) long experimental procedures, (ii) differences between experimental implementation due to humans reading a set instructions and (iii) differences between hardware equipment at different experimental locations.

1.2.1 Experiment Automation in Practice - Robotic Implementation

For coding experiments and using cloud accessed robots there are currently fewer than a handful of cloud-accessible 'robot farms' available, and so far they are exclusively located in the US. After an evaluation of the (few) different options it was decided to use the robots from Transcriptic Inc, in Menlo Park, CA. Transcriptic provides simple prewritten experimental protocols that can be accessed by any user. But they also provide the option for developers to directly code custom implementations by the robots through an API. All the protocols used in this thesis are custom-coded experiments run via this API. This thesis also became the very first user of the Transcriptic API interface for coding experiments remotely. The protocols are coded via a Python library. The Python files then produce (significantly more verbose) JSON instructions which are used to run the robotic implementation of the experiments. JSON stands for JavaScript Object Notation. It is an open-source language-independent data interchange format, and it was initially derived from JavaScript [47].

Figure 1.1 shows a conceptual layout of the code design for the experiment implementation. A first set of code is needed to successfully carry out transformations of bacteria in order to insert different DNA constructs. Afterwards a second set of code runs the experiments to produce data. The two code-sets are combined into a single program to produce a seamless order of instructions. If the code was divided into two separate units for imple-

mentations, that would enable reuse of previously transformed colonies. However, the time period between a transformation event and subsequent data collection has the potential to introduce unknown effects on the bacteria or on the stability of the construct. Therefore the two unit operations are combined into a single execution without applying reuse of previously stored colonies. The success of a transformation is assessed by two subsequent observable events. Firstly, the DNA needs to successfully enter the competent cells during transformation and, after spreading, visible colonies need to grow on the solid media plates. Secondly, colonies which have been picked from the solid plates need to grow in the liquid growth medium. It is not certain that picking of colonies from solid plates will produce growth in liquid medium. For example, with certain antibiotics (ampicillin included) there is the risk of picking so called 'satellite colonies' which are near the true transformed colony, but are instead colonies of non-transformed bacteria (this risk of 'satellite colony' picking is increased with longer incubation times). Or the colony picking robot can sometimes be poorly calibrated, such that it occasionally fails to pick and transfer bacteria from a 'true' colony into the liquid growth medium. If no colonies grow on the solid plates, and / or no liquid culture grows from the supposedly picked colonies then the transformation event has failed, and the whole experiment needs to be repeated. In order to reduce the risk of transformation failure, a surplus number of colonies (from all of the construct, positive and negative controls) are initially picked and cultivated by the automated experimental protocol. If a minimum of three liquid cultures grow for each of the construct, negative and positive control then it is regarded as an acceptable transformation and picking event sequence (*i.e.* for further data assessment of the experiment).

The code implementation does not allow for any 'if clauses' or conditional statements. This means that building a code for robust transformations (*i.e.* that works for a wide range of constructs) was more time consuming than what a manual process would require. In a manual process a wide range of conditional tests can be made to achieve transformed colonies for untested constructs. The range of conditional parameters include volumes, concentrations and times. As an illustrating example, during a manual transformation colony growth can be examined at a range of time points. When a sufficient number of colonies have grown, then colonies can be picked. However, the computer-coded and

machine-implemented protocol that is currently available does not provide the agility of conditional statements (*e.g.* to observe colony growth and then pick at an adequate time point). Instead, every parameter needs to specified up-front, *e.g.* the picking time of colonies for different constructs, even though some constructs will produce visible colonies earlier than others. Therefore, the hardcoded protocol input instructions need to be sufficiently robust for the protocol to succeed on a wide range of constructs. For this purpose, a wide range of transformation parameters were tested with different constructs in order to achieve a high level of protocol robustness across the various constructs with their different transformation properties. In summary, the limitations of non-conditionality means that more efforts are needed in order to design protocols with a robust performance to a wide range of constructs, particularly regarding the transformation event. At the same time, once such protocols have been developed, the benefit is that the reuse of identical protocols for different data samples can provide enhanced abilities for the comparison of historical design data (across data samples of different constructs, collected at different time points, but with an identical protocol).

Figure 1.2 shows an expansion of a small set of the code shown in the conceptual view of Figure 1.1. In Figure 1.2 the individual lines of code are linked to the subsets of the robotic unit operations that they instruct.

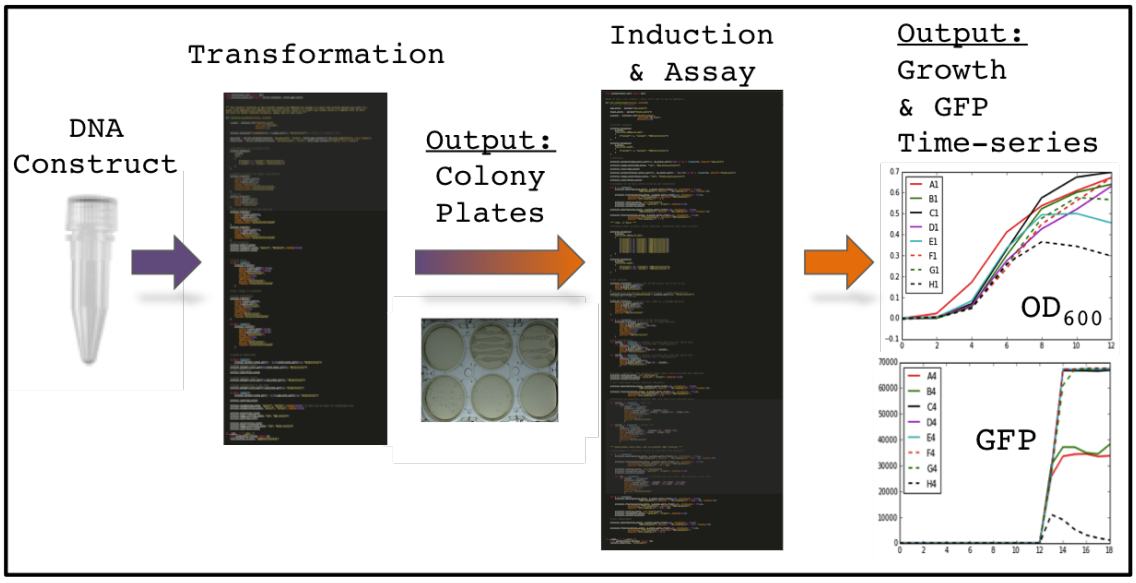


Figure 1.1: A schematic layout of the basic laboratory protocols written in Python. The initial part of the protocol takes DNA and bacteria as inputs and transforms *E. coli* DH5α with the given DNA. The output of the initial part of the protocol are transformed colonies on solid media. The second part of the code takes the colonies as inputs and incolulates them into liquid medium. The second part of the protocol contains the code for running the assays of the experiment.

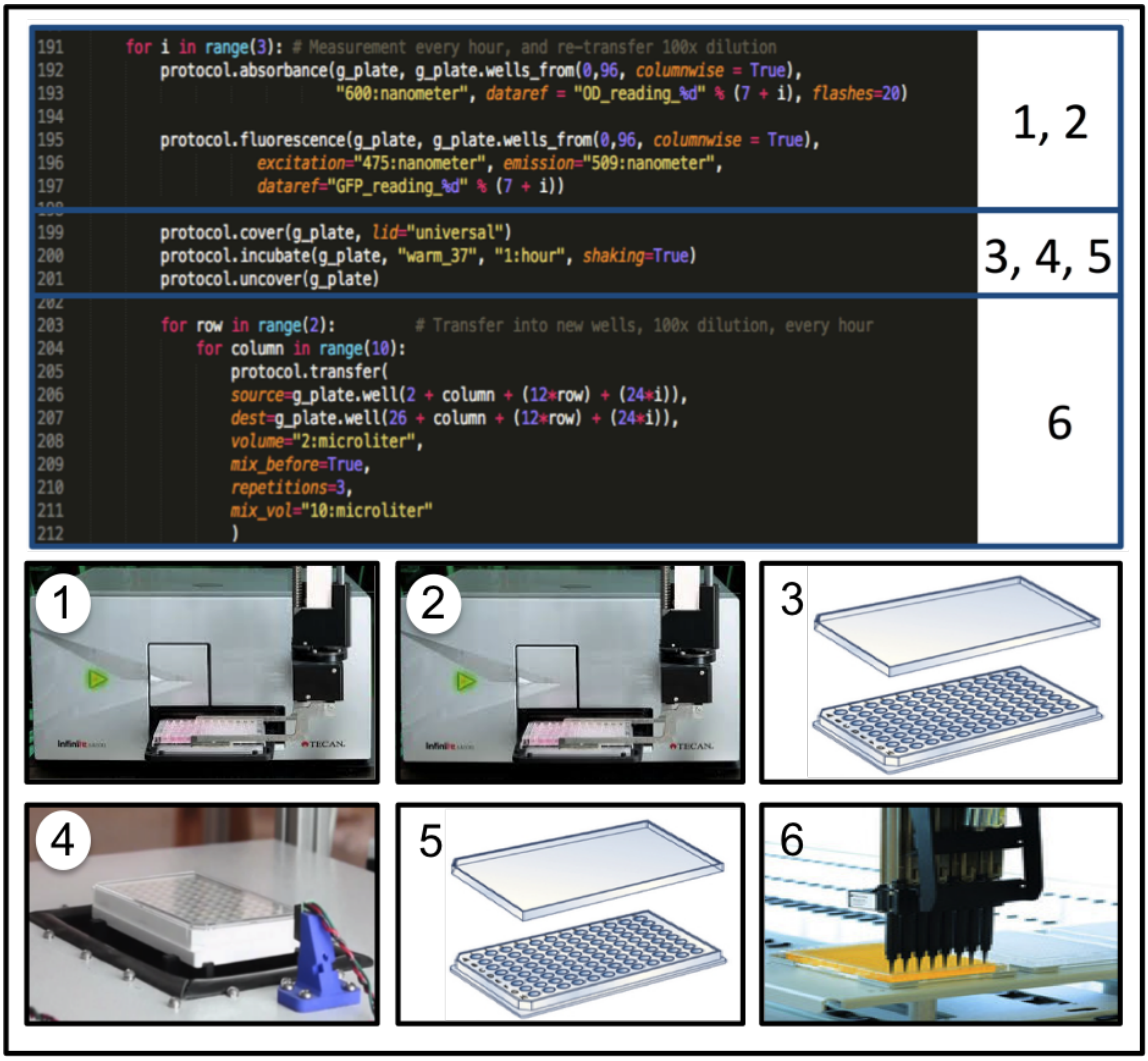


Figure 1.2: A small section of code for the experiments, highlighting through illustrations how different lines of code correspond to physical laboratory operations. The small code section shown is taken from the second part of the protocol shown in Figure 1.1. **1:** Absorbance readings to record population densities. **2:** Fluorescence readings to record expression of non-native proteins or constructs. **3:** Seal cultures with lid. **4:** Place cultures in shaking incubator (at 37°C for one hour). **5:** Remove lid from cultures. **6:** Diluting cultures by inoculating into new wells that contain fresh media.

1.3 Experimental Methods and Design Conditions

1.3.1 For all the experiments

Flat 96-wellplates were used for all growth and assay experiments. A temperature of 37°C was used for all culture growth incubations. All vectors and constructs are transformed into *E. coli* $DH5\alpha$ competent cells. Leuria-Bertani (LB) medium supplemented with 100ng/mL Ampicilin (LB-amp) was used as growth media for all cultures. Super Optimal broth with Catabolite repression (SOC) medium was used for recovery from vector transformation (during ten minutes before plating to solid media). For the fluorescence measurements of the expression of Green Fluorescent Protein (GFP): an excitation wavelength of 475nm was applied and an emission wavelength of 509nm was recorded. (The expression of GFP has been used as a circuit output due to the non-invasive method of light emission, which provides an accurate yet simple quantification of protein expression over longer time periods without the need for culture growth interruptions.) A light wavelength of 600nm was used for turbidity measurements for cell density recordings (OD_{600}). Orbital shaking with an amplitude of 3mm was used for shaking incubation growth in liquid media. This is referred to in the text as 'shaking incubation'.

1.3.2 For all plasmid transformations

Transformations prior to experiments were carried out in the following order: (i) adding competent cells to a 96-wellplate, (ii) adding DNA to the competent cells, (iii) cold incubation for 20 minutes at 4°C, (iv) addition of 50μL of room temperature SOC medium, (v) warm incubation for 10 minutes at 37°C followed by (vi) spreading of 50μL of transformed cells onto solid LB media plates with 100ng/mL of Ampicilin. After spreading, the solid media plates are incubated for 16 hours before robotic picking of transformed colonies into liquid media (LB supplemented with 100ng/mL of Ampicilin).

- Before the addition of competent cells to the 96-wellplates, the 96-wellplates are cooled at -20°C for 10 minutes.
- All plasmids except the empty control vector were transformed with a concentration

of 10ng/mL. The empty control vector was transformed with a concentration of 1ng/mL.

- For all plasmids except the empty control vector, 2μL of DNA (the concentration is given above) was added to 50μL of $DH5\alpha$ competent cells (Zymo Research). 1μL of empty control vector was added to 50μL of $DH5\alpha$ competent cells (Zymo Research).

1.3.3 For all experiments for burden predictions and model developments (Chapters 2 and 3)

- When picking transformed colonies, the cells are inoculated into 240μL of growth media (ENV_1) where they are shaking incubated at 37°C for between 15 - 26 hours with a fluorescence and turbidity reading every second hour.

- After transferring of cells via inoculation to an ENV_2 the cells are grown with shaking at 37°C with fluorescence and cell density recorded every 6 minutes.

- All constructs are expressed via constitutive expression and no inducer is used.

The annotation ENV_1 refers to the liquid growth environment immediately following picking of colonies from solid media (*i.e.* LB-plates with an antibiotic and the transformed colonies). The annotation ENV_2 refers to the growth environment after sub-culturing from ENV_1. When picking colonies from solid plates, there is a large variation in the number of cells which are transferred from each colony into liquid growth media (*i.e.* the starting population of ENV_1). This means that samples in ENV_1 will reach a detectable optical density at a wide range of time points. By the end of growth in ENV_1, all samples with successfully picked colonies will have reached detectable optical cell densities. There will also be less variation in cell concentration between each sample, as the samples which commence growth with a higher initial cell concentration from colonies will be limited in growth by the capacity of the growth medium. Thus, there is less variation in the initial cell density at the start of growth in ENV_2, compared to the initial cell concentration of ENV_1.

1.3.4 Experiments for web-interface interaction for engineering biology (Chapter 1, Section 1.4)

- When picking transformed colonies, the cells are inoculated into 240μL of growth media (ENV_1) where they are shaking incubated at 37°C for 15 hours with a fluorescence and turbidity reading every second hour.

- Seven colonies are each picked from the custom engineered construct, the empty control vectors and a positive control constitutive GFP expression vector (ProC-B0034).

- The assay experiments (*i.e.* which are displayed and accessed online) start when cells are transferred from ENV_1 into ENV_2 by inoculation of 15μL of cells into 230μL of growth medium.

- Each sample well of constructs in ENV_1 is duplicated into two different wells in ENV_2.

- In ENV_2 the cells are grown with shaking at 37°C with fluorescence and cell density recorded every 6 minutes.

- After three hours of growth, 30μL of cells are inoculated from each well in ENV_2 into 210μL of pre-warmed LB-amp supplemented with 133ng/mL of Anhydrotetracycline (aTc). This environment is ENV_3, turbidity and fluorescence are continuously recorded every 6 minutes. aTc is supplemented in ENV_3 to enable the induction of TetR repressed genes (if also supplying a vector that expresses the TetR repressor).

- After three hours of growth in ENV_3, 30μL of cells are again inoculated into new wells with 210μL of pre-warmed LB-amp but without aTc. This procedure of transferring 30μL cells for inoculation into 210μL of pre-warmed LB-amp but with out aTc is then repeated every three hours to make environments ENV_4, ENV_5 and ENV_6.

- The results from environments ENV_2 to ENV_6 inclusive are accessible to view online in order to interpret the function output of any given engineered circuit. This provides a time series for each sample over 15 hours and is split up into 150 individual

measurement points for each sample of both cell density and fluorescence readings, separated in time by six minutes between each reading.

The dilution ratios between the different environments were determined as specified above in order to provide: (i) a large enough initial cell density at each recorded environment ($ENV_2 \rightarrow ENV_6$) for cell densities to be quantified, and (ii) a fairly consistent growth phase throughout the different sub-culturing environments. The purpose of a low growth phase variation is to (i) avoid samples exhausting the growth medium and reaching stationary or death-phases, and (ii) to better enable the comparisons of cell behaviours in the earlier ENVs to those of the latter ENVs, as well as provide a more aligned behaviour between earlier to latter time point readings within each ENV. The culture time of each ENV, together with the dilution fold between the ENVs, both determine these growth behaviours. At the end of the very long incubation period in ENV_1, the samples reach a cell concentration a bit more than double the concentration at the end of each of the latter recorded ENVs. Therefore, the first dilution into ENV_2 is also around double that of the subsequent dilution folds ($15\mu L \rightarrow 230\mu L$ versus $30\mu L \rightarrow 210\mu L$). The dilution ratios and time spans between sub-cultures (*i.e.* 3 hours) were obtained empirically, through trial-and-error with an initial subset of different construct. Figure 1.7 shows the optical density readings of three different constructs during growth in $ENV_2 \rightarrow ENV_6$.

1.4 Online Automation and Digital Synthetic Biology

One of the aims of this work is to develop a novel user interface that can be used to produce Synthetic Biology data directly online. The overall project is called Digital Synthetic Biology.

The concepts, code and experimental procedures discussed in Sections 1.2 and 1.3 are useful to investigators focusing on genetic stability or burden. It is useful because, in a very simple way, it allows for repetition of experiments and improvements or alterations of the underlying data for the models. However, for the general users of synthetic biology, *i.e.* those that do not want to improve models or understanding of burden but instead want useful prediction, the method shown above is not enough. As outlined in the introduction,

one of the current issues of genetic stability and burden is that they are generally not addressed until a failure has occurred and they become part of the long list of possible issues to try to troubleshoot. In other words, methods described so far allow for data collection, but that does not necessarily stipulate they provide, facilitate nor enhance design predictions.

In order to enable the prediction and assessment of burden directly from engineering designs of biology, a different interface layer is required. The user layer presented in this Chapter provides users of Synthetic Biology with an intuitive online interface for designing and testing of biological constructs through a web browser. This enables direct automation and incorporation of data processing, which makes it a key benefit for informing burden, as well as other important ancillary data patterns.

All engineering disciplines, due to their inherent complexity, own their success to the functional construction of abstraction layers. Synthetic Biology is not excluded from this premise. Abstraction layers and hierarchies are human inventions designed to assist people in engineering complex systems by ignoring unnecessary details. A schematic outline of the abstraction layer achieved by the Digital Synthetic Biology platform is provided in Figure 1.3.

The Digital Synthetic Biology platform has many clear advantages compared to previous manual methods, however, there are also a few drawbacks with our system. For example, it provides the user with less input choices in terms of protocol setup: *e.g.* if a user wishes to provide an extra supplement into the medium. It is also a clear limitation if the user wants to use customised techniques, less clearly defined in the published literature, shaking the tubes in a particular way or other unspecified laboratory techniques described as 'green hands'. However, the compensating advantages are that it becomes much easier to compare data from experiments when using identical protocols, and facilitated data comparison makes it easier to understand each other's results. By helping users to compare designs and outputs the field will be able to progress much faster.

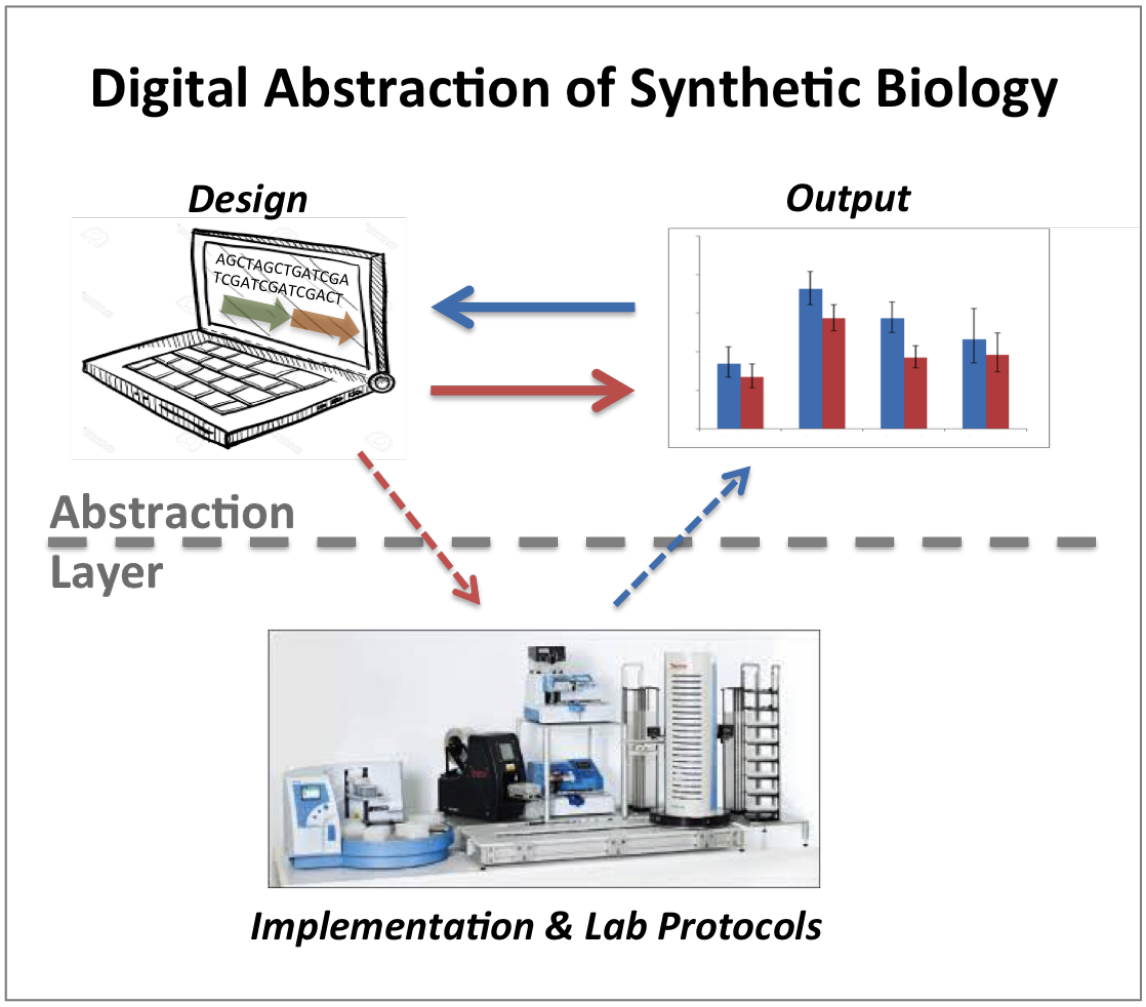


Figure 1.3: A schematic outline of the design abstraction layer implemented with the Digital Synthetic Biology interface. The entire experimental execution is removed and placed below an abstraction layer. The details and procedures for the data production follows a standardised procedure and is abstracted. The coder of biological constructs can now focus on the cycle between design and output.

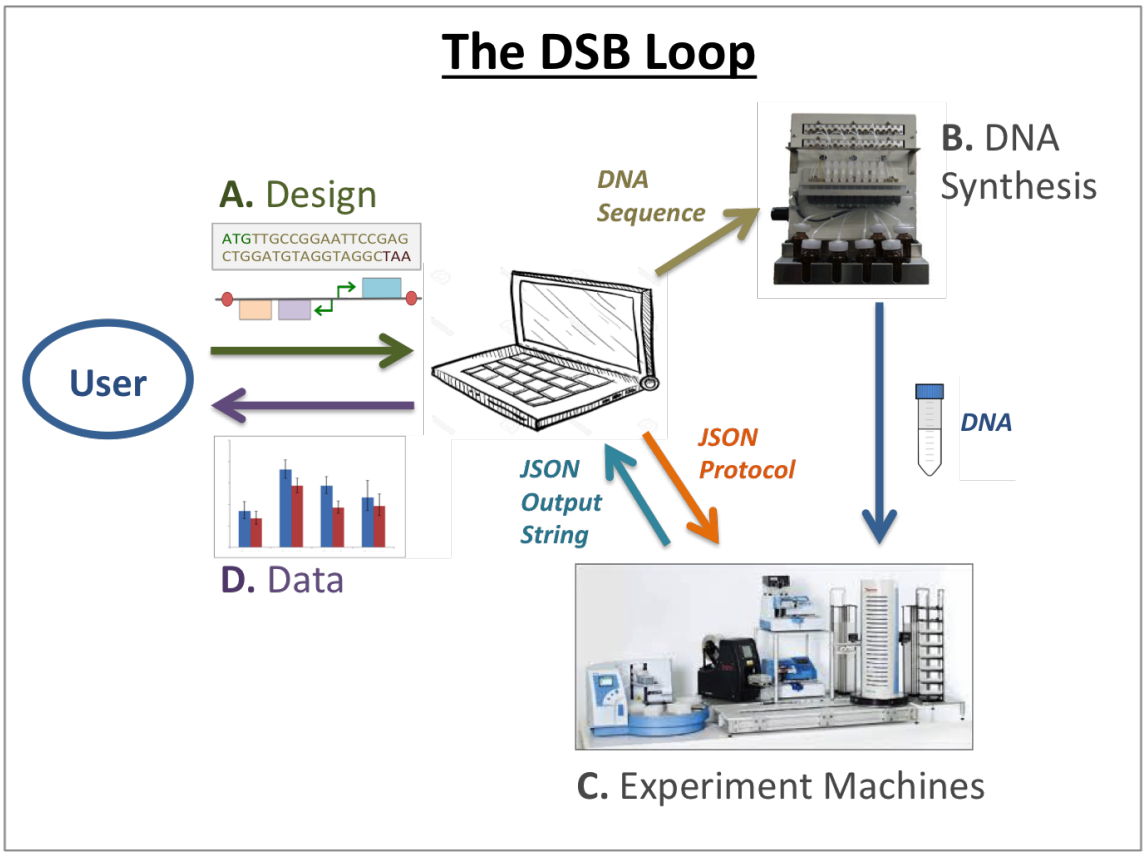


Figure 1.4: The sequence of steps and interactions for realising the Digital Synthetic Biology platform. The sequence of events and interactions starts with: (**A**) the user submits an annotated design and DNA code of a construct. The DNA is then synthesised (**B**), a predefined and computer coded experimental sequence is executed (**C**), and a JSON string of outputs is obtained (the current setup uses JSON strings, but other experimental procedures or machines could likely use alternative variants of data formats). Finally, a data output is returned to the user (**D**) which can be interacted with for future designs and iterations (implemented on the DSB platform in Javascript and Python). Current (pre-existing) cloud-lab systems allow to execute experimental via computer coded protocols. They allow to robotically transfer samples between different machines commonly used in experimental laboratories. For example, one can execute a sequence of experimental steps, as shown in Figure 1.2, while recording the machine outputs from those standard experimental machines which have been employed by the JSON encoded protocol. The two preexisting external resources which the DSB platform leverages are: (i) **B** -*DNA synthesis* and (ii) **C** -*Laboratory equipment and maintenance*. Parts of the appearances of the design input by the user and the data output to the users on the DSB platform can be seen in Figures 1.11 and 1.12, respectively.

1.4.1 Laboratory Automation Cloud Software Tools for Synthetic Biology

It has previously been argued that low-level programming languages used to steer robots (*e.g.* Autoprotocol) are not user friendly interfaces for conducting experiments [20]. In order to address this, wrapper tools have been developed to provide improved usability. One such tool is the Wet Lab Accelerator (WLB) [20] which is developed by Autodesk. The motivation of the WLB is to allow scientists without any programming knowledge to conduct biology with robotic control [20]. This is enabled via a cloud-based drag-and-drop interface for specifying laboratory unit operations and generating experimental protocols with instructions in Autoprotocol (via Python code libraries). The use of WLB was proven by showing that it could generate a protocol which was able to successfully transform *E. coli* cells with a standard pUC19 plasmid. The new abstraction layer developed by WLB is called the 'Omniprotocol' and while it is only tested for generating Autoprotocol code with a graphical interface it is intended that the method can be extended to any robotic cloud API code in the future and therefore be a generic method for generating experimental protocols. Thus, it is hoped that WLB can democratise biology by bringing automation to non-programming biologists [20].

A similar tool to the WLB is the BioBlocks software [80] which provides a visual interface for designing experiments. The experiments are designed at a high level with the help of visual blocks divided between 'container blocks', 'operation blocks' and 'organisation blocks', which specify experimental containers, experimental actions and sequence of operations, respectively. The BioBlocks software produces low-level machine-readable Autoprotocol code from the user inputs in the form of interconnected drag-and-drop blocks. Two additional software outputs from the user inputs are an English readable description of the experiment as well as a visual representation of the task. Roboliq [197] and List-of-Liquids [165] are two additional examples of recent similar tools, which enable the production of low-level experimental robotic implementation protocols from higher level experiment designs.

A distinction between the mentioned tools and the DSB platform can be made by the

different user inputs and software outputs from the two classes of tools. User inputs in the listed tools [20, 80, 165, 197] consist of high-level and visual representations of experimental procedures. User inputs on the DSB platform consist of DNA code, genetic design and construct annotations. Outputs from the listed tools consist of low-level and machine-readable experimental protocols. Outputs of the DSB platform consist of experimental data. These general differences regarding inputs and outputs for the two classes of tools are illustrated in Figure 1.5.

There is also a large number of tools available for designing DNA sequences and simulating potential experimental outputs [130, 189], or alternatively, computing and predicting a DNA sequence for realising a given desired circuit output [131]. A good overview of this type of computational and simulated design tools can be found in Appleton *et al.* [3]. A difference between these types of softwares and the DSB tool is that the latter is not able to provide computational predictions for design, instead it produces real experimental outputs. The real experimental outputs can then be used as guidance to future design implementations; either for completely novel designs (that might want to reuse a sub-component of previous systems), or for smaller changes to pre-existing designs via so called 'forks'.

1.4.2 The Nuts and Bolts of the Digital Synthetic Biology Platform

The online platform is currently hosted on Google AppEngine and is written in Python (*www.biology-circuit-builder.appspot.com*). All the interactive components are written in JavaScript and the interactive data visualisation parts are made with the D3 library. When a design submission is received the DNA code is sent to Genscript Inc for synthesis. The DNA is cloned into a previously designed vector backbone on site and then sent by mail to the robotic farm in Menlo Park. Transformation of bacteria starts upon arrival, according to the Python-encoded protocol developed and described in Section 1.3. The data producing code follows directly after transformation operations, as described in Section 1.2. The data is fed back to the database of the Digital Synthetic Biology Platform in a JSON format. The JSON format is parsed and interpreted by a prewritten script, such that each submitted design is linked to the correct data and is viewable in a user-friendly format.

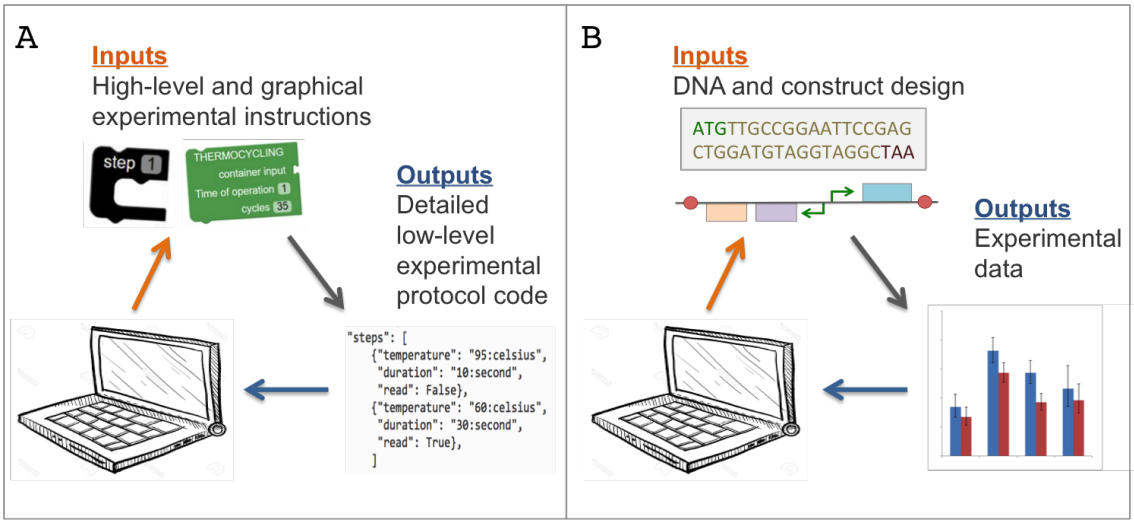


Figure 1.5: An illustrated comparisons between cloud software tools for laboratory automation. The figure highlights a distinction between (A) current available tools for cloud implementation of Synthetic Biology and (B) the presented DSB method. **A:** Users inputs consist of high level code and graphical blocks in an online browser environment (the blocks shown are example inputs in BioBlocks [80]). From these user inputs the software tools generate low-level experimental JSON code. The returned low level experimental code can be used to run experiments with robots. **B:** The user inputs consist of DNA sequences and annotated designs of biological systems in an online browser environment. The outputs consist of experimental results for the provided circuit design.

When the user submits small edits to their previous constructs, named 'forks', these are not synthesised *de novo*. Instead the edit is sent back to the DNA synthesis provider, where a copy of the original construct is stored, and the edits are made directly into the previous design before it is shipped again. This procedure of small edits is considerably faster than *de novo* synthesis and the turnaround time from design to viewing data is shortened.

The user path of the Digital Synthetic Biology platform is:

1. Submit a DNA sequence for a Synthetic Biology construct.
2. Annotate the DNA sequence with features (*i.e.* promoters, RBS, terminators etc).
3. Save drafts for later edits.
4. Submit designs for producing experimental data.
5. Get feedback on the behaviour of the construct by viewing the experimental data.

6. Make edits to the constructs and resubmit for data production (these edits are named 'forks' of the original construct).

Each user initiates the design session by logging into a personal account containing personal construct designs or drafts from previous sessions. In Figure 1.6 the user path on the platform is illustrated and supplemented with images from the online interface. Figure 1.7 provides an expanded view of the data view step. The top part of the screen shows the construct design and features together with a short description. Clicking on the 'Construct' tag in the middle provides the DNA sequence together with descriptions of all the features. Highlighting any part of the DNA sequence enables edits (forks) and immediate resubmissions for making new data. Clicking on the 'Detailed Data' tag shows detailed data for each of the different samples. The data shown in the screen in Figure 1.7 is the mean of all the values that can be seen under the 'Detailed Data' tag. The user gets back to the current screen by pressing the 'Data and Score' tab.

The DSB platform as shown here and in the following example has so far been used by users between two UK universities. Future work will aim to (i) extend and expand its use to larger groups of users, (ii) enhance to data collection capabilities for the experiments, (iii) reduce the turn-over velocity of each design-build-test cycle, and (iv) expand its applicability to more organisms and biological engineering with other purposes than circuit designs for inert protein expression (*i.e.* more than GFP expression circuit outputs).

1.4.3 Examples and Results of the Digital Synthetic Biology Platform

In this section, an example is provided to show some of the benefits of using the Digital Synthetic Biology (DSB) platform in a practical setting. The example shown here was done by outside users. The biology that they designed was not part of the thesis work. The work contributed in this Chapter is limited to the development of the underlying DSB platform. In addition, the example aims to highlight the work sequence for engineering biology on the DSB platform. The work sequence starts with an initial circuit design specification and concludes with the previously mentioned 'fork' iteration. In the example shown, a key issue encountered and addressed by the users was a 'burden' obstacle. However, the initial burden problem also makes the current example into a good and interesting user

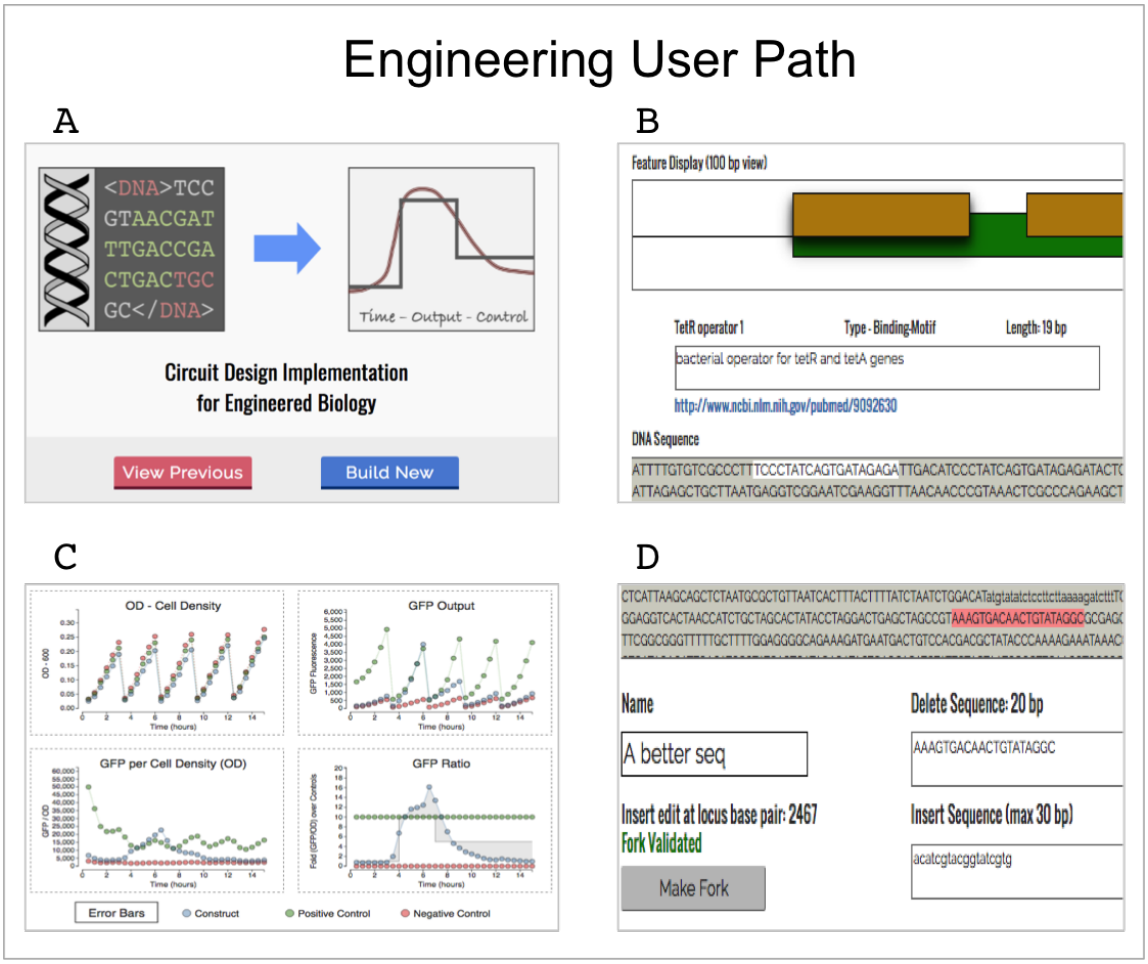


Figure 1.6: Engineering userpath for the Digital Synthetic Biology platform. **A:** Login to user account. **B:** Coding and annotation of DNA constructs. **C:** View experimental data of the construct. **D:** Make edits and resubmit construct forks for producing new experimental data.



Figure 1.7: An expanded view of the data view section shown in Figure 1.6C. The top part of the screen shows the submitted design and a short description. The four plots in the bottom part of the screen show the data produced by the design.

interaction to move earlier in the design cycle. Therefore, this user design was selected and is highlighted below. The contributors who collaborated to build the highlighted design on the DSB platform are Dr. Andreas Harris, Dr. Thomas Folliard and Dr. Dhruva Raman.

Design Target

The target design (output trace) that the engineers aimed to build a circuit for on the DSB platform was a step function followed by a (protein) signal output equal to half of an initial output spike. The target function can be seen in Figure 1.8 (the grey solid line shows the target output). The aim was to engineer a circuit that could respond to an input signal by (i) increasing the level of a protein in cells ten-fold and afterwards (ii) stabilising the expression indefinitely at a five-fold concentration level compared to the initial level (*i.e.* half of the initial concentration spike). Not only the ratios of expressions were sought, but the exact protein levels were also defined. The aim was for the maximum protein to be equal to the protein level of a constitutively expressed reference standard from a known promoter and RBS pair (ProC-B0034).

Specifically, the aim was to (i) provide a 10% protein level (of the biological reference) for the first 4 hours, followed by (ii) a 100% protein level for the next 3 hours, and (iii) ending with a 50% protein level for the last 7 hours. Functions of this shape are hard to build. Firstly, the function requires three different output levels from a single input signal. Secondly, one of the outputs is a constant level (*i.e.* the last 7 hours), at a different level from the initial output and long after the signal has been removed. This last behaviour requires some feature with memory functionality built into the circuit. In the user example provided, the memory component of the circuit is realised through an integrase which flips a DNA sequence containing promoters of different types.

Circuit Design

Figure 1.9 shows the circuit design that the users invented in order to achieve the function output. The initial state of the circuit is expressing the TetR repressor from a constitutive promoter. TetR is repressing two other promoters: one for the GFP output and one for an integrase (Bxb1 [206, 99]). If an aTc input signal is given to the circuit the repression

of the two promoters will be removed. The circuit will express GFP and the integrase simultaneously. However, two target sites for the integrase are framing the TetR promoter (constitutive) the GFP promoter (P_{tet}), such that the integrase reverses their orientation by flipping the DNA. The functionality of the circuit design can be seen in Figure 1.9A. After the Bxb1 integrase [206, 99] has flipped the DNA and reversed the direction of the promoters then the GFP expression is no longer repressed by TetR, but GFP is instead constitutively expressed from the second promoter.

A short background summary of TetR repression is provided here. In the absence of aTc in the cell the TetR repressor binds to the operators (O_{tet}) of the repressed promoter (P_{tet}). When aTc enters the cell it binds to ligand-binding domain of TetR which causes release from O_{tet}. This relieves repression and permits expression from P_{tet} [148, 1].

Design Issues and Fork Iteration

Figure 1.10 shows the results from the user design in Figure 1.8. The results are shown as a screenshot from the DSB platform. The GFP output from the circuit is noisy, and is still far from the target function. However, it can be seen from cell density readings (top left corner in Figure 1.10), that the current circuit design has a strong negative effect on growth. The data shown are produced via the automated protocol for the web-interface (described in Section 1.3.4). The experimental protocol collects data every 6 minutes. But the web-view in Figure 1.10 shows only every fifth of the total number of readings, and also it is limited to the mean (and standard deviation). The exact numeric mean value and standard deviations of each time point reading are displayed by hovering over the data point on the web-interface. The display shown in the figure is a snapshot of the data and results, while also highlighting the inter-relationships between the different readings of cell densities and function outputs. Each corresponding data point within the four subplots 'light-up' and is expanded when hovering the corresponding value in each of the other subplots. The readings of each individual sample at each 6 minute time point are shown under the header 'Detailed Data' (upper right corner of Figure 1.10, between 'Construct' and 'Comments'). The lower right corner of Figure 1.10 shows the 'GFP Ratio' for the samples together with a score of how well this output corresponds to given target

function. The gray area in the subplot (lower right corner of Figure 1.10) shows the target function. The score is calculated from a least squares fit between the actual data and the target function, with the resulting value to the power of minus one (such that higher scores represent a better fit, and lower scores mean a worse fit).

The 'GFP ratio' itself is calculated from the data by relative measures of GFP/OD of the construct to the same measures for the negative and positive controls. The negative control without any construct has the relative measure of 0. The constitutive expression of GFP from the positive control (ProC-B0034) has a value of ten. For example, if at a reading for time point = 4 hours, the OD readings are 0.3, 0.25 and 0.25 for the negative, positive control and construct, respectively, and the fluorescence readings are 600, 2 500 and 10 500 for the same samples, then the GFP/OD values are 2 000, 10 000 and 42 000, respectively (note, not true GFP measures, but fluorescence readings for the wave length of GFP; the negative control does not express any GFP). The GFP ratio for the construct at that time point is then $10 \times (10000 - 2000)/(42000 - 2000) = 2$. A blue dot (representing the construct) with 'GFP Ratio' = 2 would be seen at time = 4 hours on the bottom right subplot within the snapshot 'Data & Score' heading (shown in Figure 1.10). Meanwhile, the red (negative control) value is located at 'GFP Ratio' = 0, as $10 \times (2000 - 2000)/(42000 - 2000) = 0$, and the green (positive control) value is located at 'GFP Ratio' = 10, as $10 \times (42000 - 2000)/(42000 - 2000) = 10$.

The score is calculated from the readings shown in the snapshot view, *i.e.* the mean readings at time intervals of every 30 minutes. The first value included in each ENV time interval is the reading at 30 minutes, and the last reading of each ENV interval is at 3 hours, *i.e.* the last reading before dilution to a new ENV. This means that first readings of each ENV, at time points 6, 12, 18 and 24 minutes are not included in the score calculation shown. These excluded time points are also reading with largest technical noise, due to the less reliable readings at lowest optical densities from each ENV.

The users made edits of the design to achieve a better performance of the circuit and reduce the level of burden. The method they used to make edits on the platform is referred to as 'forks' and it is shown in Figure 1.11. This time the very same users who made the circuit also made the design iterations on the platform, but in practice any user can make

edits on any previous design.

In order to reduce burden and improve the circuit performance the engineers targeted one of the circuit RBS and reduced its rate of translation initiation. The user input consisted of retyping the sequence for the RBS of the TetR protein as a fork of the original design. The fork input enables any DNA sequence deletion in the construct, and / or DNA insertions of maximum 30 bp (as well as both deletion and insertion at the same time, as long as it is at a single locus). The decision to focus on this specific target locus by the users was likely not informed by previously existing models. But the fork redesign decision made by the users in this instance was more likely as a result of the users' skills and experience from working with similar biological circuits. A screenshot of the fork results is shown in Figure 1.12. It is seen that the burden has been alleviated, growth has been improved and the circuit function is considerably closer to the desired target.

This example of the DSB platform demonstrates two things. Firstly, the performance of Synthetic Biology systems can be substantially improved by correctly identifying and addressing design burden issues. Secondly, engineers of biology can successfully design complex circuits and interesting circuits with refined specifications using an abstraction layer of the laboratory components and data producing processes. The described DSB abstraction layer allows for a larger proportion of an engineer's time to be invested in circuit design, as opposed to implementation. It allows for specialisation. It also allows for sharing of data and collaboration — because it becomes easier to repeat and trust, as well as to contribute on each other's designs.

Finally, and to emphasise, the DSB platform does not generate the designs in place of the users. The users are still designing the constructs. But it provides design annotation, data collection and data processing in a standardised format. It also helps to inspire and guide design from users being able to view designs and DNA code which other users have implemented. In instances where users want to implement or test small design edits to other users' designs the DSB platform greatly increases the ease and speed of such design edits; by enabling to directly edit the DNA code and rerun data collection and processing with an identical setup to the preceding design iteration.

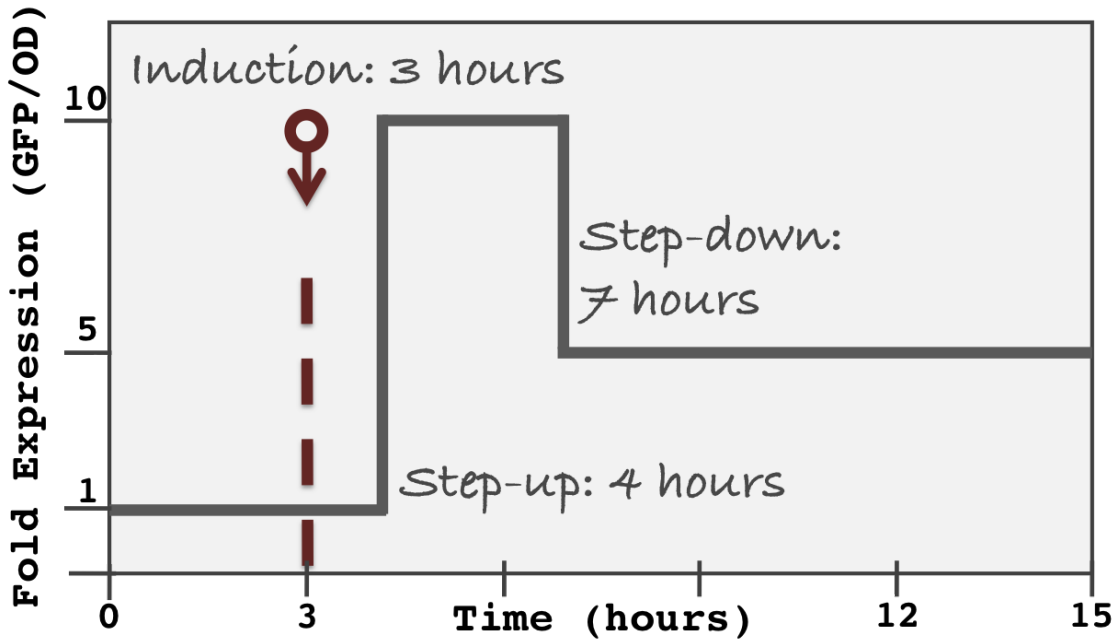


Figure 1.8: The target output for the biological circuit which the users were aiming to build on the DSB platform. The function output is measured as a protein level over time. An output of ten refers to protein level of a defined biological reference standard with constitutive expression (ProC-B0034). The function target output is one for the first four hours, it then spikes to ten for three hours before it is kept at five for the remaining seven hours of readings and redilutions. An aTc input signal is provided at time = 3 hours.

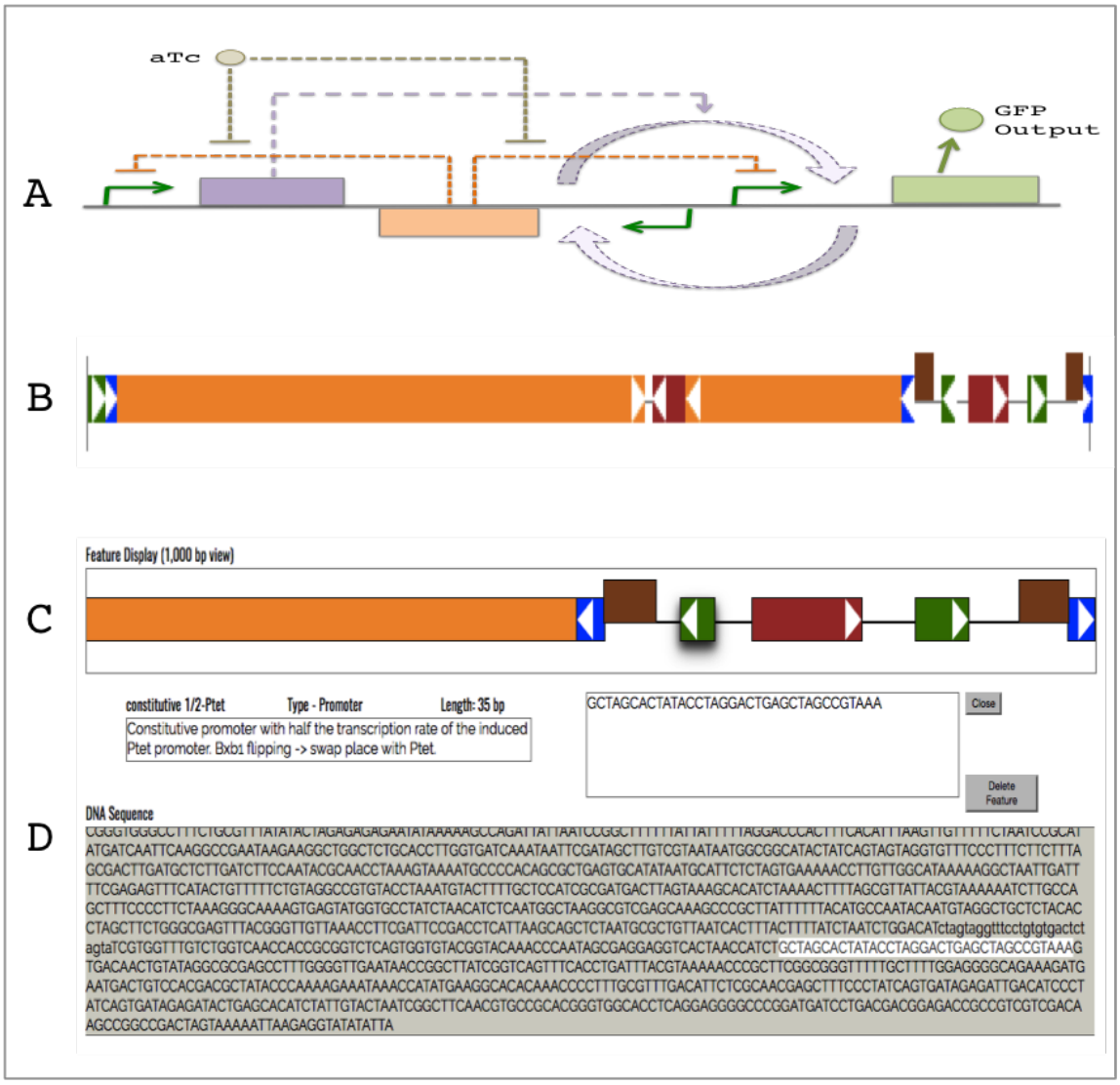


Figure 1.9: A design made on the DSB platform by users with the aim of making the function shown in Figure 1.8. **A:** The conceptual design of the circuit made by the user. The green arrows are promoters and the boxes symbolise the open reading frames: integrase (purple), TetR repressor (orange) and the GFP output (green). **B:** How the design is visualised after user uploaded to the DSB platform. **C:** Description of any currently highlighted feature from the visualisation (highlighted by hovering or clicking). **D:** The DNA sequence of the design. Any currently viewed feature is simultaneously highlighted in the DNA.

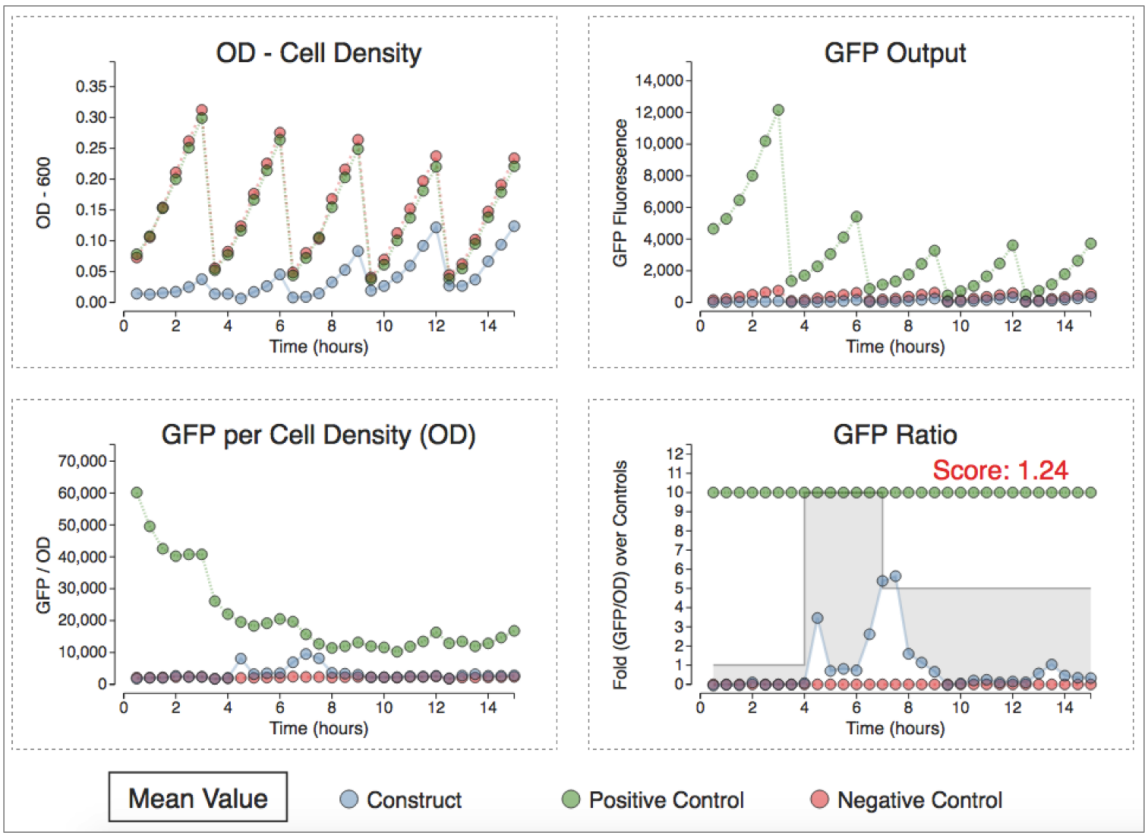


Figure 1.10: The results on the DSB platform from the user design shown in Figure 1.9. The bottom right corner shows the function output. The red 'score' value is an inverted least squares fit between the circuit function and the target output - a higher value indicates a better fit. On the OD measure (top left corner) it can be seen that the construct samples has issues with growth due to design burden.

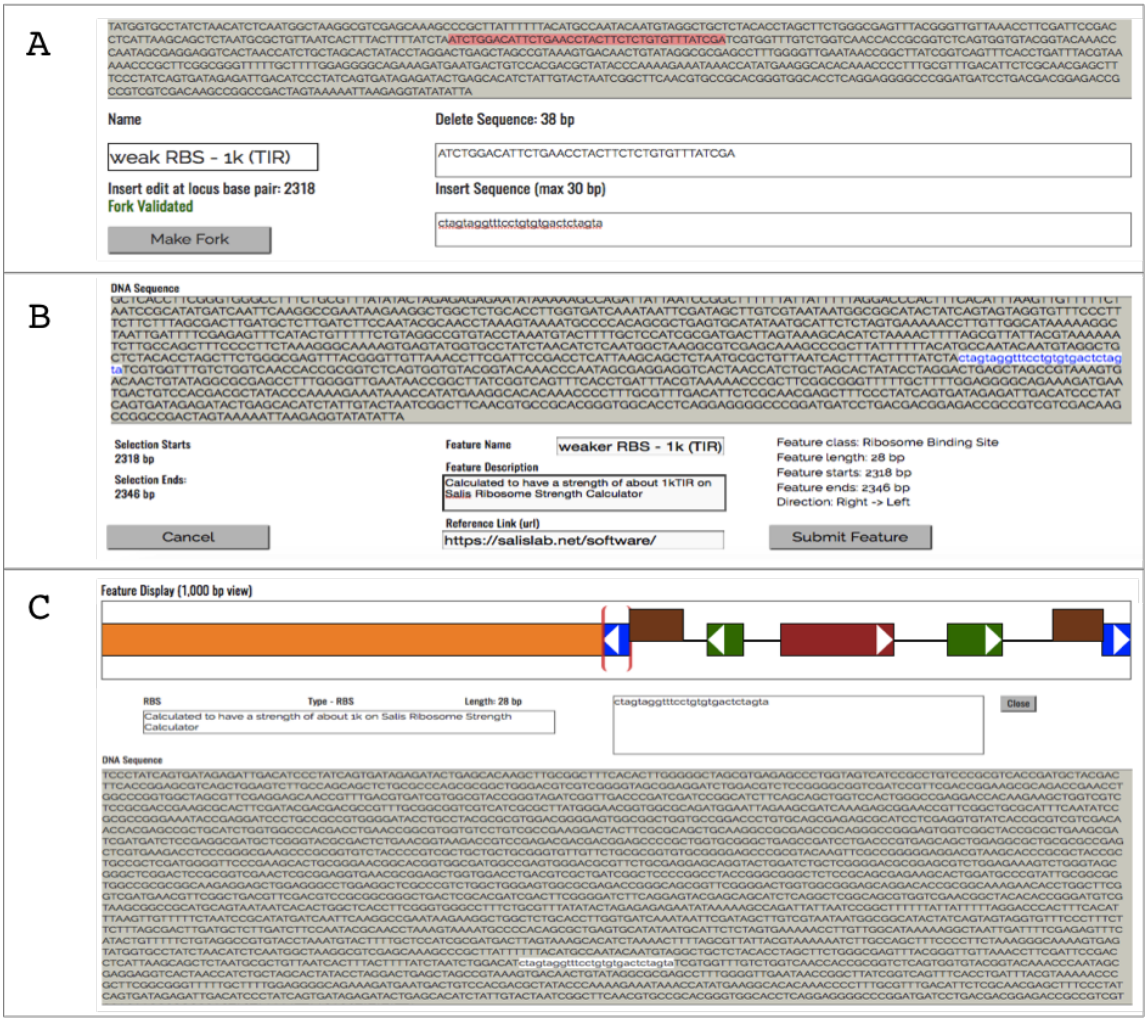


Figure 1.11: The process of submitting a fork on the DSB platform. **A:** Creation of a fork by editing in the original construct. **B:** Annotation of the fork by describing the committed edit. **C:** The display of the forked construct after the edit has been committed. All fork edits or deletions are easy to spot and locate from the red curly-brackets-frame feature display, plus lower-case letters in the DNA code.



Figure 1.12: A screenshot from the DSB platform of the output from the construct fork shown in Figure 1.11. The cell grown in the top left corner shows that the reduced strength of the RBS successfully reduces the construct burden. The performance of the circuit has been improved and the output provides a better fit to the design target specification.

1.5 Discussion and Contributions

1.5.1 Discussion

Robotic Automation

A robotic code-driven implementation was presented, which enables longer experiments while maintaining regular measurements and dilutions, and a high degree of repeatability for experiments. An identical protocol code can be reloaded multiple times to the automated environment once written. Reloading an identical computer-written protocol to a robot generally produces a higher degree of process consistency compared to asking a human to follow a written instruction. If a human has read the instruction before it is possible that they will interpret it differently compared to a first-time reader. If a human has also manually carried out the instruction before it is even more likely that they will change their implementation as they become faster at certain actions through repeated learning. One alternative could be to let new human operators implementing each procedure. However, it produces large variability from different ways of interpreting instructions — in addition, it would be very impractical. Whichever option is selected with human implementation, a fully robotic implementation run from identical computer code is likely to provide a higher degree of repeatability. Facilitated repeatability of experiments is expected to save time for research while simultaneously improve data quality. Therefore, model parameters can be improved both from higher data quality as well as from an increased number of repeated readings.

The DSB Platform & Current Limitations

In addition to pure machine automation, the method demonstrated through the DSB platform provides additional repeatability benefits to Synthetic Biology. When protocols are transferred between different laboratories, metadata of experimental parameters are frequently lost or else are misinterpreted. Using robotic machine-code instead of handwritten notes in note books is likely to alleviate some of the transfer loss of information. However, there are still a number of metadata parameters that can be hard to quantify or gauge

the importance of when setting up new experiments. These parameters can include the ambient room temperature, the brand of equivalent equipment, air humidity, etc.

The DSB platform allows for *decentralisation* of Synthetic Biology, such that engineers more easily can collaborate with each other. However, at the same time, it simultaneously *centralises* the physical implementation and testing of biology for a particular protocol. Centralisation of testing will reduce inconsistency in data resulting from laboratories in different geographical locations.

As a proof-of-concept for the DSB platform the scope was limited to a well defined subsection of Synthetic Biology. In the demonstrated example this scope is limited to the engineering and design of circuits with time series outputs of GFP expression in *E. coli*. The functional outputs of such circuit designs are informed over prolonged time periods through multiple regular sub-culturing events and frequent regular output measurements (together with historical comparisons to other design outputs from identical experimental conditions). The proof-of-concept is achieved on an important, albeit very limited part of considerably wider opportunities for Synthetic Biology research. However, it is possible to imagine the replication of the DSB method demonstrated here to other areas of biological engineering. For example, it could be extended to metabolic engineering in bacteria or yeast, or it could possibly be extended to wider varieties of organisms *e.g.* plants. A necessary requirement for these types of scope extensions is to replace, or augment, the data gathering capabilities of the underlying protocols as well as the data display and visualisation formats on the digital interface. Clearly, the transformation and growth conditions also need to be altered for any new host organism. Meanwhile, the implemented interface for inputs of DNA design and annotation could likely remain fairly constant between different outputs. At the same time, the current design and annotation interface could of course also be improved further, independently of any new outputs or organisms. With future extensions of engineering scopes then different organisms and engineering purposes (defined by their protocol output metrics) would need to be separated between different tabs (and search functions) on the digital interface in order to facilitate comparisons of different designs with the same output format. This will facilitate compatible comparisons while at the same time preventing possible confusions or mistakes from invalid design comparisons

and data outputs between different host organisms, protocols or measurable metrics.

Forks

Forks are provided as a simple and rapid method to edit preexisting biological designs on the DSB platform. Forks are restricted to single locus deletions, insertions or replacements of up to 30bp. However, this restriction is not a bug but a feature. The benefit of a short edit with a specific location is that it allows for changes in systems to be traced over time. An edit of maximum 30bp is short enough such that most edits are restricted to single component edits. The example in Section 1.4.3 showed an edit of an RBS. The fork edits can equally be targeted at exchanging a promoter, altering a terminator, modifying a degradation tag or adjusting a short RNA. However, multiple edits can not be made simultaneously, providing the benefit of tracking the effect of each individual edit. The traceability of design alterations makes each effect and reasoning easier to decipher for all subsequent and future users. Exchange of knowledge will allow (i) shorter learning curves, (ii) specialisation of skills in Biology Engineering and (iii) faster progress in the field by learning from each other's design outputs.

1.5.2 Contributions

Chapter 1 has described a robotic protocol for repeatable experiments of engineered biology. The robotic protocols and methods developed in Chapter 1 will be used in Chapters 2 and 3 for producing models and experimental data for burden from engineered biology. Chapter 1 has also developed an interconnected method for building biology which enables users from anywhere in the world to produce engineered biology data. The new method presented for online Synthetic Biology is named the Digital Synthetic Biology (DSB) platform. The DSB platform provides enhanced repeatability, access, specialisation, sharing and collaboration. The hope is that these features will propel the field of Synthetic Biology into the future at a higher velocity.

2. Chapter 2

Chapter 2 develops design models for evaluating biological constructs during the initial step of the design-build-test engineering cycle. The presented model predicts an increase in burden from elevated translation initiation rates. The model is named the 'RBS Model', where RBS refers to the Ribosome Binding Site of mRNA. To test the model, a library of constructs were built with different copy numbers and ratios of transcription to translation initiation frequencies. The constructs were tested with the automated experimental methods developed in Chapter 1. The results are followed up in Chapter 3 which proceeds to develop new methods for quantifying and measuring burden.

2.1 Introduction

The work on burden in Synthetic Biology can be divided into three broad categories: (i) development of methods to measure or quantify burden [37, 34], (ii) methods to predict burden [162], and (iii) assessment of burden from engineered biology [24].

The current method of measuring burden relies on recording the growth rate [211] of a host carrying a construct and comparing the rate of growth to a host without a construct [111, 34]. Cardinale *et al.* [34] measured the burden of engineered constructs in different hosts as the maximum growth rate of a non-engineered reference control, subtracting the growth rate of the engineered host and dividing the result by the non-engineered reference growth rate (*i.e.* burden $= (\mu_{Ref} - \mu_{Eng})/\mu_{Ref}$). Ceroni *et al.* [37] developed a new experimental method referred to as a biological 'capacitator'. The method depends on constitutively expressing a reference output in the genome of a host; as more expression resources are required and sequestered for the expression of the engineered system which

is tested, less resources are left for the expression of other host proteins. The reference output used in by the method is a fluorescent protein which is simple to quantify as a relative abundance from the absorbance and emission of light with different wavelengths. As more resources by the cell are required for the expression of an inserted construct, then there will be less expression resources available for the expression of the genome encoded reference output. Thus, the relative reduction in expression of the reference output provides a measurable metric for how much resources are required by an engineered system. Chapter 3 in this thesis provides further explanation of measurement methods for burden.

With respect to burden prediction, it has long been perceived that the expression of engineered constructs constitutes a cost to a host. Following from this assumption, it can be further deduced that the cost to a host will increase with an increased expression of constructs. However, exactly how this occurs and the basic relationship between the two parameters has been less clear. In 2010, a seminal study was published which asserts that the cost to hosts follows a linear relationship with engineered expression [162]. The model provided from the study is named the 'growth law'. It states that the growth rate of a host decreases linearly with the fractional increase of engineered constructs relative to the total proteome. The 'growth law' model provides an important tool to inform how much cost can be expected from a protein expression level required for a particular function, for example the expected cost of a metabolic pathway which requires a specific number of different enzymes [24]. Maybe more significant than the model itself are the underlying basic assumptions highlighted which are derived from simple first principles.

Another simple, and thus practical, model for the prediction of burden from engineered biology was developed by Gorochowski *et al.* [78]. The model is comprised by two delayed differential equations (DDEs) with eight parameters in total, including delays in translation initiation and elongation rates. The model assumes a constant concentration of ribosomes in cells. It purposely neglect to include a representation of growth as an output or variable, however, the model variable of endogenous protein expression could be linked to growth. The model was able to predict expression levels of synthetic constructs for different levels of temperature stress [78]. In terms of model simplifications the model provides enhanced guidance for practical implementations in that it only requires information about

the translation times for whole genes [78], but no information about the exact translation rate of individual codons [37], parameters which can drastically increase the complexity of design models. Another model for genetic loads, without direct linkage to the effect on growth, has been developed by Carbonell *et al.* [33] for the prediction of expression interdependence between synthetic constructs. Their model derives a correlation of the interconnected genetic load of synthetic construct expression and to the interconnected load of electric circuits, which is governed by Ohm's law for electric potential and resistors [33]. Later work has focused on de-coupling this interdependence between construct expression via feedback [170]. A different model for expression interactions between two constructs derives isocost lines (employed in microeconomics) between RBS strength and plasmid copy number [81]. This model does not link designs to growth impacts, but it predicts that in order to reduce the sensitivity of expression between two construct, the strongest RBS should be selected together with a plasmid copy number which adjusts the expression to a desired target for the system [81]. Weisse *et al.* [196] develop a mechanistic singel cell model for resource competition and growth rate impacts from synthetic constructs. The model aims to find a simplified and minimal description of trade-offs between cellular requirements between synthetic constructs and growth by considering the biochemical processes of gene expression, nutrient import and metabolism, while assuming a constant cell size [196].

However, the models provided to date provide few guidances as to how a specific design can be altered or enhanced for reducing the burden to hosts. Methods are still required which can evaluate the burden to cells of different designs in order to reduce burden (*i.e.* while maintaining an output according to specification). With respect to design methods, the work on burden has seen considerably less progress than the work on mutational frequency of design, for which at least one method (plus tool) has been provided which can maintain a specific design while reducing its mutational frequency [92].

A large number of studies have measured the burden contributed from the expression of engineered constructs. Many of these studies have produced data to confirm the generally accepted linear relationship predicted from the 'growth law' model [162, 163, 24, 58]. However, a significant number of studies have also measured non-linear burden effects from

expression [37, 54, 35, 168]. Meanwhile, studies in yeast have showed that burden can be attributed to either translation or transcription requirements of a synthetic construct, depending on the nutritional environment [94]. Non-linear relationships of burden are more interesting from the perspective of developing design models for burden. They provide the opportunity to enhance our current understanding of burden, as well as the scope for optimisation and fine-tuning. If the level of burden to a host organism is always proportional to the protein expression level of an engineered construct, irrespectively of the underlying design, then there is little space for improved engineering methods to reduce burden. Instead, it is where this simple rule breaks down that the opportunities arise for optimising previous design methods (*i.e.* with regards to host burden). Chapter 2 identifies a theoretical relationship which satisfies both a linear as well as non-linear relationship between burden and expression, depending on how the construct is built. The identified linear-to-non-linear relationship was developed into a model and was tested experimentally.

2.2 Design Models for Burden

As discussed in Chapter 0, mutational frequency and costs are the two sides of genetic stability. For an efficiently designed DNA sequence, without excessive repeats, the basal mutation rate in *E. coli* is $\sim 10^{-10}$ per basepair per generation [198] and tools are emerging to reduce excessive mutational loss from sequence repeats [92]. With quantified, and possibly reduced, rates for mutations of engineered constructs one of the two parts of genetic stability for engineered constructs is being addressed. However, the cost and burden of engineered biology is also necessary to address in order to improve the stability of constructs. The model developed in Chapter 3 focuses on the cost function of genetic constructs.

For engineered functions requiring a certain output, *e.g.* a protein translated per unit time, target design can be split up between different contributors. These design parameters include: (i) gene copy number, (ii) RBS affinity and (iii) promoter strength, where varying the level of one parameter can be compensated by altering the other two, providing equal expression outputs. However, combinations with the same expression will exercise different burden on the host [37], and thus have different genetic stability.

2.2.1 RBS versus Promoter Strength

An equivalent expression level can be obtained from different combinations of translation and transcription strengths. The level of transcription and translation of biological components is a re-occurring consideration in every biological engineering project. However, even if strong translation combined with weak transcription can give the same protein expression level as an opposite arrangement, the burden experienced by the host can be different.

The 'RBS Burden Model' developed below investigates how different ratios of transcription to translation produce different burden for the host. The model could inform and enable improved designs of Synthetic Biology constructs when facing the common design question of how to match promoters with RBS in order to reduce construct burden.

Background to the Growth Law Model

The cost of an engineered construct has previously been shown to be directly proportional to its expression level [162, 163, 24, 58]. However, experimentally it has also been seen that higher RBS strength for a construct can have worse fitness impact than the same expression level obtained at lower RBS strength and a higher copy number or promoter strength [37]. The linear relationship between engineered protein expression levels and growth rate has been named the 'growth law' [162], and is derived from ribosome fractions required for translation of the engineered construct, instead of being used for growth. It can be summarised as follows; the proteome, when including expression of unnecessary proteins, can be divided into four different fractions:

$$\Phi_R + \Phi_M + \Phi_E + \Phi_Q = 1 \tag{2.1}$$

where

- Φ_R**:** Protein fraction of ribosome related proteins.
- Φ_M**:** Protein fraction of enzymes for nutrient import and metabolism of amino acids.
- Φ_E**:** Protein fraction of engineered constructs.

- Φ_Q**:** Remaining non-growth related proteins. This is assumed to be a constant protein fraction.

The different protein fractions are illustrated in Figure 2.1.

The growth rate (μ) is proportional the product of the translation rate (γ) and the ribosome fraction (Φ_R) [162]:

$$\mu = \gamma_0 \Phi_R \tag{2.2}$$

The rate of translation (γ) is a function of the concentration of free amino acids and charged tRNA. However, in the same model *E. coli* efficiently regulates the costly expression of ribosomes to maintain a constant high translational efficiency, $\gamma(a) = \gamma_0$ [164, 27], where a is free intracellular amino acids, measured as a mass fraction of total protein biomass. (It has been argued that cells regulate the expression of their most abundant proteins very efficiently because sub-optimal regulations of these protein expression leads to much larger wasted production compared proteins present in small quantities [27]. Ribosome related proteins are among the most abundant proteins in fast growing bacteria, such as *E. coli*.) This means that there is a linear relationship between synthetic construct expression and the growth rate of the host [162, 164, 27]. The linearity follows from the Φ_R requirement to translate engineered fractions Φ_E in addition to host native fractions which are required for growth. The extent to which Φ_E is replacing native fractions, in particular Φ_R, directly impacts the growth rate (from Equation (2.2)). Note, in Scott *et al.* [162] Equation (2.2) is written as $\mu = \frac{\kappa_t}{\rho}(\Phi_R - \Phi_0)$, where κ_t is the 'translational capacity' and ρ is a proportionality constant. Φ_0 is a constant size the ribosome fraction when $\mu = 0$, however, for Equations (2.1) and (2.2) this fraction is considered as a constant, non-impacted by growth and thus part of fraction Φ_Q. Thus $\Phi_R - \Phi_0$ in Scott *et al.* [162] is annotated as Φ_R in Equation (2.2). The experimentally recorded Φ_0 fraction in Scott *et al.* [162] can also be though of as the baseline of ribosomes required for protein expression to compensate for protein degradation. In the simplified model developed here the protein degradation rate is not included, and the Φ_0 term to compensate for degradation is not included with in the Φ_R fraction driving growth from protein mass accumulation. In Scott *et al.* [164]

Φ_0 is written as Φ_R^{min}, and $\frac{\kappa_t}{\rho}$ is written as $\gamma(a)$.

The example above assumes equal translational efficiency between the cells' native components and engineered constructs - a property that might not be true, *e.g.* due to low optimality of codon usage or the application of an RBS with too high affinity, leading to 'overinitiation' of translation [87, 37].

A few of the other relationships which follow, or are assumed in the model above, are here considered. For example, that the growth rate (μ) is proportional to the product of the translation rate and ribosome fraction could have different interpretations. Firstly, for translation of proteins to be directly linked to growth rate, it needs to be assumed that growth in this instance refers to the rate of total biomass change. Secondly, it needs to be assumed that the total biomass fraction of proteins is constant. Thus, the rate of protein mass production becomes directly linked to the rate of change for biomass. A different method to measure growth could be through number of cell, this method would not be equal to biomass growth as cell sizes differ during different stages of growth. The growth rate μ referred to in the model is biomass growth. Further, the translation rate γ is given as the rate of protein production per total fraction of ribosomes Φ_R. Not all ribosomes will be engaged in the translation of mRNA to proteins, but the translation rate γ refers to the translation rate per the total mass fraction of ribosomes.

Derivation of the RBS Burden Model

The rate of amino acid supply and demand is given by:

$$\frac{da}{dt} = \nu(s,a)\Phi_M - \mu - a\mu, \tag{2.3}$$

where

- a**:** Intracellular free amino acids, measured as a mass fraction of total protein biomass.
- s**:** The external nutrient quality and concentration,
- $\nu(s,a)$**:** The rate of importing or metabolising amino acids from extracellular nutrients.

The first μ term refers to the amino acids utilised for protein synthesis, *i.e.* Equation (2.2), and the $a \times \mu$ term refers to the amino acids lost due to dilution at cell division (note $a \ll 1$ [164] and $\mu(1+a) \simeq \mu$). The general form of Equation (2.3) has close similarities to nutrient import equations in other models, however where the internal nutrient levels are recorded as number of molecules [196] as well as internal volume concentrations [75]. At a saturated nutrient environment $\nu(s, a) = \nu(a)$. But also the free amino acid level is assumed constant, due to efficient regulation of resource allocation (*i.e.* $\frac{da}{dt} = 0$ [164, 27]), $\nu(a) = \nu_0$, thus resulting in:

$$\nu_0 \Phi_M = \mu(1 + a) \tag{2.4}$$

From Equations (2.1), (2.2) and (2.4):

$$\nu_0(1 - \Phi_R - \Phi_Q - \Phi_E) = \gamma_0 \Phi_R(1 + a)$$

$$\Phi_R = \frac{\nu_0(1 - \Phi_Q - \Phi_E)}{\gamma_0(1 + a) + \nu_0} \tag{2.5}$$

If the rate of translation of engineered construct is different from the translation rate of native protein fractions (*i.e.* $\gamma_0 \neq \gamma_E$), then a new total translation rate (γ_{total}) has to be calculated which is representative of γ_0 and γ_E combined. In other words, γ_{total} then replaces the previous γ_0, and is the sum of products of individual translation rates and relative mRNA fractions (Ψ) which are bound to translating ribosomes. Subsequently, Ψ_x refers to the fraction of translating ribosomes in a cell which are engaged in translating mRNA, which encodes a protein belonging to the protein fraction Φ_x. Figure 2.1 provides a cartoon to illustrate Ψ and Φ fractions. The total (and mean) ribosome rate of translation in a cell is given by

$$\gamma_{total} = \sum_x \gamma_x \Psi_x \tag{2.6}$$

where x could be R, E, M, Q as before.

Also, as all non-engineered proteins are assumed to be translated with the same rate as

before (*i.e.* γ_0) and non-engineered mRNA-ribosome bound fractions which are translated with rate γ_0 are given by $= 1 - \Psi_E$, then the total sum of translation rates can be written as

$$\sum_{x \in \{R,..,E\}} \gamma_x \Psi_x = \gamma_0(1 - \Psi_E) + \gamma_E \Psi_E,$$

or equivalently

$$\sum_{x \in \{R,..,E\}} \gamma_x \Psi_x = \gamma_0 + \Psi_E(\gamma_E - \gamma_0). \tag{2.7}$$

However, when performing output measurements, we can measure the protein fraction (Φ), rather than the mRNA fraction, or mRNA-ribosome bound (Ψ). If using the same rate of protein degradation for designed constructs as the host proteome (if using a different degradation rate, Φ_E has to be adjusted accordingly in all the equations), the protein fractions are equal to the product of their mRNA translation rate and corresponding mRNA-ribosome bound concentration divided by all translated mRNA [162],

$$\Phi_x = \frac{\gamma_x \Psi_x}{\sum_{x \in \{R,..,E\}} \gamma_x \Psi_x}. \tag{2.8}$$

This again assumes a steady state of growth where fraction sizes are not increasing or decreasing, and also an equal degradation rate between the different protein fractions. As an example, if the cell is in the process of up regulating the Φ_M fraction, *e.g.* due to a worsening nutritional environment, then $\Phi_M < (\gamma_M \Psi_M)/(\sum_x \gamma_x \Psi_x)$. When protein fraction sizes are up or down regulated by the cell then $\gamma_x \Psi_x / \sum_{x \in \{R,..,E\}} \gamma_x \Psi_x$ first changes and time is required for Φ_x to reach this equilibrium (this regulation process is discussed again with further considerations in Chapter 3 of this thesis, but the assumptions of Equation (2.8) refers to steady states of growth, *i.e.* where $d\Phi_x/dt = 0$).

From Equations (2.7) and (2.8):

$$\Phi_E = \frac{\gamma_E \Psi_E}{\gamma_0 + \Psi_E(\gamma_E - \gamma_0)}$$

$$\Psi_E = \frac{\gamma_0}{\frac{\gamma_E}{\Phi_E} + \gamma_0 - \gamma_E}. \tag{2.9}$$

From Equations (2.6), (2.7) and (2.9):

$$\gamma_{total} = \gamma_0\Big(1 + \frac{\gamma_E - \gamma_0}{\frac{\gamma_E}{\Phi_E} + \gamma_0 - \gamma_E}\Big)$$

$$\gamma_{total} = \frac{\gamma_0\gamma_E}{\gamma_E + (\gamma_0 - \gamma_E)\Phi_E}. \tag{2.10}$$

This means that γ_{total} is given as a function of engineered protein fraction and its translational efficiency. Equation (2.10) shows how the total translation rate (γ_{total}) of a cell is impacted from different levels of measurable engineered output (Φ_E), when the translation rate of the the construct mRNA (γ_E) differs from the translation rate of native mRNA (γ_0). The purpose of the derivation of Equation (2.10) is to provide a relationship of γ_{total}, γ_0 and γ_E in the context of a parameter which is readily measurable, namely Φ_E.

The ribosomal protein fraction will not stay constant as the total translational efficiency changes. Instead, from Equation (2.5):

$$\Phi_R = \frac{\nu_0(1 - \Phi_Q - \Phi_E)}{\gamma_{total}(1 + a) + \nu_0}.$$

Or equally (as $a \ll 1$ and $\mu(1 + a) \simeq \mu$),

$$\Phi_R = \frac{\nu_0(1 - \Phi_Q - \Phi_E)}{\gamma_{total} + \nu_0}. \tag{2.11}$$

With Equations (2.10) and (2.11), a relationship has been derived between engineered construct translational efficiency (γ_E) and native translational efficiency (γ_0), providing the growth rate (μ) at specific expression levels of engineered construct (Φ_E):

$$\gamma_{total} = \frac{\gamma_0\gamma_E}{\gamma_E + (\gamma_0 - \gamma_E)\Phi_E}$$

$$\Phi_R = \frac{\nu_0(1 - \Phi_Q - \Phi_E)}{\gamma_{total} + \nu_0}$$

$$\mu = \gamma_{total}\Phi_R \tag{2.12}$$

According to Equation (2.12), the fitness cost of engineered protein expression ceases to be linear at lower translational efficiency (see Figure 2.2). According to the derived relationships of the model, this non-linear relationship is a simple effect of intracellular redistribution of resources due to a reduced aggregated translational efficiency. Overinitiation from high affinity RBS [37] reduces γ_E of that protein fraction and more ribosomes will be expressed to handle the same flux of amino acids. In other words, the cell 'thinks' that it has an abundance of nutritional resources and regulates itself for faster growth. Thus overinitiation from a strong RBS has an amplified negative effect as it sequesters ribosomes from other functions but also makes the cell express a larger fraction of ribosomes. See Figure 2.2 for the effect of the derived model. Φ_E shows how much engineered construct we are expressing with the engineered system, as a fraction of total proteins in the cell. γ_E/γ_0 is the ratio of translational efficiency between engineered constructs and the translation efficiency of native proteins. The translation efficiency of engineered proteins is decreased due to suboptimal codon use or excessive RBS strength causing overinitiation. The vertical axis ($\mu(\Phi_E)$) shows a measure of fitness cost given in growth rate.

Predictions from the RBS Burden Model

It is important to separate the translation efficiency seen from the perspective of the mRNA molecule and that seen from the perspective of ribosomes. For example, a high rate of initiation is efficient from the perspective of construct mRNA as more protein is translated from each mRNA. However, from the perspective of ribosomes, it leads to the overall rate of translation being reduced as some ribosomes are queuing and not operating at their highest capacity [87]. If Equation (2.12) holds experimentally, in addition to guiding stability design, it can help as a direct tool for measuring construct translational efficiency. It predicts that the slope of growth rate against protein expression is directly linked to the translational efficiency of the constructed circuit (see Figure 2.2).

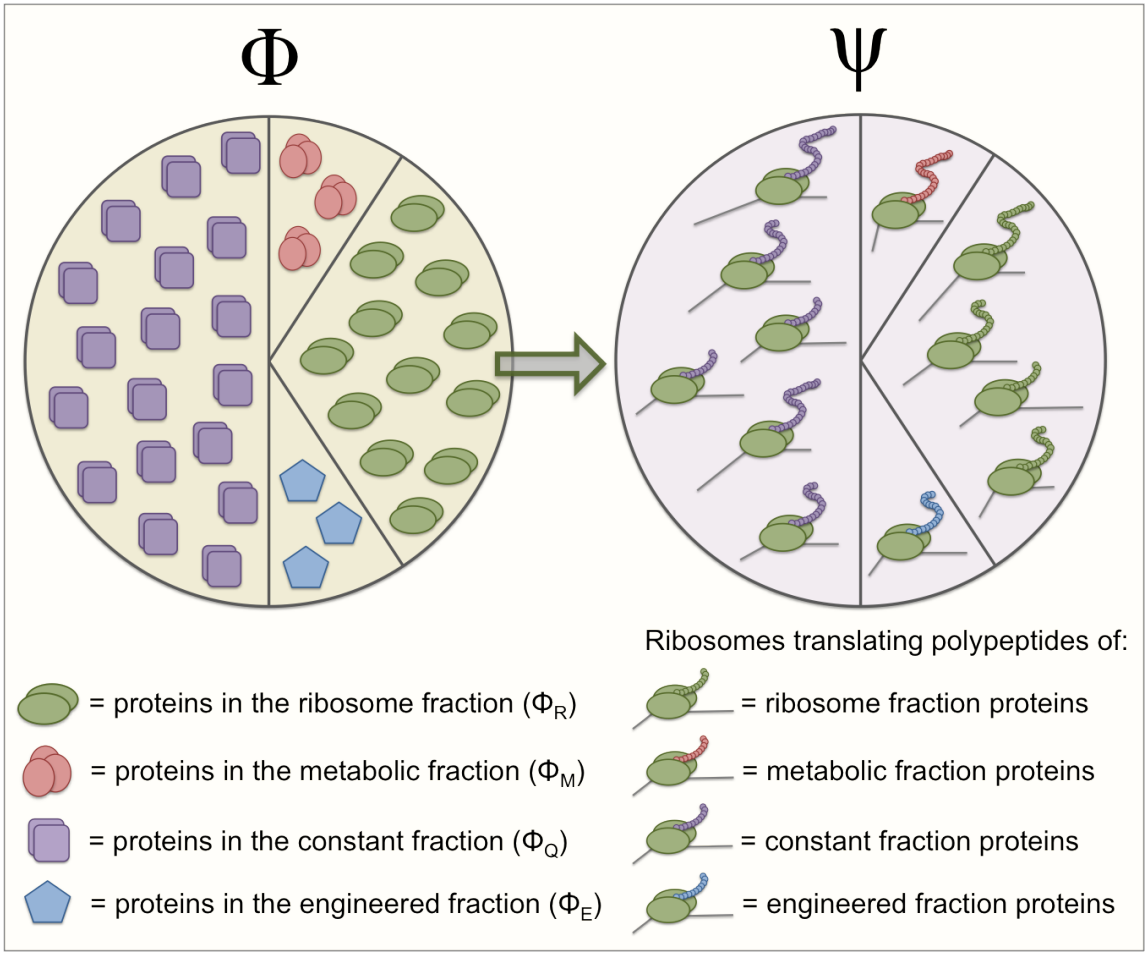


Figure 2.1: Illustration of the protein types included in the model with their different fractions for Φ and Ψ. The protein fractions in a population are annotated by Φ. Whereas Ψ determines what protein fractions that the ribosomes are currently engaged in translating. At steady state (*i.e.* no change in protein fractions), and at equal rate of translation and degradation for the different fractions then the proportion of ribosomes translating a certain fraction is equal to that current protein fraction in a cell. In other words, during such steady state $\Phi_x = \Psi_x$. At steady state but different rate of translation for the different fractions then Φ and Ψ are linked by Equation (2.8). The dark gray lines below the translating ribosomes in the Ψ fractions illustrate ribosome bound mRNA from the respective fraction.

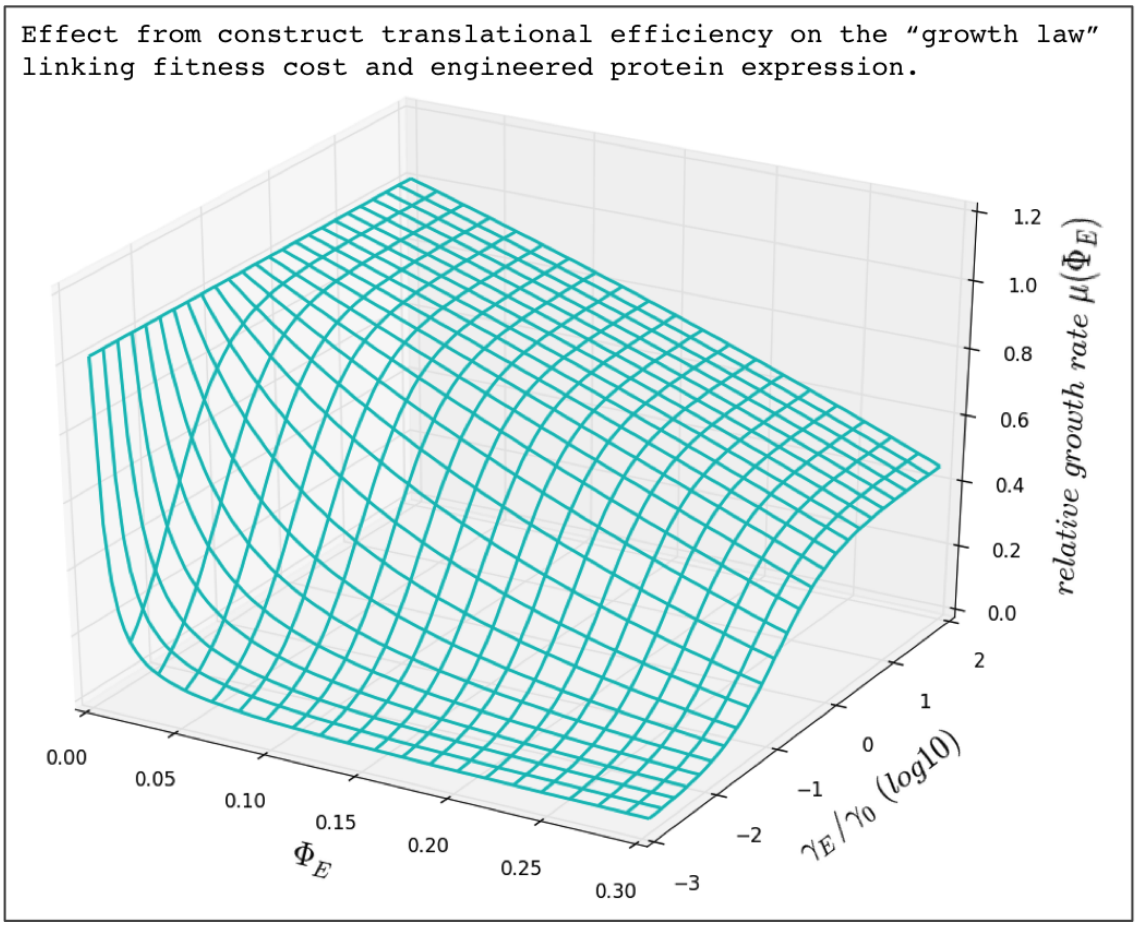


Figure 2.2: Plot of Equations (2.10) - (2.12). At reduced translational efficiency, *e.g.* from utilising excessive RBS affinity causing ribosome overinitiation [87, 37], the linear 'growth law' ceases to be linear. The previously predicted and demonstrated 'growth law' linear relationship [162, 27] is located at $\gamma_E = \gamma_0$ (*i.e.* at 0 on the axis for γ_E/γ_0 $(log10)$).

Experimental Test

To test the model prediction that reduced construct translation rate from RBS strength is a problem that harms common genetic designs, a library of promoter strengths, RBS, and plasmid copy number was designed and assembled for GFP expression. If excessive RBS affinity constitutes a high load in engineered biology, this, according to the equation, will be manifested through constructs with higher RBS strength not following a linear decrease in growth rate when total protein expression is increased. In other words, too high RBS strengths in construct designs will lead to a non-linear decrease in μ as construct expression increases. Figure 2.3 shows an illustration of experimental data which could be able to strengthen the current model hypothesis.

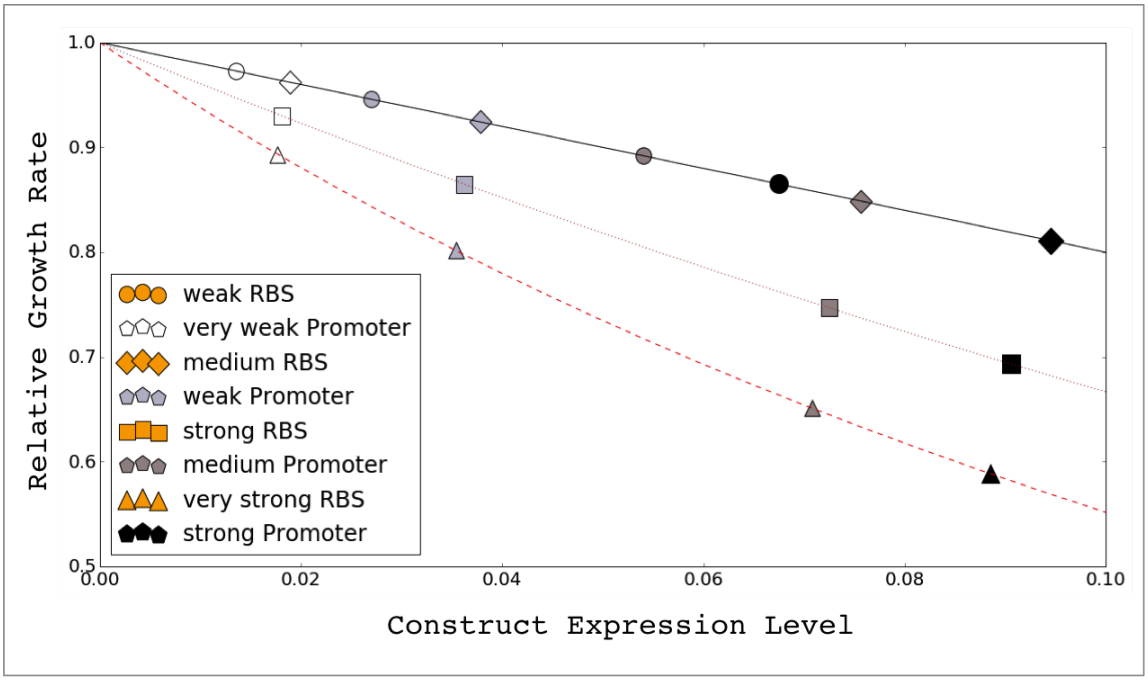


Figure 2.3: An illustration of example experimental data measurements which would be able to strengthen the model hypothesis. The example data shows the expression levels and growth rate impacts from a matrix of four promoter strengths and four RBS strengths. The two strongest RBS strengths are assumed to lead to 'overintitation' of mRNA by the ribosomes and lead to reduced translational efficiencies for the expressed constructs. The promoter strength is color coded and the RBS strength is given by the shape of each data point. The 'very strong RBS' (triangels along the red dashed line) is here assumed to have translational efficiency of the construct (γ_E) one tenth of the native translational efficiency (γ_0), *i.e.* $\frac{\gamma_E}{\gamma_0} = 0.1$. The 'strong RBS' (squares along the brown dotted line) is assumed have translational efficiency of the construct (γ_E) one fifth the level of the native translational efficiency (γ_0). The medium and weak RBS are assumed to be below the threshold for 'overinitiation' and have the same translational efficiency as the native mRNA ($\gamma_E = \gamma_0$). The black trend-line shows the linear reduction in growth rate in accordance with the 'growth law' [162]. The vertical growth axis refers to the growth rate of cells with a construct divided by the growth rate of cells without a construct. The horizontal expression axis shows a relative expression level of a construct. Each promoter and RBS strength is given a fixed value and the relative expression of each combination of promoter and RBS is estimated by promoter strength times RBS strength times translation rate. If the strongest RBS in an experiment is not too strong compared to the rate of the translation for the mRNA of the construct, then no overinitiation (*i.e.* clogging of translating ribosomes) will occur, and all expression versus growth levels would be located along the linear black trend-line.

2.3 Biology Design Parameters for Experiments

2.3.1 Promoters for Experiments

Synthetic biology is the 'engineering of biology', however, many parts utilised as components of larger systems are poorly understood, or at least poorly characterised [105]. The current reliance on poorly understood parts sets synthetic biology apart from many other engineering disciplines as it makes the design of larger circuits difficult and unreliable [105].

The experimental work in this Chapter uses some of the most reliable as well as frequently used transcriptional promoters. The rationale for this is twofold: (i) to obtain better informed input parameters for models and (ii) to extend the experimental data on parts that the research community is likely to apply for other projects.

In order to improve upon the reusability of components, and hence the ability to forward-engineer biology, libraries of 'standard' promoter sets were developed by the synthetic biology community [129, 51, 96, 32]. A large number of synthetic promoters have been generated and characterised, but they often show activities that vary with the gene transcribed or context of use [51]. One of the original and still very commonly used libraries for constitutive expression in *E. coli* hosts is the Anderson collection provided by the iGEM biobrick registry of standard biological parts [96]. Reusing smaller sets of promoters in different projects is useful for two reasons: it provides a well informed starting point when initiating novel forward-engineering projects of biology and it improves the collective knowledge of these universal components by generating an ever increasing amount of data, benefitting all future users. However, the success of reusing well-characterized components relies on a critical assumption that such devices are *functionally composable*, meaning that the properties of the device in one test context are predictive of those properties in a new context [51]. It has been shown that the commonly used Anderson collection [96] of promoters exhibit a very context-specific level of expression, dependent on varying upstream or downstream sequences [51].

Davis *et al.* [51] sought to improve reusability and hence composability of promoters as building blocks for the forward-engineering of larger circuits by further insulating them from genomic or environmental contexts. Specifically they aimed to reduce influences on

transcription from 5'-upstream binding host transcription factors. The Davis library of insulated promoters utilises a 70 bp constant insulator upstream of the -35 box and a 61 bp constant insulator downstream of -10 box, a proximity region that otherwise contains the large majority of host transcription factor binding motifs in an *E. coli* host [51]. Four insulated promoters from the Davis collection were used for the experiments in this project.

The variation of experimental setups and laboratory equipment accounts for a large fraction of measurement inconsistency for synthetic biology parts [96]. However, when parts are measured relative to biological references instead of absolute counts of molecules, *e.g.* promoter expression levels measured relative to a biological positive control, more than half of the lab-to-lab variability can be eliminated [96]. For the experimental methods in this project all expression levels and growth phenotypes have been quantified relative to biological positive controls.

Promoter	-35 box	-10 box	Relative Expression
ProA	tttacg	taggct	0.030
ProB	tttacg	taatat	0.119
ProC	tttacg	tatgat	0.278
ProD	tttacg	tataat	1.000

Table 2.3.1 *The relative expression levels initially recorded by Davis et al. [51] for the insulated constitutive promoters used in the project experiments.*

Constitutive promoters are not regulated by external signals, internal transcription factors or other time varying stimuli. However, all promoters in *E. coli* are reliant on different RNAP-binding sigma factors expressed at varying concentrations in different environments [169], meaning that none of them are truly non-regulated or active at constant levels. What is generally referred to as constitutive promoters are those reliant on the σ-70. This σ-factor is the 'housekeeping' σ-factor in *E. coli* hosts [79], which ensures essential genes are expressed and it is present at high concentration when cells are growing well and multiplying rapidly.

2.3.2 Ribosome-Binding-Sites for Experiments

For selection of translation level control in the experiments, two different sources were used for ribosome-binding site (RBS) sequences: (i) frequently reported RBS from the literature [195, 63, 51, 71] also represented in the iGEM registry of standard parts and (ii) *de-novo* designed sequences via the 'Ribosome calculator v.2.0' software [155]. The RBS from the literature are shown in Table 2.3.2, and the expression levels for the *de-novo* designed sequences are measured in Figure 2.9.

As highlighted previously, promoters with informed, reproducible and composable strengths are important for the forward-engineering of biology. Access to well-informed ribosome-binding site strengths is at least as important as access to well-characterised promoter components. Nevertheless, there is a significant practical difference between these two design 'dials'. Regulation is frequently introduced at the transcriptional instead of the translational level, and often with limited options to choose from to modulate strength of regulation from necessary signals. Therefore, the RBS becomes the common focus for tuning the magnitude of the desired signal output from promoter-encoded transcriptional regulation. Put differently, if there are different regulatory promoters to select from for a particular input signal, then they are often selected based on their fold-increase in transcription rather than their absolute level of transcription. Therefore, the RBS nodes are often the tuning points for the magnitude of the transcriptional response levels.

There are currently two common methods for choosing and implementing RBS strengths in the forward-engineering of biology: (i) libraries of pre-screened RBS [103, 129], *i.e.* similar to promoter levels [51] and (ii) software tools for predicting RNA folding and ribosome binding affinity [155]. But none of the two methods solves the complete challenge. The level of translation depends on the downstream RNA sequence for the protein coding part of the gene being expressed, meaning that the library method is dependent on a particular gene of interest already being part of a pre-existing library. Whilst the software tool method can calculate estimates for translation of any gene of interest, the method is currently very inaccurate (see Figure 2.9). In addition, it has been observed that the upstream promoter sequence of an RBS in some cases can impact the level of translation

realised *in vivo* [103, 129]. Two of the largest libraries to date of pre-screened promoters and RBS strengths were produced by Mutalik *et al.* [129] and Kosuri *et al.* [103] . Mutalik *et al.* [129] built $\sim$ 500 promoters and RBS, and subsequently combined them into 1,200 promoter-RBS combinations that were tested for expression in an *E. coli* host. Kosuri *et al.* [103] built 12,563 promoter-RBS combinations from commonly used components and tested them for protein expression as well as mRNA levels in an *E. coli* host.

Two of the RBS used in the experiments done in this thesis are also present in the library measured by Kosuri *et al.* [103] where they are initiating translation of the coding region of a GFP protein. From the total number of 111 RBS present in the Kosuri-library the two project RBS, B0032 and B0034, represent a library strength rank of 29/111 and 109/111, respectively. B0032 was further used by Davis *et al.* [51] when testing the four insulated promoter sequences discussed in the previous section. None of the project RBS are identical nor directly comparable to the Mutalik-library primarily due to a different design mechanism of 'bicistronic RBS design' instead of 'monocistronic RBS design' [129]. The bicistronic design relies on using a double set of RBS and open-reading-frames for each gene of interest. The first RBS initiates translation of a short 'dummy' polypeptide. Translation is then terminated and followed by translation initiation of the gene of interest. If reusing an identical upstream 'dummy' polypeptide and termination mRNA sequence the design method provides a more accurate design prediction of the translation initiation rate for the second RBS [129].

The authors discussed that the method with the bicistronic design will aid the forward-engineering of larger systems [129]. However, a counter argument is that for larger systems it becomes increasingly important to avoid reuse of repeated identical DNA sequences. The bicistronic method relies on reuse of a single identical DNA sequences upfront of each gene of interest. Reusing this component within a single system will drastically reduce the genetic stability [92]. Bicistronic design has not been used in the study of this thesis due to the additional burden from the upstream open-reading-frame. That initial upstream burden cannot be to linked to a measurable output, and therefore it is difficult to quantify.

Currently, the best and most widely used tool for calculating the RBS strength for any given protein sequence is the 'RBS calculator v2.0' by the Salis Lab [26, 155]. The RBS

calculator relies on a biophysical model of the free energies of translation initiation and sequence-dependent changes during RNA folding and hybridisation [155], which enables the prediction of mRNA translation rates. In addition, the RBS calculator is equipped with an optimisation algorithm that enables the forward-engineering of any given translation initiation rate for a provided gene target [26, 155]. For the experiments in this thesis, the RBS designed via the RBS calculator were produced to have very high rates of translation initiation. A total of six RBS between 349,000-1,926,000 translation-initiation-rates (TIR) were designed with the ribosome-calculator (see Figure 2.9). In comparison, with the same calculator the commonly used B0031-B0035 [195, 63, 51, 71] have much lower TIR values of 670; 1,134; 6,708 and 10,906 respectively [155] (see Table 2.3.2).

RBS	DNA sequence	Calculated TIR
B0031	TCACACAGGAAACCTACTAG	670
B0032	TCACACAGGAAAGTACTAG	1,134
B0034	AAAGAGGAGAAATACTAG	6,708
B0035	ATTAAAGAGGAGAATACTAG	10,906

Table 2.3.2 *The predicted Translation Initiation Rate of four standard RBS calculated with the RBS calculator v2.0 [26, 155]. Note, TIR values do not correspond linearly with measured expression level from experiments. Most notably, measurements give a higher expression level for B0031 than for B0032, as seen in Figure 2.5.*

2.3.3 Plasmid Replication Systems for Experiments

Plasmids are extra chromosomal DNA elements that replicate autonomously and independently of the host genome [55]. Autonomous down-regulation of plasmid copy number is required to avoid deleterious effects of excessive plasmid copy number on cell growth. Therefore, plasmids have evolved efficient control mechanisms that prevent plasmid copy number increasing above a certain threshold, which is determined by the affinity of a plasmid-encoded inhibitor for its target [188]. The regulator can be either a protein or, as for the plasmids used in this project, a nucleotide sequence (RNA).

Circular plasmids use a variety of replication strategies depending on the mechanism of initiation of DNA replication. Replication initiation depends on a section of sequence known as plasmid origin of replication (*ori*), whereas the basic *replicon* refers to the minimal sequence that supports replication preserving the regulatory circuitry [108].

ColE1-type plasmid families belong to the best-understood plasmid control systems [38]: the class B theta replication system [55], which differ from other theta replication systems in that they carry no plasmid initiation factors but instead rely exclusively on host factors for replication initiation [108].

The ColE1-type plasmids replication system is comprised of five components; two promoters, two RNA and one *ori* [108]. Replication begins with the synthesis of an RNA primer precursor, RNAII, transcribed from constitutive promoter P2 [38]. RNAII binds the *ori* and forms an RNA-DNA R-loop which is processed by RNAsae H followed by Pol I initiating DNA synthesis. RNA I, which is transcribed from an independent promoter P1, is an inhibitor of primer formation by RNAII. The antisense RNA inhibition employed in the interaction of RNAI with RNAII is one out of two main classes of plasmid control mechanisms. Regulation by antisense RNA also permits different compatibility groups to evolve from different specificity regions by the antisense RNA [38]. For example p15A and pMB1 belong to two different compatibility groups due to differences in RNAI and RNAII [166].

Plasmids with three different copy numbers were designed and tested in this study. All three plasmids have a ColE1-type origin of replication [181, 39, 205, 166], but with three variations that produce large effects upon the copy number of plasmids replicated within the cell.

The low copy number plasmid designed for the experiments belongs to the p15A replicon type [166], whereas the medium and high copy number plasmids stem from pMB1 derived replicons [181, 39, 205]. Both p15A and pMB1 origins constitute two of the most common replication groups for cloning vectors [187, 156].

The majority of experiments in this study were implemented with the medium and high copy number plasmids. The medium and high copy plasmids also produced larger phenotypic effects, which are easier to separate out from noise. See Figure 2.4 for a

comparison of expression levels from the three plasmids.

The Low-Copy Number Plasmid - p15A *ori*

A p15A [40] origin of replication was used for the lowest copy number plasmids.

p15A belongs to the wider group ColE1-type plasmids and it shares a 265bp conserved region with ColE1 [166]. However, p15A is compatible with ColE1 as well as pMB1 *ori* plasmids because it relies on a different region of interaction between RNAI and RNAII [166]. The p15A replicon generally has a copy number of 10-12 plasmids per chromosome in *E. coli* hosts [173].

The Medium-Copy Number Plasmid

The medium copy plasmid used in this study was designed as a half step between pUC19 [208, 110] and pBR322 [25].

The medium copy plasmid designed for the experiments is rop^-, same as pUC19 or pBRA322 [110], whereas pBR322 is rop^+ [110]. However, the medium copy plasmid is made with the RNAI G-point-mutation present in pUC19/pBRA322 but lacking in pBR322 [110]. The Rop protein, as a dimer, further regulates the process as it strengthens the RNAI/RNA II duplex and enhances the inhibitory activity of RNAI on RNAII, but it has no effect on primer transcription itself [38]. The Medium-Copy plasmid has the intended replicon functionality of pBR3722 tested by Lin-Chao *et al.* [110].

The High-Copy Number Plasmid - pUC19 *ori*

The High-Copy-plasmid is intended to have the same functionality as the origin of replication of pUC19 [208] as well as pBR3722 [110].

pUC19 together with its close relatives (including pUC18) is a commonly used plasmid for cloning work [208, 110]. It has a replicon derived from pBR322 [25] but with two important alterations providing drastically increased vector replication inside bacterial hosts.

- Deletion of the *rop* gene [110].

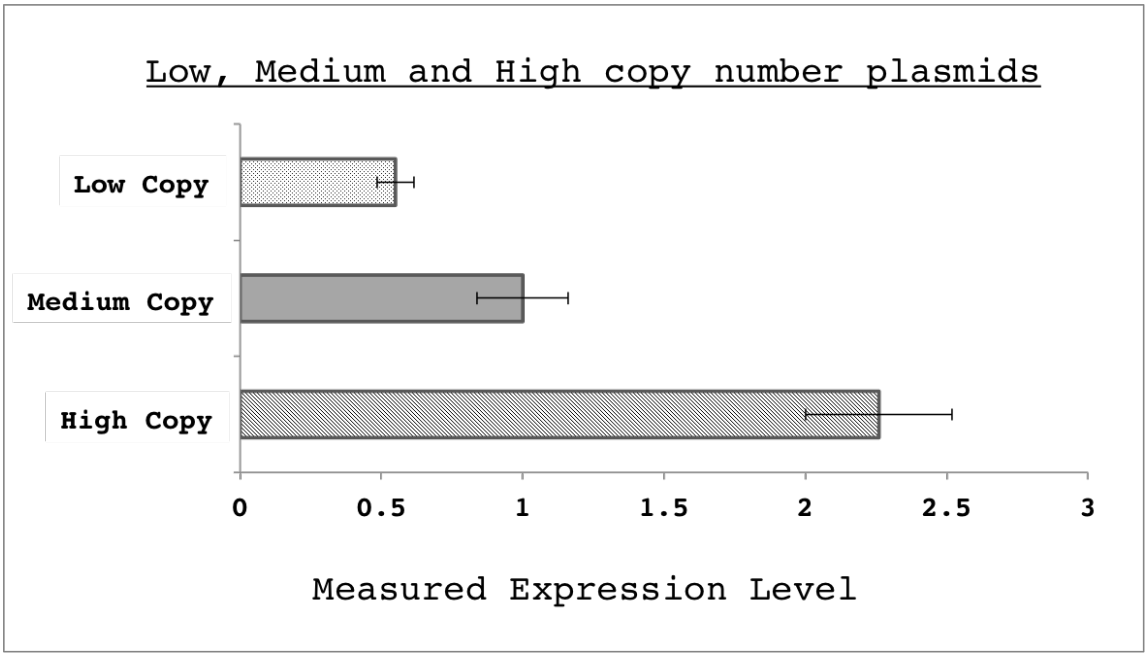


Figure 2.4: The relative expression levels of the 3 different plasmids. Medium copy plasmid is provided as the reference level of 1. Expression levels shown were measured with ProC-B0034 expression between Low and Medium copy plasmids, and with ProD-B0034 between Medium and High copy number plasmids. The expression levels compared are measured by the protein fluorescence output divided by optical cell density when $OD_{600} = 0.2$.

- A (G $\rightarrow$ A) point substitution in nucleotide 112 of the RNA II transcript of the origin of replication [110], affecting its ability to bind RNAI [186].

Replication of ColE1-type plasmids is controlled by a small plasmid encoded protein, Rop, of 63-amino acid length, providing a copy number in the order of 15 copies per chromosome [181]. However, deletion of the *rop* gene approximately doubles the plasmid copy number [181, 39]. The (G $\rightarrow$ A) base-pair substitution implemented in order to increase the copy number is only expected to increase plasmid replication in a ColE1-type rop^{-} mutants [110]. The Rop protein can suppress the relaxed phenotype introduced by the (G $\rightarrow$ A) basepair substitution [110] by catalysing RNAI-RNAII binding [186]. The *rop* gene has been deleted in the Medium and High copy plasmids designed and used in the described project experiments.

It has been shown that amino acid starvation can have positive effect on the amplification of ColE1-type origin of replication [86]. The increased amplification during amino acid starved conditions is argued to be an effect of free uncharged tRNAs binding to RNAI,

preventing it from binding and inhibiting RNAII [205]. However, the stringent response mediated by ppGpp also limits the transcription of tRNA upon amino acid starvation. Therefore, the positive effect on copy number from amino acid starvation is amplified in $relA^-$ relaxed mutants [205]. An insignificant effect on the copy number of pMB1-type replicons has been shown from amino acid starvation in $relA^+$ strains with an intact stringent response [205]. It has been hypothesised that pMB1 *ori* lack of influence from amino acid starvation and increased concentration of uncharged tRNAs is due to presence of the Rop protein catalysing the interaction between RNAI and RNAII [186, 205]. Certain uncharged tRNAs have also been demonstrated to directly cleave RNAI [188]. The rop^- mutant of pBR322 has the same copy number at fast growing cells compared with rop^+, whereas at slower growth rates the rop^- mutant has 2-3 times higher copy number than rop^+ [8]. As mentioned previously, the pMB1-replicon derived medium and high copy number plasmids used in the experiments have been designed without a *rop* gene. For pBR322 rop^+ replicons, the concentration of plasmids inside cells is decreasing with increasing growth rate [109]. The plasmid concentration decreases four fold [109] as growth rates increase from 0.6 to 2.5 h^{-1} [109].

2.4 Library Expression Levels

2.4.1 Experimental Procedures

The general experimental methods are described in Section 1.3.1. All methods for expression-level measurements use values from the initial growth from colonies before any dilutions have been allowed to occur. There are two reasons for this: (i) measurements of the initial values directly from colony growth are less affected by any loss of function due to genetic instability, especially likely to affect strong combinations of expressions, and (ii) when measuring directly from colony growth the impact of residual protein concentrations is minimised (*e.g.* if instead of measuring after a culture dilution step there will be varying levels of legacy proteins present inside the cells that were accumulated before dilution).

The first protocol method utilises measurements of expression levels every second hour (directly after colony growth) and the second method measures expression levels every

sixth minute.

There is a single negative control plasmid that has been used for negative controls in the experiments, and two different positive controls have been used in the full set of data. Expression levels are quantified relative to positive control. The negative control vector is a medium-copy (MC) plasmid without any construct for GFP. The two positive control vectors are medium-copy (MC) plasmids with either ProC-B0034 or ProD-B0034 directly upstream of the GFP coding sequence. ProC and ProD refers to the promoter, and B0034 denotes the RBS. Both reading values and the positive control subtract the values of negative control for a given cell density level in order to account for background fluorescence. The measured expression levels shown are all measured at the same cell density of $OD_{600} = 0.2$. The positive control ProC-B0034 on the medium-copy (MC) plasmid is set as value one and all other readings are presented as relative values to this reference.

2.4.2 Expression Levels

Figure 2.5 shows the relative expression level of different combinations of promoters and RBS on the medium copy plasmid. It shows that the expression level of RBS B0031 is consistently higher than RBS B0032. According to the RBS calculator B0032 should to have a higher TIR than B0031 [155] (1,134 and 670 respectively). Figure 2.6 shows the relative expression level on the high copy plasmid. The high copy values are measured relative to the same expression reference as the medium copy plasmid, in order to be directly comparable. The difference in copy number is realised through a single (G $\rightarrow$ A) basepair substitution in the origin of replication region. Figure 2.4 shows the difference in expression level from this basepair edit.

In addition to the values shown in Figures 2.5, 2.6 and 2.7 a few additional versions of RBS were tested together with promoter ProA on the medium copy plasmid. The aim was to search for very strong RBS. The RBS tested were first calculated with the Salis RBS calculator [155]. Each RBS variant in the project which was designed with help of the RBS Calculator (v2.0) has been given a name which contains its predicted Translation Initiation Rate (TIR).

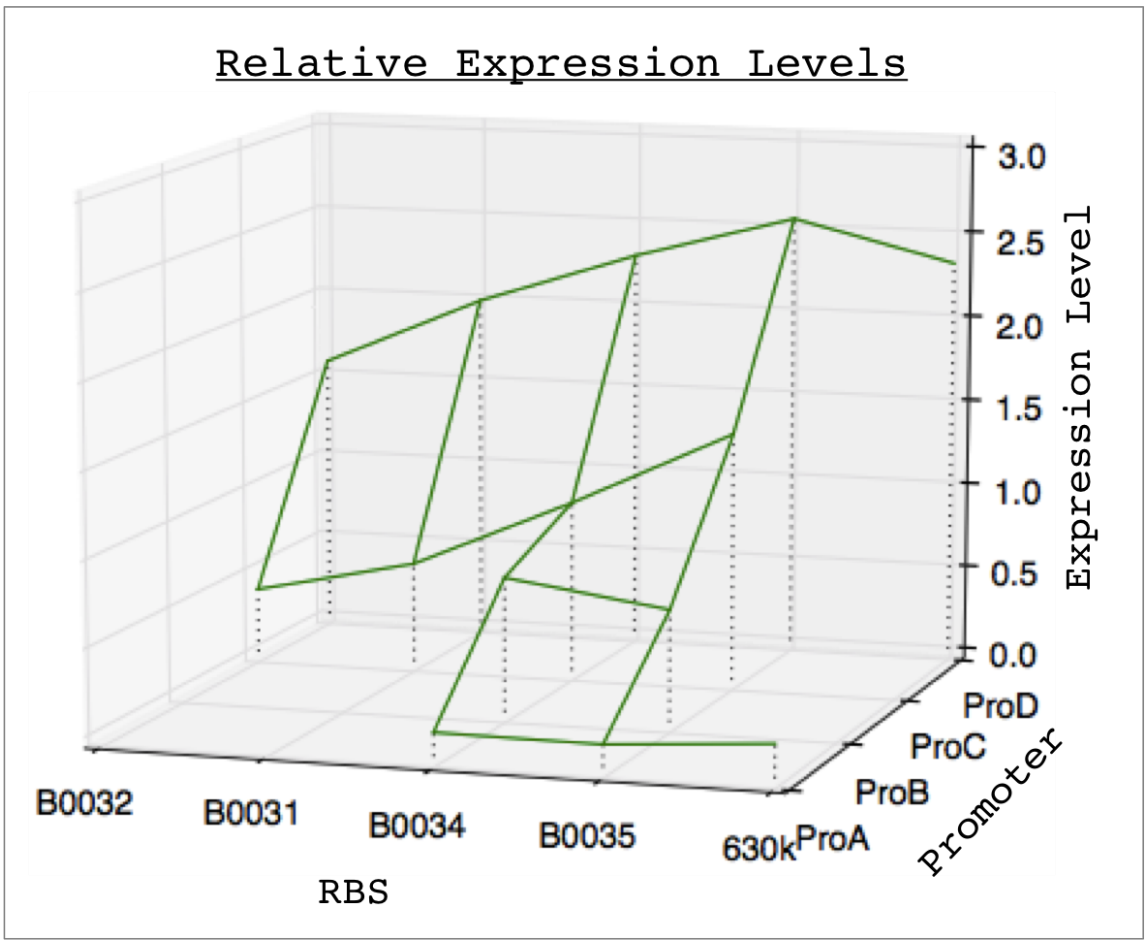


Figure 2.5: The expression levels for different combinations of RBS and promoters on a plasmid backbone with a medium copy number. Fluorescence outputs are measured when $OD_{600} = 0.2$, and normalised to ProC-B0034, which is given the relative value of 1.0. Standard deviations are shown in Figure 2.8.

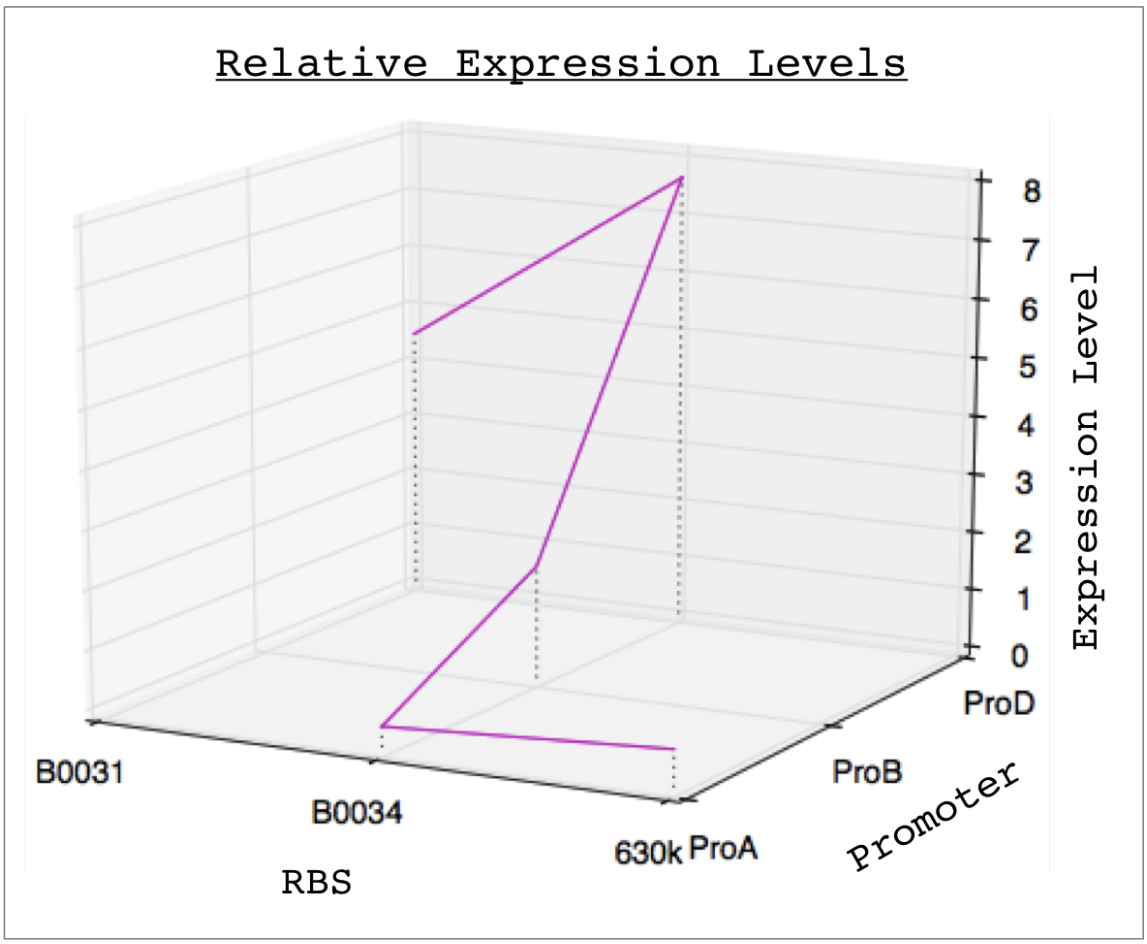


Figure 2.6: The expression levels for different combinations of RBS and promoters on a plasmid backbone with high copy number. Fluorescence outputs are measured when $OD_{600} = 0.2$, and normalised to ProC-B0034 on the medium copy plasmid, which is given the relative value of 1.0. Thus, the relative values given are comparable to the outputs in Figure 2.5. Standard deviations are shown in Figure 2.8.

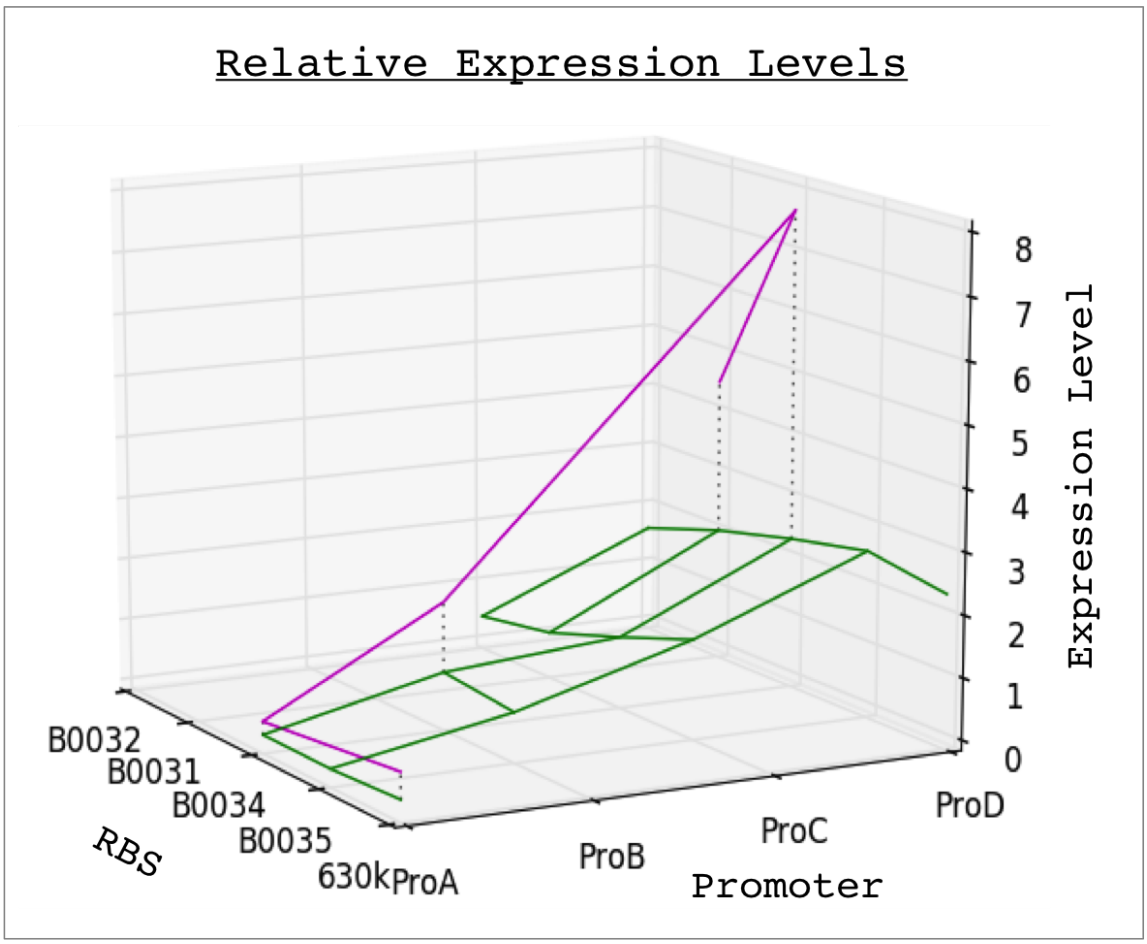


Figure 2.7: The combined expression levels for combinations of RBS and promoters on plasmid backbones with medium (Figure 2.5) and high copy numbers (Figure 2.6). Fluorescence outputs are measured when $OD_{600} = 0.2$, and normalised to ProC-B0034 on the medium copy plasmid, which is given the relative value of 1.0. Standard deviations are shown in Figure 2.8.

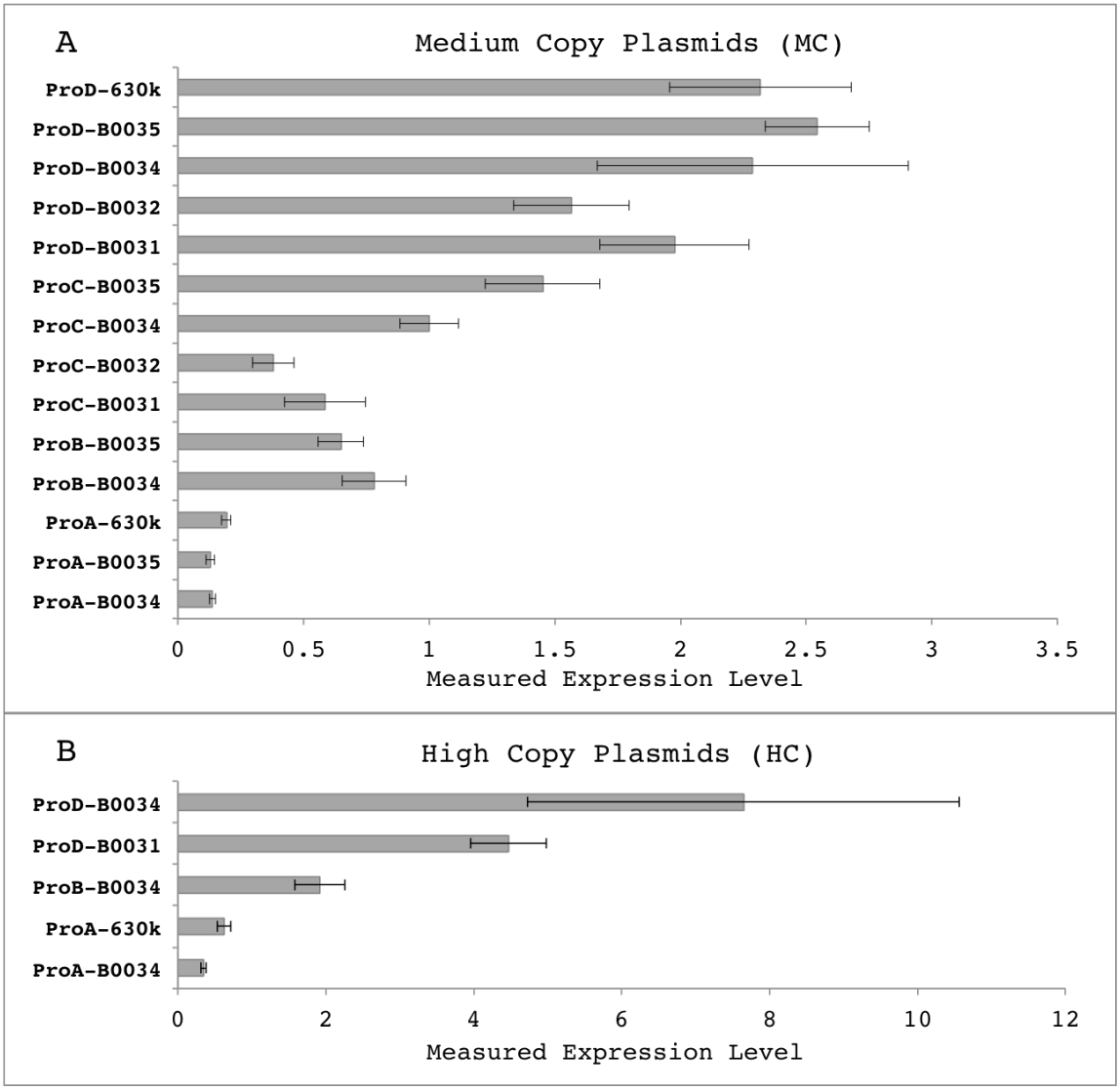


Figure 2.8: The values from Figure 2.7 plotted as two bar charts with error bars of one standard deviation. **A:** The medium copy plasmid expression level of the constructs in Figure 2.5 plotted as a bar chart with error bars. **B:** The high copy plasmid expression level of the constructs in Figure 2.6 plotted as a bar chart with error bars. Both bar charts are measured against the ProC-B0034 on a medium copy plasmid as the relative expression level equal to one.

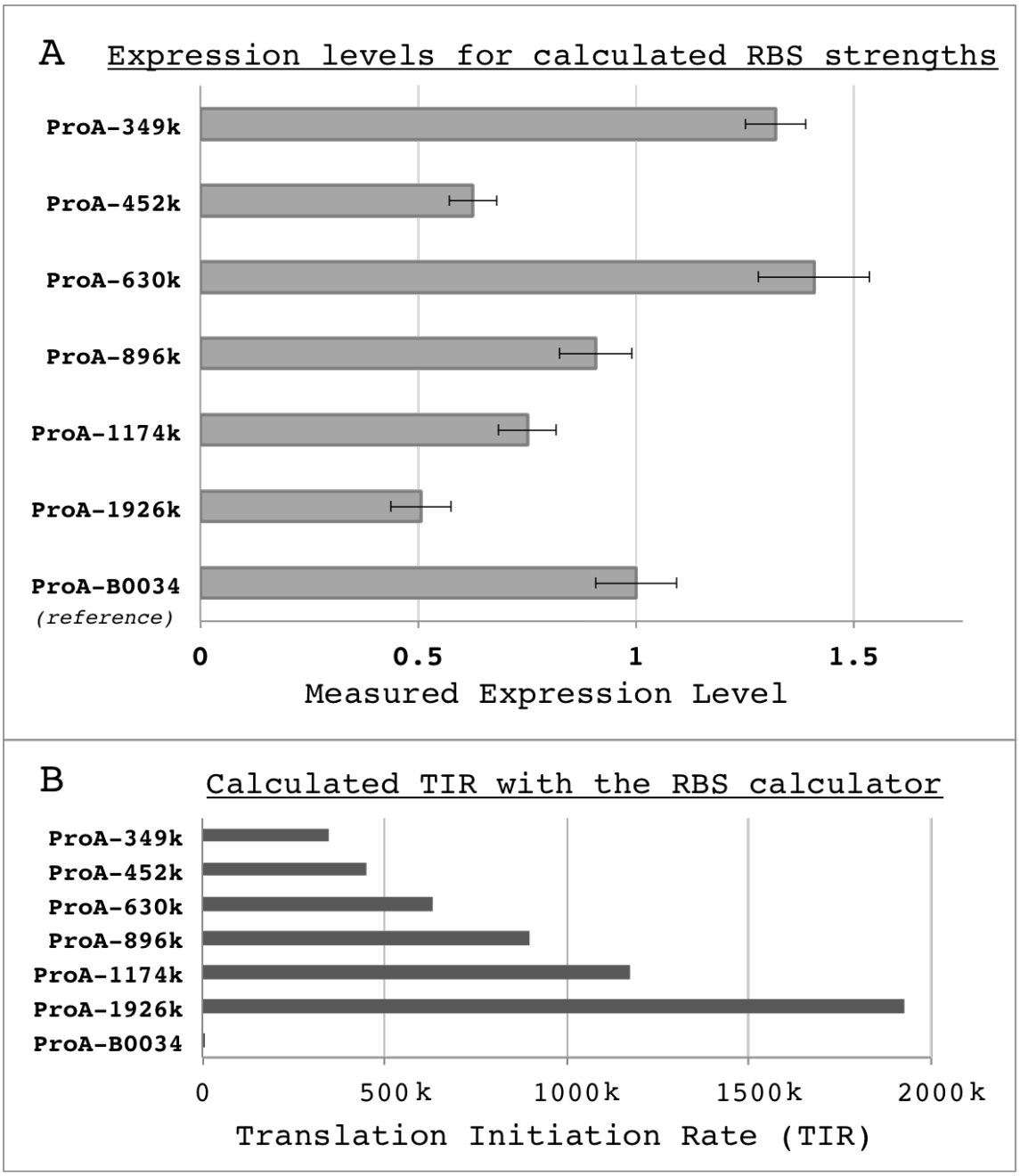


Figure 2.9: The measured expression levels for different RBS translation initiation rates that were calculated with the RBS calculator v2.0. Measured RBS versions are downstream of promoter ProA on a medium copy plasmid, and values are measured relative to ProA-B0034. **A:** The measured expression level from the different RBS, relative to ProA-B0034 = 1. **B:** The calculated TIR with the RBS calculator [155]. All values shown in kTIR (,000 TIR). The B0034 reference has a calculated translation initiation rate of 6.7k TIR. Note, some the RBS from this library were also tested with other promoters and plasmid copy numbers (*e.g.* the RBS 630k) but only ProA with a medium copy plasmid were used to test all of them. Therefore the promoter ProA on a medium copy plasmid is used to show the relative expression between the full set of the different RBS.

The RBS calculator was unable to predict the relative translation levels for the different RBS, as can be seen in Figure 2.9. The RBS-630k appears to be the strongest of the RBSs (Figure 2.9), whereas the strongest predicted RBSs (from the RBS calculator), 1174k and 1926k, instead produce two of the three weakest measured results. The RBS reference used (B0034) is predicted to have a TIR of 6.7k with the RBS calculator, *i.e.* far below the other values measured (shown in Figure 2.9B).

2.5 Library Growth Levels

To examine the effects on cellular burden from different designs, a range of design combinations were examined with respect to their impact on the rate of population growth. First the experimental procedure is described, followed by data obtained on growth effects.

2.5.1 Experimental Procedures

Recall the experimental protocol and definitions in Sections 1.3.1 and 1.3.3. Growth measurements are given as a relative value of cell growth after a single dilution of population samples into a fresh nutrient environment. Following the picking of colonies (of transformed populations containing engineered constructs) and growth in liquid media for more than 16 hours (referred to as ENV_1) the cells are diluted into fresh medium for renewed growth (referred to as ENV_2).

After a fixed time period of growth in ENV_2 the cell density of all samples is measured. The measured cell density is then divided by the initial cell density of each sample which gives a *fold increase* in cell density as a relative measure of growth. The calculated mean level of all fold increases is then compared against the fold increase of populations without construct expression. The relative fold increase between samples with and without construct provides the measured relative growth level provided by harbouring the construct. In other words, growth is quantified relative to the reading of a negative control. A relative, instead of an absolute, measure of growth was shown to provide more accurate comparisons between experimental runs. A fixed absolute and non-referenced value for growth can vary significantly, due to difficult-to-control technical variations between different experiment

realisations [96].

As discussed before, the initial population density measurements immediately following a dilution are inherently inaccurate to record due to their small value compared to technical measurement noise. Therefore, the last recorded cell density measurement prior to dilution has been used as the initial cell population density and divided by the dilution factor. After a fixed amount of time, the cell density achieved in the fresh medium is recorded and divided by the initial cell density value to obtain a fold increase in population size for the time duration. A measurement time-point of two hours after dilution was used for the data presented.

For an additional set of experiments, samples were diluted to a higher level and for those comparisons a measurement time-point of 3 hours instead of 2 hours was used. The 3 hour time-point for these experiments is because the higher dilution factor prolonged the time required to reach a comparable cell density of OD600 > 0.05. Lower OD measurements have a higher degree of measurement uncertainty. Comparisons were done between samples present in both the higher dilution experiments and in the lower dilution experiments, showing that the adjusted OD measurement between the two sets of data provided similar outputs. For example, ProC-34 present in both sets has a measured growth level of 0.88 ± 0.12 versus 0.83 ± 0.12 for the two different dilution levels, at measurement time-points 2 and 3 hours respectively.

Different versions of growth and burden measures

The benefit of recording a fold increase in cell populations over a time period, such as described above, is that it is fairly clear that the cells which have have larger fold increase also have had a faster overall growth during the time period. Another important measure of growth is the maximum growth rate (μ_{max}). However, the maximum growth rate of *e.g. E. coli* in rich media occurs at concentrations with very low optical densities [167]. At low optical densities the experimental readings are less reliable and more noisy. Chapter 3 of this thesis discusses further the topic of new potential measurement methods and metrics for burden quantification.

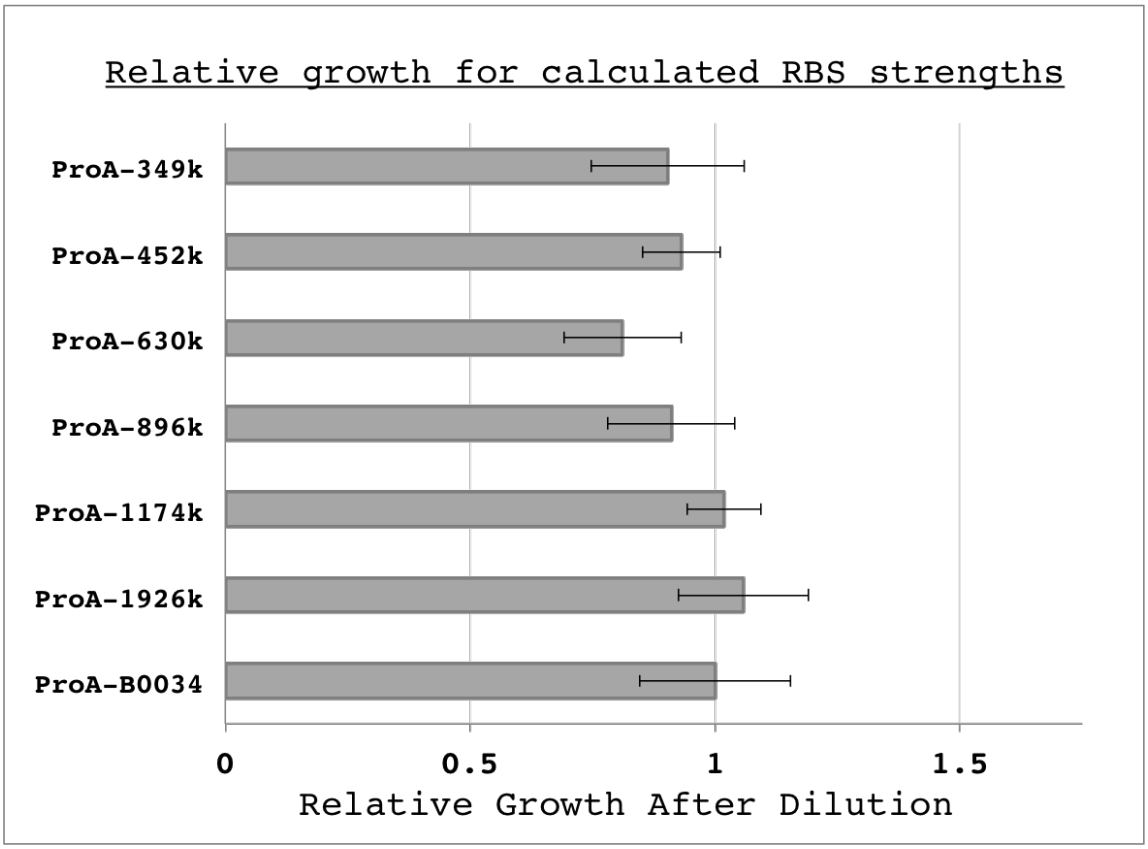


Figure 2.10: The measured growth levels for the different RBS that were calculated with the RBS calculator v2.0. All measured values shown have the same ProA promoter and are located on the medium-copy-number plasmid. The bottom bar for measured growth for ProA-B0034 is used as the relative reference growth value of 1.0, to which the other values have been compared. The measurements shown were made after a pre-incubation time of 18 hours before dilution.

2.5.2 Growth Levels

The B0034 reference RBS provided in Figure 2.10 has a translation initiation rate (TIR) of 6.7k according to the RBS calculator [155], *i.e.* with a good margin below the other predicted RBS strengths. However, as seen in Figure 2.9 in the previous section, the measured values from the RBS calculator correspond poorly with the measured expression levels. For the measured values (in Figure 2.9) RBS-630k corresponds to the highest expression level with the ProA promoter while RBS-1926k gives the lowest expression level. Nevertheless, as seen from Figure 2.10 (and compared to Figure 2.9), the actual measured expression levels correspond relatively well with the costs on growth from protein expression. The highest expressing construct ProA-630k has the highest growth cost whilst

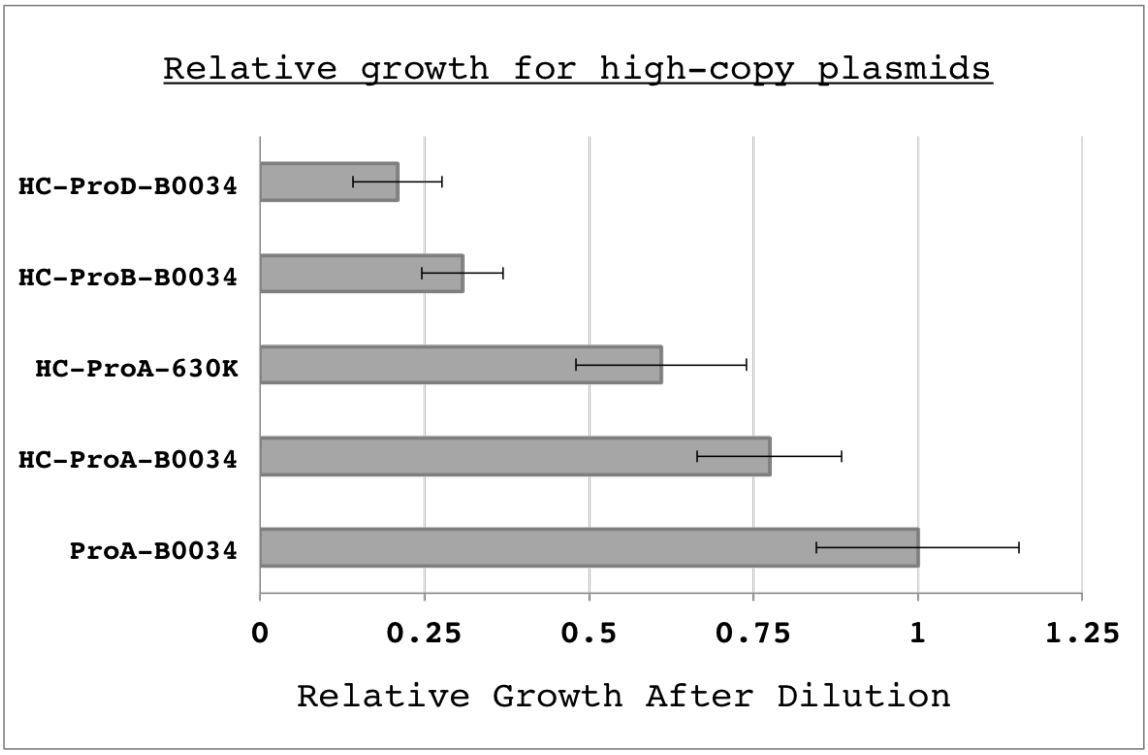


Figure 2.11: Measured growth levels for expression designs located on high-copy plasmids (HC). For comparison, the the bottom bar shows ProA-B0034 situated on the medium-copy plasmid. The measurements shown were made after a pre-incubation time of 18 hours before dilution.

the lowest expressing construct ProA-1926k also has the lowest cost on growth. Statistical tests between the measurements for these two RBS in Figure 2.9 gives a p-value of 0.0020 for Student's t-test and a p-value of 0.0093 for Welch's t-test. With either method the results are regarded as statistically significant (as well below $p = 0.05$).

Figure 2.11 shows measured growth for constructs on high-copy plasmids, and 2.12 shows growth levels for medium-copy plasmids.

The difference in relative growth measured after a short or a long incubation time (see Figure 2.13) is likely due to the longer time needed for cells to adjust to growth after a longer incubation time. In other words, that the negative growth effect from readjustment is amplified by hosts simultaneously expressing excess proteins. It was previously shown that host cells harbouring non-needed proteins have negative growth effects amplified during growth initiation, while costs are later reduced during more stable growth conditions [168]. It can be assumed that populations which have been incubated for a shorter time period,

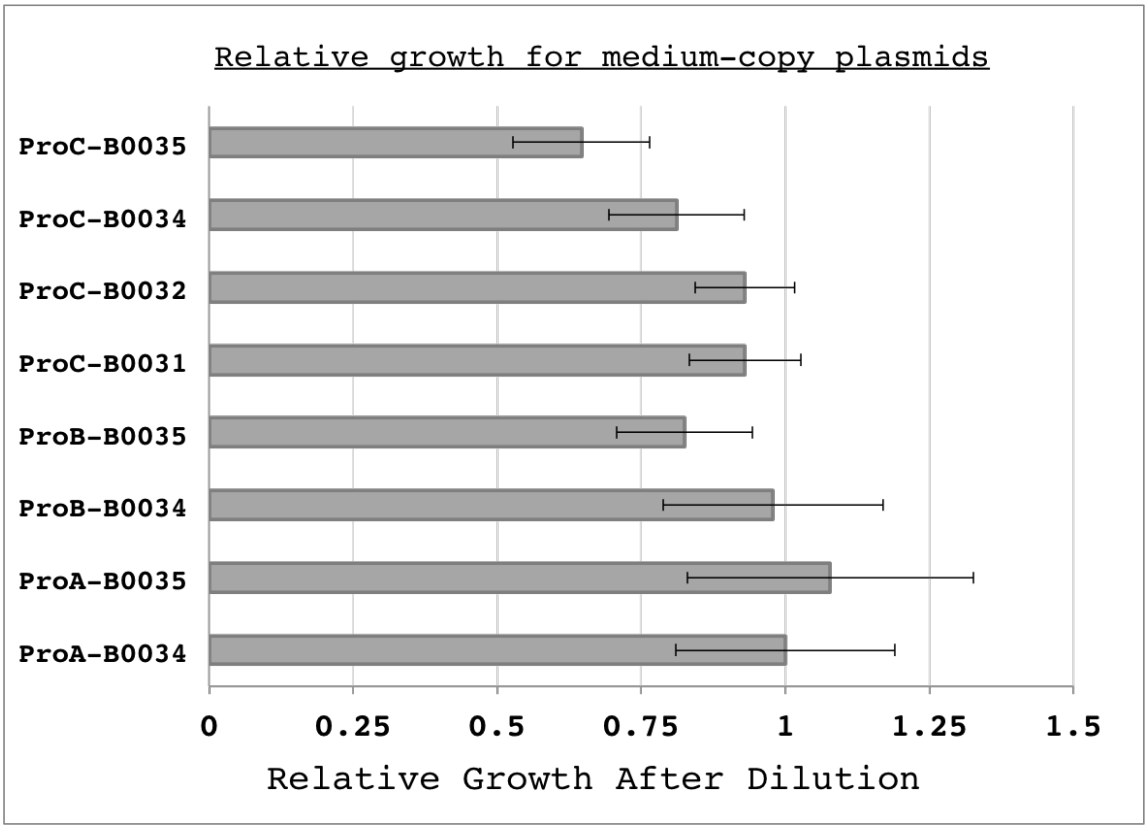


Figure 2.12: Measured growth level for library constructs on medium copy plasmids after a pre-incubation time of 16 hours. ProA-B0034, on the bottom bar, is given the relative value of 1.

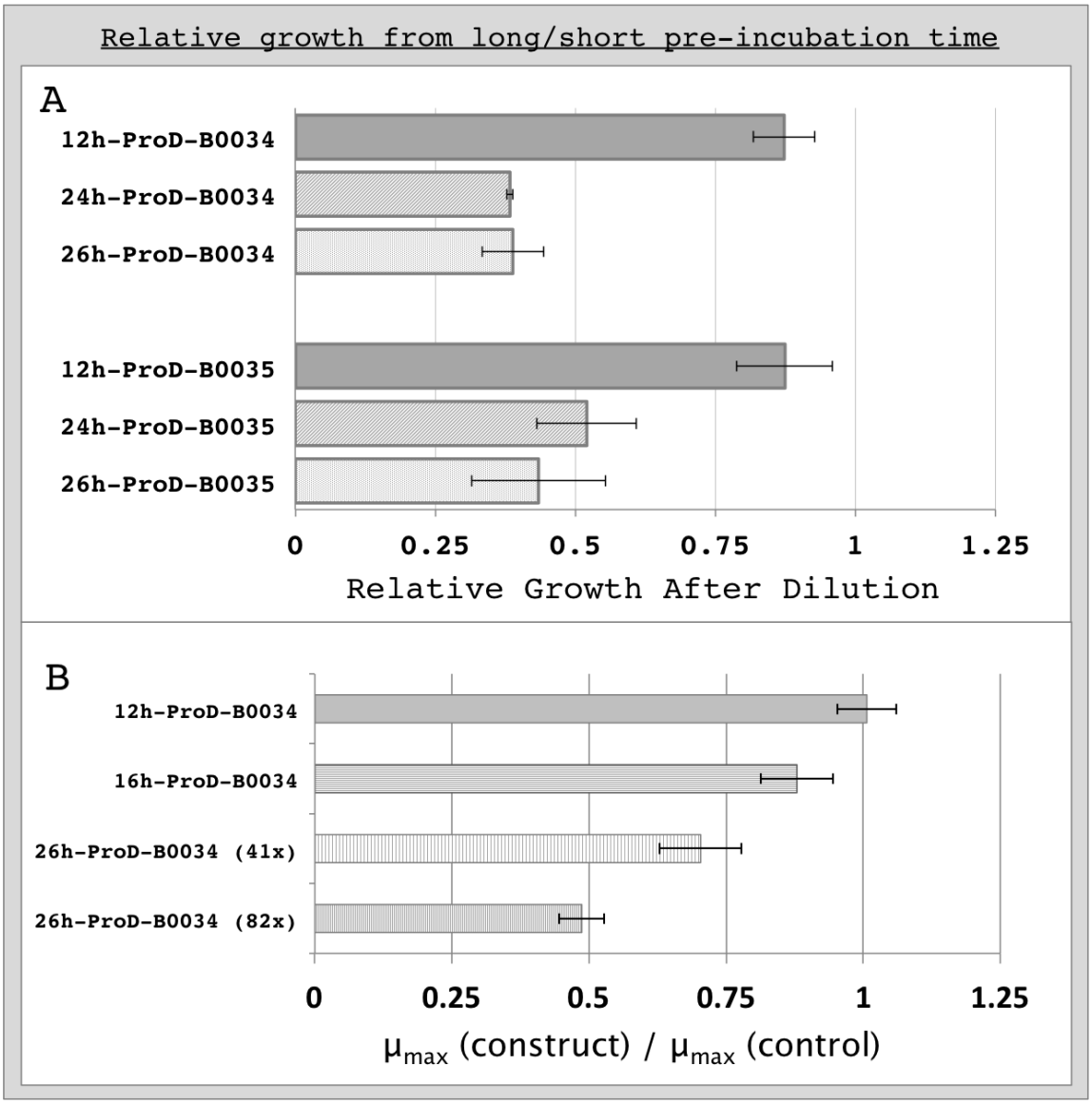


Figure 2.13: The difference in measured relative growth level resulting from short versus long pre-incubation time. The same construct, measured in comparison to a non-expressing negative control, shows a much reduced relative growth after a short pre-incubation time before dilution, compared to a larger relative growth reduction after a long pre-incubation time. *I.e.* the relative growth cost resulting from protein expression is increased after a long pre-incubation time and reduced after a short pre-incubation time before dilution. **A:** The three timespans shown and compared on the plot are 12 hours versus 24 and 26 hours of pre-incubation for the same constructs. **B:** The maximum growth rate of the same construct after three different length of pre-incubation in ENV_1. The maximum growth rate is measured as relative to the maximum growth rate of populations without construct expression. (Maximum growth rate calculations are shown in Figures 2.15 and 2.16).

before dilution to a fresh medium, have a shorter readjustment period to continued growth, compared to populations which have grown for a longer time period in the same media and depleting most of the nutrients. It has also been shown, on an individual cell basis, that cells which enter the stationary phase earlier are the last ones to leave the stationary phase when encountering more favourable growth conditions [93]. From Figure 2.13 it could be interpreted that during cell readjustment to growth in a more favourable environment, the excess expression of non-essential proteins *e.g.* engineered constructs, will compete with translation of the proteins needed in the new environment and delay the adjustment to continued growth.

2.6 Relating Model to Data

2.6.1 Relative Growth Level and Expression

Figure 2.14 shows the correlation between measured relative growth and expression. The growth-law, as well as the RBS burden model described in the beginning of this Chapter relate expression to the maximum growth rate (μ_{max}). However, the growth data shown in Figure 2.14 are describing relative growth levels after a certain time period, which is not directly comparable to μ_{max}. The next subsection continues to discuss measurement metrics in order to the link the presented model to experimental measurements.

2.6.2 Growth Rates and Fraction Sizes

The model in this chapter is derived by first linking the size of the ribosome fraction to the growth rates of populations (Equation (2.2)). The cost of engineered protein expression then follows from the reduced size of the ribosome fraction due to a competing fraction size of engineered proteins [162]. As growth rates of cells are first increasing and then decreasing when placed in environments with increased nutritional capacities the maximum size of the ribosome fraction (Φ_R) is given at the maximum growth rate. The growth rates are decreasing as the nutrients are depleted in the environment. If the size of the ribosome fraction is reduced by a competing fraction of the engineered fraction this is then manifested by a reduction in the maximum growth rate (μ_{max}). As a side point, after the growth rate has peaked in an environment subjected to nutritional decline (due to consumption from the growing population) the growth rate becomes limited by Φ_M instead of Φ_R.

The growth rates over time are measured as the rate of biomass increase divided by the biomass

$$\mu = \frac{dOD}{dt}/OD,$$

or equivalently

$$\mu = \frac{d(\ln(OD))}{dt},$$

where OD is the optical density of the bacterial population. In this case the optical density is measured as the absorbance from light with a wavelength of 600 nanometers.

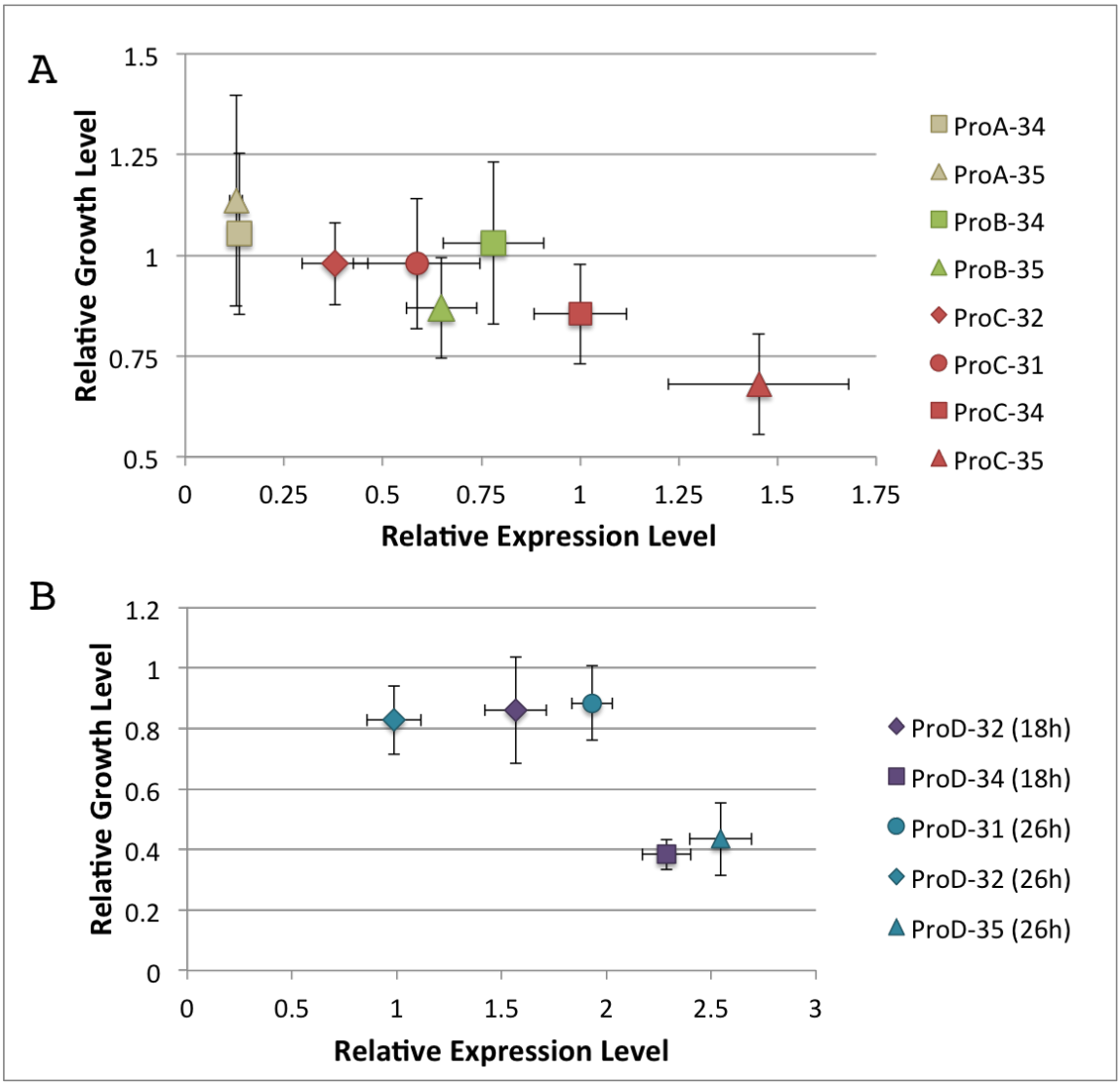


Figure 2.14: Growth levels linked to expression levels. **A:** Growth and expression for different constructs after 16 hours of pre-incubation (ENV_1). ProC-B0034 is given the relative expression value = 1. (Growth values can also be found in Figure 2.12, and expression values in Figure 2.5.) **B:** Growth and expression for the ProD promoter, after 18 and 26 hours of pre-incubation (ENV_1). ProD-B0034 is given the relative expression value = 1.

In order to fit the sequences of experimental data readings to extract growth rates a large number of growth curves have been developed which allow to extract growth parameters from full sets of reading points [211]. Due to the technical data variance of optical density readings, the fittings of growth curves (to longer sequences of data readings) are more reliable than measuring the maximum derivative between a small number of data points [31]. This is especially true for the low optical densities where the maximum growth rate occurs [167], as the measurement errors are then large in comparison to the measurement values. The most popular growth model for fitting of bacterial growth [211, 22, 154] is the Gompertz curve [76]. The original Gompertz model is given by

$$y = a \cdot exp[-exp(b - ct)],$$

where y is given as the natural logarithm of the population size x a time t divided by the original population size, *i.e.* $y = \ln(x_t/x_0)$. Mathematical inputs to the model are given by a, b and c to provide a sigmoidal growth curve [211].

A modified version of the Gompertz model that directly identifies the key relevant parameters for growth curves of microorganisms can be written as [211],

$$y = A \cdot exp\Big\{ - exp\Big[\frac{\mu_m \cdot e}{A}(\lambda - t) + 1\Big]\Big\}, \tag{2.13}$$

where μ_m is the maximum growth rate, A is the asymptote, and λ is the lag-time. The output value y is still the natural logaritm of the population size $y = \ln(x_t/x_0)$. Growth curves and the Gompertz model are discussed again with a bit more detail in Chapter 3 of this thesis.

The Gompertz model was used to extract growth parameters and μ_{max} from the different experiments and constructs in this chapter. Figure 2.15 shows the Gompertz curve fitted growth of two different samples; (i) hosts with a ProD-B0034 construct on a medium copy plasmid and (ii) control samples hosting a medium copy plasmid without GFP construct expression. The growth data is measured from ENV_2 after 26 hours of pre-growth in ENV_1 (*i.e.* growth after picking from transformed colonies). The x_0 value in the model are taken as the last measurement before dilution in ENV_1, divided by the dilution factor

(*e.g.* here a 41 fold dilution factor for the data in Figure 2.15). Measurement values for μ_{max} of different constructs fitted with this method are shown in Figures 2.17, 2.18 as well as 2.13B.

2.6.3 Engineered Fraction Sizes: Φ_E

The model, as written in Equations (2.10), (2.11) and (2.12) is linked to a measurement of the engineered protein fraction Φ_E. According to the plots in Figures 2.2 and 2.3 it is not necessary to measure and quantify the exact sizes of Φ_E. But rather a relative measurement of Φ_E suffices, in order to measure the changes in growth rate (μ) as an effect of relative changes in a the engineered fraction Φ_E. The construct used in the experiments is limited to GFP expression, such that a relative measure for construct expression can be recorded as an output in fluorescence. The relative measure of Φ_E for a construct is recorded by measuring the fluorescence of a cell population, divided by size of the population biomass, and subtracting the background fluorescence of cells without GFP expression,

$$\Phi_E \propto \Big(\frac{GFP_{construct}}{OD_{construct}} - \frac{GFP_{cells}}{OD_{cells}}\Big) \tag{2.14}$$

where GFP refers to fluorescence measures (with excitation wavelength of 475nm and emission wavelength of 509nm), OD is the optical density at 600nm. The subscript *construct* refers to samples hosting the construct and the subscript *cells* denotes the background fluorescence from control cells without any GFP expression.

In Figures 2.17B and 2.18B the expression of GFP per OD as a relative measure of Φ_E is recorded at the point in time for the maximum growth rate. Figure 2.15C shows the GFP expression over time for a sample construct (ProD-B0034 on a medium-copy plasmid). The maximum growth rate of each sample defines the time point for recording GFP/OD (shown in Figure 2.15B).

The data recordings for fluorescence has lower noise variance than the absorbance readings for optical density. However, the fluorescence values are also fitted to a curve to

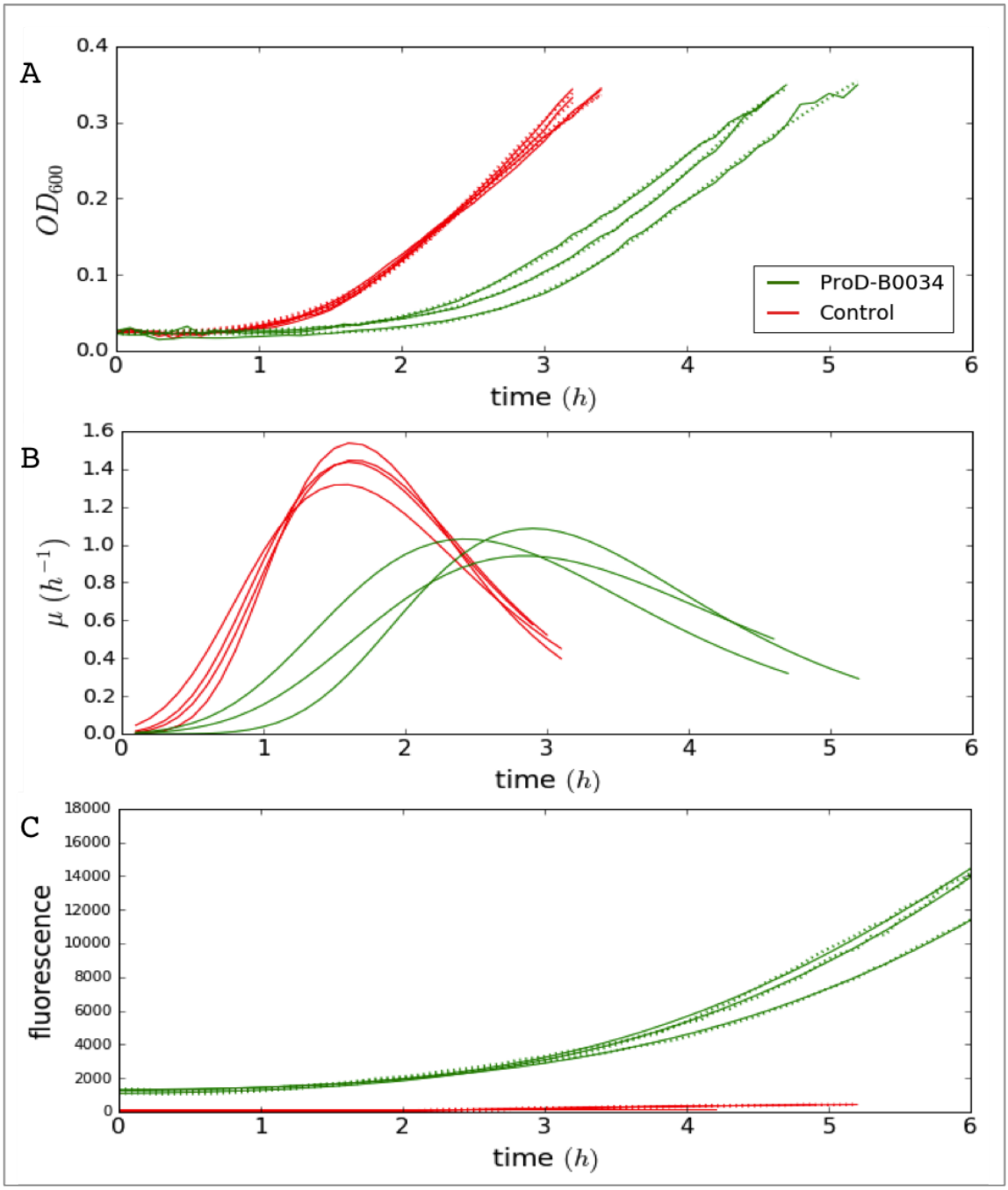


Figure 2.15: Maximum growth rates (μ_{max}) and GFP fluorescence from negative control (red) and ProD-B0034 (green) after 26 hours of pre-growth in ENV_1. **A:** The optical density of control without a construct (red dashed) and ProD-B0034 (green dashed), fitted to the Gompertz curve (black line). **B:** The time points for the maximum growth rates are obtained from the derivative of the fitted log(OD) values in A. **C:** The relative expression values for a construct are obtained from fitting the fluorescence to a Gompertz curve and taking the value of the fitted curve at the time point equal to the maximum growth rate obtained from B. Two different values for expression are calculated with this method. Firstly, $\Phi_E \propto GFP/OD$ which is taken as the fluorescence level at μ_{max} divided by OD at the same μ_{max} time point, and subtracting the same value for the negative control (*i.e.* the background fluorescence of cells). Secondly, $R_E \propto dGFP/dOD$, which is calculated as $(\frac{dGFP}{dOD}/OD)/\mu_{max}$, also at the time point for μ_{max} from subfigure B and subtracting the rate of background fluorescence from the negative control.

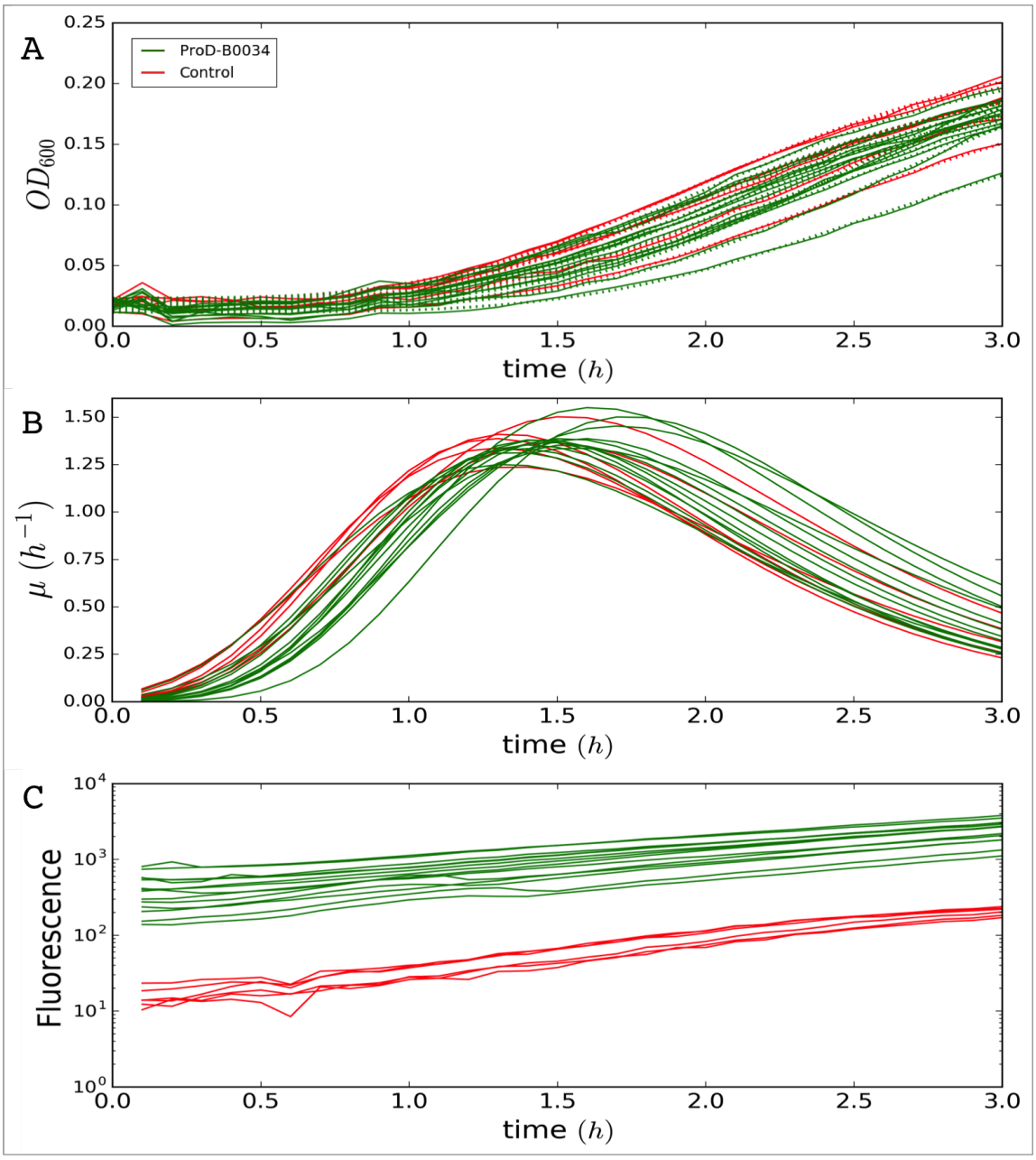


Figure 2.16: The growth rate for ProD-B0034 after 12 hours of pre-growth in ENV_1 and a 16 fold dilution to ENV_2. The mean of the maximum growth rates in this plot are shown as a bar chart value in Figure 2.13B. **A:** The optical density of control without a construct (red dashed) and ProD-B0034 (green dashed), fitted to the Gompertz curve (black line). **B:** The time points for the maximum growth rates are obtained from the derivative of the fitted log(OD) values in A. The mean μ_{max} of ProD-B0034 is 1.38 h^{-1}. The mean μ_{max} of the negative control is 1.37 h^{-1}. The standard deviations are 0.074 and 0.080 h^{-1}, respectively. The μ_{max} values are obtained directly from fitting of Equation (3.1), while subplot B shows them visually over time. **C:** The fluorescence of constructs and negative control. The plot shows that constructs expression is intact for all construct samples.

reduce the noise in data readings. They are fitted to a generic gompertz curve without the logarithm of the recordings *i.e.* $y = GFP$ fluorescence, instead of $y = \ln(x_t/x_0)$. The dashed lines in Figure 2.15C show measurement values and the solid lines show curve fits. The solid fitted curves from the recordings in Figures 2.15A (OD) and 2.15C (GFP) are used to calculate the GFP/OD at the time of μ_{max} from 2.15B (which is obtained from the derivative of the fitted curve in Figure 2.15A). However, the μ_{max} value itself is extracted directly from the equation of the fitted curve in Figure 2.15A, as shown by the Equation (3.1) used for the fitting. The derivative of the fitted Gompertz curve for cell mass is only used to identify the time point for μ_{max}.

2.6.4 Engineered Fraction Sizes: R_E

The model derivation in this chapter assumed a steady state between protein fraction sizes. In other words, that the fraction of protein mass production is equal to its current mass fraction size. This is assumption is noticeable in Equation (2.8):

$$\Phi_x = \frac{\gamma_x \Psi_x}{\sum_{x \in \{R,..,E\}} \gamma_x \Psi_x}$$

which assumes $d\Phi_x/dt = 0$. However, if protein fractions are not in steady state then a different symbol can be useful to annotate the mass fraction of protein production of each fraction. The symbol R_x is used to show the fraction of protein mass production of protein fraction x,

$$R_x = \frac{\gamma_x \Psi_x}{\sum_{x \in \{R,..,E\}} \gamma_x \Psi_x}$$

and as stated above (as well as assumed in the model derivation) during steady state of growth $R_x = \Phi_x$. A different way of writing R is

$$R_x = \frac{\gamma_x}{\gamma_{total}} \Psi_x$$

as Ψ_x is the fraction of ribosomes bound to mRNA from fraction x and γ_x/γ_{total} is its relative rate of translation. This relationship can also be seen from Equation (2.6).

Engineered Fraction Sizes: R_E from Data

The rate of engineered protein production is proportional to the derivative of GFP production. Also the total rate of protein production is proportional to the derivative of cell mass production (assuming a constant ratio of proteins to cell mass). The relative measure from data outputs which informs what proportion of ribosomes are occupied by translating a GFP construct instead of translating native biomass is given by

$$R_E \propto \frac{dGFP}{dt} / \frac{dOD}{dt},$$

which is equal to

$$R_E \propto \frac{dGFP}{dOD},$$

or also

$$R_E \propto \frac{dGFP/dt}{OD} / \mu, \tag{2.15}$$

where $\frac{dGFP}{dt}$ is proportional to constructs being translated by ribosomes, and $\frac{dOD}{dt}$ (or $\mu \times OD$) is proportional to the total protein biomass production translated by ribosomes. In other words, the biomass production of the construct divided by the total biomass production is proportional to the mass fraction of engineered protein production. Same as for the Φ_E fractions measure, the rate of fluorescence change from the background fluorescence without a construct is also subtracted from the output value.

Figures 2.15C and 2.16C show how the GFP data is fitted to a Gompertz curve. The $dGFP/dt$ rate is then obtained from the derivative of this fitted curve. The relative R_E values which are extracted from the experimental data are Equation (2.15) at the point in time when $\mu = \mu_{max}$, *i.e.* the same point in time which is used for the relative measure of Φ_E. Relative Φ_E and R_E fractions for different constructs are shown in Figures 2.17 and 2.18. The relative measure = 1 is given to the ProD-B0034 construct on a medium-copy plasmid. It can be seen from the scatter plots that the Φ_E values are not necessarily equal to R_E values, especially for the high-copy number plasmids. It is understandable that $\Phi_E \neq R_E$ for these experiments, as the growth rates are not at a static state but are changing over time (Figure 2.15B), and thus the fraction sizes are also in a state change

$\frac{d\Phi_x}{dt} \neq 0$. Chapter 3 goes back to discuss changes in growth phases from a perspective of burden, with more focus on the mass production rate fraction R.

2.6.5 Model versus Data

For fitting Φ_E from the model to data there are two measures which are interesting. If we assume a static environment and growth phase, such as the 'growth-law' then the concentration of the construct in the cells is an appropriate measure, for example GFP/OD. However, if the GFP concentration of the cell is shifting with time (*i.e.* $dGFP/dOD \neq 0$) then R_E at the point in time for μ_{max} becomes a better metric for the model Φ_E. The R_E measure (here $R_E \propto dGFP/dOD$) tells us what fraction of translational capacity is dedicated to Φ_E production at the point in time when most resources are needed by the cell to increase ribosome production (at the time for μ_{max}). On the other hand, $\Phi_E \propto GFP/OD$ could potentially give us a slightly distorted measure of the residual engineered fraction which has been expressed earlier in time, for example during a proceeding near-stationary-phase in ENV_1 prior to dilution. Figures 2.17 and 2.18 show both different metrics against the maximum growth rate for a range of different constructs after 26 hours of pre-growth in ENV_1 and after either 41 fold (Figure 2.17) or 82 fold dilution (Figure 2.18).

From the data obtained and shown it is difficult to a make a clear link between the model and the data measures. It seems as if none of the constructs represents a clear linear relationship between growth costs and construct expression expression as predicted by the static 'growth law'. Rather the entire samples as a group seems to follow an expression to growth relationship as predicted by the model for constructs with a reduced translational efficiency (model predictions shown in Figure 2.2). There is some indication from the data that the two highest RBS strengths calculated with the 'RBS calculator' (RBS-1174k and RBS-1962k) are indeed causing reduced translational efficiency of the construct, as indicated from higher cost relative to construct expression from these RBS. However, future additional experimental tests would need to be conducted in order to confirm that trend of the data for different setups and promoters.

One thing which can be observed from Figures 2.17 and 2.18 is that cells with high-copy number plasmids (data points with black frames) tend to have a higher relative construct

concentrations (Φ_E) at their peak growth rate compared to their actual expression of constructs at these growth rates (R_E). This observation could have a range of different plausible explanations. For example plasmid concentrations which are carrying the constructs can be higher at near stationary phase, and then at higher growth rates plasmids are replicating at a reduced rate compared to the host genome which would lead to reduced rate of construct expression during faster growth rates, *i.e.* simply due to relatively fewer copies of the construct DNA during these faster growth rates.

In the same manner that the fraction of the engineered fraction (Φ_E) is not necessarily equal to the mass fraction produced of the engineered fraction (R_E), but instead can vary as the growth rates are shifting, it is also intuitive that the protein fraction and mass expression fraction for the ribosome fraction would be non equal as growth rates are changing. Chapter 3 of this thesis continues to investigate changing growth phases with a focus on the relationship between R_R and Φ_R.

Future Improvements

There are a few possible reasons as to why the experimental data does not show sufficient evidence to prove or disprove the model. Firstly, a lot more constructs could be tested, or also a lot more experiments could be conducted with the exact same specification. For the total number of constructs which were designed and tested a range of different experimental setups were also tested in order to improve the data outputs and reduce measurement noice. However, as can be seen from Figure 2.13 as well as from comparison of Figure 2.15 and Figure 2.16, the data alignment between different experiments was more difficult than expected.

In addition to these rather obvious points of enhanced data collection (more constructs and more experiments) two good experimental changes to try in the future would be to (i) use codon de-optimised coding sequences for the construct and [78] (ii) use different media with slower growth [168]. Using de-optimised codons for the expression will reduce the translational rate of the construct mRNA and increase the effect 'overinitiation' from excessive RBS strengths. Less rich media might prolong the time periods of each growth phase such that more data points can be collected for each phase and thus increases the

accuracy of readings of growth (particularly the length of the growth phase for μ_{max}).

Clearly, it is also possible to try to tweak the parameters of the current experimental setup further, for example the duration of growth in ENV_1 or the dilution fold to ENV_2. But a large number of tweaks have also already been attempted during the development of the current experiments.

Thirdly, when trying to apply methods which are pushing the boundaries of technology or are at the forefront of what is currently possible or established there can often be implementation issues. In the case here, the technologies at boundaries of what is possible to implement today are those of remote and connected cloud laboratories. The purpose of applying such tools are to improve the standardisation as well as repeatability of biology (discussed with more emphasis in Chapter 1). However, it might be that for such subtle recordings as the growth rate changes between constructs of different RBS the quality of the machine protocols with regards to the calibration of the robotic setup is simply not ready yet. There were a few things with this method which was not expected when conducting the experiments. Firstly, the measurement outputs often carried unexpected errors from protocol interruptions and often needed to be implemented many times before completion. Secondly, some control is missing with regards to the sequence of steps for experiments (for example the lack of branching logic as mentioned in Chapter 1). Thirdly, that experiments with these methods required much longer time to implement than anticipated (primarily due to queuing and serialisation of experiment implementation runs instead of potentially more efficient parallelisations).

There are two things which could improve these potential issues for the future (i) the presence of more API cloud-laboratory suppliers, such that process and tools can improve naturally through competition between the different actors (ii) more users of cloud-laboratory setups which are writing code for specific experimental applications, such that these techniques can be further refined, exchanged and standardised for cloud applications through the course of time.

Finally, it also possible that improved measurement metrics for burden could enable the detection of more growth features linked to the cost of engineered protein production, or else better enable the comparison of different types of costs, constructs or growth conditions.

Chapter 3 of this thesis examines the scope for such metrics.

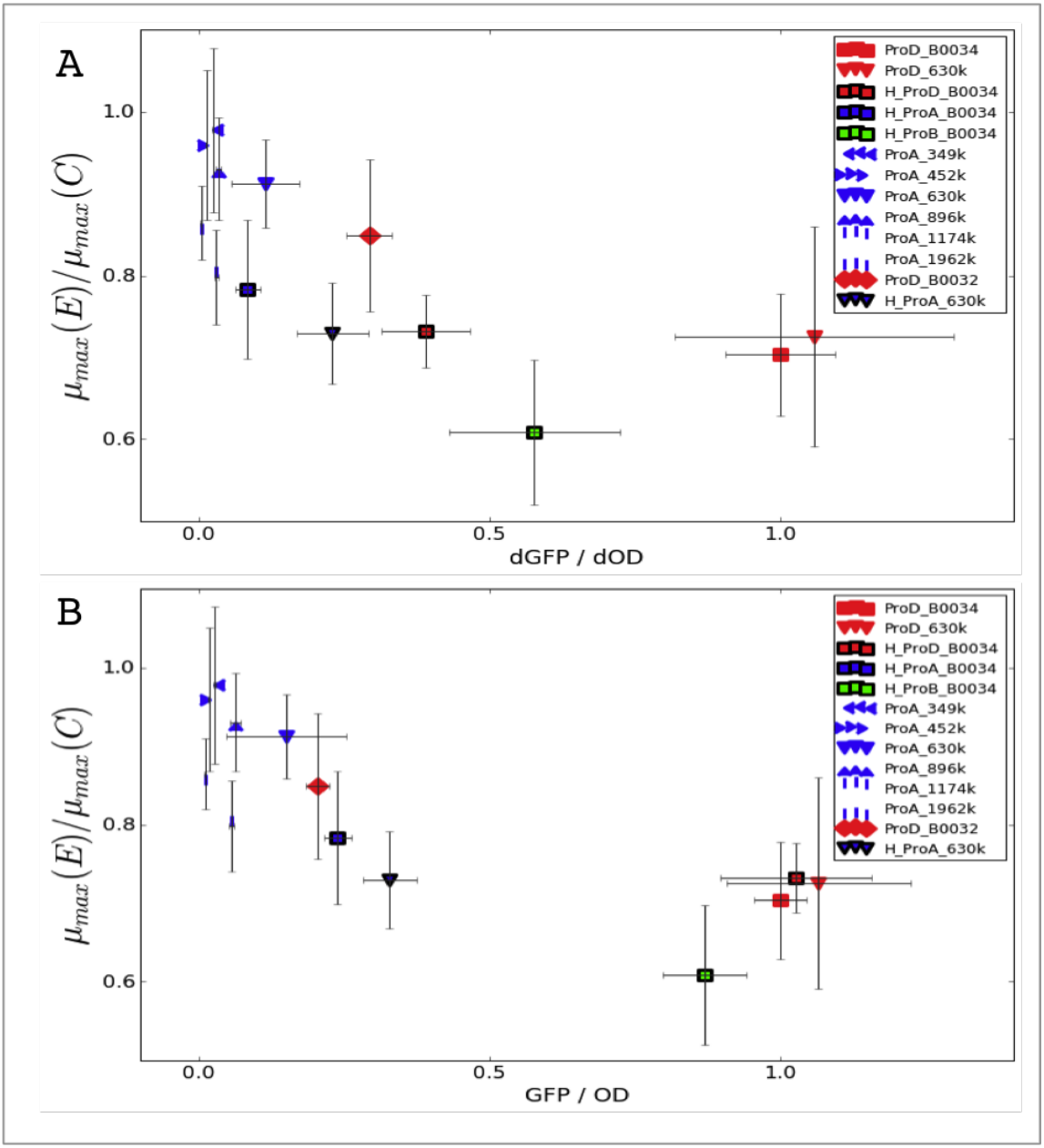


Figure 2.17: The normalised maximum growth rate μ_{max} against construct expression level after 26 hours pre-growth in ENV_1 and 41 fold dilution to ENV_2. The normalised maximum growth rate is calculated by dividing the maximum growth rate of the populations with a construct ($\mu_{max}(E)$) with the maximum growth rate of a negative control without construct expression ($\mu_{max}(C)$). The negative construct without the construct has a medium-copy plasmid without any DNA for construct expression. Constructs on a high-copy plasmid (**H**) are shown by markers with thick black frames. Constructs on a medium-copy plasmid are indicated by markers with no frame. The expression levels of ProD-B0034 is given the relative strength = 1 (for either of the two different expression measures in A and B). **A:** Growth rate versus expression level, where the expression level is measured as $dGFP/dOD$, *i.e.* proportional to R_E. **B:** Growth rate versus expression level, where the expression level is measured as GFP/OD, *i.e.* proportional to Φ_E.

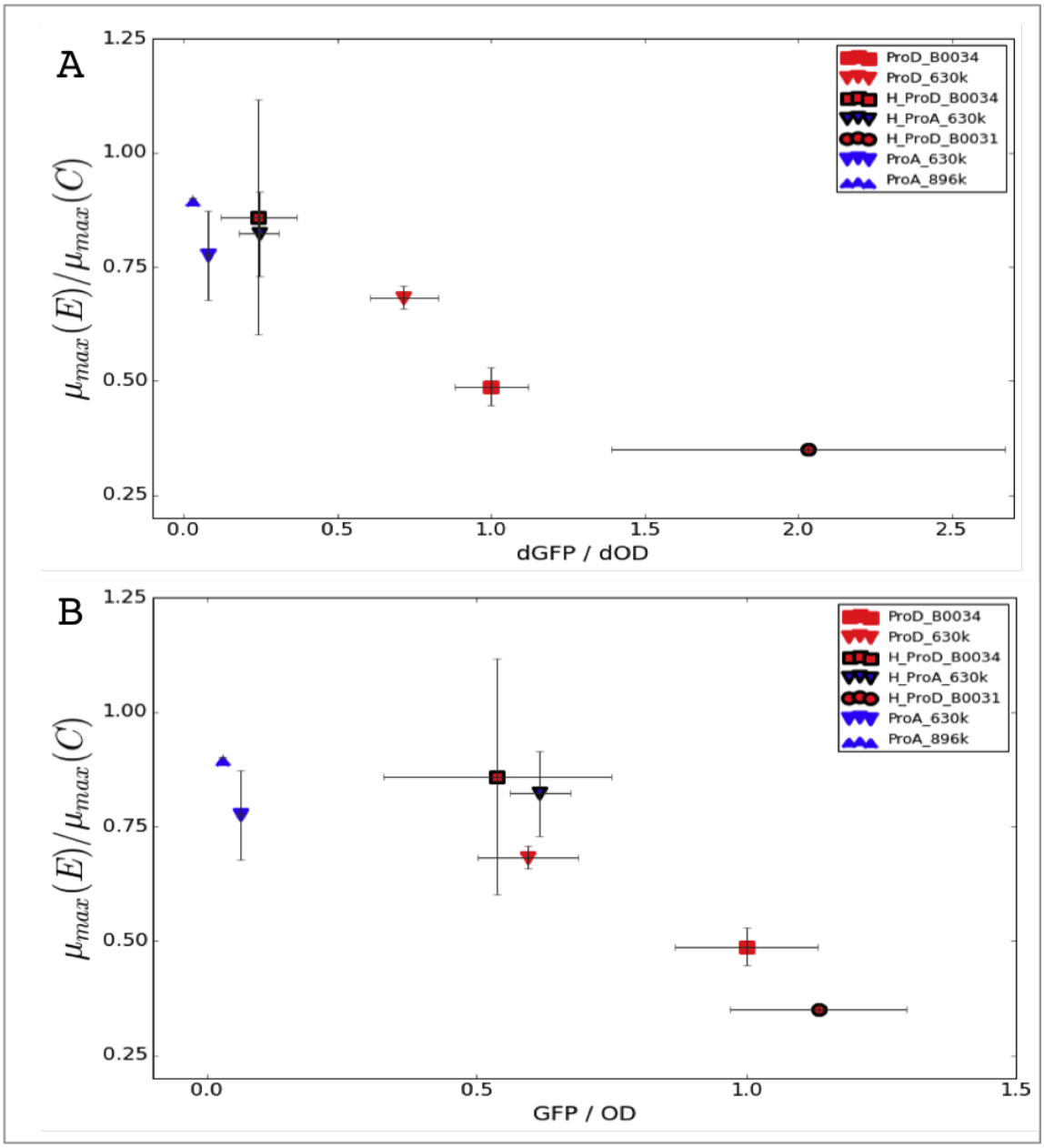


Figure 2.18: The normalised maximum growth rate μ_{max} against construct expression level after 26 hours pre-growth in ENV_1 and 82 fold dilution to ENV_2. The experimental setup is identical to the values shown Figure 2.17 except that the populations are diluted 82 fold instead of 41 fold, post the 26 hours pre-growth in ENV_1 The normalised maximum growth rate is calculated by dividing the maximum growth rate of the populations with a construct with the maximum growth rate of a negative control without construct expression. The negative construct without the construct has a medium-copy plasmid without any DNA for construct expression. Constructs on a high-copy plasmid (**H**) are indicated by markers with thick black frames. Constructs on a medium-copy plasmid are shown by markers with no frame. The expression levels of ProD-B0034 is given the relative strength = 1 (for either of the two different expression measures in A and B). **A:** Growth rate versus expression level, where the expression level is measured as $dGFP/dOD$, *i.e.* proportional to R_E. **B:** Growth rate versus expression level, where the expression level is measured as GFP/OD, *i.e.* proportional to Φ_E.

2.7 Conclusions and Contributions

Chapter 2 has evaluated burden parameters which are relevant for design of Synthetic Biology constructs. It has developed a model which aims to predict the fitness costs on the host from the translational efficiency of engineered components. The model has been developed in order to detect and inform when excessive RBS strengths are used in engineered biology. Excessive RBS strength leads to 'overinitiation' which occupies ribosomes at a reduced translation velocity [37]. Many cases of linear relationships between protein expression and fitness cost have been reported [162, 163, 24, 58]. However, non-linear relationships have also been reported [37, 54, 35] as well as nonlinear costs in time as an effect of growth rate transition periods [168]. The developed RBS-translation-rate model predicts a linear relationship to protein expression when the translation efficiency of the construct is high, and when it is not subjected to overinitiation. At the same time it predicts a non-linear relationship for expression if the translation efficiency for the construct is below the mean translation rate of native proteins. The fitness load associated with expressing an engineered construct determines an upper limit to its genetic stability. Therefore, the identification of non-linear relationships of protein burden becomes very important for informing improved design strategies. In order to test the model a library was built by combining promoters and RBS of different strengths. However, the data obtained did not succeed in finding clear relationship between the model and the samples of tested constructs. The experiments indicate that the fitness cost from engineered expression is amplified by longer growth periods before sample dilution to favourable growth conditions (Figure 2.13, and Figure 2.15 versus Figure 2.16). The burden effect from growth phase transitions is further investigated in Chapter 3.

3. Chapter 3

Chapter 3 develops a model with new predicted parameters for conceptualising and quantifying burden from engineered biology constructs. The methods focus on the transitionary construct costs during different growth phases. Parametrisation of the model builds on data from the experimental methods developed in Chapter 1 together with the constructs made in Chapter 2. Previous growth models for burden from fractions and resource allocation have focused on static, or steady state, relationships in cells. From the 'growth law' model, Chapter 3 develops a dynamic model for burden by incorporating transitionary effects from engineered construct expression. By combining the model with experimental data two new conceptual metrics for burden are identified: (i) delay in reaction time and signal processing (t_d) and (ii) the fraction of ribosomes R_R engaged in translating other ribosomes during the ramp-up phase of growth in a new environment, *i.e.* the time period before reaching maximum growth rate. The results of this Chapter have appeared in ACS Synthetic Biology [132], which also includes some additional data.

Nomenclature

P_x = Mass of protein x.

P = Total proteins (mass) [note: $P = \Sigma_x P_x$]

Φ_x = Fraction of the proteome that is x. [i.e. $\Phi_x = \frac{P_x}{\Sigma_x P_x}$]

γ = Rate of protein production by ribosomes through translation (h^{-1})

μ = growth rate. (h^{-1})

μ_L = growth rate during the ribosome limited ramp-up period up until deceleration of growth, *i.e.* $\mu_L = \mu$ for $t \leq t_{ramp-up}$ (h^{-1}).

R_x = Fraction of the total proteome currently being translated that is x.

R_R = Fraction of the total proteome currently being translated that belong to the Ribosome fraction of proteins.

$R_R(0)$: Initial R_R in a new environment.

$R_R(t_s^+)$: R_R after a step response to a new environment (occurring at time = t_s).

$R_R(t_c^+)$: R_R during the exponential growth phase after the growth ramp-up period.

t_s = Time point when R_R has a step increase in ENV_2.

t_d = Time point for growth initiation in ENV_1, given as the time point when the cell density reaches $OD_{600} = 0.2$.

3.1 Introduction

A number of models have been developed to investigate resource allocation in cells during steady-state growth [68] as part of a cellular economy. However, how allocation of resources changes over time during growth is an open problem. Host cells for engineered constructs have previously evolved to adjust this resource allocation in the absence of engineered constructs. At the same time, many experimental setups, as well as practical applications for engineered biological systems require their presence and functionality in fluctuating environments. Models and experimental investigations of how burden and 'loads' [56] from engineered constructs manifest during growth phase transitions between different environments are required to better quantify and understand the true costs of engineered biology in practical settings beyond steady-state growth.

3.1.1 Global and Coarse-grained Resource Allocation Models

One of the earlier global and coarse-grained resource allocation models was developed by Molenaar *et al.* [125], and since then an increasing number of models of similar architecture continues to emerge. Part of the motivation for this model was that many previous models for metabolic efficiencies of cells omitted the necessary cost of protein production. Its architecture can be summarised as self-replicator model of cells, which builds on simple feed-back and interactions between proteins involved in metabolism (transporters, amino acid synthesis, enzymes) and gene expression. By optimising the system for maximum growth rate, different fractions of ribosomes could be found for different static growth environments, as well as overflow metabolisms for higher growth rates [125].

Soon afterwards Scott *et al.* [162] published their 'growth-law' model together with experimental data. This model was also discussed in Chapter 2, and it derives empirical 'growth-laws' as a linear relationship between the growth rate of cells and their ribosome fraction as a function of the environment. It also includes the impact of engineered constructs during steady-state growth as well as the effect of reduced translational efficiency in the presence of toxic antibiotics.

Bosdriesz *et al.* [27] developed a kinetic model for the regulation of protein fractions

via the ppGpp stringent response alarmone [146, 179]. The model showed that the ppGpp mediated stringent response for ribosome regulation provides a very robust control for maximising the growth rate by optimising the ribosome concentrations during steady-state growth [27]. This conclusion is in agreement with previous 'growth-law' extensions [164], indicating that robust feed-forward control which senses the amino acid pool size and the ribosome fraction allows ribosome levels to be maintained at close-to-optimal values at a range of growth conditions [164].

A similar model of optimal resource allocation between the metabolic and gene expression machinery was developed focusing on energy efficiency in cells [119]. This model further reinforces the predictions of the 'growth-law' model for optimal regulation of the different fractions depending on the growth environment.

Weisse *et al.* [196] developed a coarse-grained mechanistic model on a single cellular level to investigate trade-offs of different growth strategies in different environments. They also expand the model to simulate the impact and 'load' due to the presence of engineered constructs. The model is used to simulate the impact on cells under different static conditions in the cell (of protein, nutrients and expression levels), rather than the impact from dynamic fluctuating environments. In one aspect, this mechanistic model [196] is similar to the self-replicator model by Molenaar *et al.* [125]. Both models focus on number of molecules within cells, how they are imported and how many more molecules are produced by each substrate. They also include sizes for the cells, but making different assumptions of whether the cell sizes are changing [125] or not [196]. This aspect of these previous mechanistic and molecular self-replicator type models is different from the model method developed in this chapter, which instead focuses on fractions within cells (or cell populations) and without translating these fractions to any specific quantify of molecules.

The models listed above consider resource allocation in cells at steady state of internal concentrations and external conditions (*e.g.* constant external influx rates of nutrients). (Further explanation of steady-state growth and comparisons with other similar types of growth, *e.g.* balanced and exponential growth, can be found in Fishov *et al.* [68].) Such conditions of prolonged stable environments can be achieved in a laboratory, for example in a chemostat. But many real world applications, as well as many laboratory setups and

experiments, experience frequently fluctuating environments.

The models developed by Giordano *et al.* [74, 75] have closer similarities to the models developed in this chapter, eventhough they omit any discussion of impacts from engineered constructs. For example, these models consider the distribution of metabolic and ribosome producing proteins as a dynamic process in time. The Giordano *et al.* [74, 75] models implement these through the perspective of minimal self-replicator systems in units of mass, volumes and concentrations of different molecular components, similar to the self-replicator model by Molenaar *et al.* [125]. The models in this chapter model these through dynamic interactions of proteome fractions (based on the static fractions from the 'growth-law' [162] methodology) and fractions of mass-rate expression.

Giordano *et al.* [74, 75] used tools from optimal control theory to evaluate how cells most optimally respond to changing environments. The underlying optimisation framework builds on the assumption that cells have been well optimised for rapid growth and environmental responses due to evolution, and as this would provide crucial fitness advantages. They start from a self-replicator model, similar to the model by Molenaar *et al.* [125], but then reformulate the question in terms of optimal control theory and identify cell strategies to maximise biomass production over an intervall of time, which becomes a dynamic equivalent of growth-rate maximisation.

The first study [74] finds that a so-called bang-singular control strategy is most favourable for cells in order to rapidly respond to a new environment, while a second study names it bang-bang-singular control strategy [75]. This strategy means all the cellular resources are focussed exclusively on expressing the component which is in most demand in the new environment, until all cellular components have reached their optimal concentrations. When all components have reached their optimal relative concentrations then expression levels are maintained at those levels, without the need for further sudden jumps. Meanwhile, the work recognises that cells require sensing mechanisms in order to respond to changing environments [74], and these sensing and response capabilities should require some additional cellular resources.

They find that a near-optimal control strategy has similarities with the stringent response in *E. coli* cells, mediated by the ppGpp alarmone [146, 179], which regulates ri-

bosome expression in a switch-like manner in response to mismatches between protein precursors and ribosome concentrations.

The optimal control predictions, in the form of a rapid overshoot of protein expression at changing environments, from the work by Giordano *et al.* [74, 75] are also built into the latter stages of the model in this chapter and fitted against experimental data.

3.1.2 Model Development

Most generally, the level of engineered construct expression is presumed to have an inversely proportional effect on the maximum growth rate, according to the growth law [162]. However, depending on the growth-phase, and when moving away from steady-state growth, it has also been observed that the effect on the maximum growth rate can be smaller than usually predicted [168]. A metric that traditionally has been difficult to predict or model but that could have a large correlation with the burden introduced from engineered biology is the early growth behaviour of cells, commonly referred to as the lag-phase. This Chapter is taking a deeper look at correlations between early growth effects and engineered constructs from a mechanistic point of view and is developing new dynamic models that can better inform growth effects from engineered biology.

This chapter has three subsections:

1. Development of a new Dynamic Resource Allocation Model (DRAM) model for engineered burden that links the previous 'Growth Law' model [162, 164] with how protein fractions are changing over time to predict the impact on growth when expressing engineered constructs.

2. Use of data to test the Bang-Bang control theory [74, 75] for how fractions are changing over time on the DRAM.

3. Attempts to link the growth in ENV_1 with growth in ENV_2 with the DRAM model. This has two purposes:

    - Provide and parameterise initial ribosome fractions (Φ_R) for the DRAM model that are ENV_1 dependent.

- Estimate and quantify more accurately the burden of a construct through the full data of growth behaviour in ENV_1 through to ENV_2. Providing such a measure could be particularly valuable as these are the very two environments that are normally seen in any type of experiments with engineered constructs. Previously, only the maximum growth rate in ENV_2 was used as a growth-related measure of burden for engineered constructs, however, as seen in Chapter 2 and previous literature [168], this can often provide less conclusive cost values.

3.2 Methods

3.2.1 Data Collection and Measurement Methods

For any growth curve extrapolation or fitting it is recommended that a minimum of 10 data points are collected [184].

It has been indicated that the positioning of data points collected are of high importance for the interpretation of growth behaviours. The two transition zones of entering and exiting exponential phase are the data collection time periods of highest importance for accurate growth interpretation [145]. Previously, a different method has been tested for estimating growth rates or lag times from populations that were serially diluted a multitude of times [16].

Experimental data presented in Chapter 3 are produced with the same methods as described in more detail in Chapters 1 and 2. Samples were grown in a 96-wellplate in 240 microLitre of LB-medium with 100ng/mL of Ampicillin. Optical density absorbance of a wavelength of 600nm was used for cell density measurements. Absorbance measurements were normalised against the mean value of 4 to 8 empty wells with the same volume of medium (4 or 8 depending on the layout of the plate and the time point or the measurement).

3.2.2 Data Fitting

Least squares is used for fitting functions to data series. The 'curve_fit' function from the python Scipy library was used for fitting non-linear functions. The 'curve_fit' function applies the Levenberg-Marquardt algorithm to find a minimized solution. The 'poly_fit' polynomial fitting function from the python Numpy library was used for linear fittings. Both methods apply least squares fitting.

3.3 Fitting of Data to an Existing Growth Model

3.3.1 Growth Curves and Models

When measuring and interpreting the growth behaviour of microorganisms it is recommended to use model parameter estimates from measured data, rather than using linear regression on a subset of the data [211]. This method also helps reduce measured data into a small number of meaningful parameters [31].

Growth models and growth curves have been used extensively for measuring microbial growth, not only for research purposes, but also in fields such as food microbiology [22]. Predictive models of microbial growth have a long history of wide applications by informing microbial safety, shelf life of products, the detection of critical parts of production and distribution chains as well as their optimisations [211, 175, 31]. Most of the models of microbial growth are empirical algebraic, of which the Gompertz model is the most notable [139].

In the case of predictive models for microbial populations most frequently used by food microbiologists [175], a two-step modelling approach is commonly applied. Primary models describe microbial numbers over time, for example the Gompertz equation [76, 211] can be applied to find the key parameters of growth rate (μ_{max}) and lag-phase (λ). Subsequently, secondary models are derived to relate these two parameters to influencing factors, *e.g.* temperature or pH [175, 57]. However, even if reasonable values for μ_{max} are usually estimated, poor estimates are often found for the lag-phase λ [121].

Microbial growth curves developed and used generally assume a phase where the growth rate is zero, followed by acceleration to a maximum growth rate in a certain length of time, resulting in a lag-time. In addition, growth curves contain a final stage in which the growth rate decreases to zero at a maximum cell population, at which point an asymptote is reached [211]. When a growth curve is plotted as the logarithm of organisms in a population it takes the shape of a sigmoidal curve, with an initial lag-phase, followed by at growth phase and ending with a stationary phase [211] (see Figure 3.1). The key parameters of applied use and requiring accurate estimation are the lag-phase and the maximum growth rate [22].

Various growth curves have been developed to find relevant growth data from microorganisms by fitting to data. Some of the more notable empirical models are: Gompertz [76], Stannar [174], Schnute [160], Richards [150], the logistic model [211] as well as others [151, 211]. The most commonly used and recommended model for microbial growth is the Gompertz model [211, 22, 154].

The original version of the Gompertz model (also shown in Chapter 2) is [76]:

$$y = a \cdot exp[-exp(b - ct)]$$

where y is given as the natural logarithm of the population size x a time t divided by the original population size, *i.e.* $y = ln(x_t/x_0)$. Mathematical inputs to the model are given by a, b and c to provide a sigmoidal growth curve [211].

A modified version of the Gompertz model that directly identifies the key relevant parameters for growth curves of microorganisms can be written as [211]:

$$y = A \cdot exp\Big\{ - exp\Big[\frac{\mu_m \cdot e}{A}(\lambda - t) + 1\Big]\Big\} \tag{3.1}$$

Here μ_m is the maximum growth rate, A is the asymptote, and λ is the lag-time.

Statistical comparisons have been made between different empirical sigmoidal curves, and in many cases the Gompertz model is regarded as the best model to fit data [211, 22]. These statistical comparisons are also one of the key reasons why the Gompertz equation has become so well established for microbial growth measurements. However, in practice, models with parameters with biological meaning are often considered more useful [22]. Deterministic models with only a few biologically relevant parameters are preferable and generally are more easily tested and validated experimentally [182]. It has previously been shown that the use of a correct model for fitting and interpreting growth is essential to enable the collection and comparison of growth data, and it is even more important than the techniques and methods initially used to collect the growth data [21]. The Gompertz equation, even if frequently used for fitting, is an empirically derived sigmoidal relationship that resembles microbial growth and not a physiological based model. In fact, researchers argued against the use of the Gompertz equation, as it is purely an equation with a con-

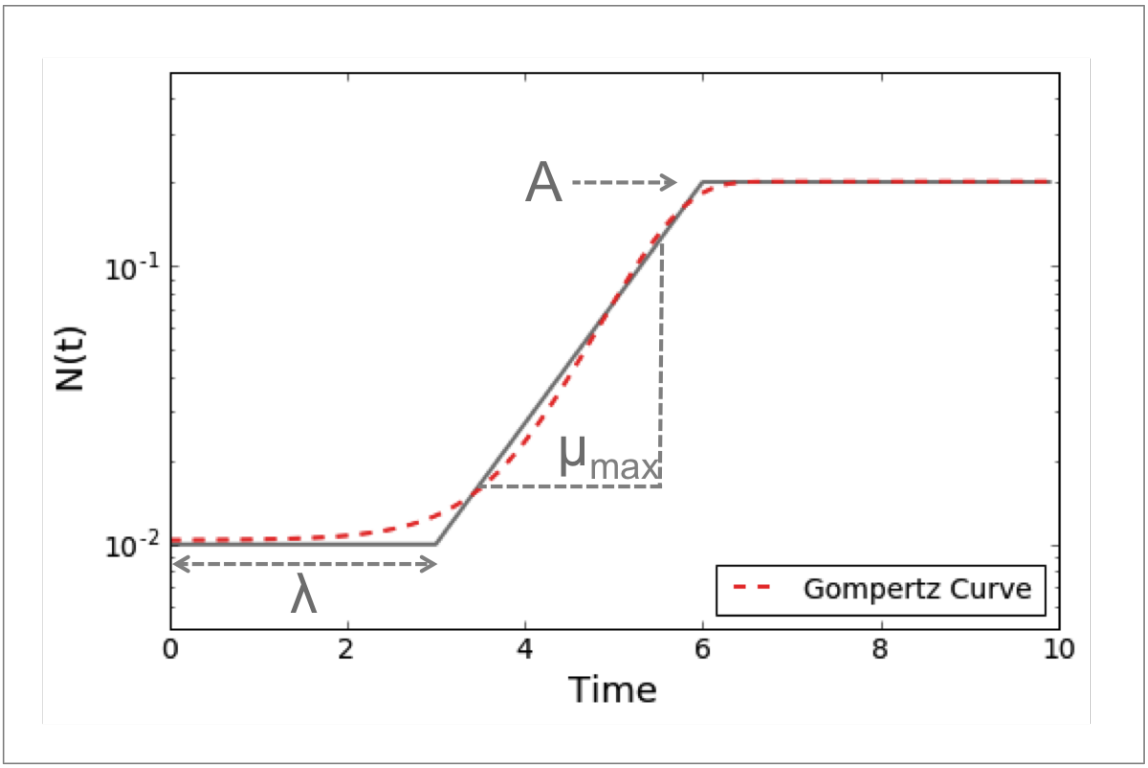


Figure 3.1: The key parameters for describing growth curves and the Gompertz growth curve (red dashed line). The grey solid line illustrates the three phases of growth that are represented by growth curves. μ_{max} = the maximum growth rate ($\frac{d(ln(N))}{dt}$). λ = the lag-phase (time). A = the asymptote (N at infinite time).

venient relationship for fitting growth curves, and not a model that can be interpreted or explained physiologically [17].

The start of growth in a new environment, either the lag-phase or an initial part of the lag-phase, is sometimes called the adaptation phase. However, the term adaptation has multiple meanings and can easily be confused with the evolutionary meaning of adaptation [115], which is a very different process discussed further in the next chapter. Therefore, this chapter uses the term acclimitisation, instead of adaptation, to refer to the regulation of internal processes within the cell in the face of a new environment.

The Lag-phase and Growth Rates

The lag-phase is inherent to the kinetics of all microbial species studied, and it is typically observed as a delay response to a sudden change in the environment [175]. In case of a sudden upshift in nutrient levels, the lag-phase is the period when cells prepare themselves

in order to take advantage of the new environment [31, 175]. The lag-phase experienced by microorganisms at the time of inoculation into stable environments is referred to as the *initial* lag-phase [175]. However, there are also other sorts of microbial lag-phases not discussed here or investigated further with experiments, for example *intermediate* lag-phase from depletion of partial nutrient sources, sudden temperature shifts, *etc* [175]. The lag-phase represents the earliest but also the least understood stage of the bacterial growth cycle [154]. There are many different definitions for the *initial* lag-phase of microbial growth [175]. One early definition used the time needed to increase the population two-fold [30]. The total time period to reach maximum growth rate has also been used as a definition of lag-time [144], (for the models and data fitting in this chapter, this time is instead referred to as the growth 'ramp-up' period). Today, the most widespread method for defining lag-phase is the intersection (on a logarithmic-scale) between the extrapolated tangent from the exponential growth phase and the horizontal tangent of initial inoculum level [175, 212, 22], as illustrated by λ in Figure 3.1. An alternative definition proposed later is the time required to reach the maximum acceleration of growth rate [28], *i.e.* when the second derivative of the logarithm of the growth curve reaches its maximum value. However, for sigmoidal logarithmic growth curves the difference in time between the last two definitions is often minimal. It has therefore been suggested that the former (extrapolated intersections of the logarithmic growth rate and the starting population) should be used, as it facilitates comparison with a larger number of literature values [212].

To add further to the diversity involved in comparing literature values for growth behaviours, there are two different definitions for the maximum growth rate [142]. One is that the maximum growth rate is the fastest rate of growth. However, for some measurement values reported, this refers to the steepest inclination of a fitted growth curve, whereas for other measures reported less frequently, it refers to the rate of maximum growth in an infinite dilution environment [142]. For the values reported in this chapter the maximum growth rate refers to the former definition.

The new models described in this chapter do not necessitate the selection of a specific lag-time definition. There is no specific parameter in the model that relates directly to lag-phase, and for our purposes they are to be thought of as descriptive. However,

when a lag-phase is mentioned, either descriptively for the new models or explicitly when comparing other models, it refers to the generally established definition of an intersection of extrapolated exponential growth and the initial population size on a logarithmic scale [212].

In addition, one of the earlier definitions of lag-time – the time taken to reach maximum growth rate – is defined as the full growth 'ramp-up' period. This ramp-up period is important specifically because the ratios of cellular components are projected to be best represented from measured data within this period. Further, for the models described in this chapter, the earlier definition of the end of lag-phase as the maximum point of growth rate acceleration [28] has a slightly different but important physiological meaning; it is the time point of maximum translation fraction of ribosomes (denoted R_R).

For growth curve and model developments, many studies have emphasised particular difficulties in estimating the lag-phase (λ) [15, 22]. One reason is the little knowledge available concerning the cells' physiological state, and as a result few models have been able to provide biological meaning regarding λ [22]. The expression of engineered constructs exerts a combined effect on growth curves by simultaneously reducing the maximum growth rate and the asymptote, while increasing the lag time, or at least that is the generally held assumption. A more recent observation shows that the timing of expression affects the maximum growth rate [168].

Cell Measurements

The lag-times observed for turbidity measurements are usually shorter than the lag-times observed for experimental techniques of viable cell counts, the reason being that cells increase in size, volume and mass as they move from stationary to rapid growth states [175]. However, turbidity measurements do not distinguish between live and dead cells, and this could in theory have the opposite effect of prolonging the lag-phase of turbidity measures compared the viable cells counts (as the non-living cells also included by turbidity measurements are not growing). Therefore, there is a discrepancy of measurement values for growth between turbidity and viable cell counts, particularly in the region of the lag-phase and the early exponential phase [211, 21]. In this region there is a non-linear relationship

between population number and population mass (and volume), as cells are changing their sizes and mass [175]. The experimental data used in this chapter measure turbidity. However, as the parameters for the models relate to total cell mass, as well as proportions of cellular components (rather than number of cells), it is likely that the models can relate equally well to both measurement types. When relating the models to the fractions of distributions within a population as a whole, rather than individual cells of different sizes, the measurement methods are equal. Put simply, a single cell of double mass will have the same total mass as two cells of the original mass. But the developed models in this chapter relate to the proportions of mass within the total number of cells, meaning that the models become indifferent to discrepancies between measurement methods of either cell numbers or cell mass.

It is well known that the lag observed by wild type populations depends on the actual experimental environment (ENV_2) [175], for example increasing with decreasing temperature [29]. A proportional relationship between lag time and generation time has generally been assumed by microbiologists [57]. However, while the generation time is largely independent of historical culture conditions, reasonable dependencies on wild-type populations' lag-phase have been observed from the pre-culture environments (ENV_1) [175, 57]. In addition, the lag time is also affected by the level of change between the two environments (ΔENV) [122, 153], for example the difference in temperature or nutrient sources. Thus, the transition period is minimal when the temperature in ENV_1 matches the temperature in ENV_2 [90, 210]. In the experiments described within this chapter, no difference in environment (ΔENV) between pre-culture and culture environment is applied, except for the difference in nutrient levels that is a natural effect of growth. In other words, growth resumes and initially accelerates in ENV_2 after having decelerated following nutrient depletion in the later stages of ENV_1. This setup for experiment is also the best accepted as well as most commonly used method for experiments of engineered biology constructs. Hence, this experimental data format is also most useful for analysing growth effects with models, especially if such models can be empowered to provide deeper insight into the levels or mechanisms of the cellular burden imposed by different engineered biology sources. Generally, it is assumed that a smaller environmental change (ΔENV) will yield shorter

transition periods [175]. Further, actively growing cells adjust more rapidly to changing environments compared to cells from either lag or stationary phase [9, 123]. It has frequently been observed that the lag-time is inversely proportional to the maximum specific growth phase of wild type populations [22]. However, this proportionality relationship might not hold true for engineered biology.

The Ramp-up Period and 'Work' from Engineered Biology

According to previous knowledge, the lag-phase is dependent on two factors: (i) how much 'work' is needed to adapt to the new environment, directly related to the change in environment ΔENV and (ii) the amount of 'work' that the cell can produce, which has been assumed to be directly proportional to the maximum growth rate in the new environment μ_{max} [153]. If the maximum growth rate is not assumed to be the traditional μ_{max} but instead the alternative (not recommended) measure of the maximum growth rate at infinite dilutions, then it can seem reasonable that μ_{max} is proportional to the maximum level of maximum 'work' that the cell can produce, as this is also the maximum rate of biomass production. This measure of 'work' capacity would then be directly proportional to the maximum Φ_R level for a given nutrient condition. However, that would still be a measure of the *maximum* rate of biomass production, or work capacity, and not the level of work capacity by cells during other phases of growth. The dynamic resource allocations models in this chapter are assuming that the 'work' capacity of cells are changing over time, and during the extend lag-phase, or 'ramp-up' period, is directly proportional to the self-expanding Φ_R fraction.

In order to build models that can describe microbial growth it is also necessary to measure biological growth. Currently there are no models for growth curves available after different times of dilutions of microorganisms, despite most experiments with microorganisms requiring dilutions before experiments start. The model and fitting methods shown in this chapter aim to provide a better way of interpreting burden caused by the presence of engineered biology directly from growth curves. Also, as the previous models for burden and engineered costs have been focused on artificial static conditions, the new models provided are looking at the more frequently encountered dynamic conditions. In most situ-

ations, expression levels are changing over time, due to fluctuating environments. The new models developed in this chapter do not assume static conditions over time but instead assume conditions are naturally changing. The models are targeting how expression level burdens can be informed from growth data.

3.3.2 Data fitting to the Gompertz function

Figure 3.2 shows the cell density readings of populations for samples with and without constructs. The growth curves are fitted to the Gompertz curve with the least squares method described in Section 3.2.2. Fitting with the Gompertz curve directly provides the maximum growth rate (μ_{max}) and lag-phase (λ), from Equation 3.1. The samples are diluted at two different levels from ENV_1 to ENV_2. Ideally, the two dilutions would provide very similar values for μ_{max} and λ. Figure 3.3 shows a considerable variation between the two dilutions, particularly for the lag-phase λ. This is consistent with previous growth curve studies which have emphasised particular difficulties in estimating the lag-phase (λ) from experimental data [15, 22]. Turbidity and optical density readings at low optical densities have much higher reading errors compared to optical density readings of higher and moderate optical densities. The lag-phase occurs at the very initial stages of growth were optical densities by default are very low (and have higher errors) compared to later stages of growth when optical densities are higher and provide more accurate readings.

That the lag-phase is negative simply means that the lag-phase has ended prior in time to the range of the fitted data. For example, if a sample has a lag-phase of one hour, then enters exponential growth where it remains for two hours before it is subcultured in an equivalent environment. The exponential growth continues in the new environment. The growth curve in the second environment (after the subculturing) would have a lag-phase which ended two hours earlier than the first time point of the growth data, *i.e.* two hours negative to the first time point post subculturing. The negative lambda values in Figure 3.3 (ENV_2) means that the Gompertz growth curve fitting projects the lag-phase to have occurred in ENV_1, prior to the fitted data values (which are fitted to ENV_2 and post subculturing). A second example of a Gompertz growth curve fitting and using an additional construct is later provided in Figure 3.4.

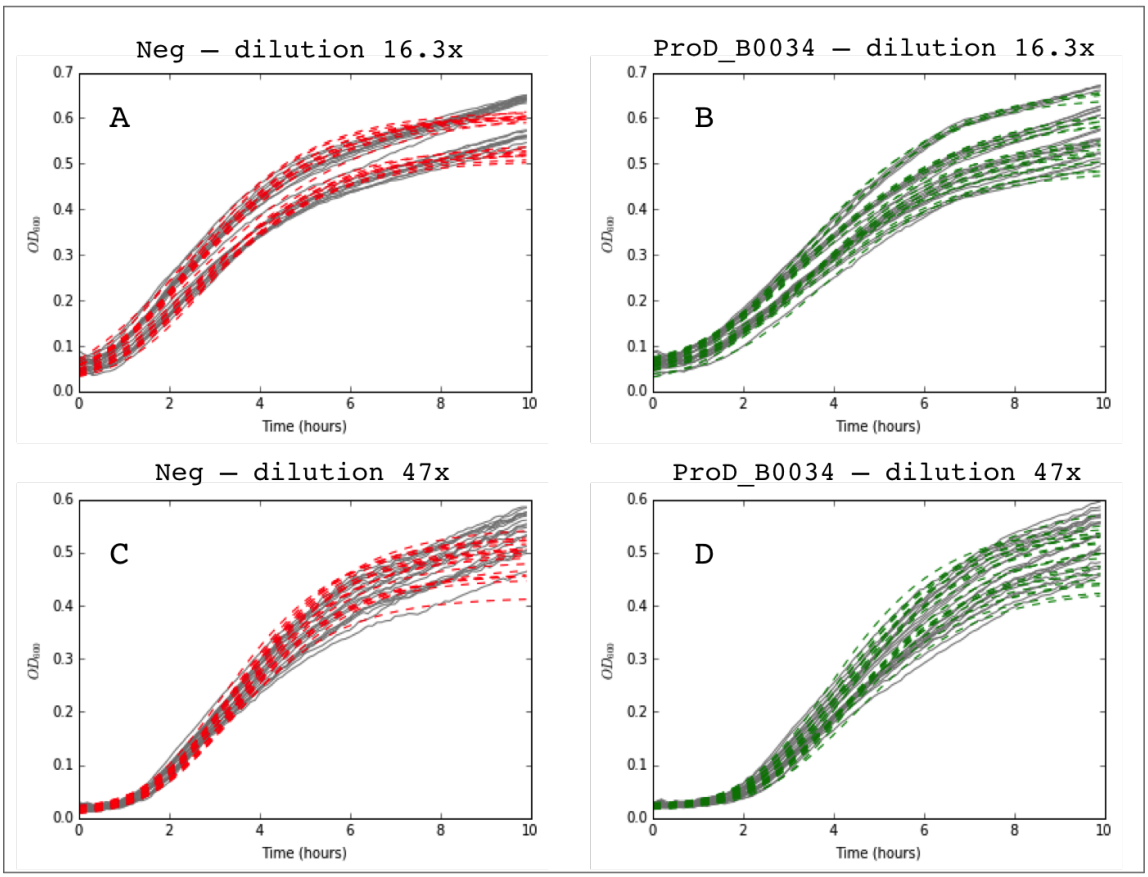


Figure 3.2: Gompertz equation fitting (coloured dashed lines) for the full set of measured growth (solid grey lines) in a single experiment with samples being diluted 16.3 times and 47 times after a 24h pre-culture. Neg are the negative control, *i.e.* no construct expression on the medium copy number plasmid, which is the same plasmid copy number as for ProD-B0034.

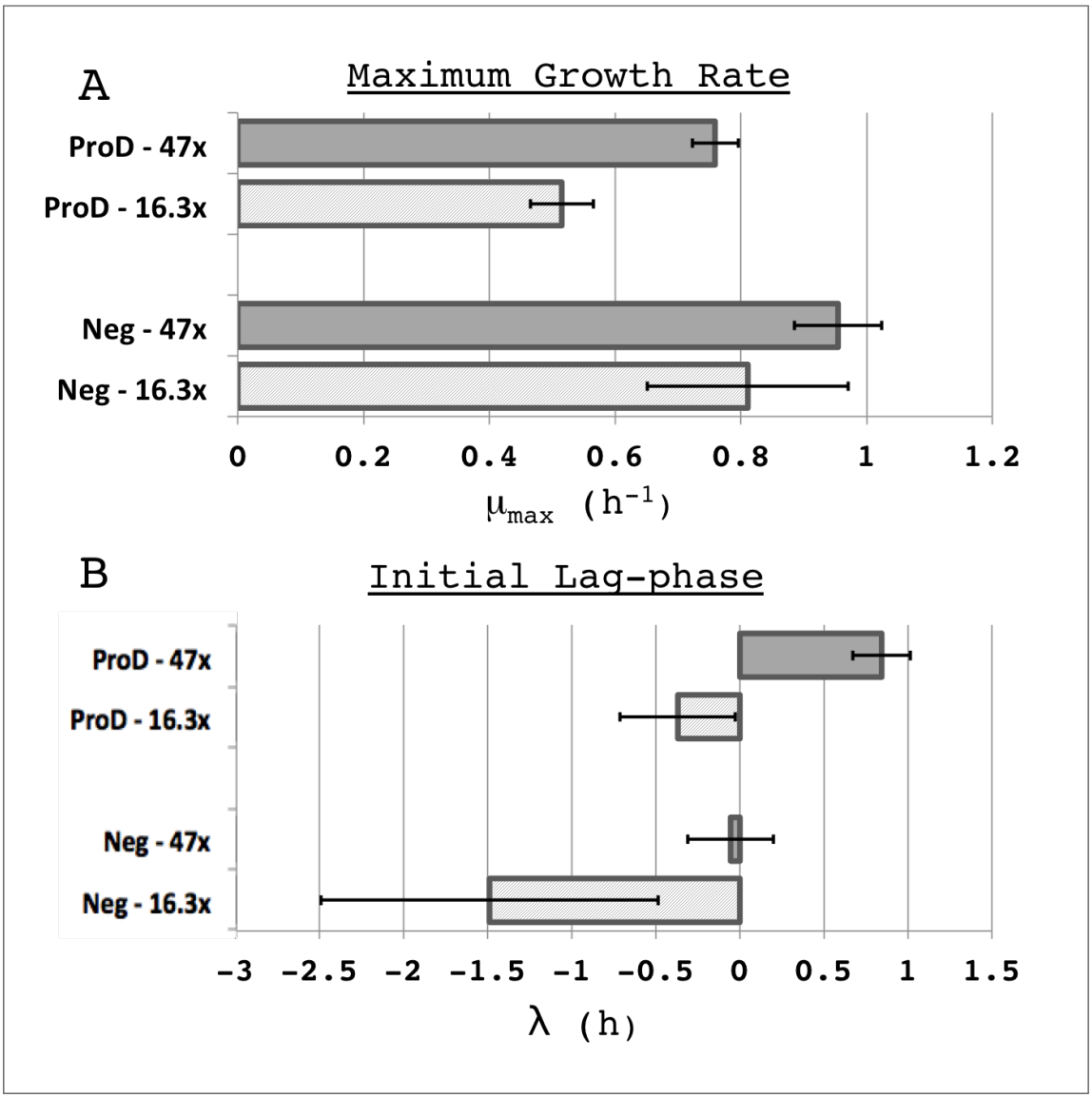


Figure 3.3: Maximum growth rate (A) and lag-phase period (B) obtained from the Gompertz fitting of the complete set of growth curves in Figure 3.2. The bars are the mean values for two different levels of sample dilutions, 16.3 and 47x during the same experiment. The error bars represent the standard deviation. ProD denotes ProD-B0034 with a medium copy number plasmid. Neg denotes negative control, *i.e.* no construct expression on the medium copy number plasmid.

3.4 Derivation of the Dynamic Resource Allocation Model (DRAM)

In order to improve the understanding of burden a dynamic model has been developed. The model is based on the 'growth law' [162] previously discussed in Chapter 2. The key difference which separates the presented DRAM from the previous 'growth law' methods is that it relates to, and interprets, the rate of fraction changes over time. The previous 'growth law' is static and refers to steady-state growth. The model developed here relates to how these fractions dynamically are changing over time.

Key parameters:

P_x = Mass of protein x.

P = Total proteins (mass) [note: $P = \Sigma_x P_x$]

Φ_x = Fraction of the proteome that is x. [i.e. $\Phi_x = \frac{P_x}{\Sigma_x P_x}$]

γ = Rate of protein production by ribosomes through translation (h^{-1})

R_x = Fraction of the total proteome currently being translated that is x.

μ = growth rate. (h^{-1})

μ_L = growth rate during the ribosome limited ramp-up period up until deceleration of growth, *i.e.* μ at $t \leq t_{ramp-up}$ (h^{-1}).

From the experimental data during the ribosome limited ramp-up period of growth we obtain: (Assuming a constant and saturated translation rate by ribosomes, $\gamma = \gamma_0$, and assumed to hold for all $t \leq t_{ramp-up}$)

$$\mu_L = \Phi_R \gamma_0 \tag{3.2}$$

$$\frac{d\mu_L}{dt} = \frac{d(\Phi_R \gamma_0)}{dt}$$

$$= \gamma_0 \frac{d\Phi_R}{dt}$$

We now proceed to find a relationship to R_R.

By definition:

$$\Phi_x = \frac{P_x}{\Sigma_x P_x}$$

By assuming the biomass of proteins is a constant fraction of the total biomass the specific growth rate can be written as:

$$\frac{dP}{dt} = P\mu$$

A different way of writing the equation is to define the growth rate as the rate of increase of total protein mass per protein mass, $\mu = \frac{1}{P}\frac{dP}{dt}$ [132].

R_x is the fraction of total translation by ribosomes that is dedicated to translating the protein fraction P_x, *i.e.*:

$$\frac{dP_x}{dt} = R_x \frac{dP}{dt} = R_x P\mu$$

Then:

$$\frac{d\Phi_x}{dt} = \frac{d\frac{P_x}{P}}{dt}$$

$$\frac{d\frac{P_x}{P}}{dt} = \frac{1}{P}\frac{dP_x}{dt} - \frac{P_x}{P^2}\frac{dP}{dt}$$

$$= \frac{1}{P}R_x P\mu - \frac{P_x}{P^2}P\mu$$

$$= \left(R_x - \frac{P_x}{P}\right)\mu$$

$$\frac{d\Phi_x}{dt} = (R_x - \Phi_x)\mu \tag{3.3}$$

During the ramp-up period for growth, the equation most relevant to growth is the ribosome fraction,

$$\frac{d\Phi_R}{dt} = (R_R - \Phi_R)\mu$$

Also, during the ramp-up period, when growth is assumed to be limited by the availability of ribosomes (from Equation (3.2)):

$$\frac{d\mu_L}{dt} = (R_R - \Phi_R)\mu\gamma_0 \tag{3.4}$$

$$= \left(R_R - \frac{\mu_L}{\gamma_0}\right)\mu\gamma_0$$

I.e.

$$R_R = \frac{1}{\gamma_0}\left(\frac{\frac{d\mu_L}{dt}}{\mu} + \mu_L\right)$$

Note, Φ_R is only assumed to be directly proportional to μ during the ramp-up period and μ during this time period is identical to μ_L. μ_L is simply a constrained version of μ (which only is valid when growth is limited by the availability of ribosomes). Subsequently, an equation containing μ_L is only valid for such data ranges, and where the value of μ_L is equal to μ.

In Equation (3.4), μ_L on the left hand side stems from $\mu_L = \Phi_R\gamma_0$ and is only valid during the ramp-up period. On the right hand side, μ relates to dilution of proteins from growth and holds for general conditions. Recall that μ_L refers to the growth phase when limited by available ribosomes for protein expression (L stands for ribosome Limited, it could also refer to Lag-phase, however, it is the whole 'ramp-up' period of growth that is referred to, *i.e.* it is not constrained to the pervious defined shorter duration of the lag-phase as an intersection of two tangents [175]).

The dynamic relationships derived above are based on basic assumptions and relationships from the previous 'growth law' model [162]. The model shown above is referred to as a Dynamic Resource Allocation Model (DRAM).

3.4.1 Fitting of a Dynamic Resource Allocation Model to Gompertz growth curves

Fitting of the Gompertz curve to the mean of experimental growth data is shown in Figure 3.4. The fitting of the Gompertz growth curve to initial points of growth measurements gives a max growth rate of $1.30h^{-1}$ vs. $0.86h^{-1}$, and a lag-phase of $0.72h$ vs. $1.26h$, for the empty vector and ProD-B0035 respectively. The fraction results of fitting the DGL model to the Gompertz fits from Figure 3.4 can be seen in Figure 3.6. The fitting of the DRAM to the Gompertz function is obtained via the growth rate and the growth rate acceleration, shown in Figure 3.5. However, the fitting of the DRAM directly to extracted Gompertz curves seems to suggest that the fraction of $R_R > 1$ (Figure 3.6). The important message is not the fitting to data, but that R_R should not exceed one, being linked to growth rates from first principles. In effect, this limit prohibits a rate of growth rate increase that is too high (*i.e.* a too high log-OD second derivative). A simple explanation for poor performance of any growth curve fitting during the initial stages of growth for cell populations is that optical density readings at low cell densities have large reading errors. However, according to the developed model, the fitting of the Gompertz curve provides a function which is unlikely to be a good estimate of the true growth of the cells, given the available data (*i.e.* as $R_R > 1$). If selecting between different growth curves to fit and extract growth parameters from early stages of growth (while also accounting for the high error rates for such low density readings), an improved model for growth fitting for early stages of growth could be a function which (i) has a lower maximum first derivative of the growth rate, and (ii) extracts more informative parameters of the biological processes within the cell population. Before moving on to alternative methods of model fitting the experimental data (without the Gompertz curve for smoothing the measurement data), a few different potential reasons for the initial growth of cell populations being able to demonstrate growth in agreement with the fitted Gompertz curve are discussed below, together with their biological meaning.

One possible explanation for the inconsistency seen in Figure 3.6, between R_R and plausible R_R (*i.e.* $R_R < 1$), is that rate of translation (γ_0) is suppressed during the first

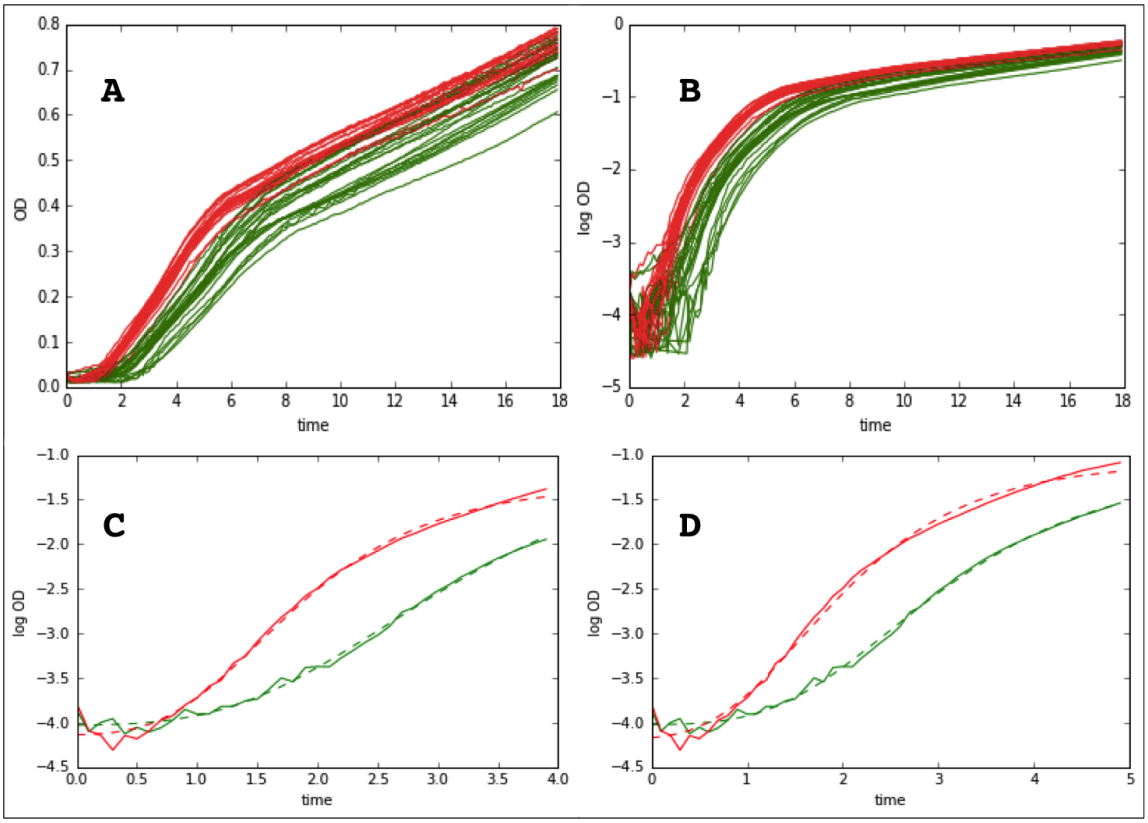


Figure 3.4: The growth in ENV_2 of ProD-B0035 (green) and empty vector (red) after a 47x dilution from a pre-growth environment (ENV_1). A: OD_{600} measurements. B: Log of OD_{600} measurements. C: Fitting of Gompertz growth curve (red dashed line) to the mean of negative control empty vectors, below log OD -1.4 (red solid line). D: Fitting of Gompertz growth curve (green dashed line) to mean of the construct ProD-B0035, below log OD -1.5 (green solid line). Note, the same data is shown in C and D, but the the Gompertz is fitted for a shorter time period in C than in D. ProD-B0035 and the empty vector are fitted to the same growth phase, as OD is increasing more rapidly for the empty vector. An additional construct and dataset was shown in Figure 3.2 and together provide two examples of Gompertz curve fitting to measured data.

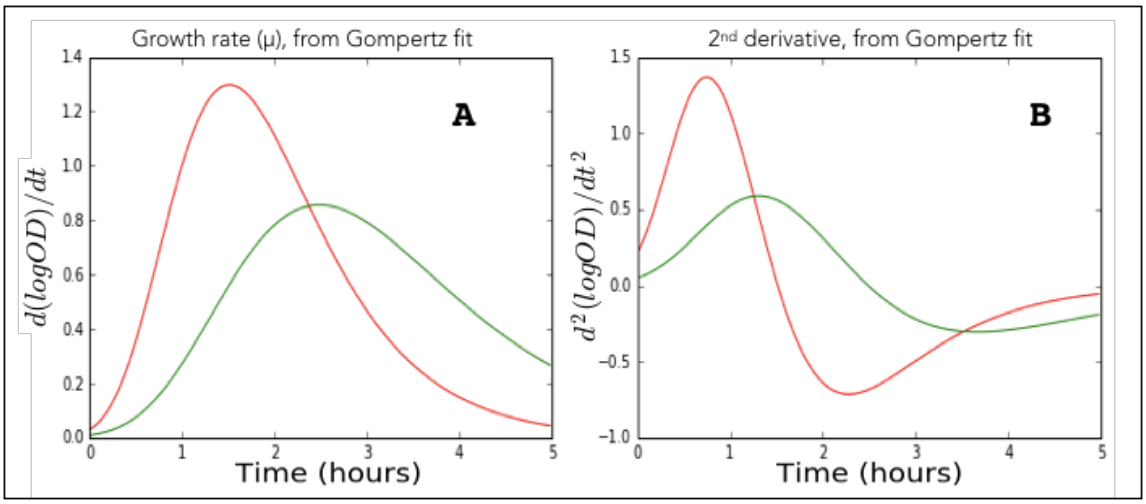


Figure 3.5: The first- (A) and second-order (B) derivative of the Gompertz function fitted in Figure 3.4. The green lines in the two subplots refer to the ProD-B0035 construct and the red lines refer to the empty control vector (*i.e.* no construct). A: Growth rates (μ) from the fitted Gompertz function (shown in Figure 3.4 and after a 47x dilution from a pre-growth environment ENV_1). B: Rate of growth rate change ($\frac{d\mu}{dt}$) from the fitted Gompertz function. The growth rate (A) and the growth rate acceleration (B) are the first and the second derivative of the Gompertz function in Figure 3.4.

time points of shift from slow to fast growth, *e.g.* from a delay in import of nutrients. This means that the true level of ribosomes (Φ_R) is underestimated at these time points and therefore so is the rate at which ribosomes initially increase. This would still give $\mu = \Phi_R\gamma(t)$ but:

$$\frac{d\mu_L}{dt} = \frac{d(\Phi_R\gamma(t))}{dt}$$

$$= \Phi_R\frac{d\gamma(t)}{dt} + \gamma(t)\frac{d\Phi_R}{dt}$$

$$= \Phi_R\frac{d\gamma(t)}{dt} + \gamma(t)\mu(R_R - \Phi_R)$$

So that,

$$R_R = \frac{1}{\gamma(t)}\left(\frac{\frac{d\mu_L}{dt}}{\mu} + \mu_L - \frac{\frac{d\gamma(t)}{dt}}{\gamma(t)}\right) \tag{3.5}$$

With those two equations one can adjust the initial translation rate [$\gamma(t)$] as well as how fast it reaches its saturation rate [γ_0] to fit a plausible (slightly higher) initial fraction for Φ_R as well as its reduced rate of initial increase. This assumption of reduced translation

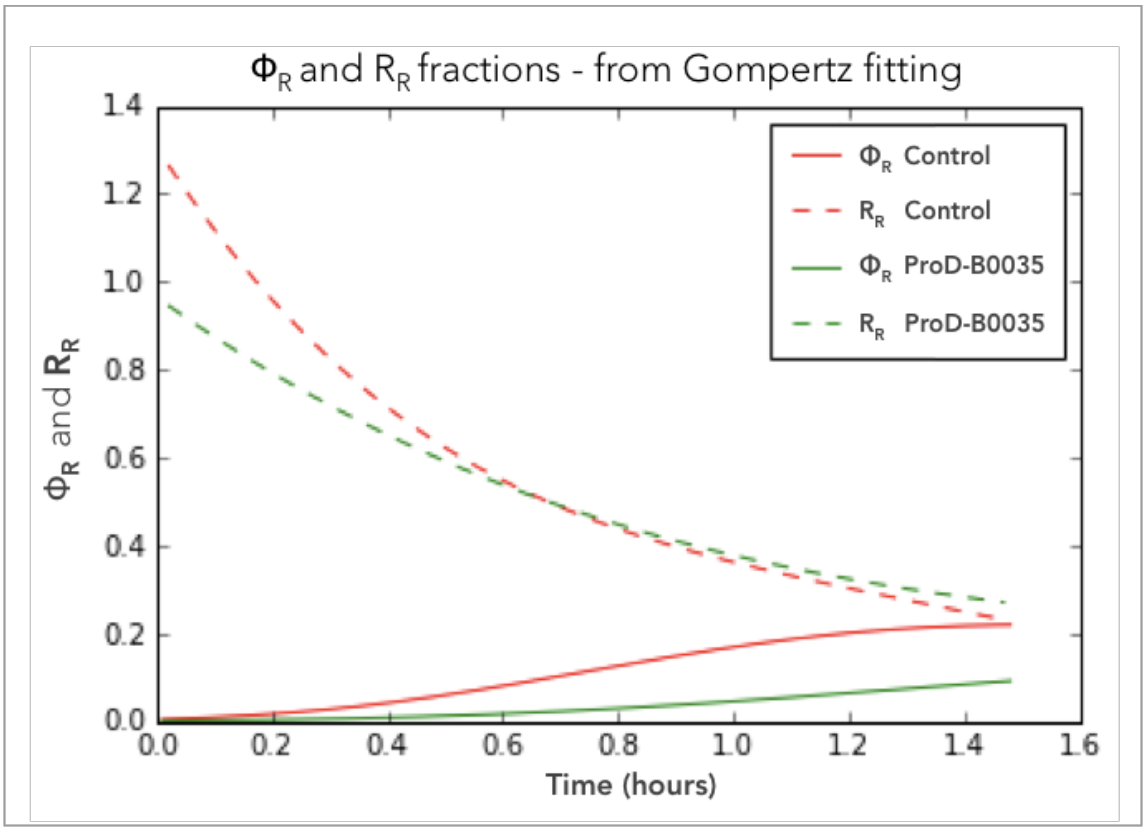


Figure 3.6: The Φ_R and R_R fractions obtained from the μ and $\frac{d\mu}{dt}$ rates shown in Figure 3.5 (and after a 47x dilution from a pre-growth environment in ENV_1). The fractions are obtained from applying the Dynamic Resource Allocation Model with Equations 3.2 and 3.5, and assuming $\gamma_0 = 5.9h^{-1}$ [164]. The fitting does not give a sensible prediction, because the fraction R_R can not exceed 1.

rate could in theory easily provide a reduced $R_R << 1$.

In a recent report that systematically characterised protein synthesis rates in *E. coli*, it was established that during slow growth the translation rate decreases with reduced growth rate [49]. At poor nutrient conditions, cells maintain a balance between the high translation rate and the reduced nutrient availability by decreasing the number of fraction of active ribosomes [49]. The translational elongation rate decreases as a Michaelis-Menten relationship with respect to the abundance of ternary complexes (TCs, comprised of aminoacyl-tRNA, elongation factor TU and guanosine triphosphate). However, a significant elongation rate is maintained, even towards zero growth and including stationary phase ($> 50\%$ compared to rich medium) [49].

The report concludes that the maintained higher rate of elongation also during stationary phase is critical for the cell's ability to respond in harsh environments [49]. It is possible that rapid acclimitisation to more favourable conditions also necessitates a higher basal translation level, as a rapid response time can have large evolutionary benefits during feast-and-famine fluctuating environments.

However, the translation efficiency quickly jumps to a high rate immediately after a nutrient upshift [49, 102]. The transition from a low proportion of active ribosomes increases (4-5 fold) to a maximum capacity of ribosomes within minutes [49, 102]. Due to the very fast response time of the translation efficiency that has been demonstrated, it can be assumed for the Dynamic Resource Allocation Model that $\gamma = \gamma_0$. In other words, applying Equation 3.5 is not the solution but is instead an over complication of the model.

A second option for reducing R_R (seen in Figure 3.6) is to introduce initial fast degradation of non-ribosomal proteins. That assumption does provide a faster readjustment of Φ_R without requiring as high proportion of R_R. At the same time it also requires a larger absolute level of Φ_R for a given protein mass accumulation rate. This is a plausible scenario, but there is currently no theory or data that would support how the cell's degradation machinery (which was previously in a slow-growth mode) could be directed towards degrading growth-related proteins, including ribosomes. The shift necessary to boost Φ_R from a lower R_R would be to rapidly shift degradation target from growth related proteins to the metabolic fractions of the proteome. This second scenario seems unlikely, or it can

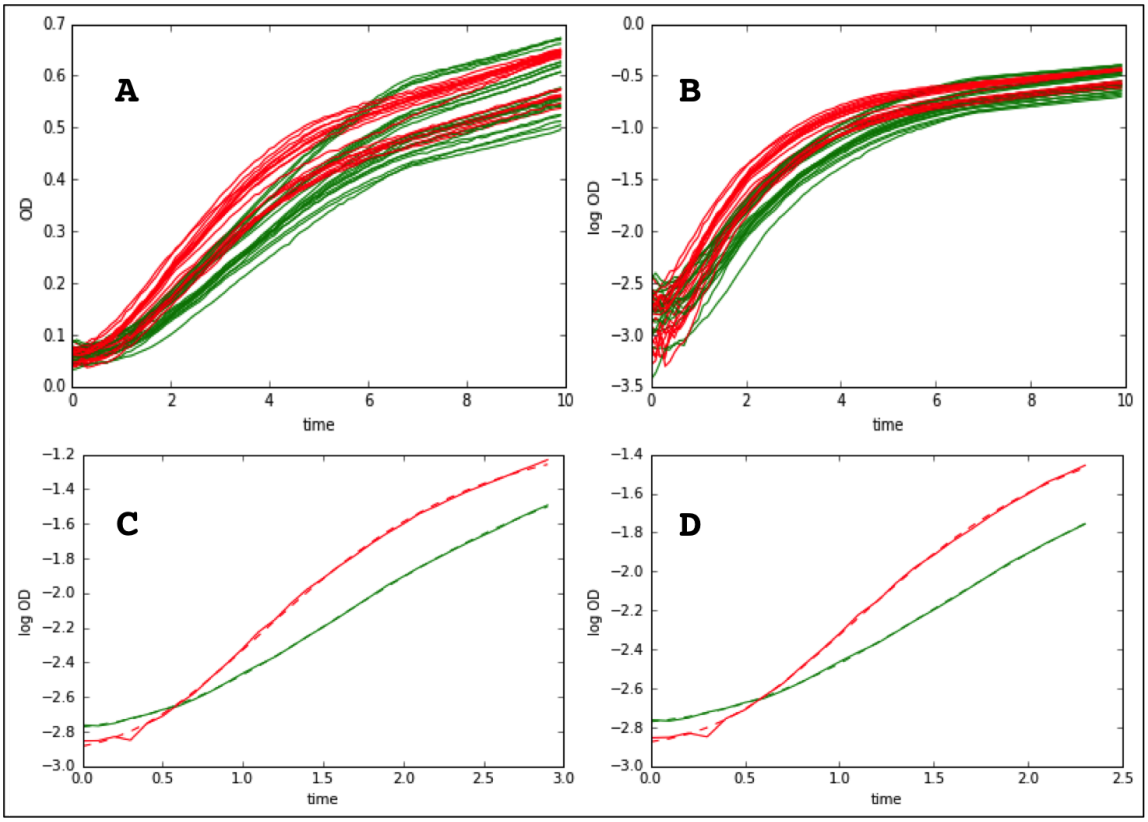


Figure 3.7: The growth in ENV_2 of ProD-B0035 (green) and empty vector (red) after a 16x dilution from a pre-growth environment (ENV_1). The same samples and gompertz fittings are shown in Figure 3.4 for a 47x dilution. The lower dilution means that populations have less time to readjust to higher growth rates before nutrients become limiting. However, they provide more reliable data readings for the very initial time points by providing more accurate estimates of the very initial growth rates. A: Growth rates (μ) from the fitted Gompertz function (shown in Figure 3.4). B: Rate of growth rate change ($\frac{d\mu}{dt}$) from the fitted Gompertz function. Green: Construct ProD-B0035. Red: Empty control vector.

at least not be supported by available data.

Finally, a third and most likely scenario is that the Gompertz growth-curve, as well as similar currently applied growth-curves are underestimating the initial growth rate of the bacterial populations at the same time as overestimating the rate of growth rate increase during the lag-phase. Indeed, when re-running the same experiment but at a lower dilution rate it becomes apparent that the growth rate at the initial time points is in fact higher than what Gompertz fitting estimates.

Figure 3.7 shows fitting of the Gompertz model to the same constructs as in Figure 3.4 but with a lower initial dilution. In theory, the initial growth rate should be the same but then increase to a lower maximum compared to the previous growth rates in Figure 3.4 (as

nutrients are depleted faster for lower dilutions with higher initial cell densities). However, this data set also provides more reliable data points for the initial time points, showing that the very initial growth rates are in fact higher than estimated by the Gompertz curve fitting in Figure 3.4. Figure 3.8 compares the fitted growth rates from the Gompertz model of the same samples with two different dilutions (16x fold dilution from Figure 3.7 compared to 47x fold dilution from Figure 3.4).

It has previously been argued that the Gompertz model has a definite curvature around the inflection point of the sigmoidal curve which causes too high estimates of the maximum specific growth rate, as well as the lag time [17, 22]. The model discussion above refers to the Gompertz curve having too high rate increase of the growth rate in the early stages of growth. A derivative of the growth rate which is too high would lead to a maximum specific growth rate which is too high, and possibly also a lag-phase which is too long as the growth rate then can increase too rapidly after a prolonged period of near stagnant growth (*i.e.* rater than the growth rate having to start increasing earlier to reach its peak value). In this respect, the model predictions are in agreement with previously observed inconsistencies of the Gompertz curve compared to data [17, 22]. Meanwhile, the model together with the points and further assumptions discussed above puts these behaviour of the Gompertz curve in a biological context connected to microbial growth. Furthermore, the Gompertz function is not based on microbial considerations [175, 22], but was originally developed based on ecological considerations (in particular, the life expectancy of humans during the 19th century [76]).

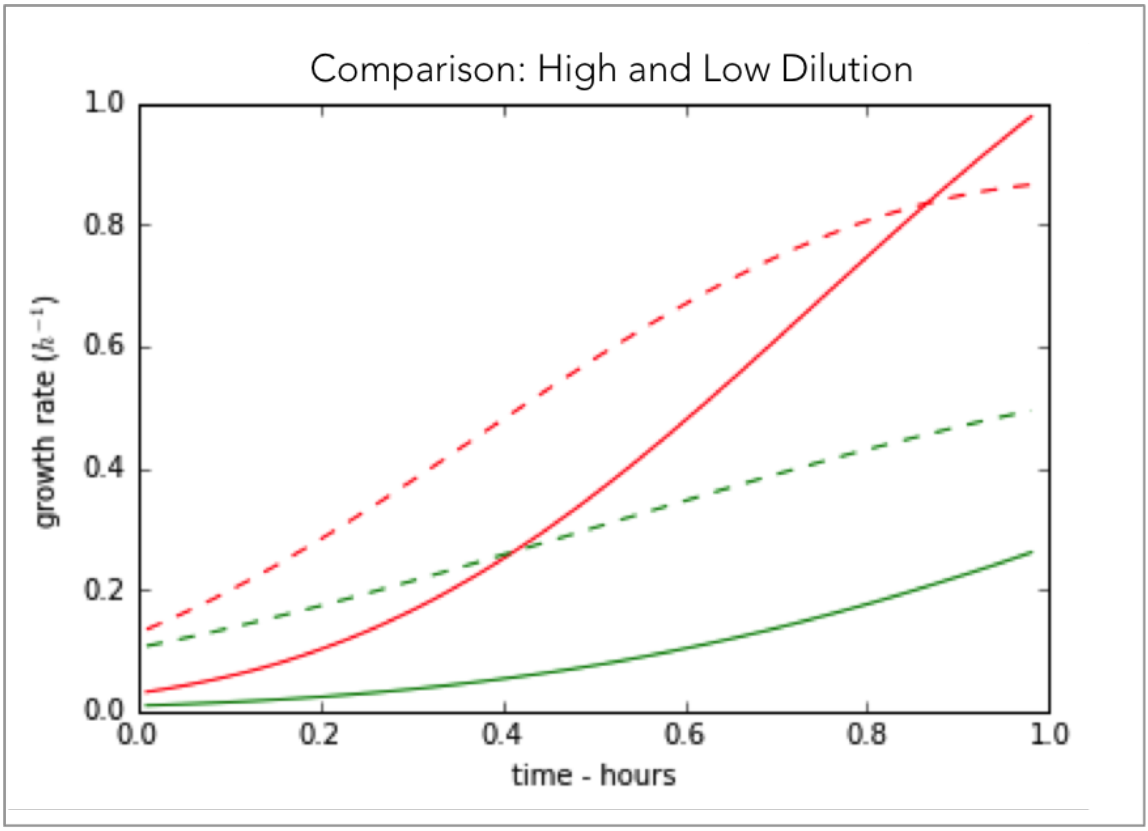


Figure 3.8: Plots the initial growth rates from Gompertz fitting in Figure 3.4 (solid lines) and the initial growth rates from Gompertz fitting from lower dilution in Figure 3.7 (dashed lines). The plot shows how initial growth rates are fitted to lower values at higher dilutions. Green: Construct ProD_B0035. Red: Empty Control.

3.4.2 Fitting of DRAM directly to cell growth measurements

By solving the differential equation in the Dynamic Resource Allocation Model (previously derived Equation (3.4)) an equation as a function of time is produced which can be fitted against data. The initial part of the method (Equation 3.7) makes the assumptions that the cells: (i) respond to a high level of ribosome expression when faced with excess nutrients, (ii) maintain a maximum level of protein expression until ribosomes no longer are the growth bottleneck (*i.e.* until the end of μ_L) and (iii) cells respond instantaneously to the changing environment by altering their expression levels. Because of these properties, this initial version of the model will be referred to as a bang-singlar response version of the model. The next two versions of the model can be referred to as a delayed bang-singular response and a delayed bang-bang-singular response. The initial assumptions are modified later on with the next iterations of the model, but they serve to highlight the basic principles of the model, as shown below.

Extending from Equation (3.3):

$$\frac{d\Phi_x}{dt} = (R_x - \Phi_x)\mu$$

Note also that

$$\mu = \frac{d\ln(OD)}{dt}$$

We consider the case in which R_x has a constant value. This could correspond to a particular growth phase. In particular, if R_x follows an immediate step function in regulation at transition from one growth phase to another (*i.e.* R_x has a constant value at a given growth phase), then during that growth phase:

$$\frac{d\Phi_x}{dt} = (R_x - \Phi_x)\frac{d\ln(OD)}{dt}$$

which can be solved using separation of variables:

$$\frac{d\Phi_x}{R_x - \Phi_x} = d\ln(OD)$$

$$\int \frac{d\Phi_x}{R_x - \Phi_x} = \int d\ln(OD)$$

$$-\ln(R_x - \Phi_x) = \ln(OD) - C$$

Set $k = e^C$

$$R_x - \Phi_x = \frac{k}{OD} \tag{3.6}$$

During the ribosome limited ramp-up period until the deceleration of exponential phase, we have $\Phi_R = \frac{\mu_L}{\gamma_0}$. Note that $\mu = \frac{d\ln(OD)}{dt} = \frac{1}{OD}\frac{dOD}{dt}$,

$$R_R - \frac{\frac{dOD}{dt}}{\gamma_0 OD} = \frac{k}{OD},$$

and

$$\frac{dOD}{dt} = \gamma_0 OD R_R - k\gamma_0. \tag{3.7}$$

This is an equation which can be fitted against data, without initially requiring to filter and extract growth rates or other 'modified' data sets for fitting (*e.g.* growth rate extractions).

3.4.3 Static R_R DRAM fitting from μ_L data

The algoritm for fitting the Dynamic Resource Allocation Model to single curve data proceeds in two steps: (i) finding a time point for the end of 'ramp-up period', defined as the first point at which growth rate is no longer increasing, (ii) fitting R_R and $\Phi_R(0)$ to the full shape, up until the end of the ramp-up period.

As an initial example, a flat and constant value of R_R can be fitted with the DRAM, as shown in Equation 3.7 and Figures 3.9 and 3.10. Figure 3.9 shows two different methods of selecting data for fitting: anchored and normal (or non-anchored). The anchored data set simply means that an additional data point is added at time zero in ENV_2. The additional initial data point is the last cell density reading from the previous environment before dilution (ENV_1), and divided by the level of dilution. Cell density absorbance readings are more accurate at higher cell densities compared to low cell densities. Including the last reading from ENV_1 as a starting point in ENV_2 reduces the reading noise from the

very initial readings at low cell densities in ENV_2. The anchoring method is used for all subsequent fittings of ENV_2 data sets.

The reduced values of R_R for cells expressing engineered constructs (seen in Figure 3.10) are due to reduced level of expression of ribosomes from competition of expression with the engineered construct. R_R is the fraction of translational capacity currently expressing ribosome proteins. (If assuming that the translational efficiency is equal between all mRNA, then R_x is equal to the fraction of ribosomes bound to mRNA of x.) The fitted R_R values in Figure 3.9 (displayed in Figure 3.10) refer to a static value for R_R during the entire growth ramp-up period, *i.e.* until the specific growth rates decelerates. An entirely flat value for R_R during the growth ramp-up period is likely not the most plausible assumption, even if predicted the most favourable response from control theory predictions [74, 75]. Later on it will be showed that further improved interpretations of the growth behaviour can be predicted by a delayed response of R_R, as opposed to a current static value for R_R.

The parameter k

The only free parameter in Equation (3.7) is k. This can be obtained through fitting to data, or by specifying and providing additional initial parameters to the system. For example, k can be applied to supply an initial state of Φ_R, if known; or Φ_R can be assumed carried over from the previous growth phase. If information is known at $t = 0$, then using Equation (3.6):

$$k = OD(0)\Big(R_R(0) - \Phi_R(0)\Big) \tag{3.8}$$

3.4.4 R_R delayed response for DRAM fitting from μ_L data

Cells require time to adjust themselves to process signals and adjust their transcription as well as translation in a new environment. Therefore, it is justified to assume a delayed response in translation as well as transcription levels in response to new cell environments. For the specific example of increased ribosome expression, or rRNA transcription, a relaxation of the ppGpp mediated stringent response is necessary in response to higher nutrient

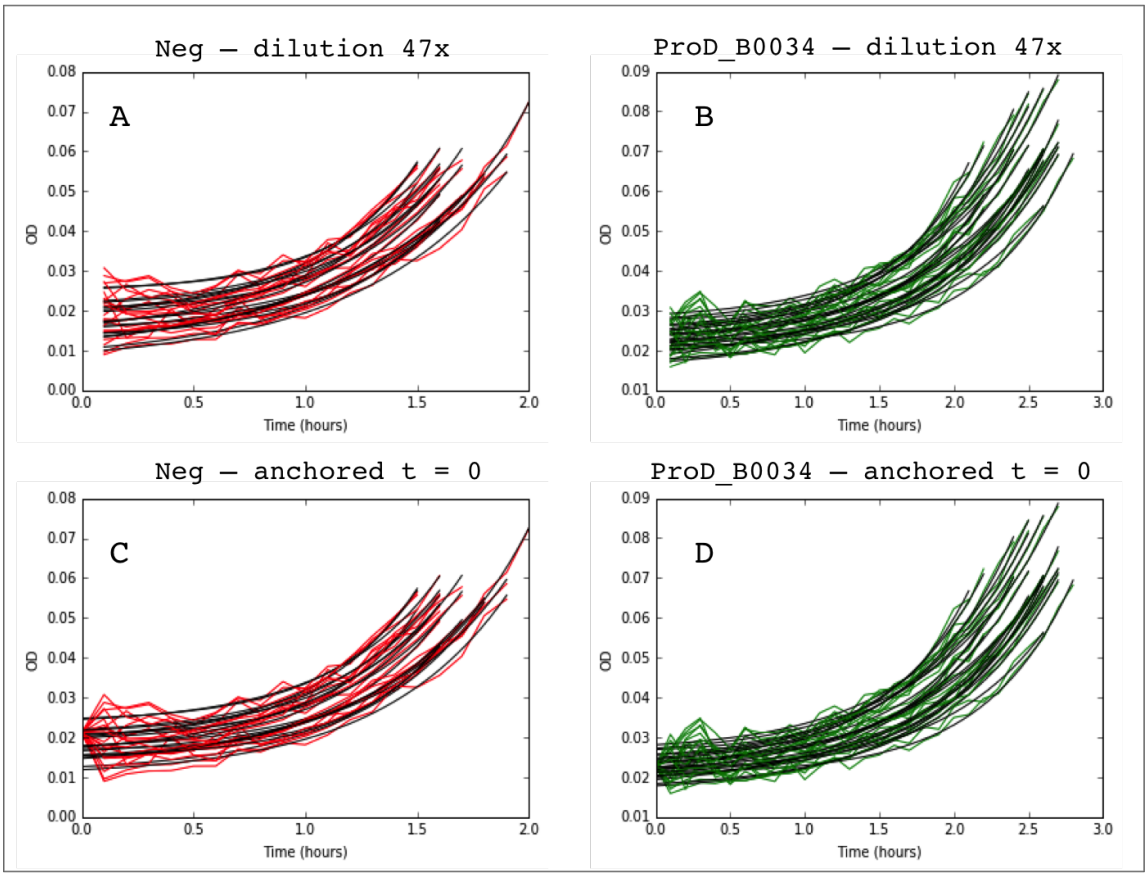


Figure 3.9: The Dynamic Resource Allocation Model fitting (black lines) for the full set of growth curves in experiments with samples being diluted 47x after a 24h pre-culture. Neg are the negative control (red lines), *i.e.* no construct expression on the medium copy number plasmid, which is the same plasmid copy number as for ProD_B0034 (green lines). A and B shows fitting to the μ_L growth ramp-up period (i.e. full data until μ_{max} is reached). C and D show the same curves but anchored with an initial value at $t = 0$ which is taken as the last density value before dilution and divided by the dilution factor. The growth data is identical to the data for Figure 3.2 in this chapter fitted to the Gompertz growth curve.

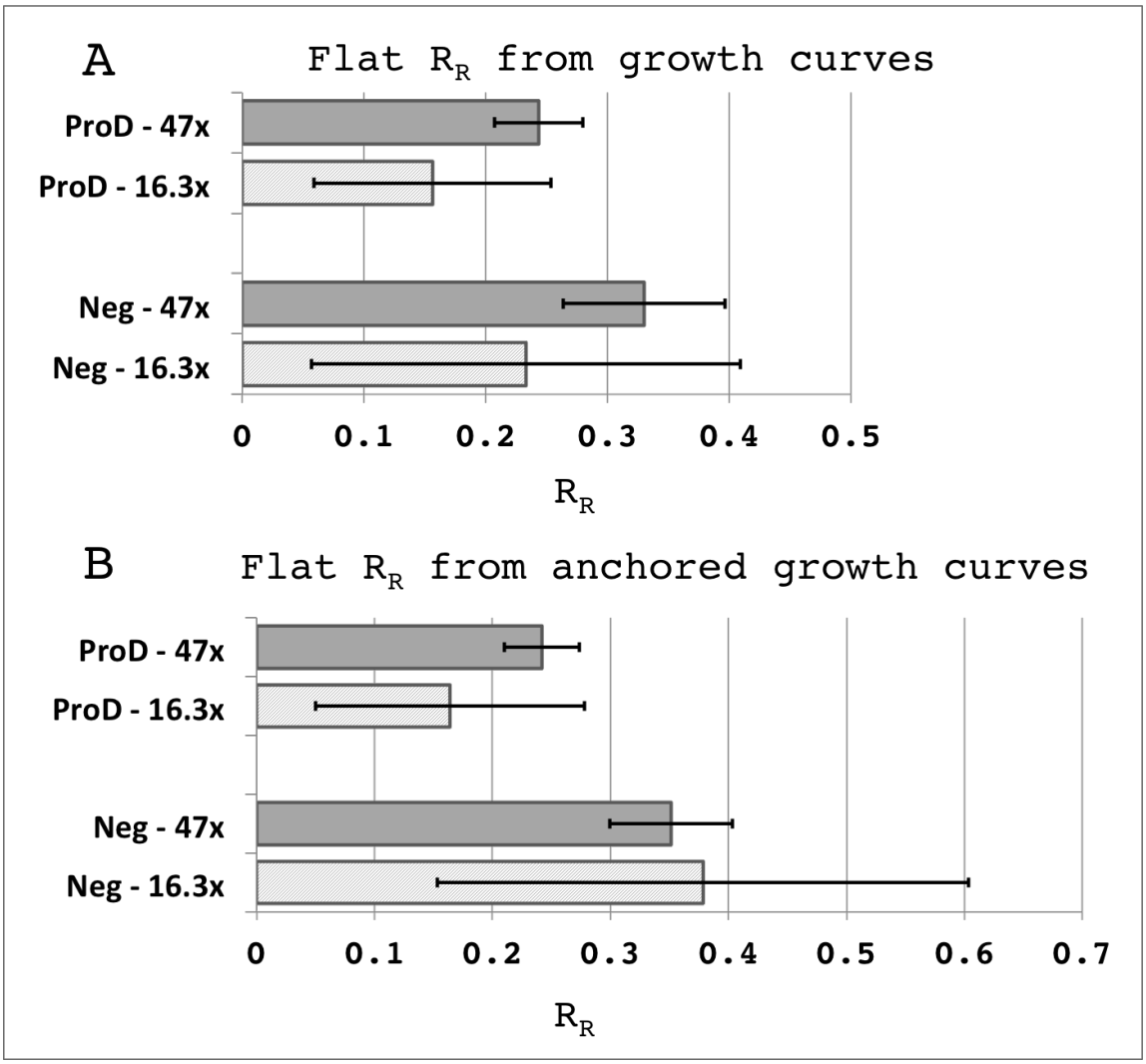


Figure 3.10: The interpreted flat R_R level from ramp-up growth curves (A) as well as fixed ramp-up growth curves (B) obtained from the DRAM fitting of the complete set of growth curves in Figure 3.9. The bars are the mean values for two different levels of sample dilutions, 16.3 and 47x during the same experiment. The error bars represent the standard deviation. ProD are the ProD-B0035 with a medium copy number plasmid. Neg are the negative control, i.e. no construct expression on the medium copy number plasmid. Note, in Figure 3.9 only the data for 47x dilutions are shown, the fittings for the 16.3x dilutions are made with the identical method.

levels [115, 196, 101]. Figure 3.11 shows the conceptual difference between assuming an instantaneous or delayed response by R_R in a new environment. The term t_s refers to a delayed time point by which cells make a step change in expression of ribosomes in order to adjust themselves to a new more favourable nutrient environment.

In order to fit a delayed step function for R_R to data some modifications have to be made to Equation (3.7). After long periods of growth in a constant environment Φ_R and R_R are assumed to reach equilibrium. After shorter durations of stationary or decelerating growth Φ_R is assumed to be larger than R_R, as Φ_R is then in excess of what is required by the cell. To modify Equation (3.7) to an initial state of Φ_R and R_R equilibrium, the k parameter can be targeted via Equation (3.8). The initial, pre-step-change phase of the delayed R_R response, then becomes a simplification of Equation (3.7) as $k = 0$.

$$\frac{dOD}{dt} = \gamma_0 OD\Phi_R(0) \tag{3.9}$$

Even if the cells originate from a state with a shorter previous history of stationary phase (or still declining growth rate), such that $R_R(0) < \Phi_R(0)$, assuming that $R_R(0) = \Phi_R(0)$ simply means that R_R has a negligible impact on the initial growth rate. After a delayed time period, t_s, a step increase in R_R occurs and the growth equation resumes the form stated in the initial Equation(3.7).

Fitting the *delayed response* Dynamic Resource Allocation Model to growth data shows that hosts which are carrying engineered constructs reach a lower peak level for the expression of ribosomes R_R (see Figures 3.12E,F and 3.12A). It can also be seen that the response to increase the expression of ribosomes is delayed for organisms expressing engineered constructs (t_s in Figure 3.12B). Further, it shows that the populations with a construct have a lower concentration of ribosomes already from the start of growth in ENV_2 (see Figure 3.13C). The hosts that carry an engineered construct have been expressing the same construct in ENV_1, and the expression of the construct during this period has competed for resources with ribosomes, as well as other host proteins, resulting in a lower initial level of ribosomes by the start of ENV_2. After the end of μ_L in ENV_2, and during the last phases of growth phases, the growth is not limited by ribosomes, but instead by the avail-

ability of nutrients. This means that having a lower concentration of ribosomes (compared to negative controls) is invisible from growth, and the limitation of R_R does not manifests itself until a shift to higher nutrient levels.

It is interesting to note in Figures 3.12 and 3.13 that not only is the R_R reduced from the presence of engineered construct, as also was indicated previously if assuming an immediate response of cells to a new environment. But the response time to start expressing more ribosomes is also delayed for cells hosting a construct, seen by an increase in t_s. The fits are improved in Figure 3.13 compared to Figure 3.10, but at the same time the model complexity has also increased. The main reason for increasing the model complexity is the biological assumption that cell need time to respond to a new environment.

An estimated 10-20% of proteins expressed in *E. coli* cells are expressed only to be prepared in case of a rapid change in environmental conditions [147]. Hypothetically, the imposed requirement on host keep engineered construct is competing for the space of spare resources the cells keep in order to be prepared for alternative or changing environments. In that case t_s could also be a measure which captures the competition for resources of response readiness of hosts, a cost from engineered biology which is all but hidden in a constant environment.

In Figure 3.12, the $\Phi_R(0)$ value has been fitted to the DRAM from the growth data. To control that the increased levels of t_s from engineered constructs are not an effect fitting factors affecting $\Phi_R(0)$ a test can be made to freeze the assumed input value $\Phi_R(0)$ between both sets of samples. *I.e.* that is the same as assuming that starting value for Φ_R is equal for organisms with and without a construct, in other words, that the concentration of ribosomes is the same within the two different populations after growth completion in ENV_1. If assuming an initial $\Phi_R = 0.03$ for all samples at the start of growth, then the mean of the best fitted delayed response by R_R would be 0.3 minutes for samples without a construct, versus 23.4 minutes for samples with a construct. In comparison, the fitting values shown in Figure 3.13B, with a best fit $\Phi_R(0)$ value, give a delayed response of carrying construct of $t_s = 12$ minutes. Thus, the differences in t_s would increase rather then decrease if assuming an equal $\Phi_R(0)$ between both sample sets, rather than assuming an $\Phi_R(0)$ according to best fit of the data. Therefore it can be assumed that fitting $\Phi_R(0)$

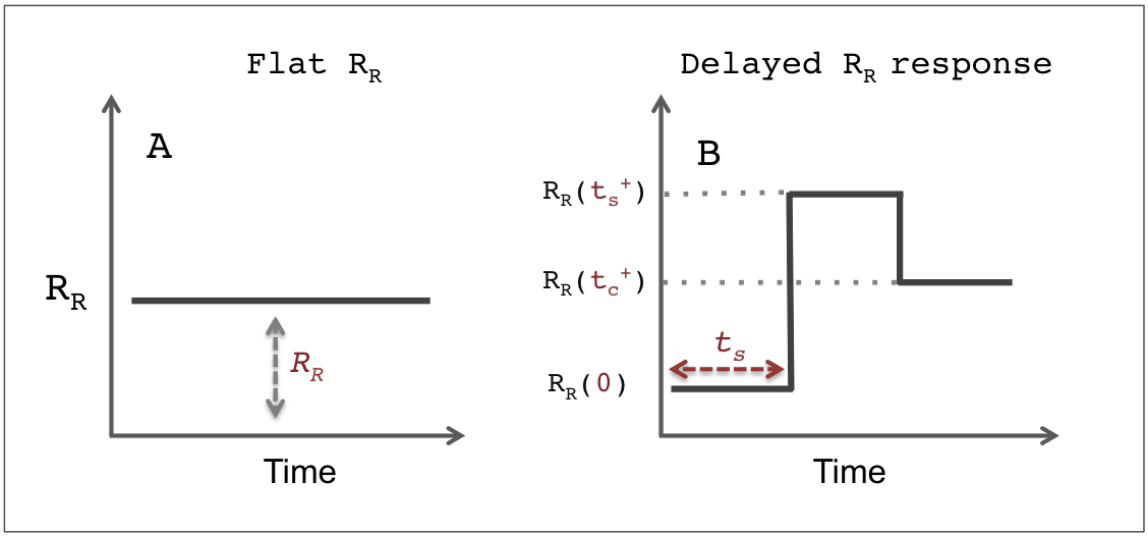


Figure 3.11: Difference in model behaviour between assuming an immediate full R_R response (A) and a time delayed R_R response (B). Both model assumptions can be aligned with the bang-bang control assumption. The delayed response assumes there is an inherent delay in how fast the cell is able to respond to the new environment, a delay which might be increased by the presence of burden from engineered protein fractions. $R_R(0)$: assumed to be the legacy level from the previous environment. t_s: the response delay to step increase in expression. $R_R(t_s^+)$: the step change response in expression according to the new environment. $R_R(t_c^+)$: ribosome expression after adjusting to the maximum growth rate in the new environment.

to the model, rather than assuming a fixed $\Phi_R(0)$ from the start, does not increase the difference in the fitted t_s values between the construct and the negative control.

R_R as a bang-bang response

Despite the substantial amount of work on microbial growth, little is known about the dynamics of proteome change and expression strategies [138, 74]. Previous theoretical work has suggested that the most beneficial environment acclimitisation for cell populations is to overcompensate the expression of a required cellular component in a new environment [74, 75, 138, 119]. According to this theory, to reach most optimal ratios of cellular components for a new environment the cells would direct all their cellular resources towards a component that is undersupplied and express none of the other components until a new optimal internal state is reached. This control method is referred to as 'bang-band control' [75], and would minimise the time required to reach optimal growth in a new environment. However, the bang-bang strategy would rely on very accurate and fast response mechanisms in order to

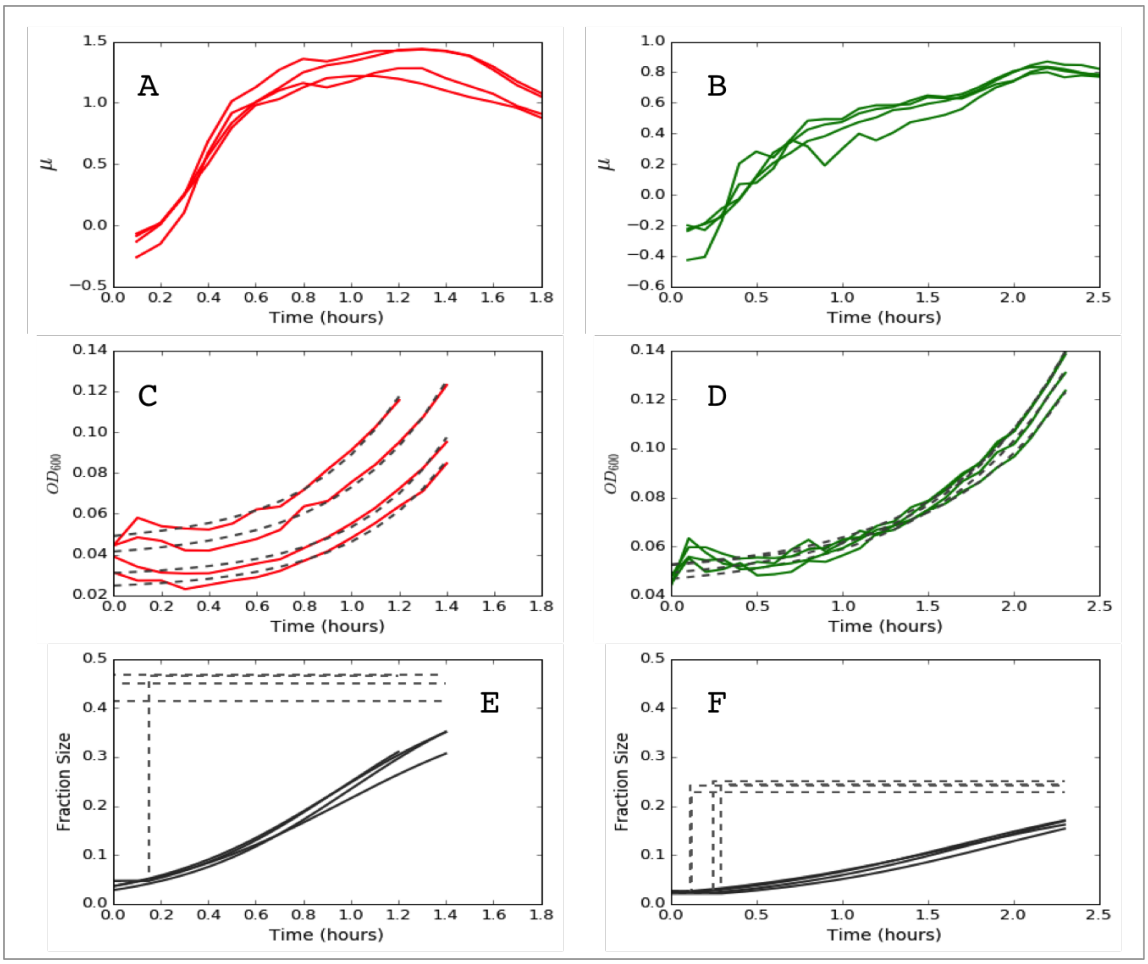


Figure 3.12: Fitting of the DRAM with delayed R_R response to samples with (green) and without (red) a construct. The data shows the ramp-up period in ENV_2 after a 25 times dilution from ENV_1. **A & B:** The growth rate of samples without (A - red) and with (B - green) a construct from start of growth in ENV_2 following a 25 times dilution from colonies grown for 26 hours under the same conditions in ENV_1. The period between time = 0 until maximum growth rate (A and B) provides the time period for μ_L, which limits the period of the fitting (in C and D). **C & D:** Fitting of the delayed R_R DRAM (dashed lines) to the measured cell density data during μ_L (solid lines). **E & F:** The Φ_R (solid lines) and R_R (dashed lines) fractions extracted by fitting of the model. The construct shown is ProC-B0034 on the medium copy plasmid (green). The negative controls carry the same medium copy plasmid without a construct (red).

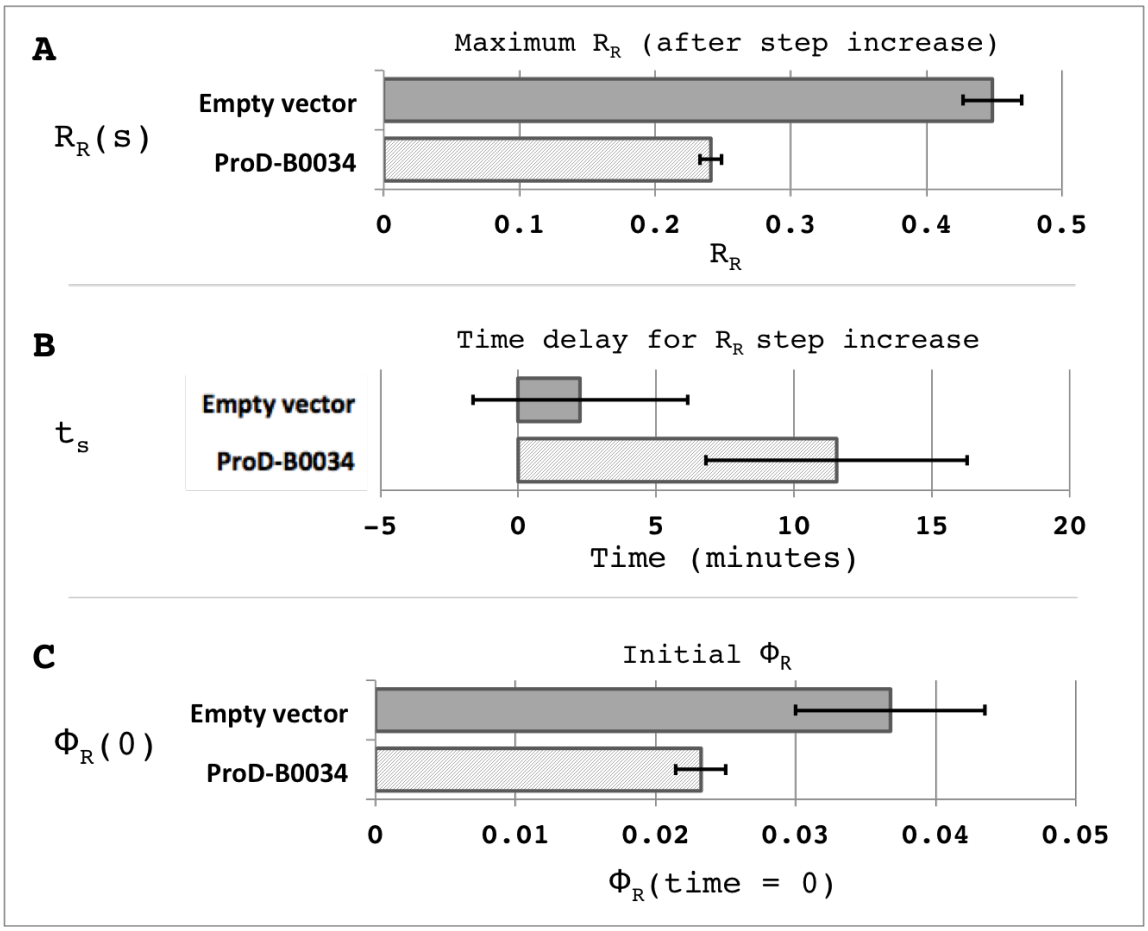


Figure 3.13: Comparison of parameters for samples with and without construct for the delayed response Dynamic Resource Allocation Model. The model parameters shown are fitted to data in Figure 3.12. **A:** Expression level of ribosomes after a delayed response to a new environment. **B:** The delay of response time before the step increase in ribosome expression (defined as t_s). **C:** The initial ribosome fraction within the cells at the start of growth, after sample dilution into a nutrient-rich environment.

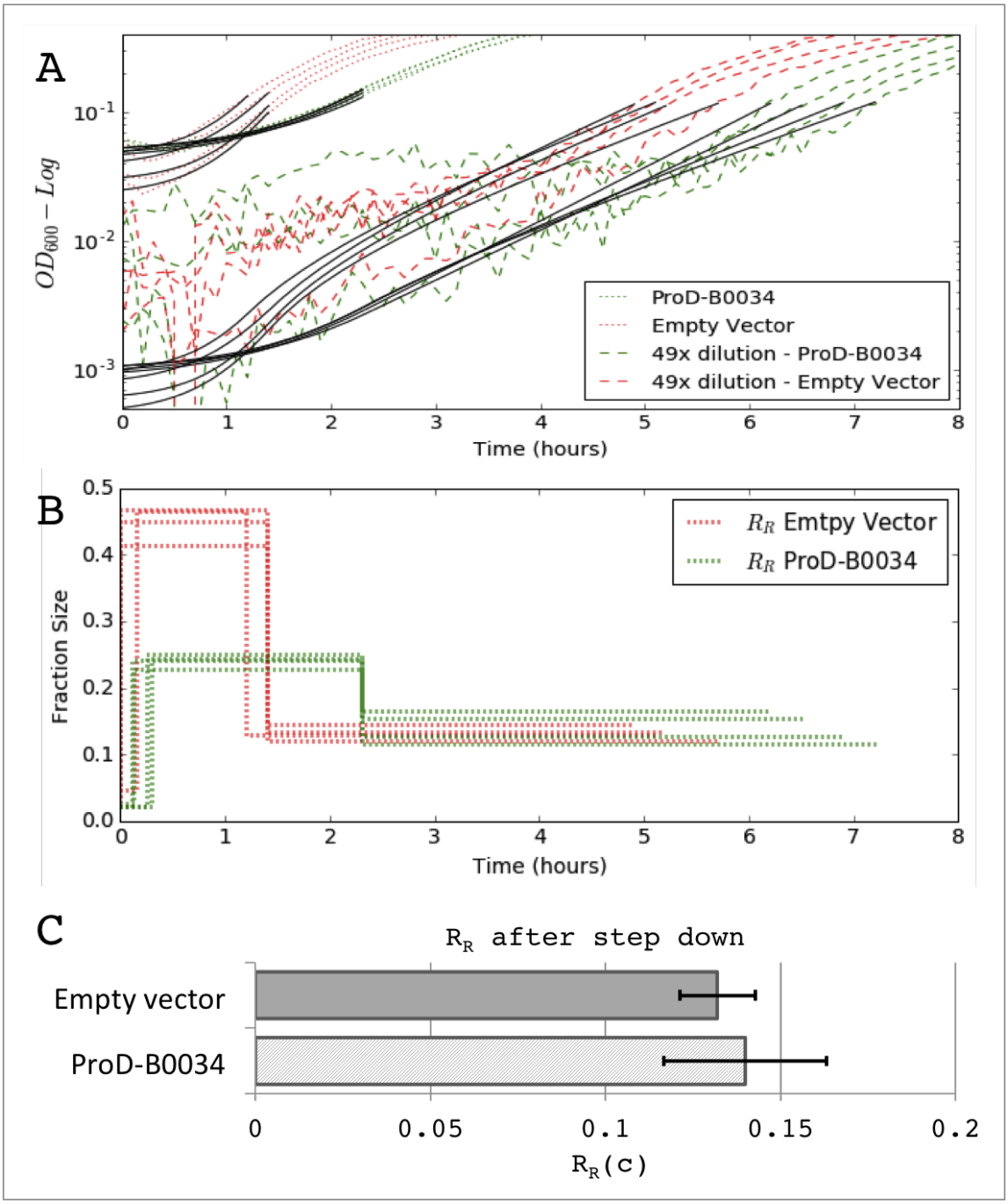


Figure 3.14: Fitting of DRAM to data samples diluted twice in order to capture the level of R_R after the initial growth adjustment has been made to reach μ_{max}. **A:** The parallel growth of samples in ENV_2 where the duplicate of the original sample has been diluted a further 49 times. Solid lines show fitting of the DRAM and dashed lines show measured growth data. **B:** The expression fraction of ribosomes (R_R) extracted from the fitting of the DRAM to the full data set of both dilutions. The initial values of R_R, denoted $R_R(0)$ and $R_R(t_s^+)$ are obtained from the first diluted samples. The subsequent constant level of R_R at μ_{max} equilibrium expression, called $R_R(t_c^+)$, is obtained by combining the initial shape of the growth curve with the subsequent growth of the duplicates that have been further diluted. **C:** The mean values for $R_R(t_c^+)$, showing that the growth behaviour of host with and with engineered constructs do become very similar after the initial adjustment period of cells to growth in a new environment.

respond, an assumption that might be limited by physical constraints.

In order to interpret calculated values of R_R from the Dynamic Resource Allocation Model over longer time periods of growth simultaneous comparisons were made between samples of different dilution levels. In Figure 3.14 samples with and without constructs were diluted first 25 times, and then immediately 49 times again. The initial parts of the growth curves and DRAM fittings are taken from the growth data of the first dilution, and then fitted back to the second dilution.

$R_R(t_c^+)$ is the expression of ribosomes at constant conditions of exponential growth, and after a transition period in a new environment whereby an equilibrium is reached between R_R and Φ_R (assuming that the external environment remains constant). Therefore, after initial adjustments have been made to a new environment with excess nutrients, R_R is also proportional to the exponential growth rate.

Interestingly, the combination of fittings to different sample dilution levels (as shown in Figure 3.14), indicates that the level R_R initially increases to a very high level during a transition period, but then decreases substantially. Even if R_R rapidly reaches a very high value in order to fit the first set of less diluted samples, R_R also needs to be reduced rapidly in order to fit the samples of higher dilutions. This observation of over reacting response of R_R is in agreement with the bang-bang control regulation, which has previously been suggested for resource allocation in bacteria with theoretical arguments [74].

Further, according to the data obtained from this DRAM fitting, the expression levels of ribosomes and the growth rate of organisms are very close to each other after equilibrium has been reached for growth in a new environment. This relationship can be seen in Figure 3.14, which shows a comparison of $R_R(t_c^+)$ values for hosts with and without burden from engineered expression. The $R_R(t_c^+)$ values for hosts with and without constructs are very similar to each other, in particular compared to the other new metrics for burden shown previously, *e.g.* t_s and $R_R(t_s^+)$. The observation that $R_R(t_c^+)$ values are very similar to each other for hosts with and without constructs strengthens a previous observation made that the burden of engineered biology is reduced after a few generations of exponential growth [168] - an observation which highlights that μ_{max} is an imprecise measure of burden. The effect of small impact on $R_R(t_c^+)$ from engineered burden shown by the fitting of the

DRAM helps to explain why comparisons of maximum growth rates (μ_{max}) are non-ideal and error prone measures for burden. Particularly when compared to either t_s or $R_R(t_s^+)$, which relate to the transition phase, rather than long term equilibria.

The computational method shown here is using two sets of dilutions from an initial single set of sample cultures as the data input. However, more involved and labour intensive laboratory techniques could allow for less computations from a single set of dilution samples from sample cultures. One such method would be to culture samples in a chemostat. Another method could be to use batch culture, with a single very high dilution. Meanwhile, it would be possible to count the number of live cells from frequent sampling (by plating or microscope), instead of using the much faster but more error prone method of optical density measurements.

The model and computations as demonstrated here are not restricted to these more labour intensive laboratory techniques. But it is the intention that the model and the general computational procedure should be possible to apply to as broad data sources as possible. This includes the frequently used experimental method of growing samples in batch cultures with rich media together with turbidity measurement data.

Comparison of the three fitting procedures

As can be seen in Figure 3.15 the fitting of the delayed bang-bang-singular model is an extension of the model fitting for the delayed bang-singular model, but with more data. The fitting of the delayed bang-bang-singular model uses the input values of $\Phi_R(t_c)$ and $OD_{600}(t_c)$ from less diluted duplicate samples fitted with a delayed bang-singular model. The fitting of the bang-singular and the delayed bang-singular response applies least squares fitting. However, for the extension of the delayed bang-singular response to the delayed bang-bang-singular response the fitting of the model only uses an extrapolated growth line between two cell density points.

One key rationale for using data from diluting a single sample with two different dilution fold magnitudes in parallel in a single experiment is that there will be a similar initial growth behaviour for both of the two duplicates of the sample at the two different dilution folds. This rationale is assumed to be valid if the original sample comes from an environ-

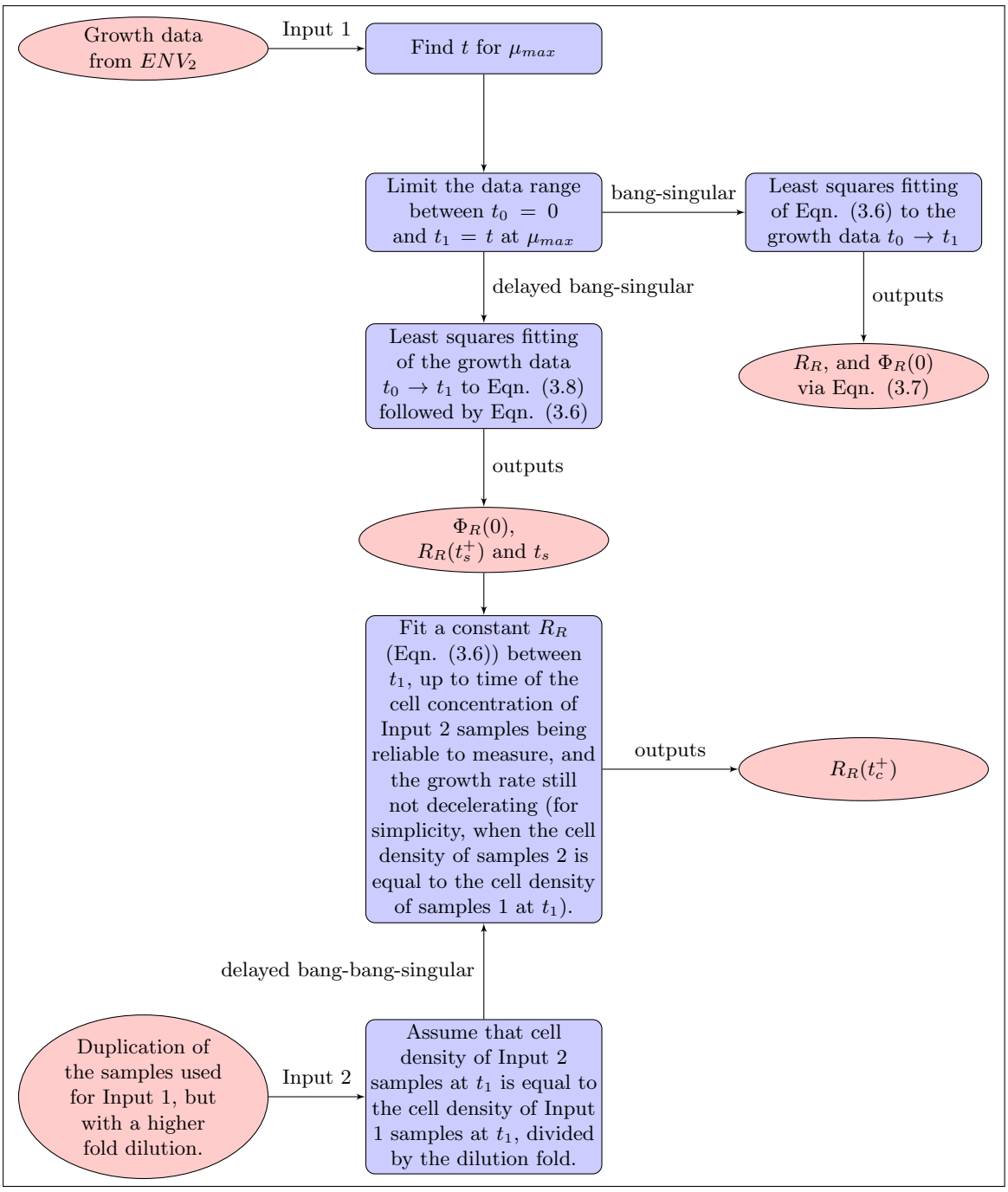


Figure 3.15: Flowchart comparisons of the three different models, input fitting data and model outputs. Note, the duplicate samples of Input 2 grow in parallel to the samples of Input 1. In other words they are also from growth in ENV_2, and are the duplicates of the same samples as Input 1, but at a higher dilution fold. The assumption of the growth made for the Input 2 samples is that they initiate growth in the same way as the Input 1 samples (the cell densities of these duplications with a higher dilution fold are too low to record accurately with turbidity measurements). The rationale for this assumption is that both sets of samples are limited by their ribosome fraction size up to t_1, when the growth rate peaks for the Input 1 samples.

ment with is nutrient limited (here manifested by the stagnant growth of the samples in ENV_1 and after 26 hours of incubation in their growth medium which has been exhausted), and the two duplicate samples are both inoculated into identical environments with much more favourable growth conditions. Then both samples will initially be limited by their ribosome content and their growth will initially accelerate accordingly. After some period of time, the sample duplicate which has been less diluted (and therefore starts at a higher initial cell concentration) will have consumed more nutrients and be more limited by the nutrient content in the medium compared to the sample duplicate with a higher dilution fold (and therefore a lower initial cell concentration). At that time point the growth rate will stop to accelerate, or accelerate more slowly compared to the duplicate with a higher dilution fold. The time point at which the less diluted sample stops to accelerate is $t(c^+)$ (also written as t_1 in Figure 3.15). In Figure 3.14 the duplicate samples with a higher dilution fold are assumed to have the same growth profile as the less diluted samples up to this time point of the first growth deceleration.

The growth of the high diluted samples (Input 2) is assumed to have initiated with the same growth behaviour as the less diluted samples of Input 1 up to time point t_1. The end point of fitting the growth line (in Figure 3.14) for the high diluted samples (Input 2 in the flowchart Figure 3.15) is when the cell density reaches an optical density which is high enough to be accurately recorded, but not so high that the growth rate is already decelerating. The optical density used for the fitting in Figure 3.14 is $OD_{600} = 0.05$. In other words, for the three constant phases shown graphically in Figure 3.14B, the two earlier phases are also shown in Figure 3.13. They were obtained via least squares fitting of the delayed bang-singular model in Figure 3.12. The $R_R(c)$ value produced via the fitting in Figure 3.14 is obtained by assuming the same growth of the samples during the two earlier phases of growth for the low (25x) diluted samples and the high ($25 \times 49 = 1225$) diluted replications of the samples.

From the fitting in Figure 3.12 the initial starting values of $\Phi_R(t_c)$ and $OD_{600}(t_c)$ have been provided. The $OD_{600}(t_c)$ for the high diluted samples (annotated Input 2 in the flowchart Figure 3.15) are assumed to be equal to the $OD_{600}(t_c)$ cell concentration of the low diluted samples (annotated Input 1 in the flowchart Figure 3.15) divided by the dilution

fold (*i.e.* $OD_{600}(t_c)$ divided by 49 in the fitting in Figure 3.14). If assuming a constant value for $R_R(t_c^+)$ (*i.e.* between t_1 and t_2) then only a single value for $R_R(t_c^+)$ satisfies the growth between $OD_{600}(t_1)$ up to $OD_{600}(t_2)$.

The $R_R(c)$ is fitted between a single optical density at $t_1 = t_c$ (*i.e.* when the fitting of the samples in Figure 3.12 ends) divided by the extra dilution fold (*i.e.* 49x in Figure 3.14) up to a single optical density value at a time point of reliable optical measurements for the high diluted samples (annotated t_2 in the flowchart Figure 3.15).

3.4.5 Initial Φ_R levels and correlations from ENV_1 to ENV_2

The current method shown provides metrics and output parameters for how Φ_R and R_R respond to more favourable growth conditions as a measure for burden. However, what the model so far does not show is how these fractions behave in conditions where growth rates are declining. No investigations have previously shown the effect of burden on the internal environment of hosts during growth decelerations. If it was possible to inform and predict the rate of decline of Φ_R during growth decelerations or stationary phases as an effect of engineered biology constructs then this information would be very useful to predict how different constructs will behave in a host over longer periods of time.

3.4.6 Parameterisation of the decline of the Φ_R fraction

In order to parameterise the decline of the Φ_R fraction in populations with and without a construct, an initial indication of behaviour was sought from relationships of when populations initiated growth in ENV_1 to how they grow in ENV_2. If a cell population started growth early, it would then have longer time to reduce its ribosome fraction before being diluted, and that is the hypothesis for finding a relationship from ENV_1 to ENV_2. Thus, it would take longer for these 'early growth' samples to start growing after dilution, and this would occur even if late growers and early growers were diluted at very similar population densities and growth rates.

In Figure 3.13 it is seen that the fitted model predicts a markedly reduced initial ribosome level ($\Phi_R(0)$) in cells with a construct compared to cells without a construct. However, if relating the $\Phi_R(0)$ back to the history of individual samples in ENV_2 another

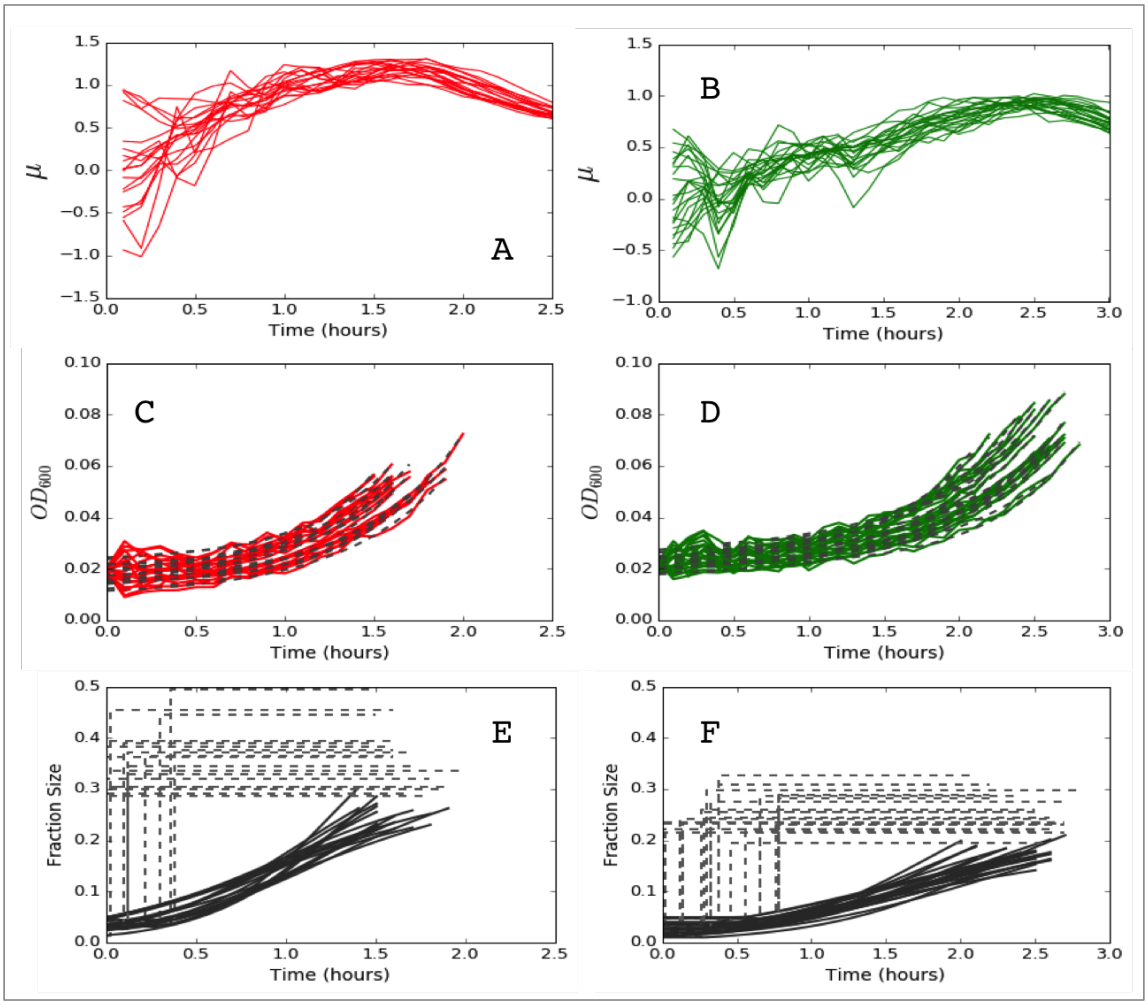


Figure 3.16: Fitting of the Dynamic Resource Allocation Model with delayed R_R response to samples with (green) and without (red) a construct. The figure shows the same method for fitting data as in Figure 3.12, but with (i) more samples, (ii) a more strongly expressed construct and (iii) populations diluted 47 times from ENV_1 instead of 25 times. **A & B:** The growth rate of samples without (A - red) and with (B - green) a construct from start of growth in ENV_2 following a 25 times dilution from colonies grown for 26 hours under the same conditions as in ENV_1. The period between $t = 0$ until maximum growth rate (A and B) is used as μ_L, which limits the period of the fitting (in C and D). **C & D:** Fitting of the delayed R_R DRAM (dashed lines) to the measured cell density data during μ_L (solid lines). **E & F:** The Φ_R (solid lines) and R_R (dashed lines) fractions extracted by fitting of the model. The construct shown is ProD-B0035 on the medium copy plasmid (green). The negative controls carry the same medium copy plasmid, but without a construct (red).

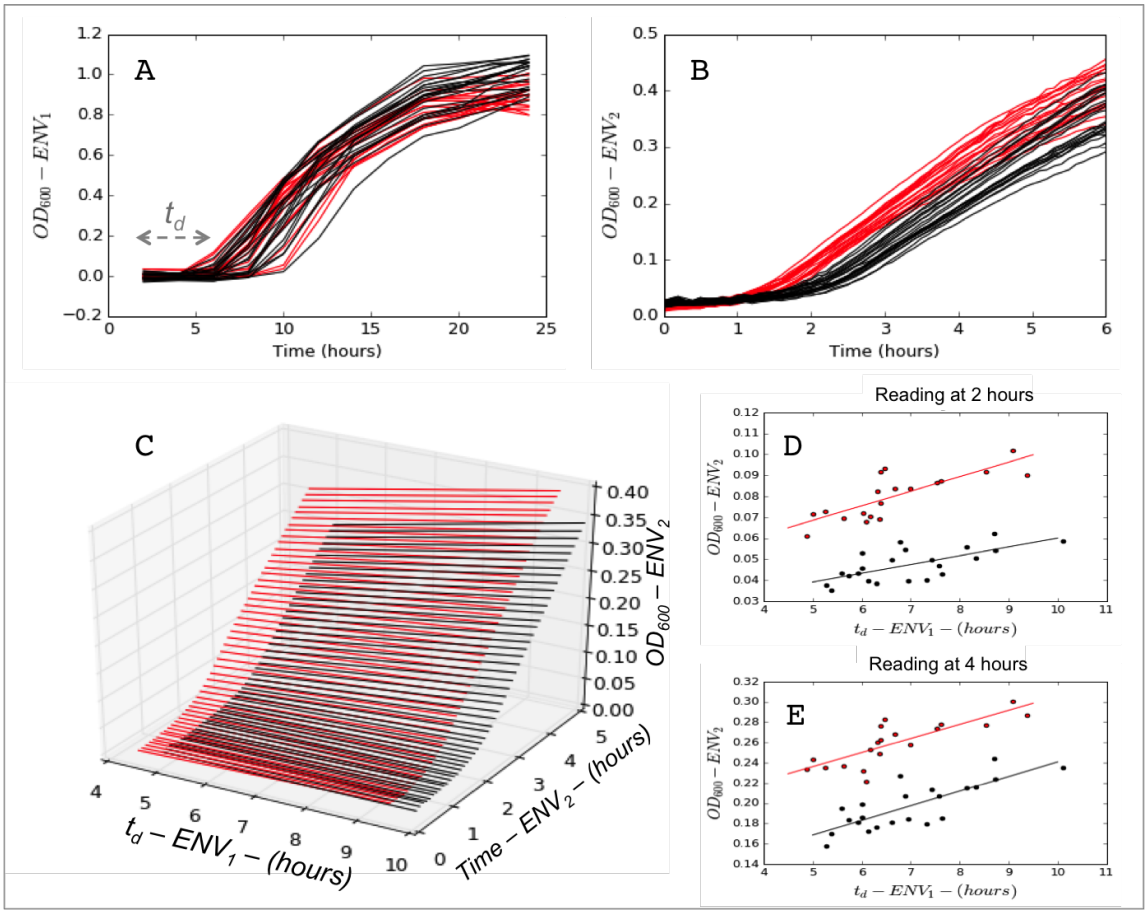


Figure 3.17: The extraction of a time variable (t_d) for start of growth in ENV_1 and comparison of the effect of that variable for growth in ENV_2. The experiment data shown is the same as in Figure 3.16. In all subplots, **red color** refers to the empty vector and **black color** refers to the construct, which is ProD-B0035. **A:** Growth in ENV_1, measured every second hour for 24 hours. The parameter t_d refers to the time delay of individual samples until they initiate growth in ENV_1. The time point at which a sample passes $OD_{600} = 0.2$ has been selected as the value for growth initiation. The time value is obtained from data with linear interpolation (between two time points before and after $OD_{600} = 0.2$). **B:** Growth in ENV_2, measured every 6 minutes. **C:** The correlation between t_d in ENV_1 and subsequent growth in ENV_2. For each growth reading in ENV_2 a linear relationship has been drawn between t_d in ENV_1 and OD_{600} in ENV_2, according to the method shown in D and E. **D & E:** Shows the method for linear fitting at time 2 hours and 4 hours, respectively (*i.e.* two of the lines shown in C). It can be seen that going from ENV_1 to ENV_2 the time of growth initiation is reversed. Samples that start growth earlier in ENV_1, and therefore have lower t_d, grow slower in ENV_2. This relationship can be explained by samples that grow earlier in ENV_1 have had more time, after passing μ_{max}, when Φ_R is declining (see Figure 3.18).

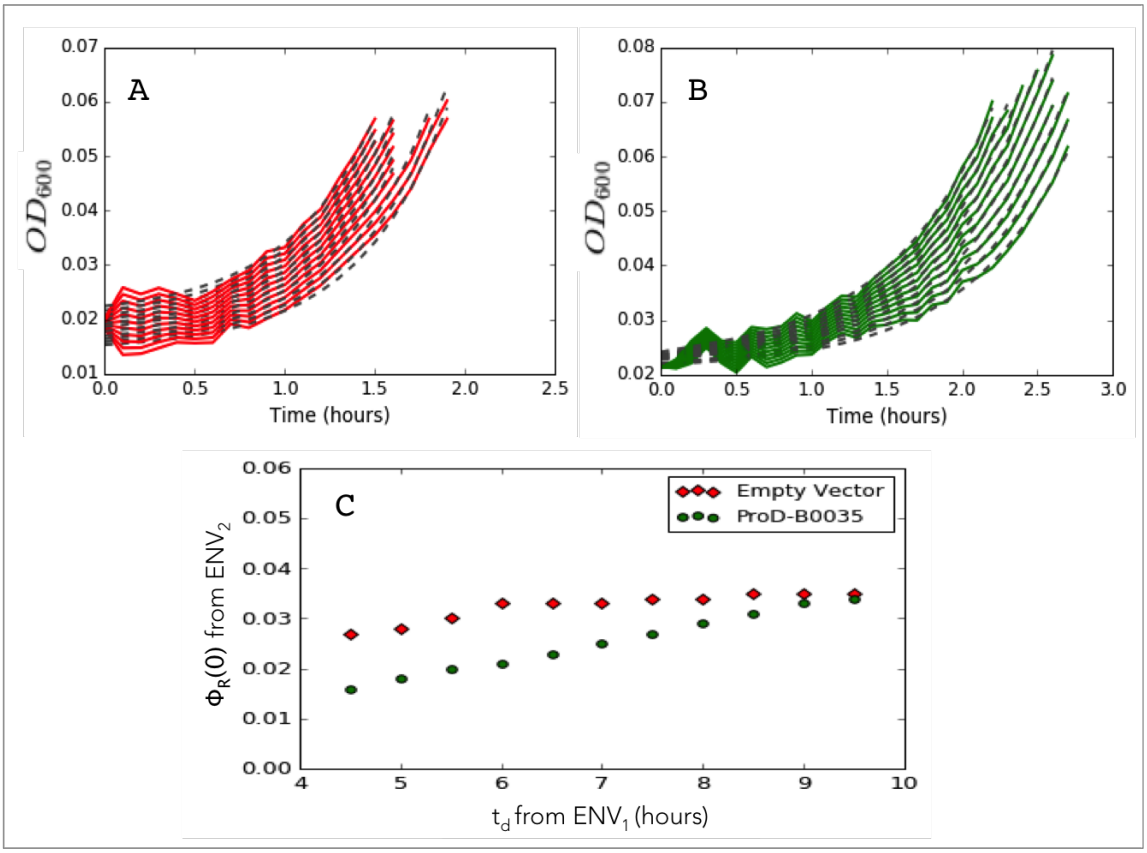


Figure 3.18: DRAM fitting to the t_d extrapolated lines drawn in Figure 3.17C. For each 0.5 hour time increase in t_d from the lines in Figure 3.17C a fitting to data has been made with the DRAM in order to extract the initial value of Φ_R by the start of growth in ENV_2. For the purpose of finding $\Phi_R(0)$, the parameters of t_s and R_R have been frozen at the previously extracted mean for the complete set of samples shown in Figure 3.16 ($R_R(neg) = 0.361$, $R_R(pos) = 0.252$, $t_s(neg) = 0.09h$ and $t_s(pos) = 0.369h$). **A:** Fitting by the DRAM to the Empty Vector data for each 0.5 hour time intervall of t_d in Figure 3.17C. **B:** Fitting by the DRAM to the construct data (ProD-B0035) for each 0.5 hour time intervall of t_d in Figure 3.17C. **C:** The initial ribosome fractions ($\Phi_R(0)$) at the start of growth in ENV_2, as extracted with the DRAM model (shown in A and B) and plotted against the t_d values in ENV_1. It is seen in that Φ_R is declining faster (after reaching μ_{max}, or t_d) for host expressing a construct (green - ProD-B0035) compared to samples without a construct expression (red - Empty Vector).

interesting relationship emerges. Data plotted in Figures 3.12 and 3.13 only covers a small set of samples. Therefore, in Figure 3.16 a larger number of data readings are shown and are fitted with the same method as in Figure 3.12 and Figure 3.13. The empty vector is the same for both figures, but the construct carried in Figure 3.16 has a stronger promoter (ProD) and RBS (B0035) than the construct in Figure 3.13 (ProC-B0034). The larger number of samples used for the experiment in Figure 3.16 allows the growth in ENV_2 to be correlated with data from ENV_1 prior to DRAM fitting. A clear correlation can be seen between the time point of growth initiation in ENV_1 and the $\Phi_R(0)$ level at the start of ENV_2 (see Figure 3.18). The inverse relationship between time of growth initiation in ENV_1 and the level of $\Phi_R(0)$ is logical as Φ_R starts to decline after the end of μ_L in ENV_1 [6, 143]. Procaryotes have previously been shown to follow a 'first in last out' behaviour with regards to growth [93], where the last individuals to stop growing are also the first ones to resume growth.

From a burden-perspective it is interesting to note that Φ_R (extracted from the model) declines more rapidly by the end of growth in ENV_1 for a host with a engineered construct, compared to cells without construct expression (see Figure 3.18C). Standard deviation error-bars have not been included on Figure 3.18C as these are output values from fitting of the model to previous least squares linear interpolations of cell populations commencing growth at different time points (t_d) in ENV_1 (the linear fits of two time points are shown visually in Figures 3.17D and 3.17E). The resulting effect is that cells with a construct not only accelerate their growth slower (as seen previously from a reduced R_R) but also have a lower initial growth rate when diluted to an ENV_2.

The correlation between t_d and $\Phi_R(0)$ from the fitted model shown in Figure 3.18C provides the rate of decline for Φ_R at the final hours of ENV_2. As highlighted previously, very little is previously known regarding the dynamic adjustments of protein fractions in bacteria [138, 74]. The method as shown in Figure 3.17 and Figure 3.18 enables the interpretation of the internal proteome fractions during the deceleration of growth rates. The effect shown of more rapidly declining R_R provides a new and additional parameter for burden. An burden effect which might have a significant impact on the traditional method of measuring *half-life* [32] through a series of multiple dilutions. In other words, the half-life

effect measured from multiple dilutions could be directly impacted from the time periods between each dilution. The effect of the length of these time-periods could be directly impacted from the original cost of the construct, such that some burden levels amplifies the cost for certain time-periods between subculturing events, and other combinations of time-periods and costs might diminish the apparent half-lives from measurements.

With the help of Equation (3.3) together with degradations rates, the dynamics of R_R during these late-stage growth-phases could also be interpreted (*i.e.* in addition to the Φ_R fraction). However, that research work is part of future investigations.

3.5 Conclusions and Contributions

Resource allocation in Synthetic biology has to strike a balance between directing cellular resources to engineered processes of interest and preserving smooth functionality of the cellular machinery [52].

The purpose of the Dynamic Resource Allocation Model is to extract information from measurement data, particularly information relating to the burden of expressing and engineered construct. Ideally this information can provide better and more reliable measures for how we quantify and describe burden.

Previous work on growth has focused on static fractions of proteins within a cell or cell populations and then related those static conditions to burden [162]. In a static environment this approach is probably fine, as over time the fraction of a protein in a cell will approximate its relative expression level. However, the static assumption requires that the expression levels are constant for a long time, which requires that the cell density levels and nutrient levels in the environment are kept constant for an extended time. Such static conditions can be realised artificially in a laboratory, *e.g.* with the help of a chemostat. A different argument against static conditions for burden quantification is that the burden exercised by engineered proteins is not linked to their concentrations *per se* (*e.g.* there is no evidence for the presence of the GFP molecule itself having a cost to cell growth). Instead burden often originates from the resources occupied in producing the engineered proteins, and in a fluctuating environment these resource requirement can also fluctuate over time, even if changes in protein concentrations are lagging. This is also the effect which the Dynamic Resource Allocation Model aims to elucidate.

The original 'growth law' model developed by Scott *et al.* [162] refers to static growth conditions, and for the purposes of measuring burden in static growth conditions (*e.g.* in a chemostat) this simpler model is likely to be preferred. Meanwhile, this chapter developed models for the dynamic response of cells to changing growth conditions. These methods can hopefully provide additional measurable values for growth and fitness cost which are useful in fluctuating environments, such as commonly encountered in practical applications of engineered constructs (*i.e.* outside a chemostat).

The first model developed and shown can be referred to as a bang-singular response model, where cells are assumed to change their expression levels instantaneously following a new environment. This is different from the original 'growth-law' model in that it takes into account what expression levels are required in a cell prior to reaching steady state. However, this first version of the model assumes that cells do not require any time to adjust to their new expression levels in a new environment, an assumption which is unlikely to be fully accurate.

The second version of the dynamic resource allocation model can be referred to as a delayed bang-singular response model. It assumes that there is indeed a time requirement for cells to respond and change their expression level when responding to a new environment. This assumption of a time requirement to respond to the changing environment does increase the model complexity (by including the delay parameter t_s). However, the assumption also improves the fit of the model to data, as well as reduces the standard deviation of the expression level for the ribosome fraction. Additional fitting examples for this model with more construct examples can be found in Nyström *et al.* [132].

The third version of the model can be referred to as a delayed bang-bang-singlar response model. It is used to test the theoretical optimal control prediction by Giordano *et al.* [74, 75] which states that cells most optimally respond to new environments by initially over-expressing the fraction which is required, but then reduces the expression level when an optimal level has been reached for the required fraction. Fitting of the DRAM to identical cell samples diluted at different ratios finds that the prediction of initial over-expression corresponds well with the growth behaviour of the cell samples. The fitting also shows that expression levels of ribosomes is fairly equal between cells with and without constructs after the initial spike in ribosome expression during growth rate acceleration (see Figure 3.14). This result of the model fitting corresponds well with the previous experimental finding by Shachrai *et al.* [168] which states that the cost of engineered constructs is reduced after completed adjustment to higher growth rates.

For practical uses of the DRAM the version which is likely to be most applicable is the delayed bang-singular response model. It has higher complexity than the non-delayed bang-singular response model, but at the same time is thought to be more realistic by inclusion

of the cells response time. Further, the output from fitting the model also provides the response time (t_s) which can be an interesting and additional parameter for evaluating burden. An obvious improvement to the current state of the model could be to include smoother and more graded responses of expression, in contrast to the current step-wise responses of expression. However, other types of experimental data would be required to develop such assumptions accurately, and hopefully the current step-wise approximation can serve as good estimates until such models have been developed.

It is very likely that the total cost of an engineered biological component is a combination of different factors. Taking enzyme expression as an example, the cost of expression could be divided between (i) producing toxic products or intermediates, (ii) sequestering vital precursors or simply (iii) disturbing the balance of production and consumption. However, it is currently difficult to disentangle the burden effect because of the expression of proteins from absolute fitness costs, including toxicity or metabolite imbalances. Therefore, it would be very useful to find modelling methods that are able to distinguish burden caused by protein expression from other sources. The aim of the Dynamic Resource Allocation Models is to provide information specifically on the level of burden caused from protein expression, under natural dynamic growth conditions.

3.5.1 Conclusions

We are only at the very beginning of understanding costs of engineered biology through host machinery interactions. This chapter has identified novel conceptual features for assessing burden caused by engineered biology. Not only are these parameters derived from metrics of direct biological meaning (in contrast to the previous measures of μ_{max} and the less defined λ) but they also enable interpretation from short time series and broader sets of experimental data.

It is intriguing how the specific effect of over-compensation (or lack thereof as shown in this chapter) can be used to measure and quantify the level of burden that is exercised by engineered biology through competition for resources within a cell. Two new conceptual features for burden measurement are identified: (i) the level of peak expression of ribosomes during adjustment to growth in a new environment ($R_R(t_s^+)$), as well as (ii) the

time required to respond to a new environment (t_s). The altered initial level of ribosome concentration ($\Phi_R(0)$) was also predicted from the model as a result of engineered biology. However, this parameter might be less reliable as a measure for the specific level of burden, as $\Phi_R(0)$ is also strongly influenced by the time period after μ_{max} during the history of a sample in ENV_1.

There are previously no reports indicating that the time required by cells in order to interpret and respond to new environmental conditions can be affected by the presence of engineered constructs. However, the effect shown on t_s from engineered constructs (by the model fitted to data) indicates that there is a signalling and response delay. The imposition of signalling delays within hosts is a new area of research which would be interesting to investigate with a deeper focus in the future. It is possible that the time required by cells to interpret and respond to more favourable nutrient conditions is not the only information processing capability which is delayed. Potentially engineered functions, such as cell-cell signalling or the responsiveness to natural cues more broadly might also be delayed in their information processing, due to the very presence of the engineered components. If such a broader delay would exist as an effect of burden then these effects would need to be accounted for and calculated when designing interacting circuits.

However, in order to progress our current understanding forward it will be crucial to initiate discretisation of costs in order to break them up between different sources. This chapter has focused on costs derived from sequestration of host machinery for the expression of the protein components necessary in an engineered biological system. More than the additional conceptual features for burden that this chapter has provided, it is the aim and purpose of the chapter that the derivation of *burden* specific costs shown (*i.e.* specifically due to the resource requirement of protein expression) will enable them to be distinguished from other sources of engineered costs, such as toxicity, signalling or metabolic flux imbalances. This would enable future methods for quantifying these costs by subtraction (or other simple estimations) of the cost related to protein expression burden.

3.5.2 Contributions

The key contribution of this chapter has been the development of a new model (the Dynamic Resource Allocation Model, DRAM) for conceptualising burden from engineered biology. The most important contributions that this work has enabled include the following:

- Derived a new dynamic model for burden from engineered biology directly from population growth curves - the Dynamic Resource Allocation Model. The new dynamic model is derived from the previous static 'growth-law' model combined with simple assumptions from first principles.

- By applying the DRAM and observations from experimental data, the Chapter has developed new model parameters for how burden from the expression of proteins can be measured for engineered biology.

- The derived model parameters specific for burden are: (i) R_R, the fraction the proteome that can be dedicated to ribosome production during the ramp-up period, (ii) t_s, the signalling delay in response to a new environment.

- The Dynamic Resource Allocation Model and experimental data presented in this chapter have provided support for the previous theoretical assumption of over-compensation of expression at transition phases of growth, suggested by the bang-bang control models (and shown by the lower level of $R_R(t_c^+)$ during the exponential phase as compared to $R_R(s)$ during the ramp-up period).

- The chapter has identified and proposed a method for finding the rate of decline of Φ_R after reaching maximum growth rate by applying DRAM. Applying the method to data shows that hosts with engineered constructs have faster rate of decline for Φ_R compared to non-engineered hosts (see Figure 3.18). The identified effect can have implications for future measure methods of engineered burden and genetic stability that rely on series of dilutions into new growth phases. Further, the method allows better comparisons of growth effects in an experimental ENV_2 by relating the data

back to initial growth in a previous environment, enhancing the understanding and interpretation of growth data.

Compared to the existing burden measures of μ_{max} and λ, the new model features of t_s and R_R have a biological meaning and thus enable the variables to be used in other predictive models of engineered biology. The new parameters proposed allow for the assessment of burden from the early stages of growth, which is faster and requires fewer collected data points. In the same way, the Dynamic Resource Allocation method showed in this chapter enables the assessment of burden directly from experimental data without needing separate experiments for maximum growth rate.

Ideally, the work shown in this chapter is only the beginning of dynamic interactions between engineered biology and their hosts; a research direction which can further inform how the constructs we are designing will impact their host environments during natural and dynamic fluctuations. In turn, this will allow for predictions of how the altered host environment, imposed from our engineered constructs, affects the behaviour of our intended system designs. Robust forward-engineering of biology will be enabled by accurately quantifying the feedback loops of interacting (or interfering) signals between engineered constructs and hosts.

4. Chapter 4

Chapter 4 develops a model for identifying how host organisms can evolve to reduced costs when they carry engineered biology constructs. It develops the model by assuming a similar process to how multi-resistant 'superbugs' arise, and attempts to better formalise and reconstruct this process in order to utilise it for obtaining longer lasting engineered constructs. The developed model is parameterised with data from the Long Term Evolutionary Experiment. The purpose of the model is to inform experimental methods while identifying beneficial and enhanced conditions for how to evolve host genome architectures with reduced cost for engineered biology.

4.1 Introduction

The previous chapters have focused on rational forward-design and new interpretation methods for burden. Chapter 4 takes a very different approach to address the problem of genetic stability. It investigates evolutionary methods that could be applied for enhancing the stability of engineered biology. No known previous research has investigated evolutionary approaches, absent of additional selection pressure markers, for sustained cost reduction of engineered constructs. The closest example is work by Herbert Sauro's group which have used rational design as well as construct expression directly linked to antibiotic resistance marker expression in order to toggle selection pressure on-and-off in the presence of an antibiotic [207].

If a robust and repeatable evolutionary method for construct stability is developed, then it would enable the creation of radically more stable engineered biological features, which would also have positive implications for wide ranging sectors of applied biotechnology.

As a short background, it is known that populations adapt themselves to new external environments [84, 107, 198]. However, populations are also able to adapt to new *internal* environments [161, 157, 117]. When a procaryote pathogen acquires a genetic element of antibiotic resistance, usually carried on a transmissible plasmid, this entails a fitness cost to the host in the absence of the antibiotic [183, 118]. There is a selective pressure against keeping the element. However, pathogens have been shown to evolve second-site *compensatory mutations* that reduce the costs of carrying an antibiotic marker also in the absence of antibiotics [161, 118, 157]. Evolutionary adaptation to the deleterious effects of resistance by compensation has been observed for several bacterial species and types of resistance [157, 112, 118]. Compensatory mutations are a key reason for the emergence of multi-resistant 'superbugs' [127, 113], which preserve their antibiotic resistance during prolonged periods in environments without a selection pressure for the plasmid-encoded antibiotic markers [157, 112, 118, 23]. Following fitness improvements caused by compensatory mutations, resistance markers have been preserved in populations for more than 10,000 generations in the absence of antibiotics [161]. Specific compensatory mutations for internal adaptation have been shown to nearly fully restore fitness costs of $S = 0.2$ (for fitness cost see Equation (0.2)) due to hosting a resistance marker [157], while other studies have shown that hosts can overcome the additional cost of carrying antibiotic resistance in as little as ten days [158]. Work on multi-resistant *superbug* pathogens shows that costs produced by plasmids are transient and can be ameliorated rapidly, suggesting that new genetic elements produce substantial burdens only when recently acquired by bacterial hosts [158, 117].

The models in this chapter are inspired by the naturally observed process of pathogens adapting to novel internal genetic elements with the aim to harness the same principle for reducing the cost of engineered elements within carrier hosts.

To make a distinction between (i) the natural occurrence of *compensatory* mutations in the host genome of *multiresistant pathogens* and (ii) the dedicated process of accommodating engineered elements with *suitable carrier hosts* for reduced burden (*i.e.* the focus of the models), this thesis refers to the latter as *accommodating* mutations.

Chapter 4 develops a theoretical model plus a methods framework to leverage evolu-

tionary accommodating effects for enhancing the stability of engineered biology. In essence, the quest becomes one of adapting a host to a new internal environment, as opposed to adapting it to a new external environment. The beneficial outcome of adapting a strain to new internal conditions (manifested by the presence of an engineered construct) is that the construct will become more stable in the host. The stability is increased due to the reduced cost to the host of coexisting with the engineered construct. However, note that the existence of such accommodating mutations is an assumption made by this chapter.

The presented model develops *accommodating mutations* selection methods for engineered construct stability. The naturally occurring process of compensatory mutations is aided by selective pressure of the resistance phenotype. However, for the accommodating mutations sought for maintaining an engineered phenotype, a conservative assumption is that there is no selective pressure available which can be leveraged (nor a fluctuating selection pressure). If a selective pressure is available for the engineered phenotype sought, then the task can be regarded as plain directed evolution. Directed evolution is outside the topic of this investigation, great resources on directed evolution can be found elsewhere [137, 48, 126, 46].

Evolutionary adaptation is driven by the accumulation of beneficial mutations [77] and depends on the effect of new mutations as well as their dependency on the genetic background in which they occur [85]. These two factors of adaptation can be summarised by random walks through fitness landscapes [85, 135], which were first introduced by Sewall Wright [204] and later adapted by Kauffman and Levin [95] to represent nodes in a DNA sequence. However, experimental verification of specific fitness landscapes is difficult [85], if not impossible, as it would require the testing of infinite numbers of potential mutations and how they will impact each other in the future.

A method for simplifying the representation of fitness landscapes is by aggregating the effect of all potential mutations into a probability density function of their Distribution of Fitness Effects (DFE) [19, 72]. In other words, DFEs simplify the paths of fitness landscapes by reducing them into two dimensions of frequency and fitness effect. This way, the information of how the history for each individual potential mutation can impact the future effect of individual mutations is removed. Detailed information of future impacts

is currently impossible to verify, and the DFEs instead provide an aggregate distribution for all potential mutations. The model presented in this chapter derives evolving DFE shapes from available data and then extends those DFEs by including engineered biology constructs.

4.1.1 Fitness Costs of Engineered Constructs

When inserting an engineered construct into a wild type genotype its fitness is reduced by the fitness cost of the engineered circuit. If the circuit is completely removed again (*e.g.* through natural mutations) then the *wild type* fitness is restored. The fitness gain realised by full removal of an engineered construct from a host is denoted S_0. When adding an engineered construct into a host, an additional set of mutation possibilities are introduced into the host DFE; the mutations that can destroy the construct (see Figure 4.1B1). If an engineered construct has a cost $S_0 = 0.2$ and a natural mutation completely destroys this construct then the fitness of the mutated individual compared to its peers in the engineered population is $w = 1 + S_0 = 1.2$. However, many mutations would only partly restore the fitness by partial destruction of the engineered function. The area of the probability density distribution of destructive mutations (the red coloured area in Figure 4.1B1) is equal to the combined probability of any mutations within the DNA sequence of the construct that has been introduced. In the absence of DNA sequence repeats (which generate increased mutation rates [92]), the total area of destructive mutations is proportional to the sequence length of the construct. The size of the area is independent of cost. The location and distribution of the area (on the x-axis of the DFE) depends on the fitness cost to the host from the construct. For example, a non-induced construct would be centered further to the left on the x-axis, while adding an inducer to increase expression levels would draw the distribution to the right on the x-axis. But the total cumulative area of destructive mutations remains constant.

The hypothesis made for the model relies on an important assumption, namely that the DFE of mutations *within* the host genome is changing when introducing engineered constructs. Specifically, it is assumed that introducing a construct (with a negative fitness effect) has a qualitatively similar effect to the DFE as when moving the organism to a

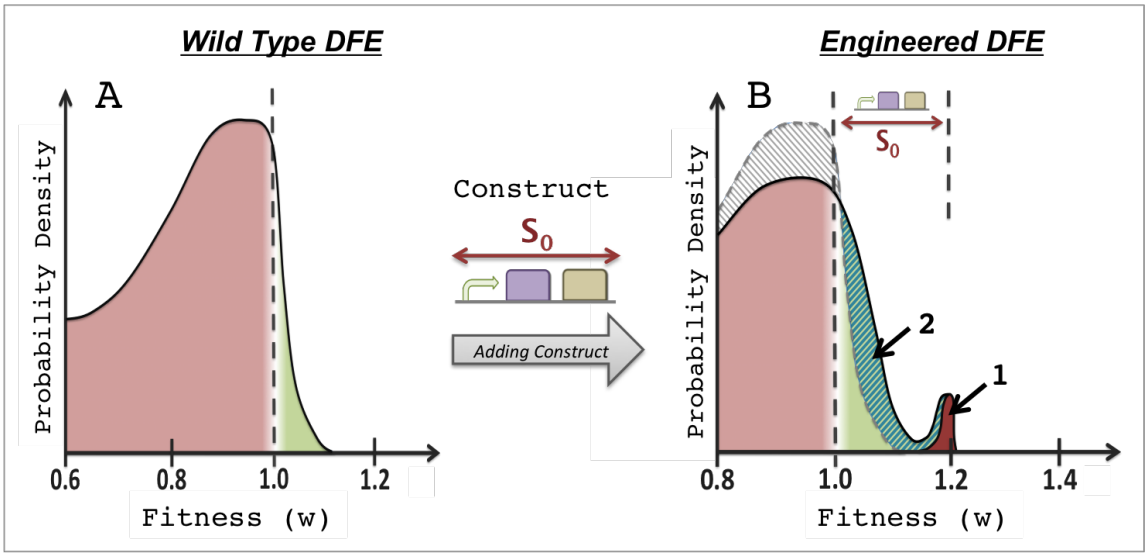


Figure 4.1: Illustration of the key hypothesis of how DFEs are altered as an effect of introducing engineered constructs. When inserting an engineered construct into a genotype the fitness is reduced compared to the wild type genotype. If the circuit is completely removed (*e.g.* through mutations), then the original fitness is restored again. The fitness gain from completely removing the construct is denoted S_0 (in the figure example $S_0 = 0.2$). The dashed vertical line in A and B divides mutations with a positive impact on fitness from those with a negative fitness effect. **A:** The wild-type distribution of fitness effects (DFE), the x-axis represents the resulting fitness of a mutation and the y-axis the probability density. **B:** The DFE after the introduction an engineered function (*e.g.* via transformation). Completely removing the construct yields a new relative fitness of $w = 1 + S_0 = 1.2$ compared to the engineered genotype. **B1:** The probability density distribution of mutations deleterious to the function which has been introduced to the host. **B2:** The change in the probability density distribution of mutations as an effect of the new internal environment resulting from the engineered construct. The shape of the change in the host DFE from a new deteriorated internal environment (due to an engineered construct) is assumed to be of similar shape as the effect on the organism of a deteriorated external environment. Note, (i) the DFE and x-axis in A refers to wild type, and the DFE in B refers to the engineered type, *i.e.* the relative fitness $w = 1$ for wild type is not the same as $w = 1$ for the engineered type (which has been reduced by the cost of hosting the construct). (ii) The red arrow above the construct illustrates the cost of the construct, and not the size of the construct. The size of the construct (and thus its probability of mutations) is only related to the area of B1. Also note, w refers to relative fitness (and it does not mean the absolute fitness). The fitness (w) in this chapter always refers to relative fitness.

worse external environment (meaning an environment to which the organism is less well adapted). An illustrating example of this effect is given in Figure 4.1B2; as a larger number of potential mutations with positive fitness effects become available within the host genome. Qualitatively, introducing an engineered construct affects the host DFE in the same way as when the organism is placed in a novel and less favourable external environment. In other words, the introduction of an engineered construct alters the internal environment of a host, not the external environment, but the effect is assumed to be similar. However, the quantity of the effect on the DFE depends on the type of construct introduced, in the same way as the changes on the DFE from a new environment depends on the type of environment.

Data from the Long Term Evolutionary Experiment (LTEE) [201, 107] has been used in order to parametrise the model with the important underlying assumptions regarding the change of probability distributions over time.

4.1.2 The LTEE data source

In the famous long-term-evolution-experiment (LTEE), started in 1988 and still running [107], 12 *E. coli* replicate populations are daily diluted into a defined and constant medium. A single carbon source of glucose limits growth of the diluted population. A single day of the experiment consists of 6-7 generations, followed by entry into stationary phase before cultures are propagated into fresh medium again the next day. Today, after >25 years, this amounts to over 60,000 generations [116] of isolated growth and adaptation. Together with sequencing data and saved populations from each time step the LTEE experiment provides a deep resource for inferring evolutionary properties. In addition to providing data for estimations of change in the DFE over time, several unexpected phenomena have been observed; including one population adapting to utilising a different carbon source (citrate) and other populations being overtaken by mutator strains with drastically elevated mutation rates [116].

One interesting observation from the LTEE is that high mutator phenotypes naturally evolved in the experiment [107]. This suggests that experimental conditions favour higher mutation rates than the mutation rates that have evolved in natural habitats. A selective

advantage of a higher mutation rate suggests that mutations occurring during the experiment have a comparatively more positive effect in comparison to natural environments. The selective advantage of a higher mutation rate within the LTEE experiment can have at least three different (or combined) explanations:

- The stable environment maintained in the LTEE allows for higher mutation rates than fluctuating natural environments. In a fluctuating environment a more stable genotype has to be maintained in order to safeguard against infrequent but reoccurring environmental conditions.

- The environment provided in the LTEE is sufficiently different from the normal environments encountered by the strains involved in the experiment. This allows for a much larger proportion of beneficial mutations able to adapt to the new environment.

- The experiment maintains a significantly larger population size compared to natural habitats, thus increasing the effect of positive mutations while diminishing the effect of negative mutations. (Instead, a smaller population size would allow for a faster fixation of deleterious mutations.)

Irrespective of the precise reason for the emergence of hypermutators in the LTEE, their presence and emergence during well characterised conditions provide valuable data and parameters for fitting and testing the distribution of fitness effects required for the model implemented in this chapter.

In summary, Chapter 4 aims to identify criteria for enabling host organisms to evolve to reduced costs for keeping engineered circuits (and without losing the construct while getting there). It develops a model by assuming a similar process to pathogen antibiotic-resistance genome-compensation arising in nature, and attempts to better formalise and reconstruct this process of construct-genome adaptation in order to use the process for obtaining longer lasting engineered circuits.

As Ronald A. Fisher emphasized in 1930 [135, 67]: *"Adaptation is characterized by the movement of a population towards a phenotype that best fits the present environment. The result is an often astonishingly precise match between an organism and the world in which it lives."*

The challenge presented for creating accommodating mutations for engineered circuits is akin to letting the organism adapt to a new *'internal environment'* provided by a new present circuit, without losing the construct along the way.

4.1.3 Distribution of Fitness Effects

Theoretical models produce formalised hypotheses for evolutionary processes [12], and in the best cases, experimental evolutionary experiments can be designed from these theories in order to characterise what is biologically reasonable [12]. Evolutionary data for informing theories can be split up into two broader sections. Firstly, genetic and genomic data from natural populations provide a basis for identifying key factors having played important roles in steering evolution in the natural world [19]. Secondly, experimental evolution studies provide the means to test theories being difficult or impossible to understand from natural population data, generating new insight to refine evolutionary models and theories [12].

The rate at which mutations occur in natural populations and their distribution of fitness effects are of key importance in evolutionary genetics [133]. Classic mutation accumulation studies have shown that mutations with deleterious effects far outnumber those that increase fitness [82]. Parameters for beneficial mutations are of particular interest as they constitute the driving force of adaptation and survival of populations in new environments [83]. Experimental evolution studies have shown that although the majority of mutants have nearly neutral effects, the mutations of intermediate positive effect tend to drive adaptation [19].

Wiser *et al.* [201] develops a DFE model for the LTEE which only takes beneficial mutations into account, and this is achieved by assuming an exponential distribution for beneficial mutations. In contrast, the model developed in Section 4.2.1 of this chapter includes the full spectrum of deleterious to beneficial mutations. Developing a model with a full DFE spectrum was necessary in order to enable model fitting to data with different mutation rates (discussed in more detail in Section 4.2.1 and Figure 4.9).

Beneficial Mutations

A large number of aspects regarding adaptive evolution depend on the beneficial fitness effects and their distribution [133]. The most direct method for studying the distribution of beneficial mutations is empirical [19], however, even if feasible in principle it has proved to be difficult in practice [134]. Two of the main problems can be summarised as follows: (*i*) beneficial mutations are rare and (*ii*) mutations of small effects are difficult to detect [134]. These are also the reasons why microbial populations are commonly employed to study DFEs. They enable a combination of large population sizes and short generation times, thus providing many beneficial mutations to be sampled during shorter time spans. However, the results which can be obtained are normally compromised by stochastic loss and clonal interference. While some experiments have sought to assay beneficial mutations before they are subjected to stochastic loss they face a second problem of not being able to detect mutations of small effect [134].

The Gamma Distribution

Another method, other than experimental techniques for understanding beneficial mutations is to apply population genetics techniques. Among the mathematical distributions used to model DFEs the most common is the gamma distribution [65] (see Figure 4.2). The gamma distribution is governed by a shape parameter: k, and a scale parameter: θ. The kurtosis of a DFE is the extent to which it is peaked; greater kurtosis meaning that more variance is contributed by infrequent mutations of large effect. A leptokurtic DFE normally refers to a gamma distribution shape being more leptokurtic than an exponential distribution, *i.e.* has a shape parameter $k < 1$. However, this is in contrast to normal, non-DFE contexts, where kurtosis would be defined relative to that of a normal distribution [65].

- $k > 1$: L-shaped distribution.
- $k = 1$: Exponential distribution.
- $k < 1$: Skewed normal distribution.

Theoretical work by Gillespie [72] and Orr [134] suggests that Extreme Value Theory (EVT) can be applied to inform DFEs. This predicts that the distribution of beneficial mutations should always be exponential. However, the method is conditioned on the distribution of absolute fitness values being invariant and the population being reasonably well adapted to start with [72, 134, 65].

The EVT assumes a constant distribution of mutations in time, however, as populations adapt, the DFE is likely to change, thus violating the central time-constant assumption of the system [65]. Accounting for effects of epistasis might provide further insight into the trajectory of the DFE shape during adaptation towards higher fitness in a certain environment. It has also been proposed that the adaptive dynamics of a population can be simplified and approximated through an equivalent model to a state where all favourable mutations carry the same fitness advantage, called the "effective selection coefficient" [83]. However, this argument seems to assume a single and fixed population size.

Negative Mutations

In general, the distribution of deleterious effects is more complex than positive mutations, and sometimes negative mutations appear centred around more than one maximum. In both RNA viruses and yeast there is a distinct class of mutations that are lethal, generating a bimodal shape of deleterious mutations [65]; in yeast 30-40 % of laboratory detectable mutations are lethal [203].

Trindade *et al.* [180] measured the rate and effect of spontaneous mutations in mutator strains of *E. coli* through mutation accumulation (MA) experiments and found a deleterious rate of 0.005 mutations/genome/generation with a mean fitness effect of $s = 0.03$ [180].

Mutations that substitute amino acids in expressed proteins generally belong to a group of mutations with very mild negative effects. Most amino acid mutations have very small effects on protein function and an even smaller impact on the host in general [65].

The model developed by Wiser *et al.* [201] was able to exclude negative mutations. The model uses a logarithmic distribution of positive mutations as a DFE where the positive fitness impact (s) of each mutation is given by $f(s) = \alpha e^{-\alpha s}$, which also means that $\frac{1}{\alpha}$ is the mean advantage. They assume the distribution of beneficial mutations decrease after

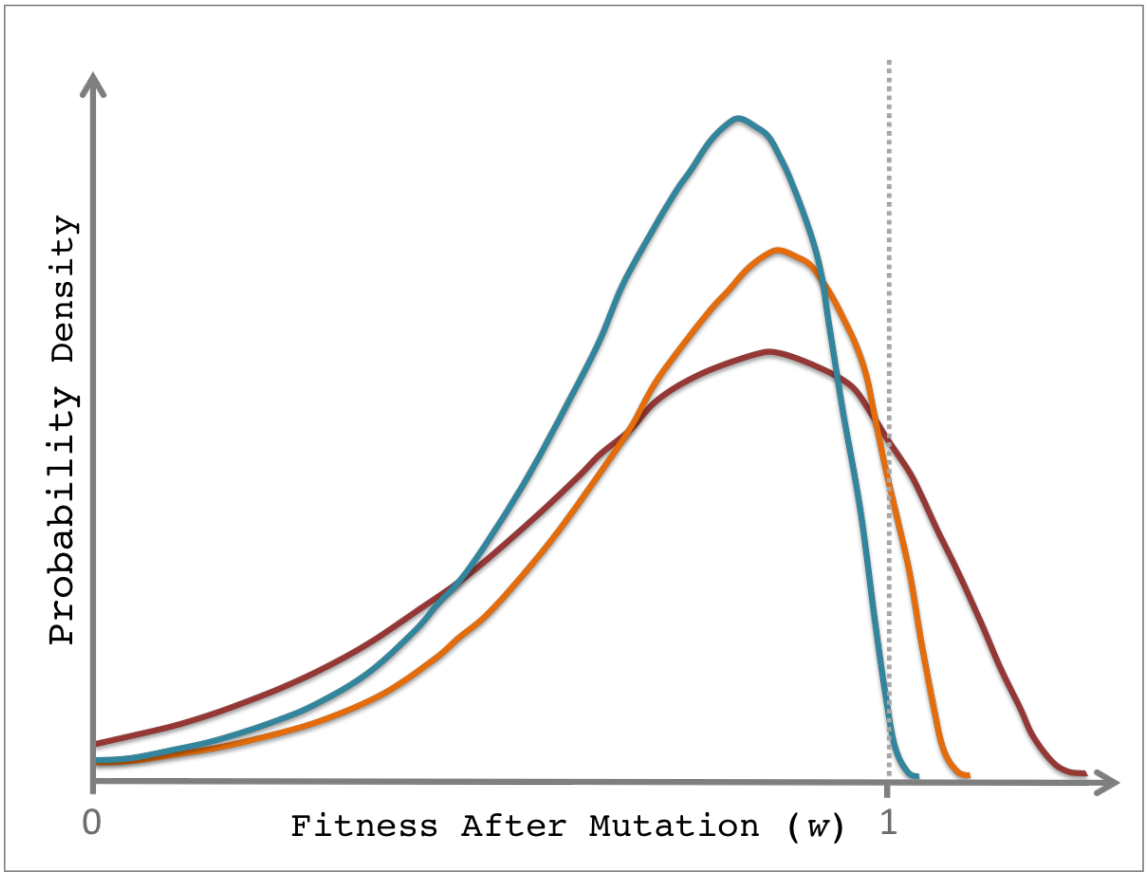


Figure 4.2: Example DFE gamma distributions. Three distinct DFEs are shown for populations with varying levels of adaptation: perfectly adapted (**blue** - high k), partly adapted (**orange** - intermediate k), and poorly adapted (**red** - low k). DFEs are here shown as reverse gamma distributions [12]. Similar variation in DFEs has been found empirically in *E. coli* [88, 141]

a mutation of advantage $\langle s \rangle$ fixes in the population, such that α increases linearly related to $\langle s \rangle$: $\alpha_{n+1} = \alpha_n(1 + g\langle s_{n+1} \rangle)$, where g is the diminishing-returns parameter, and α_n is α after n fixations. Subsequently, they are able to find the μ value for positive mutations (*i.e.* the probability of a positive mutation) from fitting the model to data. Additionally, they are able to solve their model analytically to obtain mean fitness (w) over time, while accounting for diminishing-return epistasis and clonal interference. In summary, it is an impressive model.

However, a model which excludes negative mutations can not account for the genetic load contribution by hypermutators. By the time of the publication of the model by Wiser *et al.* [201] it had already been shown that the genetic load from negative mutations does have an effect on the hypermutators, and that the effect is large enough to be able reverse hypermutability [199]. More evidence has since shown the importance for the genetic load (from deleterious mutations) on the hypermutators [177], which indicates that all hypermutators in the LTEE subsequently evolved reduced mutability (due to genetic load from deleterious mutations).

It is the purpose of the method developed in this chapter that it should be able to account for and simulate the effect of different mutation rates, with the potential use case of more rapidly obtaining accommodating with reduced construct costs (*i.e.* through the deliberate use hypermutators). However, this is not the only reason for including the distribution of negative mutations in the DFE. As can be seen in Section 4.3, one of the experimental parameters which the model development is used to simulate is the culture volume of population samples. Genetic drift increases the likelihood of near-neutral deleterious to spread in smaller populations. To enable the model to simulate the effect of different culture volumes, including very small populations, it is necessary to account for the effect of mutations with negative fitness effects.

Beneficial Mutations - Impact from Adaptation

Estimating DFEs remains experimentally challenging even with model organisms in well controlled environments [134], therefore inferring and quantifying the rates of change of a DFE is even more challenging. Intuitively, in a very well adapted environment virtually

all mutations are deleterious and the probability of beneficial ones is vanishingly small. In contrast, in a new environment there should be a larger initial proportion of mutations carrying beneficial effects [12]. This view is also supported by Fisher's Geometric Model (FGM) [176], which predicts that the fraction of favourable mutations available in a given environment is strongly conditioned on the current level of adaptation. Experimental evidence also supports this prediction [18, 89].

That the proportion of mutations carrying beneficial effects declines with increased levels of adaptation can also be supported by experimental evidence that mutator genotypes (*i.e.* genotypes with increased mutation rate) are more likely to fix in a population that is less adapted to a new environment [201].

According to two papers investigating the rate of evolution in the LTEE experiments, the rate of evolution can be best described as following a power-law model [107, 201], meaning that evolution does not reach local maxima where it stops but continues indefinitely, albeit with a reduced velocity. It is important to note that while underlying data is provided for very long experimental timespans it only covers a single population size, thus providing clear but still limited insight into the general process. Nevertheless, the finding of a power-law relationship for the rate of evolution ought to have a clear and significant effect for informing the change of shape of the DFE over time and at different levels of adaptation.

Epistasis

Epistasis describes the phenotypic effect of a mutation in concert with other mutations present in the genotype [43]. If two mutations act independently on a phenotype they are referred to as having no epistatic effect. From the perspective of DFEs, the global level of epistasis becomes crucial in order to inform how the DFE changes during adaptation to a new environment while mutations accumulate.

One general trend of epistasis is that a lesion within a single pathway has a greater effect alone compared to when there are already other deleterious mutations present in the same process. This is referred to as *antagonistic epistasis*. Conversely, lesions in parallel pathways producing similar or equal outputs tend to cause *synergistic epistasis*, meaning

that the phenotypic effect is increased with previous mutants present, sometimes resulting in a nonviable phenotype [43]. Construction and testing of all combinations from five amino acid combinations in the *E. coli* β-lactamase enzyme showed a single fitness peak. But identical mutations decreased fitness in some background while increasing it in others [194], referred to as *sign epistasis*.

For assumptions informing the trajectory of DFEs during population adaptation to a new environment, the following observation is of significant importance: one of the most consistent findings across evolved laboratory populations has been the rapid deceleration of evolutionary velocity [61]. These observations point towards a reduced area (or effect) of beneficial mutations within a DFE during the process of adaptation, *i.e.* a general pattern of *synergistic epistasis*. This effect is consistent with the behaviour of the model implemented in this chapter, producing a reduced area as well as magnitude of beneficial mutations as adaptation progresses in time.

4.2 Approach

A sequence of steps were implemented in order to model and simulate the occurrence of accommodating mutations from engineered biology constructs. First a gamma DFE is fitted and parameterised with data from the Long Term Evolutionary Experiment [107] (Section 4.2.1). Second, general assumptions are included for how the DFE changes in the presence of engineered constructs (Section 4.2.2). Finally, results from the model are produced for how accommodating mutations can be favoured, in which the effect from different types of constructs and selection approaches are analysed (Section 4.3).

4.2.1 Model Step 1 - DFE fitting to LTEE data

The first step of the model setup was to construct a model which can represent environmental adaptation of a population in an environment and then to parameterise this model with experimental data from the LTEE to obtain a realistic basic model. This initial and representative model of experimental data can then be used as the foundation for making further assumptions and application of the model to the investigation of accommodating mutations.

The Computational Model Procedure

The general model setup starts with a set of individuals with an initial fitness level $w = 1$ and then evolves them through a predetermined number of generations. The population of individuals reproduces from one generation to the next, and the number of offspring from each individual depends on the fitness level of that individual. With the initial fitness level of $w = 1$, an individual produces one offspring at each generation, while the parent also remains in the population. The number of offsprings produced per each generation is in this simulation given by w, to give a close value to the measurement method of the relative fitness w used in the LTEE [200] (here the number offsprings is $e^{w \ln 2} - 1$, excluding cell death). Each time an offspring is produced there is a certain probability that a mutation will occur which will alter the fitness of the offspring compared to the parent. The probability of a mutation to occur is denoted μ. When the total

population reaches a specified maximum population size ($N = 4 \times 10^8$ individuals [107]) then the whole population is diluted 100-fold (the same dilution as the LTEE [107]), before replications resume. The dilution process is implemented by a random sample selection from the original population. The event of cell death is not included in the model. This is because the likelihood for a cell to be lost at subculturing is much greater than the probability of cell death. Each time a mutation occurs in the replication of an offspring a new fitness level is drawn from a DFE. The shape of the DFE depends on the fitness level of the parent. See the flow chart in Figure 4.3 for a visual representation of the basic model implementation. All computational results are realised through Markov Chain Monte Carlo (MCMC) simulations and the models are written in Python. The initial starting populations of the simulations are $N = 10,000$ with a fitness $w = 1$. The mutation rate for normal populations is $\sim 3 \times 10^{-3}$ mutations/genome/generation [59, 114], and up to two orders of magnitude higher for hypermutators [201]. The probability that a mutation m occurs is given by the mutation rate μ: $P(m = 1) = \mu$, where a the occurrence of a mutation is a binary event $m \in \{0, 1\}$.

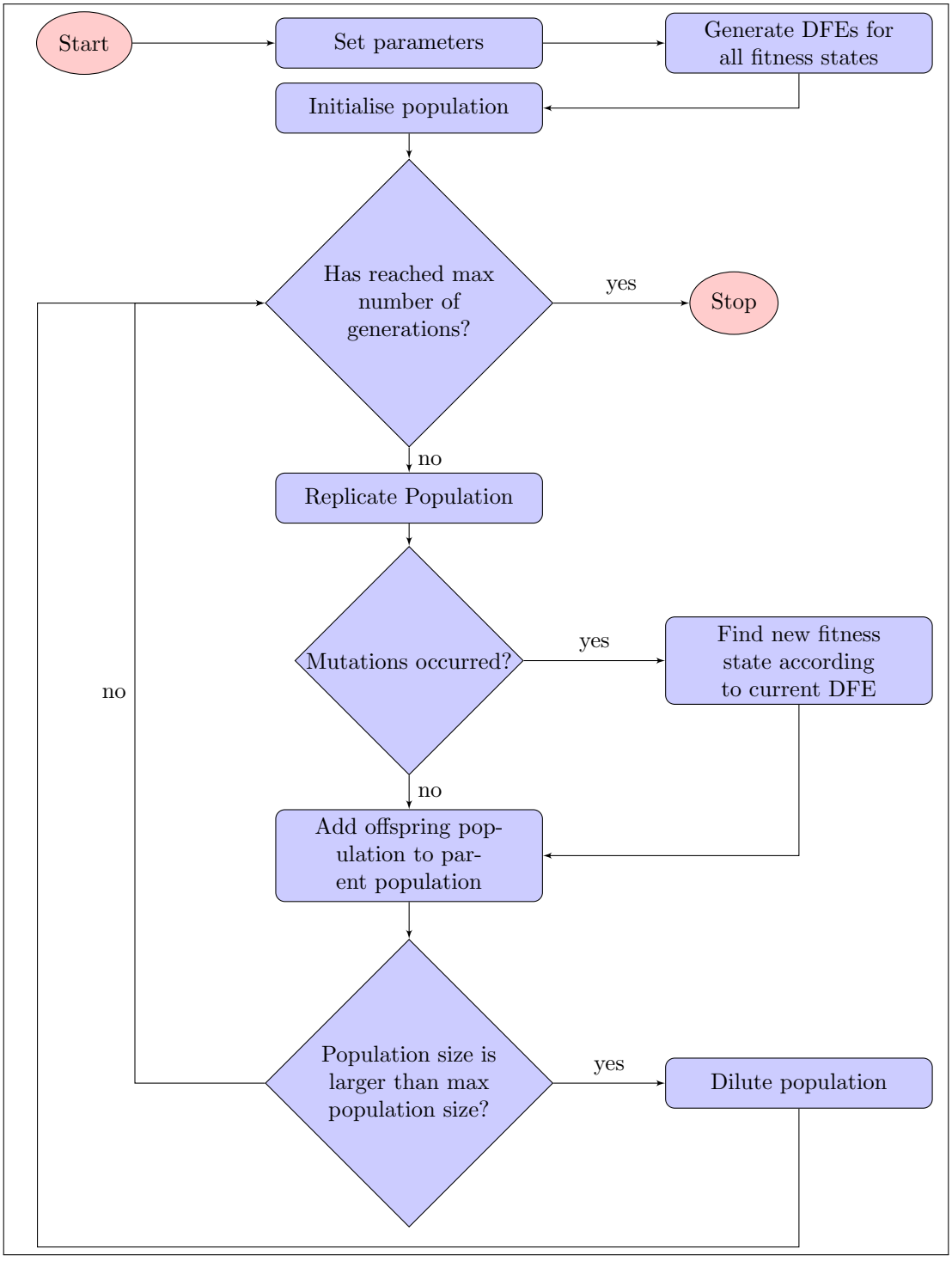


Figure 4.3: Flow chart for the model realisation of environmental adaptation. The model is initialised by specifying mutation rates, the DFE as a function of fitness, max population size before dilution, number of generations to simulate growth, the size of the starting population, *etc* (see Sections: The Computational Model Procedure *and* Computational Replication and Dilution Procedures). At "initialise population" the specified starting population has a relative fitness $w = 1$. The size of the starting population used for the model realisation of this chapter is $N = 10,000$.

Model Calibration

A sequence of steps were necessary to calibrate the general model (as shown in Figure 4.3) such that it can represent the general evolutionary process seen in LTEE, and also provide a function for how the DFE depends on the fitness state of an individual. The following sequence of steps was carried out in order to fit the general model as well as the gamma DFE to the experimental data:

1. Test a range of different model fitness discretisation levels to find a level which approximates a continuous distribution of fitness (Figure 4.5).

2. Find a suitable time span to measure the fitness velocity (*i.e.* the rate at which fitness improves) as an effect of static DFEs (Figure 4.6).

3. From multiple simulations of static DFEs, find how the gamma shape parameter (k) impacts fitness velocity (Figure 4.7D).

4. Combine the results from step 3 together with the fitness velocity of the LTEE to find a preliminary function for how k changes with increased adaptation, *i.e.* $k = f(w)$ (Figure 4.7E).

5. Use least squares fitting of gamma means ($E[X]$), shapes (k) and the identified function parameters for $k = f(w)$ to find how the DFE changes as a function of w by simultaneous fitting of normal mutators and hypermutators from the LTEE (Figure 4.8).

The Gamma Distribution as a DFE

The gamma function is an integral component of various probability distributions and widely occurring in the fields of statistics and probability. The gamma function (Γ) is an extension of the factorial function but with its argument shifted down by 1,

$$\Gamma(n) = (n-1)! \qquad \{n \in \mathbb{N}^+\}$$

and the probability density function of the gamma distribution is given by

$$f(x;k,\theta) = \frac{x^{k-1}e^{-\frac{x}{\theta}}}{\Gamma(k)\theta^k} \qquad \{x,k,\theta > 0\}$$

where k is the shape parameter and θ is the scale parameter.

The mean of the gamma distribution ($E[X]$) is equal to the shape parameter (k) multiplied by the scale parameter (θ). It is assumed that most mutations are slightly deleterious or neutral irrespectively of the current environment and that the global mean of the DFE does not change during adaptation. With a fixed $E[X]$ the shape and the scale parameter are interdependent and it is sufficient to provide the mean and shape parameters to specify the gamma distribution. Thus the probability density function of the gamma DFE distribution is equivalently given by

$$g(x;k,E[X]) = f\Big(x;k,\theta = \frac{E[X]}{k}\Big). \tag{4.1}$$

Figure 4.4 shows two gamma distributions with a fixed mean and two different shape parameters. Other groups have used reverse gamma functions as a DFE function, however, that method necessitates fixing a starting point of maximum potential fitness increase. The gamma functions discussed (in Figure 4.4) and used here for the models are the standard gamma distributions.

As the population adapts to a new environment the shape of the DFE is changing, such that the gamma shape parameter (k) is a function of fitness level of the parent (w):

$$k = f(w) \tag{4.2}$$

The precise relationship of k to w at this point remains to be determined from data (later shown in Section 4.2.1, Figure 4.8 and Equation (4.7)).

Assuming that the function $k = f(w)$ and $E[X]$ are known then the new fitness of an offspring $w(\tau+1)$ from a mutation ($m = 1$) is given by

$$P(w(\tau+1)|m=1) = X \sim g(x, k = f(w(\tau)), E[X]),$$

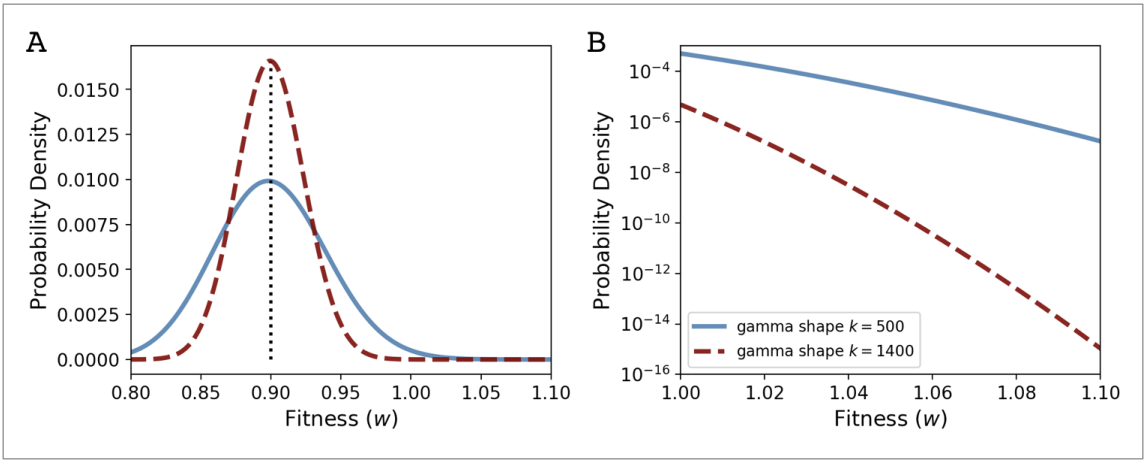


Figure 4.4: The gamma distributions of two different shape parameters and a fixed mean $E[X] = 0.9$ (*i.e.* the same mean as used for the fitting in Figure 4.7). **A:** The DFE for a k of 500 and 1400, and with a fixed mean. **B:** The distribution of mutations with positive fitness effects on a log scale. (Note, according to the plot in Figure 4.7E a Gamma shape $k = 500$ would correspond to a DFE shape at the beginning of adaptation (*i.e.* $w = 1$) while $k = 1400$ corresponds to a DFE near a fitness of $w = 1.4$.)

and for simplicity, going forward, this is written as

$$P(w(\tau + 1)|m) = X \sim DFE_0(x, w_k = w(\tau)), \tag{4.3}$$

where DFE_0 denotes a gamma function with a specified $E[X]$ and the value of w_k determines the shape parameter k as a function of the parent fitness w (from Equation (4.2)). Note that DFE_0 means an original (nominal) DFE due to environmental mutations, and in the absence of any construct (this terminology will be expanded upon in the next section).

The fitness, by definition (Equation (0.1)), always has a positive value, and for the practical purpose of the model realisations all fitnesses $w(\tau) \leq 0.2$ are approximated to $w = 0$. This assumption means that these individuals do not replicate at all. This slightly increases the computational processing speed (while the very low relative fitnesses which are approximated to 0 are near impossible to fix in a larger population). Also, in the process of determining a function for k in Equation (4.2), a few of the initial model realisations were implemented with a fixed and static gamma shape. Model realisations with static k values were used for Figures 4.5, 4.6 and 4.7D.

Fitness Discretisation

The model implementation employs discretised bins for fitnesses in order to better utilise computational capacity to avoid tracking each individual organism in the full population over the course of evolution. As a large number of individual cells (close to a billion) are processed in parallel during each model realisation, discretisation drastically spares computational resources during simulations of large populations. However, the discretised bins of the model have to be sufficiently small in order to successfully approximate the continuous evolutionary distribution effects of *in vivo* populations. At the same time, smaller sizes of discrete steps with larger number of bins to divide them require larger amounts of computational resources.

The first step of the model calibration process consists of testing what sizes of fitness bins are necessary in order to approximate a continuous distribution of fitness levels for individuals in a model population. Figure 4.5 shows the effect of different intervals of fitness resolutions. The word *resolution* here refers to the meaning of the smallest interval which is measurable and recorded. A high resolution means that the fitness is divided into many small bins, whereas a low and more coarse resolution means that fitness levels are less precisely recorded into fewer discrete bins. A low resolution of 10 means that the minimum recorded fitness interval is $w/10$, and equally, a high resolution of 10,000 means that the minimum recorded fitness interval is $w/10,000$. For example, with a model resolution of 1,000 then the fitness levels of the population are discretised into bins with fitness differences of $10^{-3}w$. Also, each fitness bin covers a slice of the DFE which has a width of $10^{-3}w$; the probability of a mutation with a new fitness of *e.g.* $w(\tau+1)=1$ is given by

$$P(w(\tau+1)=1|m)=\int_{0.9995}^{1.0005} DFE_0(x, w_k=w(\tau))dx. \tag{4.4}$$

As shown in the figure, utilising a resolution size in the range of 100 up to 10,000 has little effect on the output of the model, whereas very low resolutions of 10 or 20 distorts the results of the simulation. Numerical comparison of the values confirms the result shown visually on the plots. In addition, numerical comparisons show that a resolution of 50

marginally distorts simulation outputs, an effect which is too small to distinguish visually in the figure. A fitness resolution of 1,000 has been used for future investigations and model expansions. A resolution of 1,000 performs $\sim$ 1.5x slower than a resolution of 100, but $\sim$ 5x faster than a resolution of 10,000.

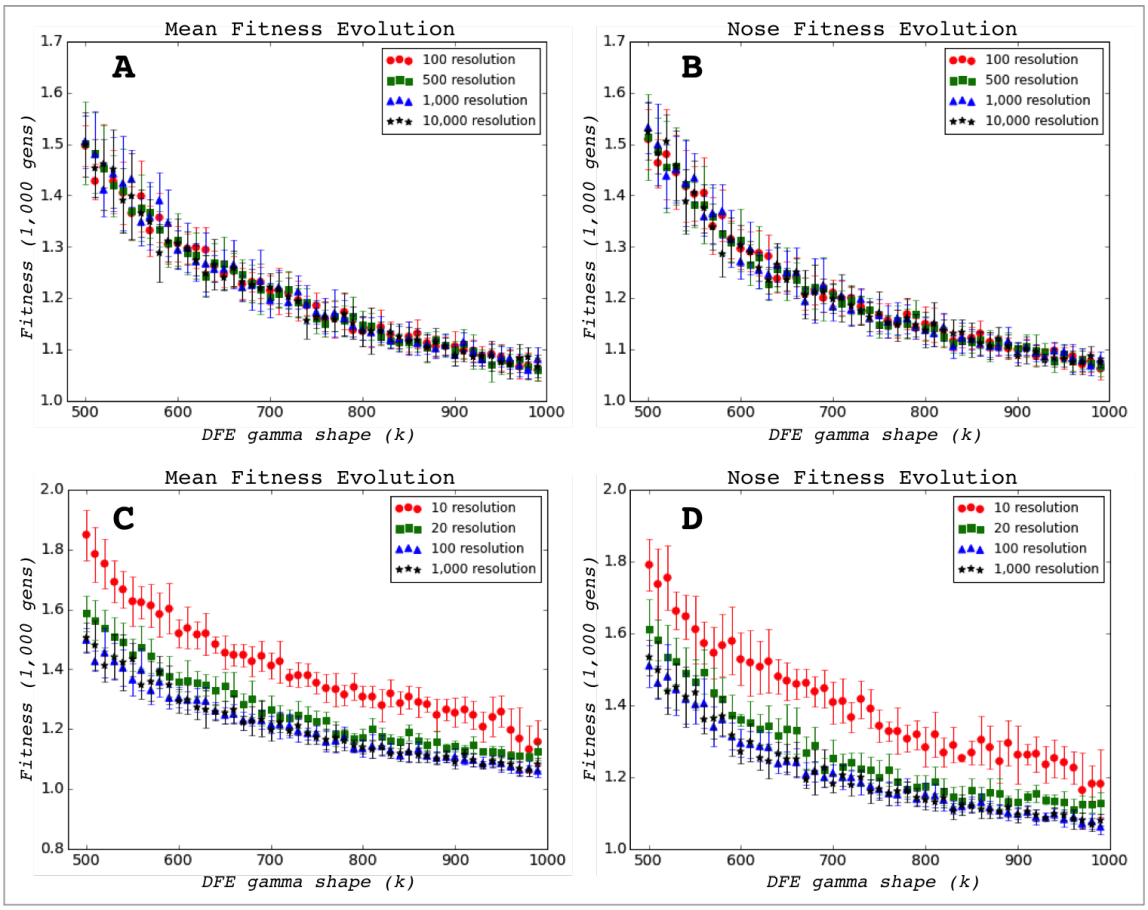


Figure 4.5: The effect of discretising the fitness levels in the model to different resolution sizes, *i.e.* dividing the population between different numbers of discrete fitness bins. A population discretisation resolution of 100 means that each fitness level interval is $w/100$ (i.e. 1%) of the initial relative fitness. The figure plots the final population fitness after 1,000 generations of evolution of either mean fitness $\bar{w}$ (**A, C**) as well as leading nose fitness w_{max} (**B, D**). The plots show comparison of the fitness outcomes over a range of gamma shapes and discrete resolutions. The plots illustrate that equivalent outputs are realised between resolutions of 100 to 10,000 (**A & B**), whereas lower resolutions distort simulation outputs (**C & D**).

Computational Replication and Dilution Procedures

The previous section specified how continuous fitness values are approximated to a continuous set of fixed fitness intervals. This section details how the specified set of fitness values are processed by the model.

An index-from-lists method is used to keep track of all individuals in the evolving population. Each index position in a list corresponds to a fitness value, and the value at the index determines the number of individuals at that fitness level. The index positions are referred to as fitness bins L_i in the fitness list L, and the index value (i) corresponds to a fitness value. In the example of a fitness resolution of 1,000 (Figure 4.5) then the position L_1 corresponds to the subpopulation with a fitness of $w = 0$, and the list position L_{1001} corresponds to the subpopulation with a fitness $w = 1$. In the model flowchart (Figure 4.3), at the step "initialise population" the fitness bin $L_{1001} = 10,000$ and all other positions are equal to zero.

At the replication step of the model, a copy of the original list is produced to represent the offspring. However, the values of each index in the list change from the first list according to the index, and thus the fitness of each subpopulation at that index. The original list with the parent population is denoted L, and the replicate list with the offspring population is denoted Lr. With a fitness resolution of 1,000 the replication process is written as

$$Lr_i = L_i \times \frac{i-1}{1000}.$$

However, only discrete values of individuals are stored in the indices, and decimal values are randomly rounded upwards or downwards with a probability according to their magnitude:

$$Round(Lr_i) = \begin{cases} \lfloor Lr_i \rfloor & \text{with probability } 1 - (Lr_i - \lfloor Lr_i \rfloor) \\ \lfloor Lr_i \rfloor + 1 & \text{with probability } Lr_i - \lfloor Lr_i \rfloor. \end{cases} \tag{4.5}$$

The replication event is subjected to the random occurrence of mutations. Mutated offsprings are identified and selected from the replication list (Lr) by a binomial distribution (B), and the list of mutated offspring is annotated Lm. From each position of Lr the number of mutated offspring ($N_{mutated}$) is drawn with a probability of the mutation rate

(μ):

$$N_{mutated} = B(Lr_i, \mu),$$

and the mutated offspring are then immediately subtracted from the original Lr value:

$$Lr_i(new) = Lr_i(old) - N_{mutated}$$

The mutated offspring ($N_{mutated}$) are inserted into a third list of mutated offspring (Lm) by distributing their fitness value (*i.e.* list index) according to the fitness value of the parent (i):

$$Lm_X \rightarrow Lm_X + 1$$

where X is drawn from the DFE of the parent fitness:

$$X \sim DFE_0\Big(x, w_k = \frac{i-1}{1000}\Big)$$

and i is the index of the parent fitness bin, and the fitness resolution = 1000. This process is repeated in a loop $N_{mutated}$ times.

After replication of each fitness bin in L the three population lists are merged for each index position to produce the new population at generation $\tau + 1$:

$$L(\tau + 1) = L(\tau) + Lr(\tau) + Lm(\tau) \tag{4.6}$$

After merging the temporary offspring lists of Lr and Lm with L, all values of Lr and Lm are reset to null. The lists are recreated and repopulated with a new set of offspring (Lr) and mutated offspring (Lm) at the next generation cycle.

Before the replication process resumes, a test is made of whether the size of the total population ($N_{total} = \sum_i L_i$) of L exceeds the maximum allowed population of the growth container (N_{max}). If $N_{total} > N_{max}$ the population is diluted 100-fold. The dilution is implemented with a random binomial selection at each index of L, similar to the mutation event for Lr:

$$L_i(remaining) \sim B\Big(L_i, \frac{1}{\text{dilution-factor}} = 0.01\Big)$$

where $L_i(remaining)$ is the subpopulation in each fitness bin after the dilution and L_i is the corresponding subpopulation size before the dilution. The process resumes the replication with L (or $L(remaining)$ if diluted) as the initial population.

This sequence of events can be summarised as follows:

1. Initiate the population as an integer value at an index position i in a list L.
2. Replicate each fitness bin (i) from L into a discrete number of offspring as a new list (Lr), where the replication multiple is given by the index position.
3. Find the number of mutated offspring ($N_{mutated}$) from each replicated fitness bin in Lr.
4. Find the a new fitness for all $N_{mutated}$, by sampling from the DFE of the parent.
5. Distribute all the mutated offspring into a new list Lm, at index positions according to their new fitness value.
6. For each fitness bin index in L: sum all parents, offspring, and mutated offsprings (*i.e.* summation for each index of L, Lr and Lm).
7. If the total size (N_{total}) of the population L exceeds N_{max} then the population L is diluted (by a random binomial selection of each L_i, with a probability according to the dilution factor).
8. Repeat from step 2.

This is the general computational processing of the model in the absence of engineered constructs.

Measurement Time Periods

In order to parameterise Equation (4.3) and obtain the function for $k = f(w_n)$ (*i.e.* Equation (4.2)), a first step is to look at how populations evolve at a range of fixed k values. Fitting a series of *fixed* gamma shapes allows for the subsequent derivation of an approximate initial function for how the DFE is evolving over time (see Figure 4.7D and E).

When a population expands, as well as when it initiates adaptation from a non-mutated homogenous initial state, there is a time-delay before the population reaches a steady state of fitness velocity (w/τ). At the same time, recording the fitness velocity (w/τ) over very large number of generations is computationally expensive, especially as a large number of static DFE shapes need to be tested in order to obtain good correlations between the gamma shape (k) and the fitness velocity (w/τ). In addition, if the evolutionary speed is measured against the generation time of the initial starting population fitness, then the populations that increase in fitness rapidly appear to adapt at an increasing velocity, and populations that deteriorate rapidly would appear to reduce their rate of fitness decline. This is an effect of increasing or decreasing amounts of evolutionary generation time passing between each measurement point, while the measured clock time interval is constant (*i.e.* the generation cycle τ of the starting population). An example of this potential distortion can be seen in Figure 4.6 where a slight increase in fitness velocity over time is observed for the populations with the highest rate of adaptation and a gamma shape $k = 500$ (red circles).

In summary there are two trade-offs: (i) record timespans that are short enough to represent the initial velocity of a *fixed* shape DFE, while at the same time (ii) record a timespan that is long enough to enable the population to disperse into an efficient and representable fitness distribution.

The fitness velocity shown in Figure 4.6 is recorded as the fitness increase over 100 generations in a timespan of 3,000 generations (*i.e.* the fitness velocity, with unit w/τ, is calculated by $\Delta w/\Delta\tau = \frac{\Delta w}{100\tau}$ where Δw is the fitness at the end of a $100 + \tau$ generation point subtracting the fitness at the τ generation point, $\Delta w(\tau) = w(\tau + 100) - w(\tau)$).

The time span selected for fitting of fitness velocity against static DFEs (in Figure

4.7D) are the 1,000 generations from 400 generations to 1,400 generations. *I.e.* the fitness velocity (w/τ) referred to in Figure 4.7D is $\Delta w/\Delta\tau = \frac{w(1400)-w(400)}{1000\tau}$. The selected velocity reference timespan is highlighted with a black line in Figure 4.6B.

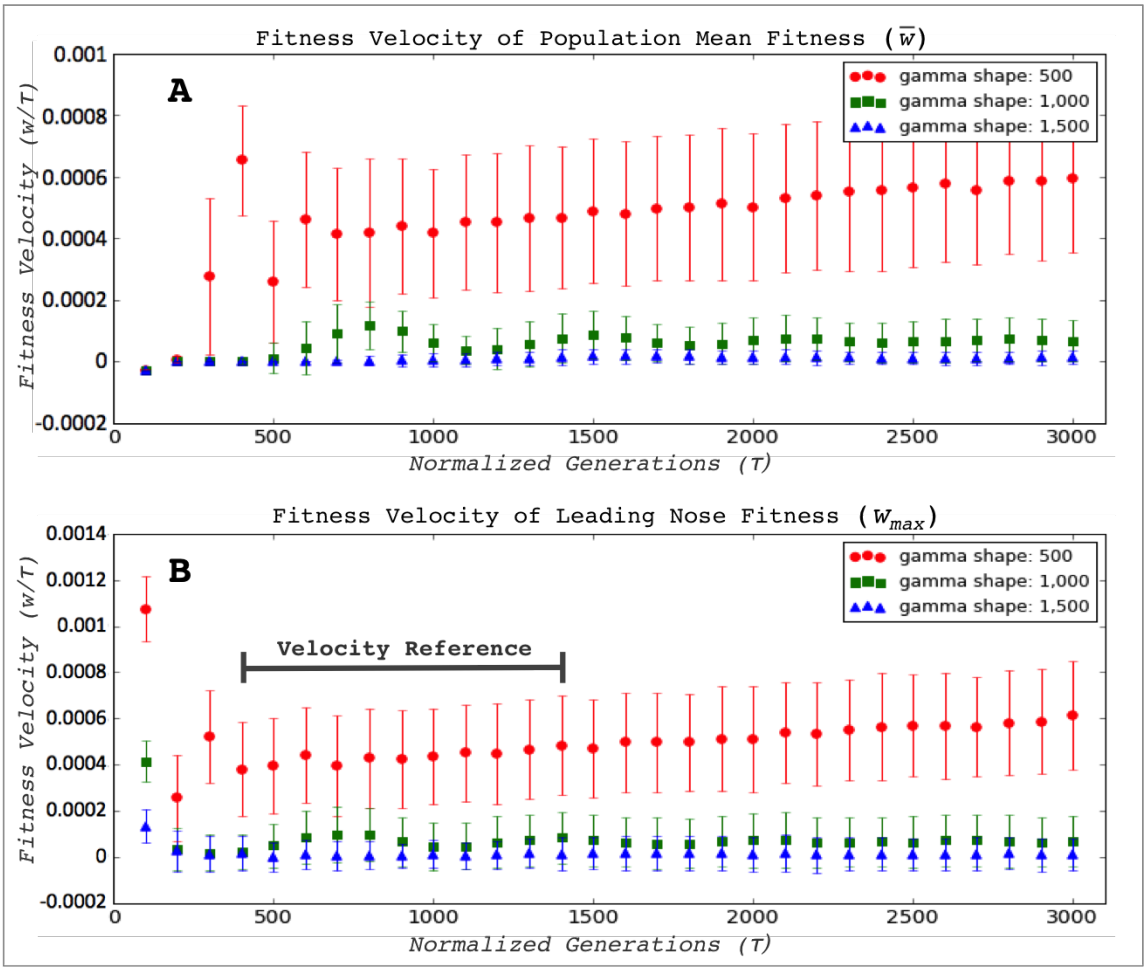


Figure 4.6: The fitness velocity plotted against age of population (measured as generations of the initial population) for three different gamma shapes (k) and a fixed gamma mean ($E[X] = 0.9$). Each data point is the mean value of 1,000 model realisations with the error bars showing the standard deviation. **A:** The fitness velocity of the mean fitness of the populations ($\bar{w}$). **B:** The fitness velocity of the leading nose fitness of the populations (w_{max}). The fitness velocity over shorter time periods (*e.g.* 100 generations) is too short for identifying consistent adaptation velocities. The black line (in **B**) shows generation time span selected for the fitting of DFE gamma shape evolution to fitness velocity. The time points chosen for later fitting of fitness velocity (in Figure 4.7D) are the 1,000 generations from 400 generations to 1,400 generations.

Model Parameterisation using the LTEE

The first step of obtaining a function for the gamma DFE evolution consists of fitting static gamma distributions to experimental time points (Figure 4.7E). This is in order to identify an approximate function for the DFE, and knowing how k depends on w, *i.e.* the general shape for $k \approx f(w)$. The following steps consist of fine tuning the indicated gamma shape trajectory by least squares fitting, as well as verifying the mean of the gamma distribution ($E[X]$) against experimental data for hypermutator phenotypes.

Finding an Approximate Gamma Function

Wiser *et al.* [201] have previously showed that the population fitnesses of the normal mutators in the LTEE follows a power law, where the relative fitness of the populations can be described by $w = (0.00871 \times \tau + 1)^{0.0830}$. This power law function is shown in Figure 4.7A. Next, the provided fitness as a function of number of generation (w as a function of τ) in Figure 4.7A has been converted to fitness velocities in Figures 4.7B (w/τ as a function of τ) and C (w/τ as a function of w), from derivation of the power law in Figure 4.7A. Secondly, the fitness velocities from a range of fixed gamma functions have been simulated and are plotted as w/τ as a function of k in Figure 4.7D. The fitness velocities are recorded as specified in Section 4.2.1 and Figure 4.6B. Subsequently, in Figure 4.7E the model gamma shapes from Figure 4.7D are plotted against the LTEE population fitness from Figure 4.7C. The data points in Figure 4.7E are produced by aligning the y-axis from the Figures 4.7C and D, *i.e.* aligning the model fitness velocity and the experimental fitness velocity (w/τ). This alignment of fitness velocities provides an initial representation of gamma shapes *versus* current fitness, and thus a guide for the shape of the function $k = f(w)$.

A visual inspection of Figure 4.7E suggests a linear relationship between k and w. The next step of finding a function for the non-static DFE consists of least squares fitting of ranges of evolving DFEs against extended time periods of experimental data from the LTEE. For this purpose it has been assumed that $k = f(w)$ is an affine function, as observed in Figure 4.7E.

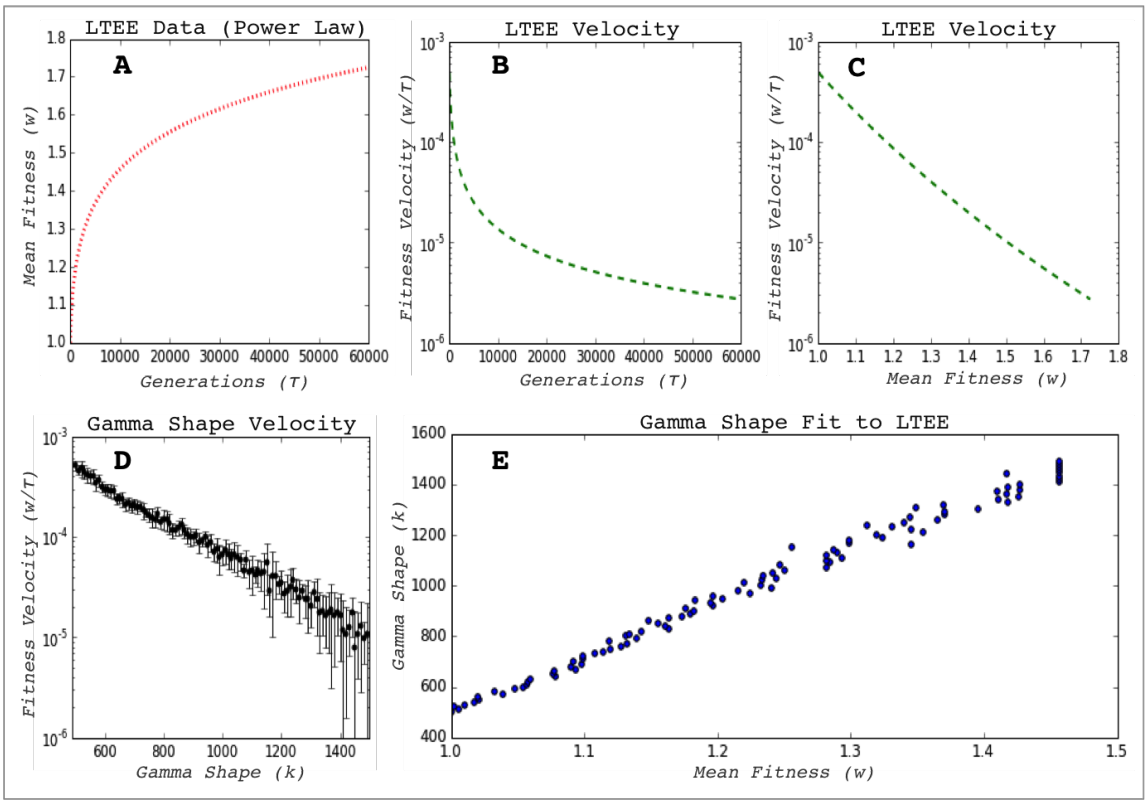


Figure 4.7: The relationship between fitness velocity from individual fixed gamma shapes and measured experimental evolution states from the LTEE. Alignments between these two independent data sets are used to infer a function for how the DFE (in the shape of a gamma distribution) is evolving as populations continuously are adapting to a new environment. More precisely, the process plotted is used to guide the general shape of the function $k = f(w)$ (Equation (4.2)). **A:** Fitting of a Power Law to experimental data of relative fitness (w), originally used by Wiser *et al.* [201] (the power law plotted is included in the reference). **B:** Conversion of the fitted Power Law to the fitness velocity over time (from the derivate of A). **C:** The rate of adaptation in the LTEE as a function of previous adaptation to a new environment (from combining A with B). **D:** The fitness velocity as a function of the shape of the gamma distribution. The black circles are the mean of 100 model realisations, and the error bars show the standard deviations. **E:** The experimental fitness velocity in the LTEE (in plot C) is aligned with the fitness velocity for gamma shapes (in plot D) to show what previous adaptation levels from experiments correspond to different gamma shapes. It provides a guide for the shape of the function $k = f(w)$, which can be approximated by an affine function.

Fitting of the DFE

Fitting an evolving DFE against data assumes that the DFE shape is a function of the current fitness level. In other words, a replicated genome which (by chance) becomes subjected to a mutation will draw its new fitness level from a distribution which depends on the current fitness level of the parent genome.

A time extract of 10,000 generations from the LTEE was used to fit the evolving DFE function against experimental data. The evolving DFE was fitted via least squares fitting against the reference fitness at intervals of 1,000 generations. Both normal mutators and hypermutators were fitted to simultaneously to yield a total of 20 fitting points. The simulated normal mutators have a mutation rate $\mu = 0.003\tau^{-1}$ and the hypermutators $\mu = 0.3\tau^{-1}$.

It is assumed that the mean of the DFE stays constant, while the shape parameter is changing over time. The scale parameter is then obtained from $\theta = E[X]/k$, where $E[X]$ is constant. Three parameters are fitted to describe the evolving DFE:

- $k_{w=1}$: The initial gamma shape parameter of the DFE, and when $w \leq 1$.
- $E[X]$: The mean of the gamma function.
- Ω: A linear scaling factor for k.

The initial gamma shape parameter ($k_{w=1}$) and the linear scaling factor (Ω) are linked as follows to provide the Equation (4.2) ($k = f(w)$):

$$k = k_{w=1} + \Omega(w - 1) \qquad \{w \geq 1\} \tag{4.7}$$

Where k is the shape parameter of the gamma distribution, w is the current level of fitness of an individual in an environment, and Ω is simply a linear factor for the current level of adaptation.

In Wiser *et al.* [201] the fitness trajectory for normal mutation rate samples is given by the power law $w = (0.00871 \times \tau + 1)^{0.0830}$. The hypermutator samples, with start at 4667 generations, can also be given by a power law function in a shape of $w = (0.02144 \times (\tau -$

$4667)+41.65)^{0.0830}$, where $\tau \geq 4667$ (see Figure 4.8B). When fitting the DFE function to the LTEE, then the leading nose fitness (w_{max}) is used instead of the mean population fitness ($\bar{w}$). This is because introducing a higher mutation rate in the middle of the simulation creates a drastic jump in $\bar{w}$, whereas w_{max} provides a smoother value to fit against. Also, the size of the jump in $\bar{w}$ is very much dependent on the exact shape of the distribution of mutations with negative effects, and therefore provides a less consistent output between different input parameters. As shown visually in Figure 4.8 the identified parmeters for the evolving DFE are $E[X] = 0.85$, $k_{w=1} = 340$ and $\Omega = 800$. The function which was fitted for 10,000 generations was subsequently simulated for 50,000 generations, and the results are shown in Figure 4.8C. The fitness trajectory for the normal mutator populations are seen to be consistent with the experimental data for the longer time period. The fitness for the simulated hypermutator populations increases less rapidly than the reference hypermutator function when extended for 50,000 generations. However, this adaptation rate reduction for hypermutators is not fully inconsistent with the LTEE experiment which showed that the hypermutators lost their fitness advantage during the experiment, but also that elevated mutation rates subsequently could be reversed [199] and that all four populations which evolved hypermutability early in the LTEE subsequently evolved a reduced mutation rate with time [177].

The Presence or Absence of Negative Mutations

It has been argued that the effect from mutations with adverse fitness effects is very limited in the LTEE [199, 201]. Therefore mutations with negative fitness effects are excluded from the model developed by Wiser *et al.* [201]. This might be true for experimental populations with low mutation rates and a constant population size. However, as mutation rate increases, *e.g.* from fixation of hypermutators, or when population size reduces, considering also the effect from the large majority of mutations with non-positive effects has a higher importance. This effect from the distribution from negative mutations can be seen in Figure 4.9 where two alternative gamma means ($E[X]$) are fitted to the normal mutators, but resulting in very different trajectories for the hypermutator populations.

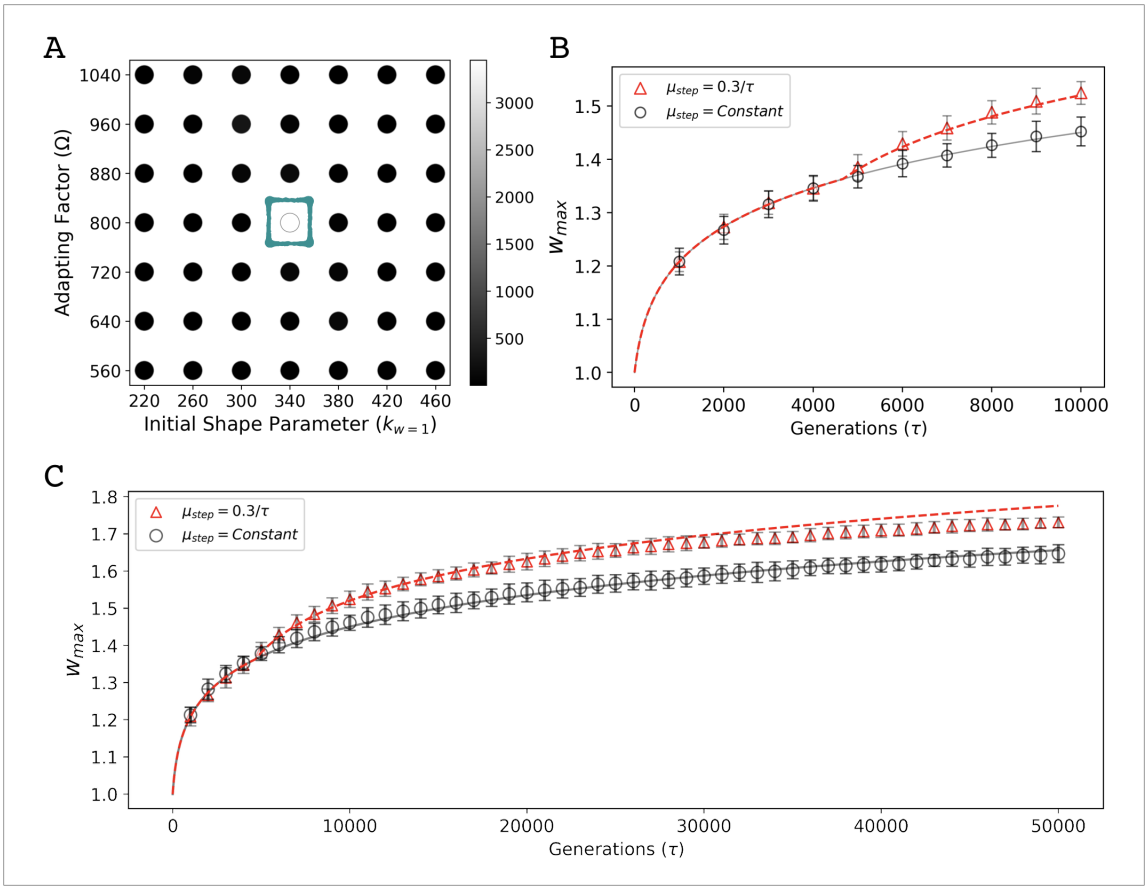


Figure 4.8: Fitting of the gamma shape parameter (k) and a linear Adapting Factor (Ω) to the LTEE fitness trajectory data. The fitting is implemented via a grid search of k, Ω and $E[X]$ to the first 10,000 generations of growth of the normal mutators simultaneously with the hypermutator genotypes. **A:** A two dimensional grid search of Ω and $k_{w=1}$ for gamma mean $E[X] = 0.85$. The gray color shade shows the inverted Least Square fit of the mean for w_{max} from 50 model realisations of simulated growth for each parameter combination. The fit is taken at steps of 1000 generations for the first 10,000 generations of growth. The teal coloured box indicates the best parameter fit. **B:** The fitting of the best parameters ($\Omega = 800$ and $k_{w=1} = 340$) to the first 10,000 generations of growth of the LTEE. **Gray line:** the fitness trajectory of the LTEE normal mutators (*i.e.* $\mu = 0.003$). **Red dashed line:** the fitness trajectory of the LTEE hypermutator, which increases the mutation rate from $\mu = 0.003$ to $\mu = 0.3$ at 4667 generations. **Gray circles:** the mean of w_{max} of the normal mutator from the simulated parameters ($\Omega = 800$ and $k_{w=1} = 340$) at each 1000 generation step used for the fitting. **Red triangles:** the mean of w_{max} of the hypermutator from the simulated parameters ($\Omega = 800$ and $k_{w=1} = 340$) at each 1000 generation step used for the fitting. **C:** A plotted extension of the LTEE fitness trajectories together with same the model parameters from **B** re-simulated for 50,000 generations. Symbols are the same as in **B** and with 20 model realisations. The fitness trajectory of the normal mutator follows the extended trajectory of the LTEE. The fitness trajectory of the hypermutator is not sustained, however, this is not completely inconsistent with the experimental findings of the LTEE which shows that the hypermutation populations can evolve back to reduced mutation rates as the fitness advantage of hypermutators are reversed with time [199, 177].

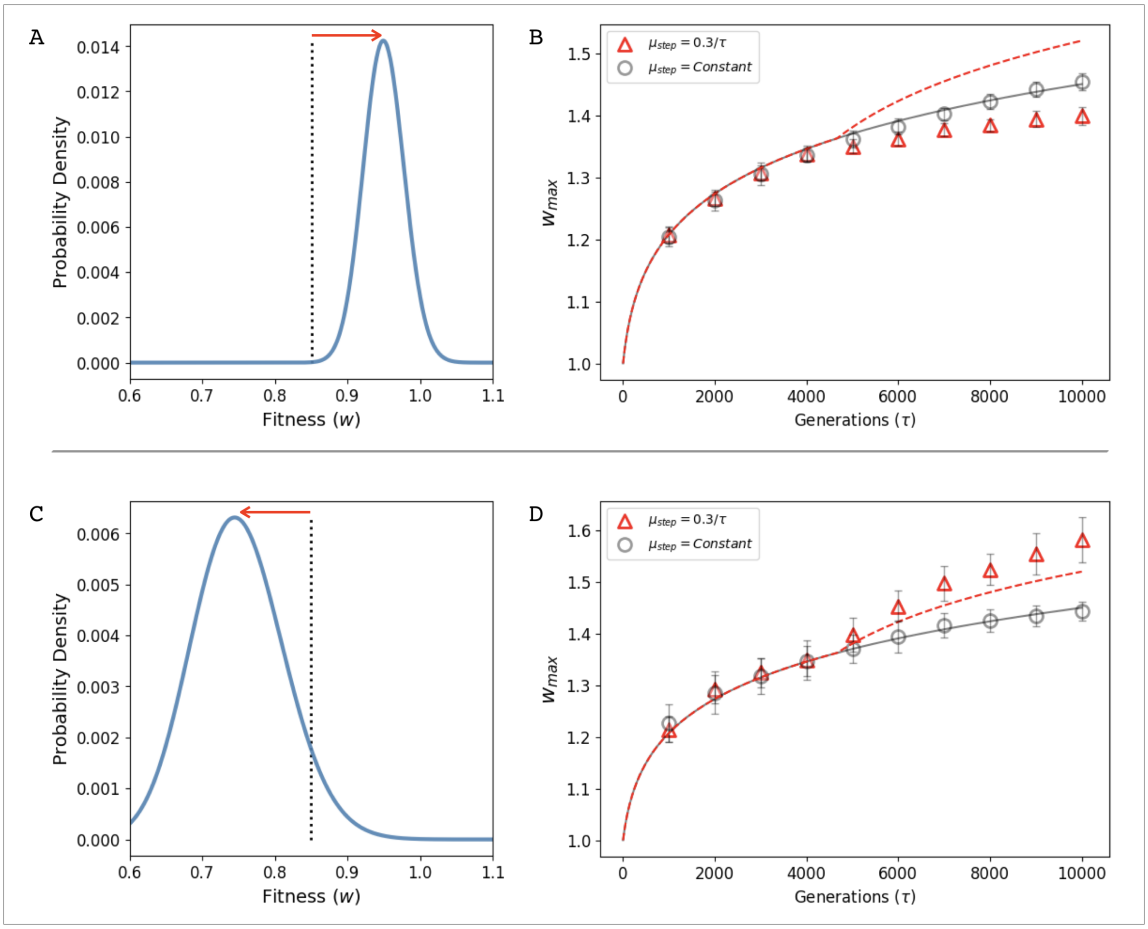


Figure 4.9: Demonstration of what happens to the DFE fitting with a changing gamma mean ($E[X]$). Multiple combinations of $E[X]$, k and Ω can be fitted to populations with a single mutation rate (μ). But when including more than one mutation rate the degree of freedom is reduced such that a single combination $E[X]$, k and Ω fits the data (*i.e.* the results shown in Figure 4.8). **A:** The starting position ($DFE_0(w = 1)$) of a gamma function with $E[X] = 0.95$ after least square fitting to the normal mutators of the LTEE. The gamma shape $k = 1150$ with a linear Adapting Factor $\Omega = 7000$. **B:** The DFE function with $E[X] = 0.95$ was fitted to the normal mutators (**gray circles**). When realising the same DFE function with a 100 fold increase of μ at 4667 generations then the rate of fitness velocity decreases instead of increases (**red triangles**). **C:** The starting position ($DFE_0(w = 1)$) of a gamma function with $E[X] = 0.75$ after least square fitting to the normal mutators of the LTEE. The gamma shape $k = 140$ with a linear Adapting Factor $\Omega = 240$. **D:** The DFE function with $E[X] = 0.75$ was fitted to the normal mutators (**gray circles**). When realising the same DFE function with a 100 fold increase of μ at 4667 generations then the rate of fitness velocity increases too fast compared to the LTEE data (**red triangles**).

Discussion and considerations of fitting mechanics: finite versus infinite mutation space

It has been argued whether populations do or do not 'run out' of ways to adapt [107]. However, with the implemented model, the question is not so much whether populations 'run out' of new mutations, as there will always be available mutations with positive fitness effects in the DFE (in Equation (4.3)). Instead, what is happening over time is that the diminishing proportion (and magnitude) of positive mutations become counteracted by the fairly constant proportion of slightly negative mutations. This means that even if the population is still evolving it increases in fitness very slowly due to the downward pressure of negative mutations. The result is that the effective population size very much determines the velocity at which the population is able to continue improving its fitness. Another observation of the model is that a higher mutation rate will be favoured initially at a low level of adaptation in a new environment and then an increasingly reduced rate of mutations will be favoured as the population becomes better adapted. This reduces the negative effect of the increasingly larger proportion of slightly negative available mutations.

If only taking mutations with positive fitness effects into account, such as for the proposed model by Wiser *et al.* [201], the fitness velocity will always be favoured by higher mutation rates, contrary to what data suggest [199, 177].

It would be very interesting to explore and measure the rate of adaptation for a few different population sizes from a later stage of the LTEE, together with less well adapted populations from the LTEE. For instance, a 100 times smaller and a 100 times larger population size compared to the standard LTEE culture size (currently $10mL$), and measured over a medium-long time period. Such experimental data could provide very valuable information for providing more insights of the shape of the DFE and how its shape is altered as a function of adaptation.

4.2.2 Model Step 2 - Including Engineered Constructs

The approach taken for building a model which includes engineered constructs consists of using the model derived from the LTEE data in Section 4.2.1, and then extending it with a small number of well defined assumptions regarding engineered biology. The extended model follows the same general outline as the flowchart in Figure 4.3, where the key extended assumptions can be summarised as follows:

- When introducing an engineered construct into a host organism the DFE of the host organism changes.
- The DFE of an organism can be divided into sub-areas (component DFEs), each with its corresponding frequency of mutations μ_i (and $\mu_{total} = \sum_i \mu_i$).
- The DFE of a host carrying an engineered construct can be divided into sub-areas of (i) 'normal environmental mutations', (ii) 'construct destructive mutations' and (iii) 'construct accommodating mutations'.
- The distribution of fitness effects (DFE) for the subset of mutations for *environmental adaptation* takes the form of a gamma distribution whose shape depends on the level of adaptation to the current environment, but the shape is independent from effects of engineered biology (*i.e.* the shape is the same as the DFE function identified in Section 4.2.1).
- Reverse mutations, *i.e.* mutations that restore function of a construct previously lost, are non-existent.

In the previous section (4.2.1), the fitness impact from a new mutation (m) were drawn from a DFE (*i.e.* DFE_0) with a probability of $P(m) = \mu$. In this section, the sub-areas of the DFE are annotated DFE_n, DFE_d and DFE_A, and the probability frequencies of the three sub-areas are given by μ_n, μ_d and μ_a. The reason for dividing different types of mutations into different subareas of fitness effects is simply for the purpose of being able to separate and model different types of mutations. A general outline for the new assumptions is illustrated in Figure 4.1 in the introduction of this chapter. The following

text in this section provides a precise definition for these assumptions and how they are linked.

Fitness Contributions

There are two possible distinct subpopulations at any time: individuals that have lost the specified function of the construct due to mutations in the DNA sequence, and those that have not lost their bespoke function.

The total fitness of an individual is given by the sum of environmental fitness adaptation and the fitness benefit of reducing the cost of the construct,

$$w_{total} = w_n + s_d + s_a \tag{4.8}$$

where w_{total} is the cumulative fitness, w_n is the fitness from environmental adaptation, s_d is the additional fitness from construct loss and s_a is the possible fitness gain from construct accommodation.

Equations for the distribution of fitness effects

Environmental Distributions ($DFE_n = DFE_0$)

When an environmental mutation occurs ($P(m_n) = \mu_n$), the new fitness level ($w_n(\tau + 1)$) resulting from the mutation is drawn from a distribution of fitness effects (DFE_0) which depends on current fitness level from environmental adaptation ($w_n(\tau)$).

$$P(w_n(\tau + 1)|m_n) = X \sim DFE_0(x, w_k = w_n(\tau)) \tag{4.9}$$

The DFE function for environmental adaption (DFE_n) is the same function as the parameterised DFE function (from Section 4.2.1 and Equation (4.3)) for environmental adaptation in the absence of an engineered construct. The DFE function is determined by $w_n(\tau)$, which is the environmental fitness level of the parent genome before the mutation occurs, and $w_n(\tau + 1)$ is the fitness level in the environment after the mutation.

Destructive Distributions (DFE_d)

The fitness benefit of a destructive mutation ($s_d = w_d - 1$) given that a destructive mutation has occurred (m_d, where $P(m_d) = \mu_d$), is assumed to follow a uniform distribution and is given by

$$P(w_d|m_d) = X \sim DFE_d(x) \tag{4.10}$$

and

$$DFE_d(x) = \begin{cases} \frac{1}{S_0} & \text{for } x \in [1, 1 + S_0] \\ 0 & \text{for } x < 1, \text{ or } x > (1 + S_0) \end{cases}$$

where the fitness is drawn from a distribution between 1 (*i.e.* no fitness improvement) up to a full recovery of the fitness cost initially imposed by the construct (*i.e.* $1 + S_0$), and $\int_1^{1+S_0} \frac{1}{S_0} dx = 1$. The uniform function is assumed to be independent from previous levels of destruction as well as from possible previous levels of accommodations of the construct by the host. Apart from being a simple assumption, a uniform distribution is motivated by the following argument (i) mutations exist which distort the function of the construct without providing any fitness benefits to the host, *e.g.* some amino acid substitutions of a protein, (ii) some mutations of the construct sequence completely restore the fitness to the host, *e.g.* mutations in the promoter sequence which lead to complete silencing as well as full deletions of the whole construct sequence. Finally, a full range of possibilities in the range between these two points are also plausible. Otherwise, a normal distribution of fitness effects from loss of construct could also have been plausible, or a skewed normal distribution. However, in the absence of further information a uniform distribution is the best starting point. Clearly, construct mutations are also possible which can reduce the fitness of the host further, *e.g.* by increasing the expression level of engineered proteins. But these mutations face increasing selection pressure against them and therefore they are considered less important in the model.

The model assumes that there is no epistatic effect internally within the distribution of mutations which are destructive to the construct. This means that even if a construct is partially destructed, the distribution of fitness effects for destructive mutations is the same

as if the construct did not have any destructive mutations. As an example, if the probability density for the fitness effect change of a destructive mutation was evenly distributed between full construct loss ($s_d = S_0$) and full function ($s_d = 0$) and there had already been one destructive mutation reducing the cost by 50% ($\frac{s_d}{S_0} = 0.5$), then the next destructive mutation does still have the same probability density function, evenly distributed between zero ($s_d = 0$) and full loss ($s_d = S_0$), as if there had been no prior destructive mutation. The construct will always remain destructed, as there are no reverse mutations, but in this example of an even distribution, the probability of fitness gain versus fitness loss would be 50%.

Accommodating Distributions (DFE_A)

True 'accommodating mutations' will be through adaptations to the engineered construct. These mutations are found within the subsection of the DFE for accommodating mutations (DFE_A). This subsection of the DFE (*i.e.* DFE_A) is due to alterations of the host *internal environment* and it is assumed to change similarly to the change of a DFE during the adaptation of a host to a new *external environment*. However, the extent of initial DFE change depends on the type of construct that has been introduced. Due to the lack of existing data for the novel question of internal construct accommodation, this is the simplest assumption. If DFE data from simple genetic alterations (*e.g.* gene knock-out strains) are produced in the future, then such data would provide an additional, and likely enhanced, approximation for the shape of DFE_A.

The distribution of accommodating mutations (DFE_A) is derived from DFEs of environmental shape with different levels of adaptations. The DFE is assumed to be the difference between the shape of the curve at full level of fitness recovery from construct loss ($w = 1 + S_0$), and the current fitness cost (S) imposed by the construct ($w + (S_0 - S)$, or equally $w + s_a$). The cost of the construct ($S = S_0 - s_a$) can be at a reduced level by accommodating mutations that previously occurred, such that $s_a \neq 0$. In other words, DFE_A changes from accommodating mutations to the internal environment (imposed by the engineered construct) in the same way as a DFE_n changes as an effect of adaption to a new external environment. The accommodating mutation is assumed to have a positive fit-

ness impact, and it can not exceed the fitness advantage of losing the construct $s_a \in [0, S_0]$ (and $s_{a+1} \in [s_a, S_0]$, where s_{a+1} refers to an additional accommodating mutation in a host which already has an accommodation level = s_a). If the construct cost $S_0 = 0.2$ and there is no cost alleviating impact from accommodation (*i.e.* $s_a = 0$), then the shape of the distribution for accommodating mutations is assumed to be equal to an environmental DFE if the environmental fitness level had been equal to one (*i.e.* $DFE_0(w_k = 1 + 0)$) subtracting the shape of an DFE had the environmental fitness been equal to $w = 1.2$ from full construct recovery (*i.e.* $DFE_0(w_k = 1+0.2)$). The distribution curves are both limited by the initial fitness as the lower bound ($w_a = 1 + s_a$) and full recovery from construct cost ($1 + S_0$). The fitness benefit from an accommodating mutation ($s_a = w_a - 1$) of an accommodating mutation (m_a, where $P(m_a) = \mu_a$) is given by

$$P(w_a|m_a) = X \sim DFE_A, \tag{4.11}$$

where

$$DFE_A = \frac{DFE_{0\left[1+s_a,1+S_0\right]}(x, w_k = 1 + s_a) - DFE_{0\left[1+s_a,1+S_0\right]}(x, w_k = 1 + S_0)}{\int_{1+s_a}^{1+S_0}(DFE_0(u, w_k = 1 + s_a) - DFE_0(u, w_k = 1 + S_0))du}. \tag{4.12}$$

The distribution is normalised by dividing the area with its own size (*i.e.* $\int_{1+s_a}^{1+S_0}(DFE_0(u, w_k = 1 + s_a) - DFE_0(u, w_k = 1 + S_0))du$) such that the resulting total probability distribution is equal to one. $DFE_{\left[y,z\right]}$ means the function of $DFE(x)$ between a lower limit $x \geq y$ and an upper limit $x \leq z$. The shape of DFE_A is plotted in Figure 4.10.

Equations for the probabilities of different types of mutations

The mutation rate for the organism can either be given as a rate per replicated basepair or as a rate per replicated genome. The mutation rate for an organism is here given as μ, which is simultaneously represented as the probability of a mutation per replicated genome per generation (τ^{-1}). Many organisms, including *E. coli*, have a mutation rate of $\mu \simeq 0.003\tau^{-1}$ [59, 114]. However, mutations can also cause this basal mutation rate to increase by one or two orders of magnitude, as has been observed to occur repeatedly in

the LTEE [201].

Destructive Probabilities (μ_d)

When inserting an engineered construct into the genome of the organisms the total number of basepairs increases. Thus, if the underlying mutation rate per basepair is assumed to stay constant, the total mutation rate per genome replication increases,

$$\mu_{total} = \mu + \mu_d, \tag{4.13}$$

where μ is the original mutation rate per genome and μ_d are the additional mutations from mutations occurring in the additional DNA sequence constituting the construct. The construct sequence can often have an elevated mutation rate per basepair compared to the rest of the genome [92]. However, if assuming a well designed DNA sequence then the construct mutation rate per basepair can approach the rate per basepair of the genome. In such instances the mutation rate per genome of the construct is proportional to the length of the construct DNA sequence. As an example, if the genome size of the host is 3 million basepairs and the size of the construct is 15k basepairs then the destructive mutation rate $\mu_d = 0.005\mu$, and $w_{total} = \mu(1 + 0.005)$. This assumption (of equal rates per base pair of construct and genome) also means that the mutation rate of the construct will increase proportionally with an increased underlying mutation rate (*i.e.* $\mu_d \propto \mu$). If the DNA sequence of the construct is badly designed and contains an excess of repeats with elevated mutation rates independent of the underlying mutation rate per basepair then mutation rate per construct might not follow a linear relationship to the genome mutation rate (*i.e.* if the genome mutation rate were to change). But for simplicity (or if assuming well designed DNA sequences of the construct void of mutational hotspots) it is assumed in the model that $\mu_d \propto \mu$.

Environmental and Accommodating Probabilities (μ_n and μ_a)

In the absence of an engineered construct, destructive mutations and accommodating mutations, then the rate of normal environmental mutations (μ_n) is equal to the underlying

mutation rate, as well as the total mutation rate ($\mu_{total} = \mu_n = \mu$). However, in the presence of a construct two things are assumed to happen. Firstly, μ_{total} is assumed to increase in line with the additional rate of mutations within the construct DNA as described in the previous paragraph ($\mu_{total} = \mu + \mu_d$, Equation (4.13)). Secondly, the mutations occurring within the original genome of the organism are assumed to be divided between two different effects; (i) those mutations which alter the fitness of the organism independently of the presence of the construct, denoted m_n (where $P(m_n) = \mu_n$), and (ii) those mutations which alter the fitness of the organism only in the presence of the current construct, denoted m_a (where $P(m_a) = \mu_a$). To summarise, in the presence of an engineered construct the total rate of mutations is assumed to be divided as

$$\mu_{total} = \mu_n + \mu_a + \mu_d, \tag{4.14}$$

with

$$\mu = \mu_n + \mu_a. \tag{4.15}$$

Equation (4.15) also follows from Equation (4.13) together with (4.14). The proportion of μ_a to μ is assumed to be a function of the current cost of the construct (S) times some proportionality constant Υ.

$$\mu_a = f(S, \mu, \Upsilon) \tag{4.16}$$

More precisely,

$$P(m_a|m) = \Upsilon \int_{1+s_a}^{1+S_0} (DFE_0(x, w_k = 1 + s_a) - DFE_0(x, w_k = 1 + S_0))dx. \tag{4.17}$$

In other words, the probability of a mutation with accommodating effects, given a mutation in the genome, is assumed to be equal to additional area of mutations with positive fitness effects created by $DFE_0(w_k = 1 + s_a)$ subtracting $DFE_0(w_k = 1 + S_0)$, limited by the total cost of the construct and multiplied by some proportionality constant Υ. This means that engineered constructs with a larger cost to a host will have a larger impact on the

DFE. Note that Equation (4.17) without the proportionality constant Υ is equal to the area of Equation (4.12) prior to the normalisation by area division, see Figure 4.10. Also note, $\mu_a = P(m_a) = P(m_a|m) \times P(m) = P(m_a|m) \times \mu$. The proportionality constant Υ is assumed to depend on the type of cost which the construct imposes on the host. For example, costs primarily related to over expression of proteins and directly competing with resources for growth by the cell through *burden* are assumed to carry a low Υ, whereas cost primarily related to metabolic imbalances in the cells, which more easily could be compensated for by tuning the expressions of other metabolic pathways in the cells, are assumed to carry a higher Υ. The Υ dictates the type of fitness cost imposed by the construct and the degree to which these costs can be alleviated by random mutations. The construct mutation simulations later on in this chapter have assumed $\Upsilon = 1$. To put this assumption in context, an $\Upsilon = 1$ together with a construct cost $S_0 = 0.1$ gives $P(m_a|m) = 7 \times 10^{-7}$ (illustrated visually in Figure 4.10). In the same example, if an accommodating mutation occurs which improves the fitness by $s_a = 0.05$, then the new $P(m_{a+1}|m) = 5 \times 10^{-9}$. (Note, the probability of the first mutation of $s_a \geq 0.05$ was $P(m_a \geq 0.05|m) = 2 \times 10^{-8}$, or equally with a $\mu = 0.003\tau^{-1}$ then $\mu_a = \mu \times P(m_a|m) = 6 \times 10^{-11}\tau^{-1}$).

The shape as well as the area ratio of DFE_A versus DFE_n changes upon induction of a construct. At a higher induction level of a construct there is a higher fitness cost due to larger perturbations of the internal environment of the organism. The larger change of internal environment means that the probability area of accommodating mutations (μ_a) is increased relative to the probability area size of normal environmental mutations (μ_n). At the same time μ_d remains constant, while DFE_d is elongated between $s_d = 0$ and the *new* increased cost (S_0) to cover the new extended distribution of potential fitness impacts.

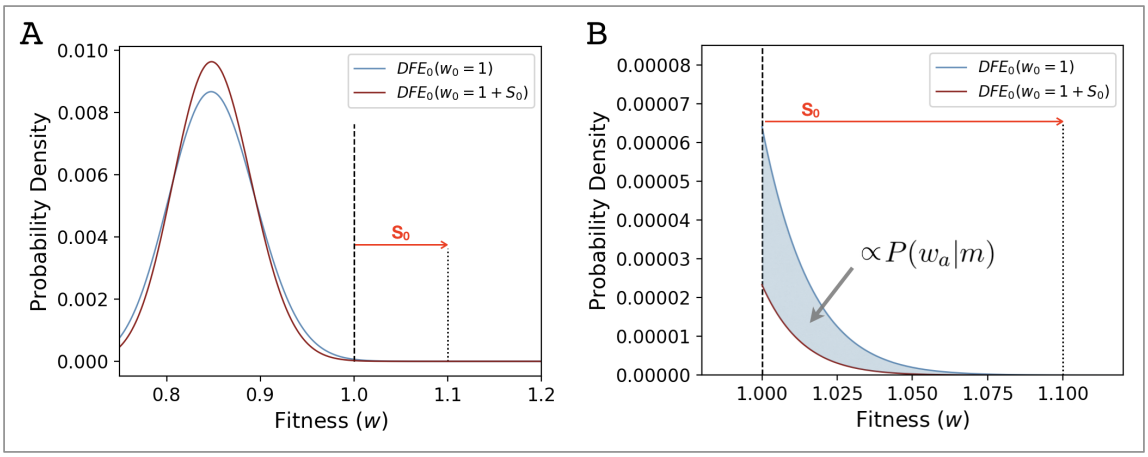


Figure 4.10: A visual representation of the DFE_A assumption and calculation. DFE_A is calculated as the difference between $DFE_0(w_k = 1)$ and $DFE_0(w_k = 1 + S_0)$ between the limits of $w_a \in [1, 1 + S_0]$, where S_0 is the fitness improvement if the construct were removed. At previous partial initial accommodation levels then the lower limit of 1 is replaced by $1 + s_a$ (not shown in the figure). The example illustrated here is for a DFE_0 with an initial $k = 340$ and $\Omega = 800w^{-1}$ as found from fitting in Figure 4.8. The construct fitness cost shown is $S_0 = 0.1$ with no previous accommodating mutation (*i.e.* $s_a = 0$). **A:** The DFE_0 with $w_k = 1$ (**blue line**) and DFE_0 with $w_k = 1 + S_0$ (**dark red line**). **B:** An expanded view of A. The distribution of fitness effects from accommodating mutations is equal to the area between the red and the blue line between $w = 1$ and $w = 1 + S_0$ (Equation (4.11)). If the proportionality constant $\Upsilon = 1$ then the highlighted area in B is also equal to $P(m_a|m) = 7 \times 10^{-7}$ as given in Equation (4.17), and $\mu_n = \mu - \mu_a$. Note, $P(m_a|m)$ refers to the probability of an accommodating mutation, whereas $P(w_a|m)$ refers to the probability density of fitness outcomes.

Computational Replication of Engineered Populations

The general flowcharts for the non-engineered model (Section 4.2.1) and the engineered model (Section 4.2.2), are identical. In other words, both procedures follow the same flowchart as outlined in Figure 4.3. But a few details of the steps within the flowchart are implemented differently.

The two processes differ for the replication event, which was specified in detail for the non-engineered model in Section Computational Replication and Dilution Procedures. Also, the engineered model expands the list structure for how the population is first initiated (see Figure 4.3 and Section Computational Replication and Dilution Procedures). In the previous list structure (without a construct), each index (i) in the list L contains a value which represents a subpopulation size (with a fitness determined by i). Instead, in the model implementation with constructs the subpopulation size value is moved two hierarchies further down in the list L. Each index L_i contains a sublist with indices j. There are two positions of j in this sublist: $j = 1$ indicates hosts with non-destructed constructs, and $j = 2$ are hosts with destructed constructs. In each of the two j positions there is a second sublist with indices q, and with an equal length for $j = 1$ and $j = 2$. Note, the matrix indices ijq (instead of more intuitive ijk) are used here to avoid any double annotation with k as the shape parameter for the gamma distribution. Each of the two sublists in j (with indices q) denotes the fitness recovery from either construct loss ($j = 2$) or construct accommodation ($j = 1$). In other words, the q index determines the value of s_d or s_a. For example (and with a fitness resolution of 1000), $L_{i=1101,j=2,q=51}$ gives the subpopulation with a destructed construct and a fitness of $1.15 = 1.1 + 0.05$ ($w_{total} = w_n + s_d$ and from a fitness resolution of 1,000 then $w_{total} = (i-1)/1000 + (q-1)/1000$). The simulation initiates with a relative fitness of $w = 1$, a functional construct and no fitness recovery from loss or accommodations, *i.e.* at list position $L_{i=1001,j=1,q=1}$. The range of q indices is determined by the original cost of the construct S_0 and the fitness bin resolution, such that the maximum value of q represents a full fitness recovery from the construct cost. For example, with a fitness resolution of 1000 then $q \in \{1, 2, ..., S_0 \times 1000 + 1\}$.

During the replication event, the list L is replicated into Lr where the fitness of each

subpopulation determines the number of offspring in Lr. A summation of from the indices i and q determines the total fitness (from Equation (4.8)):

$$Lr_{ijq} = L_{ijq} \times \left(\frac{i-1}{1000} + \frac{q-1}{1000} \right),$$

where subpopulations are rounded to whole integers as before (Equation (4.5)).

During the mutation event, three different mutations can occur: μ_n, μ_d and μ_a.

$$N_{mutated}(n) = B(Lr_{ijq}, \mu_n),$$

$$N_{mutated}(d) = B(Lr_{ijq}, \mu_d),$$

$$N_{mutated}(a) = B(Lr_{ijq}, \mu_a),$$

and the mutated offspring are immediately subtracted from the Lr list:

$$Lr_{ijq}(new) = Lr_{ijq}(old) - N_{mutated}.$$

The mutated offspring are inserted into a third list of mutated offspring (Lm) by distributing their fitness according to the type of mutation.

For normal environmental mutations $N_{mutated}(n)$:

$$Lm_{Xjq} \rightarrow Lm_{Xjq} + 1$$

where X is drawn from the DFE_n of the parent fitness:

$$X \sim DFE_0 \left(x, w_k = \frac{i-1}{1000} \right).$$

For destructive mutations $N_{mutated}(d)$:

$$Lm_{i2X} \rightarrow Lm_{i2X} + 1$$

where X is drawn from the DFE_d (which is independent of parent fitness or previous

construct loss, Equation (4.10)):

$$X \sim DFE_d(x).$$

These are destructive mutations, and therefore $j = 2$ (which also are irreversible, $j = 2$ can never reverse to $j = 1$).

For accommodating mutations $N_{mutated}(a)$:

$$Lm_{ijX} \rightarrow Lm_{ijX} + 1$$

where X is drawn from DFE_A, which depends on the q of the parent and the initial construct cost S_0:

$$X \sim DFE_A(x, q).$$

The DFE_A is given by Equation (4.12), where the initial construct cost S_0 is a constant variable for each model realisation and $s_a = \frac{q-1}{1000}$. Each mutation process is repeat for the number of mutated individuals of each mutation type. The whole replication process is repeated for each subpopulation index ijq in L, and then the three population lists are merged (same as Equation (4.6)),

$$L(\tau + 1) = L(\tau) + Lr(\tau) + Lm(\tau).$$

The dilution process is also identical to the model implementation without constructs. If $N_{total} > N_{max}$ each subpopulation is diluted according to the dilution factor ($N_{total} = \sum_{ijq} L_{ijq}$):

$$L_{ijq}(remaining) \sim B(L_{ijq}, p = 0.01).$$

Then the replication process resumes (see the model flowchart in Figure 4.3).

4.3 Engineered Construct Accommodation and Results

The model which was developed and described in Section 4.2.2 is used to simulate the occurrence of accommodating mutations. Ranges of (i) construct fitness costs, (ii) frequencies

of destructive mutations (related to their DNA sequence length) and (iii) population sizes (defined by culture volumes), are simulated with 100 model realisations for each parameter combination. The destructive mutations are assumed to have an even probability distribution between: (i) a zero fitness gain and (ii) a full fitness restoration from a complete deletion of the engineered function (Equation (4.10)). The time span simulated is here capped at 3,000 generations. If accommodating mutations are to be realised in practise as useful method then experimental time scales of weeks or months, rather than years are assumed to be more useful. The maximum simulated time limit of 3,000 generations is equivalent to 2-3 months for a fast growing microbe under good laboratory conditions.

Experimental Assumptions

A small number of experimental assumptions are made which relate to the physical conditions of organisms as well as the availability of measurable outputs.

It is assumed that the experimental process consists of continuous subculturing into a fixed sized volume container (similarly to the LTEE, which has a culture volume of 10mL) and the population size is diluted 100-fold each time it reaches its maximum limit for the container. The population size limit is given as an input argument to the model (an example of four different maximum population sizes N_{max} can be seen in Figure 4.11).

Each time a subculturing event occurs it is possible to plate part of the culture onto solid plates in order to obtain colonies from a subset of individuals within the population. The model aims to inform the probabilities for obtaining individual clones with different properties during the timecourse of the subculturing process.

However, not all biological properties and parameters are possible to observe or infer directly, or can only be observed retrospectively; *e.g.* the fitness level of an individual in a population.

Two different general conditions are assumed for the practical setup of evolving reduced construct costs through host accommodation. The two general conditions are specified as either of the two cases:

1. Cultures and plated colonies with functional constructs can be easily distinguished from populations which have lost the construct (*e.g.* due to a fluorescent marker or

other visual cue due to a functional construct).

2. Cultures and plated colonies with functional constructs are difficult to distinguish from populations that have lost the engineered construct (*e.g.* the output is a difficult to measure metabolite or a circuit with a range of different responses in time, in which case DNA sequencing of the construct might be the fastest available screening method).

In Case 1, after plating out colonies from the subculturing process, it is assumed that one can chose to pick only those colonies which still have a functional construct. This means that the sub-sample of colonies which need further investigation is narrowed. In Case 2, no distinction is made between colonies with and without constructs, and all growing colonies would need to be assayed. An example of Case 1 is seen in Figure 4.12, and an example of Case 2 is seen in Figure 4.11.

4.3.1 Simulation Results

Figure 4.11 shows the mean construct accommodation level of a construct if plating from *any* population sample, and subsequently picking *any* colony from the plated population. The process of plating any sample and picking any colony has the same effect as if the phenotype of the engineered construct is invisible (*i.e.* Case 2 from above). The mean level of accommodation is calculated by adding all accommodation levels of individuals in a population and dividing by the total population size, *i.e.* it refers to the mean of all sample means of accommodation levels. The accommodation value of an individual in the population is given by s_a/S_0, where s_a is the accommodation level of the individual and S_0 is the initial total cost of the engineered construct. An accommodation value equal to one represents a fully accommodated construct, *i.e.* the construct no longer constitutes any fitness cost (and $s_a = S_0$). A host with an impaired construct or without accommodation ($s_a = 0$) has a value of 0. In Figure 4.11 it can be seen that the ideal time of picking (in the figure shown as the maximum level of mean accommodation level), as well as the best container size for culturing, is very much dependent on the cost contributed by the engineered construct. Small fitness costs are more feasible to accommodate in larger

containers, whereas larger fitness costs are favoured by smaller population sizes.

If assuming that colonies with functional constructs can be distinguished easily and rapidly from non-functional constructs it allows for a more selective approach of picking (Case 1 from above). A rapid phenotype distinction can be enabled by visual cues from fluorescent proteins or a metabolic pathway with a colorimetric end-product [2, 66]. Figure 4.12 shows the mean accommodation level (similar to Figure 4.11), but while only picking colonies with a functional construct. The mean accommodation level is here (*i.e.* Case 1) calculated similarly to above (Case 2) by summing all accommodation levels by individuals in the population, but then divided by the total number of hosts without destructive mutations of the construct (instead of the total population, used in Case 2). In other words, those individuals (or colonies) with a destructed construct can be readily distinguished from the rest of the sample and need not be picked.

When selecting only colonies with a functional phenotype, then the largest simulated population size instead become the most favourable for obtaining accommodating phenotypes for construct costs of $S_0 = 0.05$ and $S_0 = 0.1$, but not for the higher selected construct cost of $S_0 = 0.2$. The intermediate population size of $N = 4 \times 10^6$ is still the best for most rapidly obtaining accommodated constructs with a cost of $S_0 = 0.2$. This population size corresponds to a volume of $100\mu L$, which is a normal culture size for 96-wellplates. In Figure 4.12B it is interesting to see that constructs of a very large size and high probability of destruction (red lines) are equally fast to accommodate as the small constructs (grey lines) for the two largest container volumes. Selection of only functional constructs has a larger benefit for constructs with high mutation rates. This is understandable, as a higher mutation rate means that a larger proportion of colonies lose their construct. Thus, those are the lost population colonies which are avoided.

Both Figures 4.11 and 4.12 show the mean level of accommodation. A different metric from the 'mean accommodation level' is to refer to the probability of picking a colony with a certain level of minimum accommodation. Figure 4.13 shows the probability of picking a colony with either a 10% or a 50% minimum accommodation level and for a construct with a low fitness costs of $S_0 = 0.05$. The accommodation level is given by s_a/S_0, in other words; how much of the initial total construct cost (S_0) has been ameliorated by accommodating

mutations (s_a). Figure 4.14 shows the same probability relationship as Figure 4.13, but for a high construct cost of $S_0 = 0.2$. Results are shown both for non-selective and selective picking. From the plots it is seen that the method of colony selection has less of an impact for success, *i.e.* in comparison to using a construct with lower fitness cost (Figures 4.13 and 4.14 — top subplots versus bottom subplots). Clearly, odds of success are helped by avoiding colonies which have already lost their construct (*i.e.* Case 1, shown in bottom subplots). However, a much greater effect is contributed from the population size (*i.e.* the size of the culturing vessel). The selection of functional colonies (Case 1) has a larger impact only for designs with a very high mutational frequency (see red lines in Figure 4.13).

Figures 4.15, 4.16, 4.17 and 4.18 show a grid of probabilities for selecting hosts with minimum accommodation levels between 2% $\rightarrow$ 50% during 3,000 generations. Each of the four plots shows a single population size subjected to a combination pair of three different μ_d and three S_0.

The general pattern seen in Figures 4.15 to 4.18 is that the probability of picking hosts with accommodating mutations, as well as the velocity for accommodating mutations to spread in the population, are increasing with increased population sizes. However, there are a couple of exceptions. Figure 4.19 shows an over-lay of three different population sizes for a construct with a cost of $S_0 = 0.2$ and mutational frequency of $\mu_d = 6 \times 10^{-7}\tau^{-1}$. After long time periods, the smallest population ($N = 4 \times 10^5$) is superior for picking colonies with marginal levels of accommodation. However, the second smallest population ($N = 4 \times 10^6$, *i.e.* 96-wellplate cultures size) is better at earlier time points, and at all time points it is superior for obtaining higher accommodation levels (30% construct fitness cost recovery or more; $\frac{s_a}{S_0} \geq 0.3$).

Figure 4.20 shows the full distribution of samples from Figure 4.11C. It shows that a general pattern, within a larger number of sample populations, is producing the values shown as means in the figures. This figure is shown to highlight that the mean values from the model realisations are not the product of a small number of outliers.

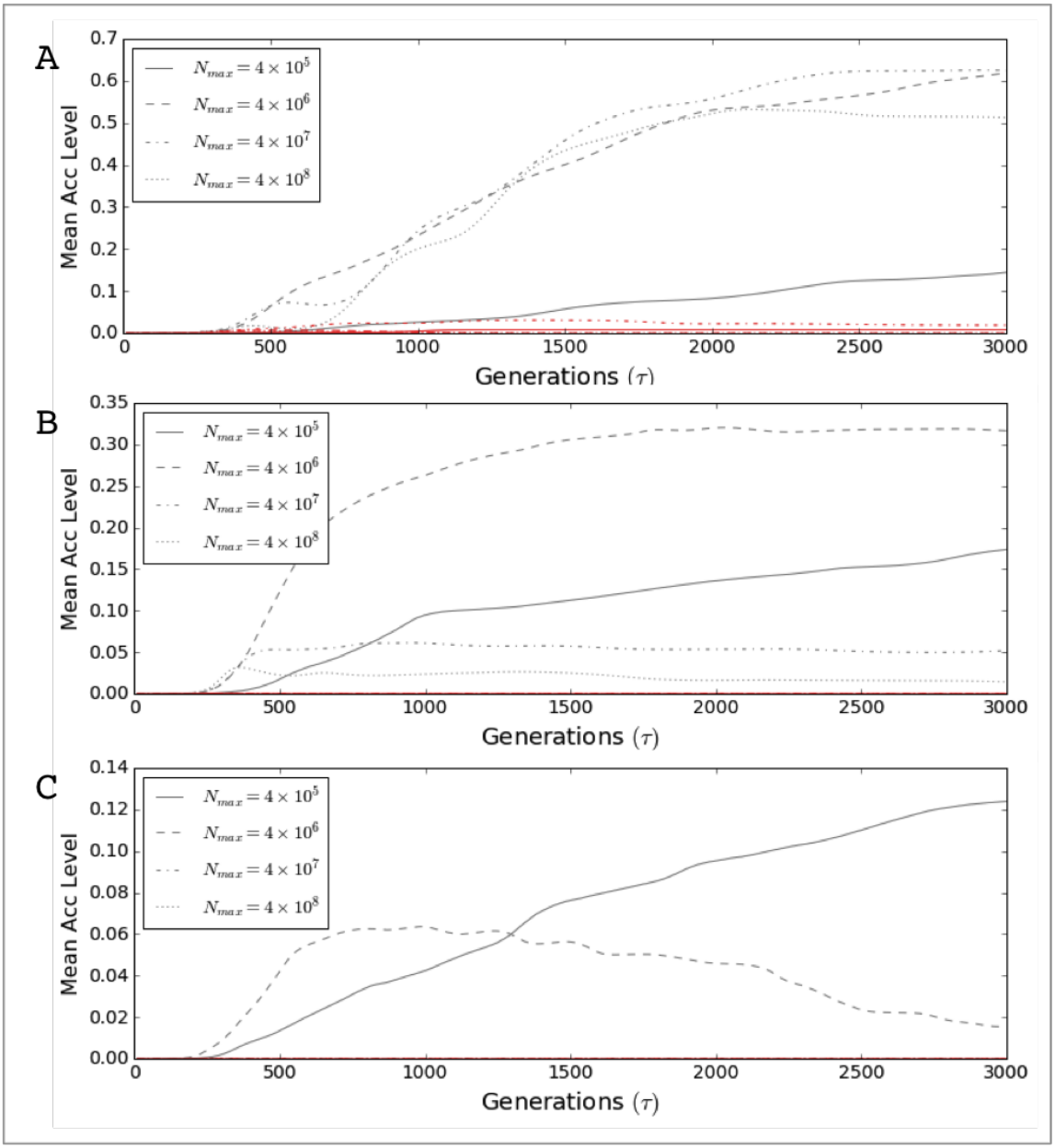


Figure 4.11: The level of accommodating mutations if picking any colonies. *I.e.* no distinction is made between phenotypes with functional or disrupted construct sequences. **Grey lines** show $\mu_d = 6 \times 10^{-7}\tau^{-1}$ and **red lines** show $\mu_d = 6 \times 10^{-5}\tau^{-1}$. $\mu_d = 6 \times 10^{-7}\tau^{-1}$ corresponds to a construct size of $\sim$ 5kbp. **A:** Low fitness cost, $S_0 = 0.05$. **B:** Medium fitness cost, $S_0 = 0.1$. **C:** High fitness cost, $S_0 = 0.2$. The Accommodation (Acc) Level on the y-axis is given by s_a/S_0.

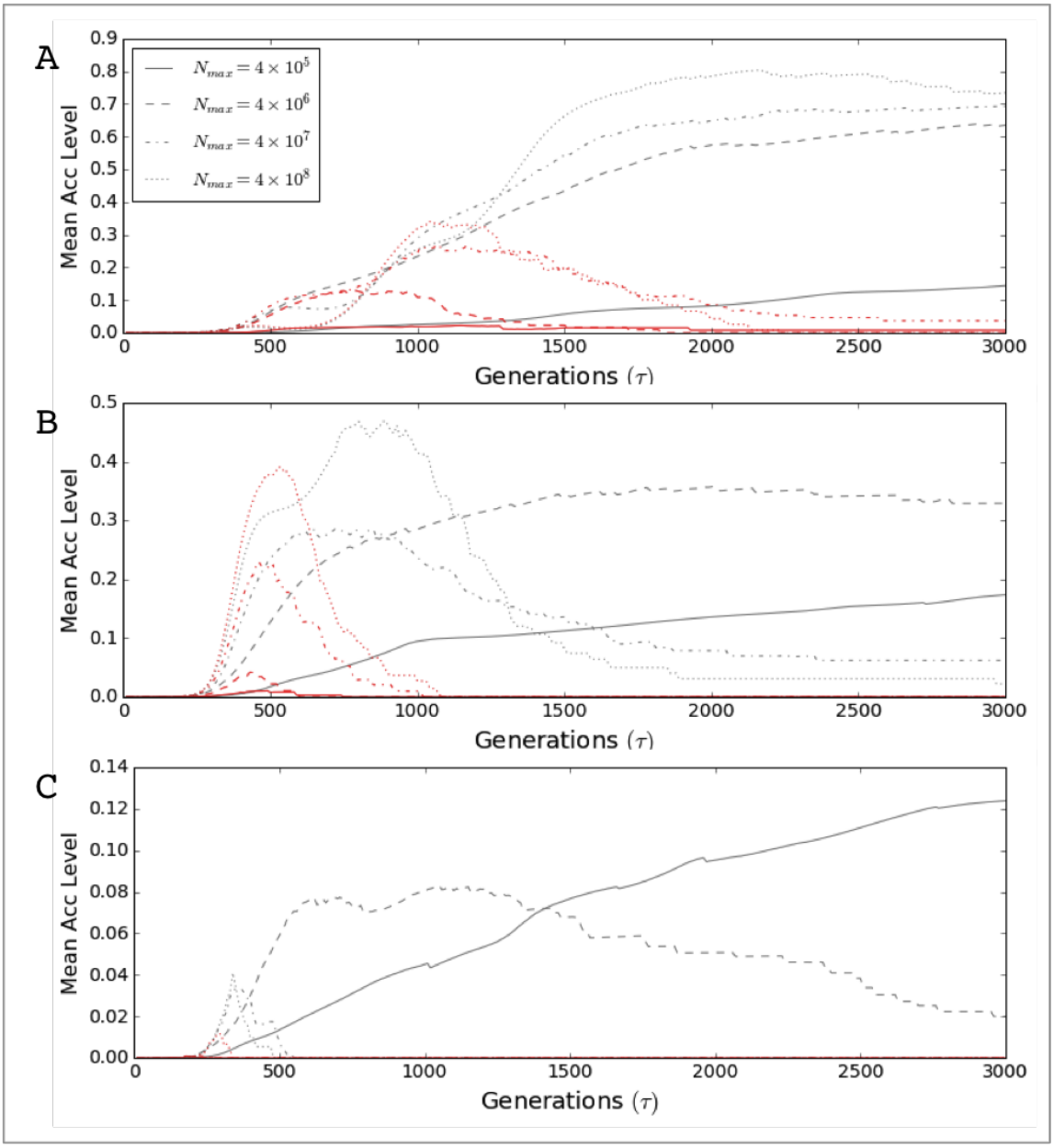


Figure 4.12: The level of accommodating mutations if selecting and picking colonies that have not been destructed. **Grey lines** show $\mu_d = 6 \times 10^{-7}\tau^{-1}$ and **red lines** show $\mu_d = 6 \times 10^{-5}\tau^{-1}$. $\mu_d = 6 \times 10^{-7}\tau^{-1}$ corresponds to a construct size of $\sim$ 5kbp. **A:** Low fitness cost, $S_0 = 0.05$. **B:** Medium fitness cost, $S_0 = 0.1$. **C:** High fitness cost, $S_0 = 0.2$. The Accommodation (Acc) Level on the y-axis is given by s_a/S_0.

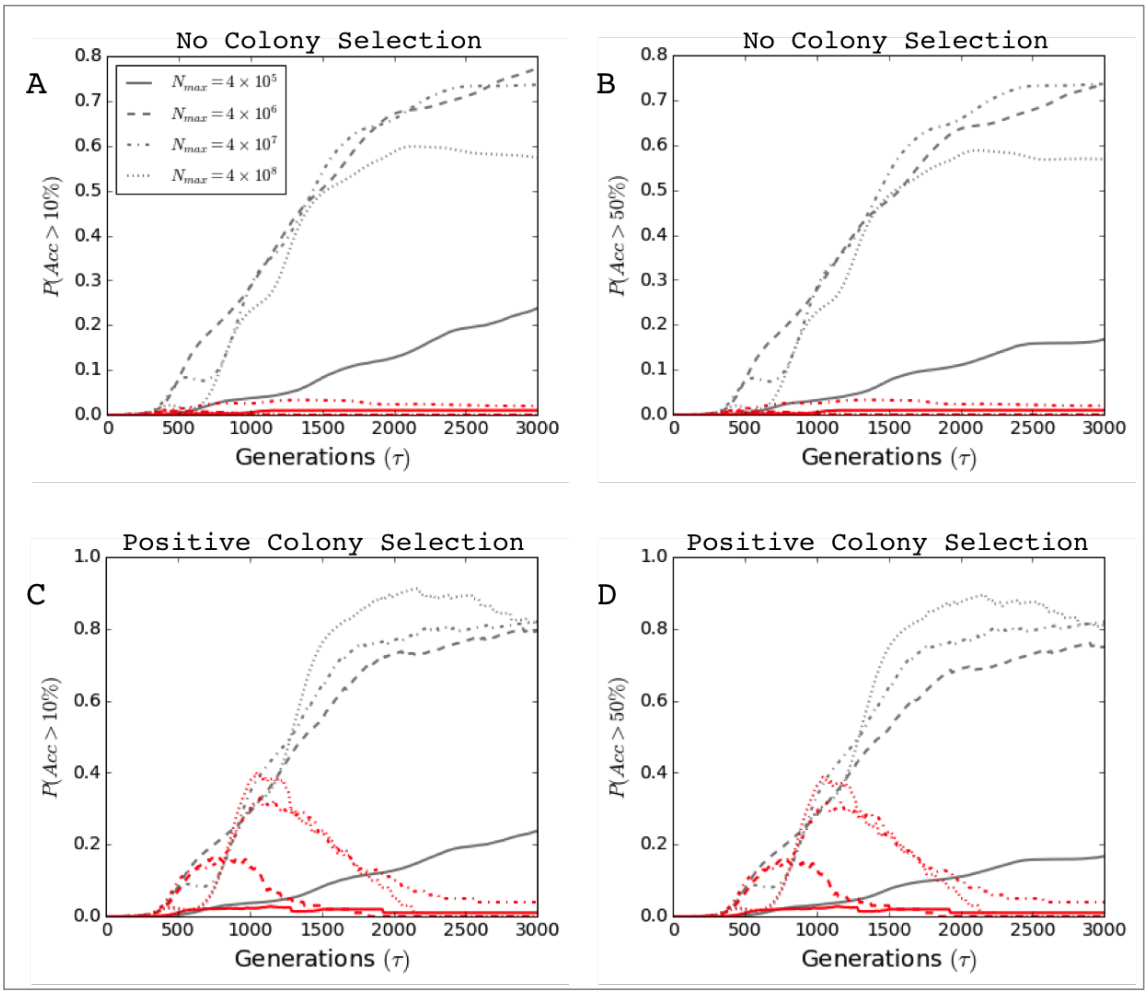


Figure 4.13: The probability of picking an accommodated colony for a construct with fitness cost of $S_0 = 0.05$. **Grey lines** show $\mu_d = 6 \times 10^{-7}\tau^{-1}$ and **red lines** show $\mu_d = 6 \times 10^{-5}\tau^{-1}$. $\mu_d = 6 \times 10^{-7}\tau^{-1}$ corresponds to a construct size of $\sim$ 5kbp. **A:** The probability of picking a colony with at least 10% accommodation level ($\frac{s_a}{S_0} \geq 0.1$), if picking any colony. **B:** The probability of picking a colony with at least 50% accommodation level ($\frac{s_a}{S_0} \geq 0.5$), if picking any colony. **C:** The probability of picking a colony with at least 10% accommodation level, if picking only colonies with a functional construct. **D:** The probability of picking a colony with at least 50% accommodation level, if picking only colonies with a functional construct.

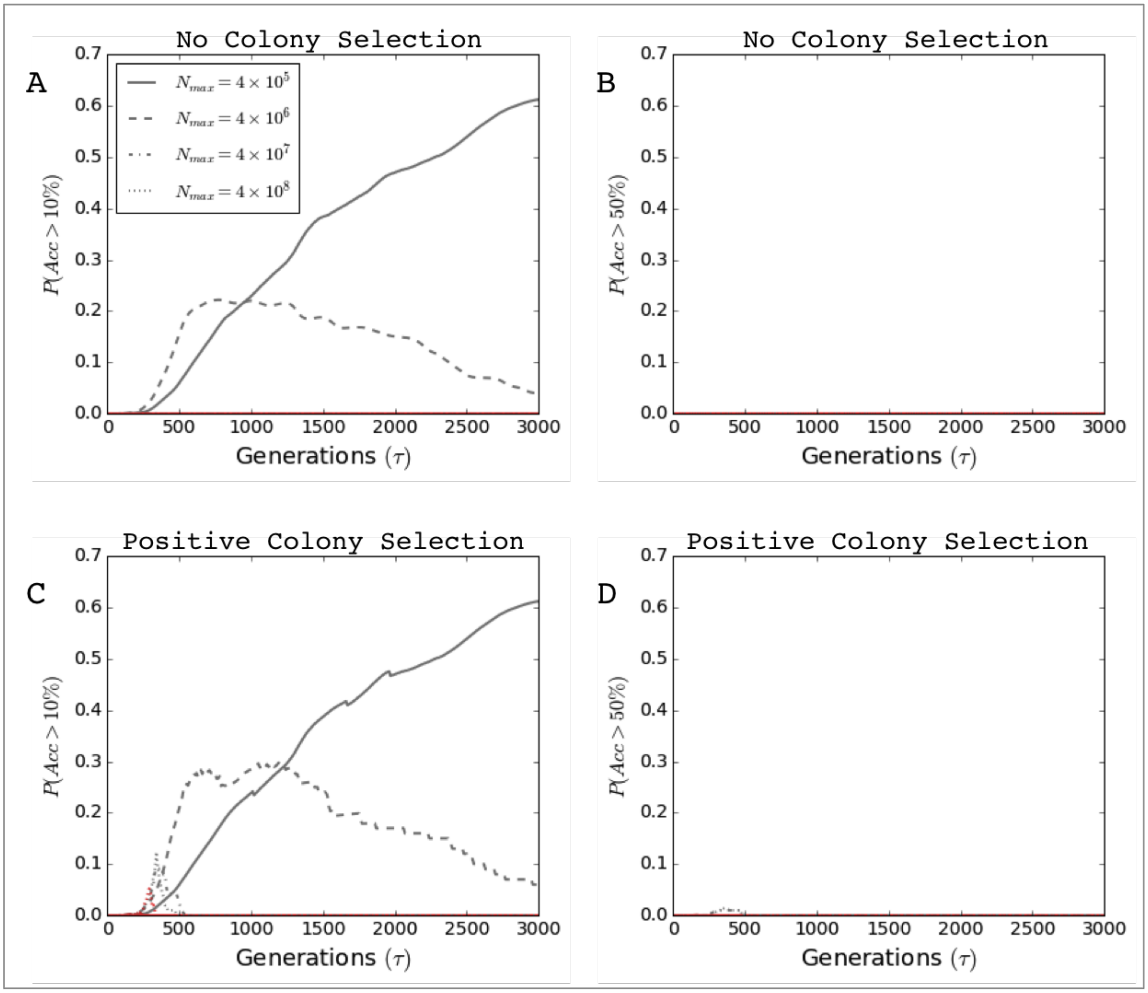


Figure 4.14: The probability of picking an accommodated colony for a construct with fitness cost of $S_0 = 0.2$. **Grey lines** show $\mu_d = 6 \times 10^{-7}\tau^{-1}$ and **red lines** show $\mu_d = 6 \times 10^{-5}\tau^{-1}$. $\mu_d = 6 \times 10^{-7}\tau^{-1}$ corresponds to a construct size of $\sim$ 5kbp. **A:** The probability of picking a colony with at least 10% accommodation level ($\frac{s_a}{S_0} \geq 0.1$), if picking any colony. **B:** The probability of picking a colony with at least 50% accommodation level ($\frac{s_a}{S_0} \geq 0.5$), if picking any colony. **C:** The probability of picking a colony with at least 10% accommodation level, if picking only colonies with a functional construct. **D:** The probability of picking a colony with at least 50% accommodation level, if picking only colonies with a functional construct.

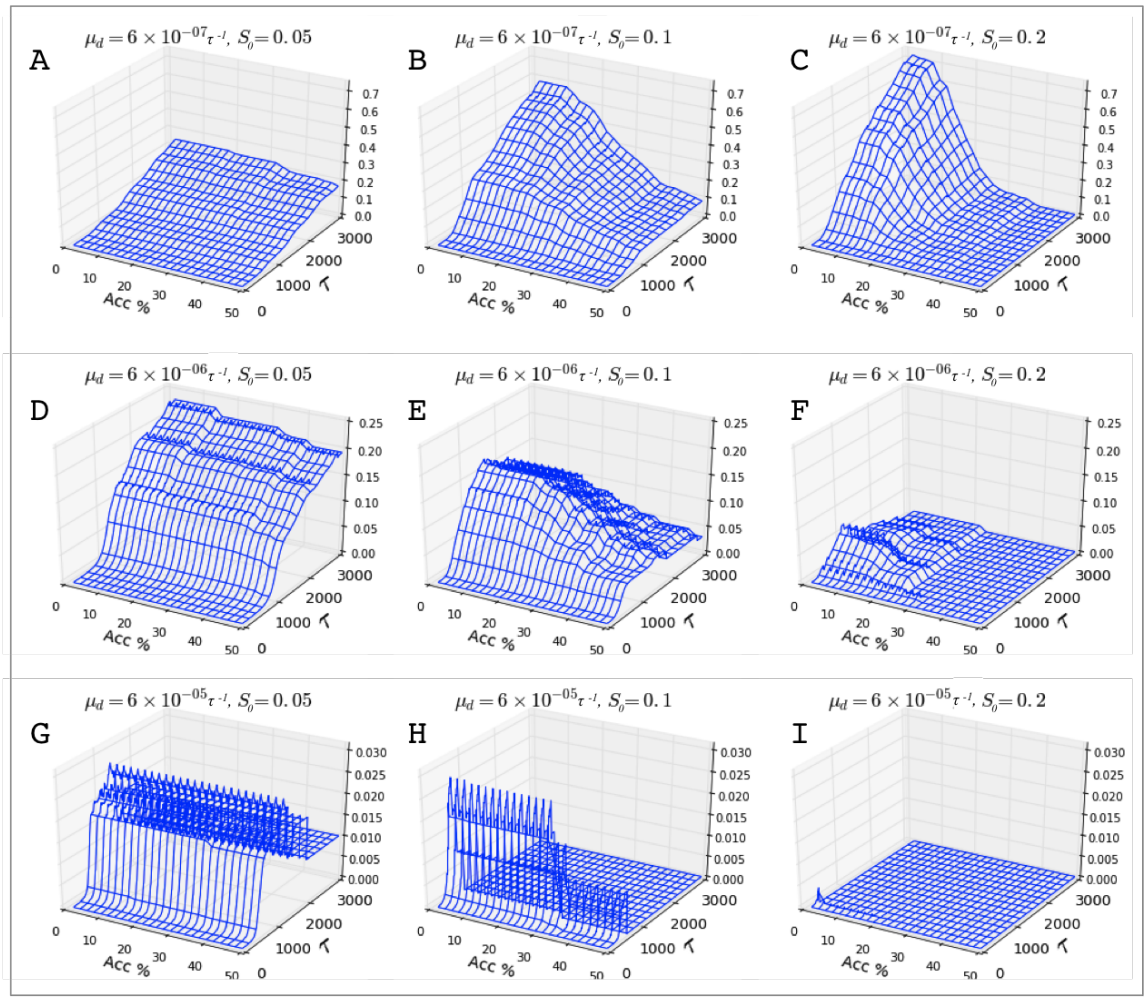


Figure 4.15: Probabilities for picking a colony with an accommodation level between 2% $\rightarrow$ 50% for population size of $N = 4 \times 10^5$. The population size is equivalent to an *E. coli* culture of $10\mu L$, *e.g.* a culture volume in a 384-wellplate. The vertical z-axis shows the probability of picking a colony with minimum level of accommodation (Acc % given on the x-axis), after τ number of generations (y-axis). The plots assume picking only colonies with functional construct (Case 1).

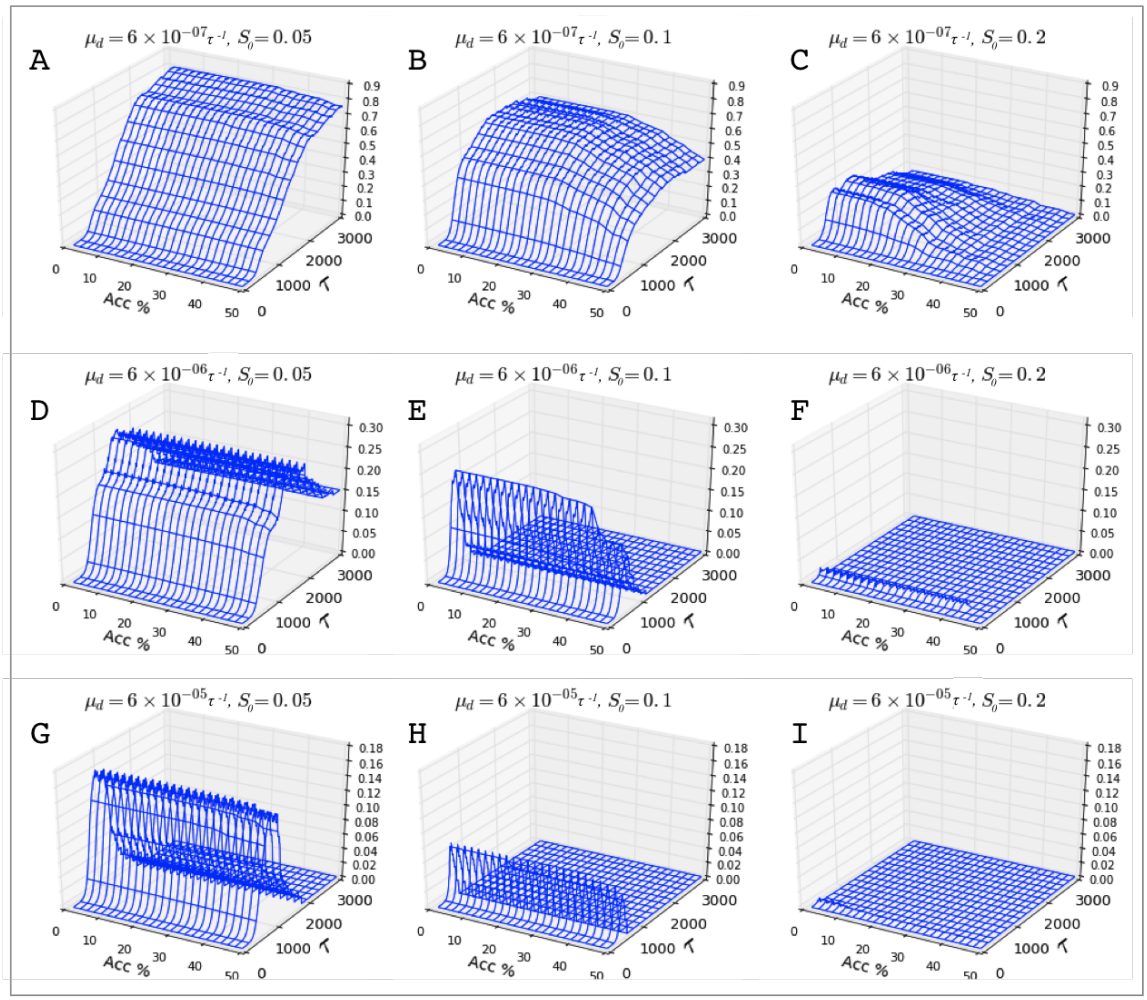


Figure 4.16: Probabilities for picking a colony with an accommodation level between 2% $\rightarrow$ 50% for population size of $N = 4 \times 10^6$. The population size is equivalent to an *E. coli* culture of $100\mu L$, *e.g.* a culture volume in a 96-wellplate. The vertical z-axis shows the probability of picking a colony with minimum level of accommodation (Acc % given on the x-axis), after τ number of generations (y-axis). The plots assume picking only colonies with functional construct (Case 1).

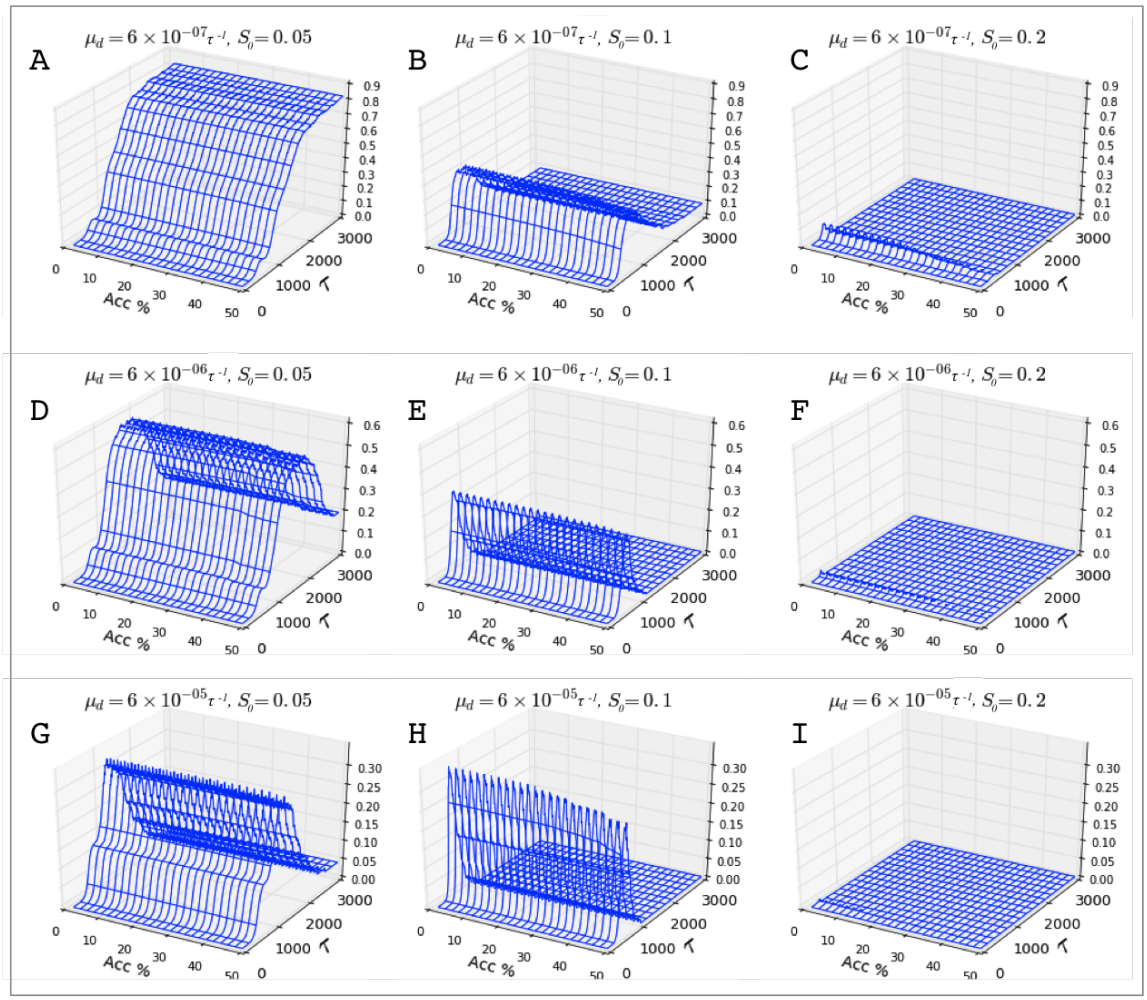


Figure 4.17: Probabilities for picking a colony with an accommodation level between 2% $\rightarrow$ 50% for population size of $N = 4 \times 10^7$. The population size is equivalent to an *E. coli* culture of $1mL$, *e.g.* the culture volume in a laboratory test-tube. The vertical z-axis shows the probability of picking a colony with minimum level of accommodation (Acc % given on the x-axis), after τ number of generations (y-axis). The plots assume picking only colonies with functional construct (Case 1).

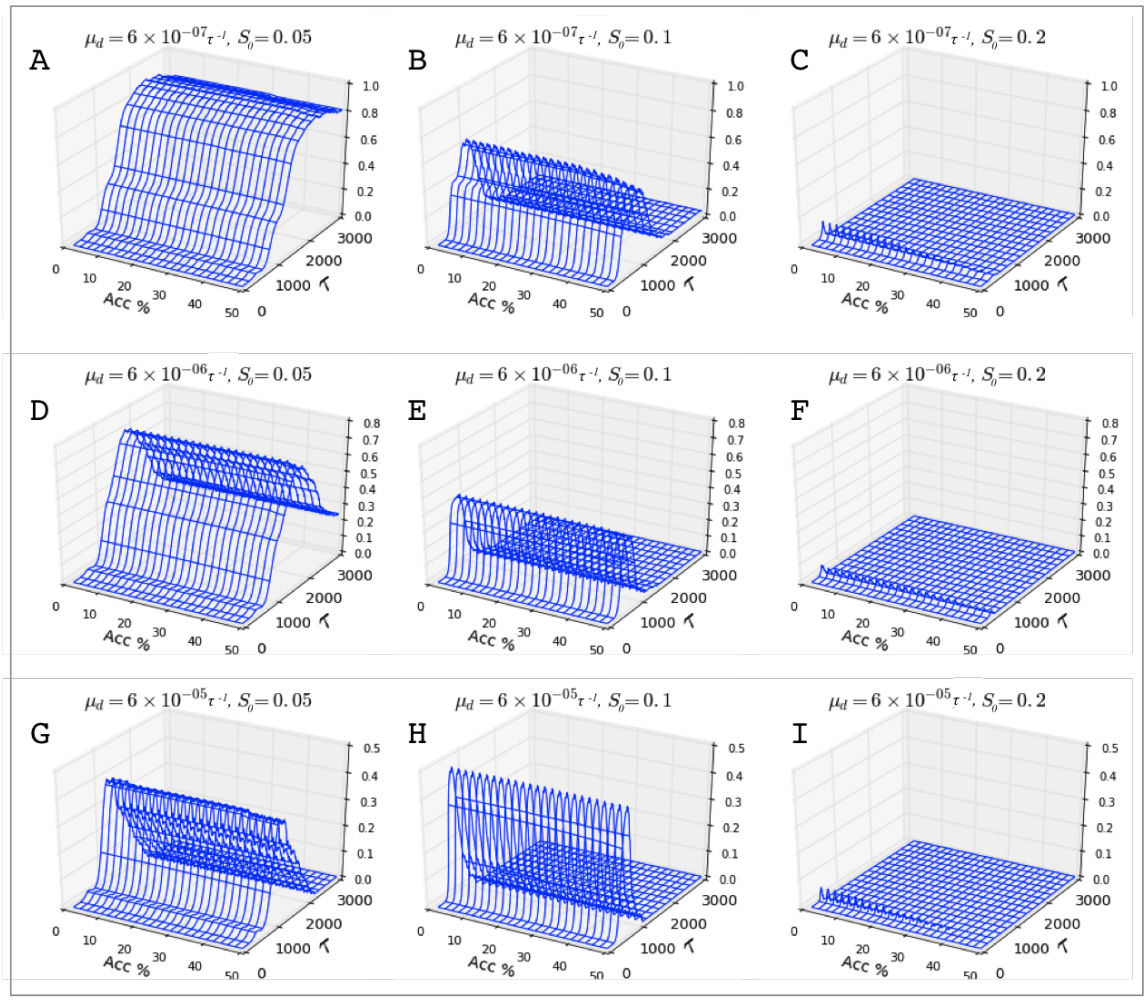


Figure 4.18: Probabilities for picking a colony with an accommodation level between 2% $\rightarrow$ 50% for population size of $N = 4 \times 10^8$. The population size is equivalent to an *E. coli* culture of $10mL$, *e.g.* the culture volume of a small shake-flask. The vertical z-axis shows the probability of picking a colony with minimum level of accommodation (Acc % given on the x-axis), after τ number of generations (y-axis). The plots assume picking only colonies with functional construct (Case 1).

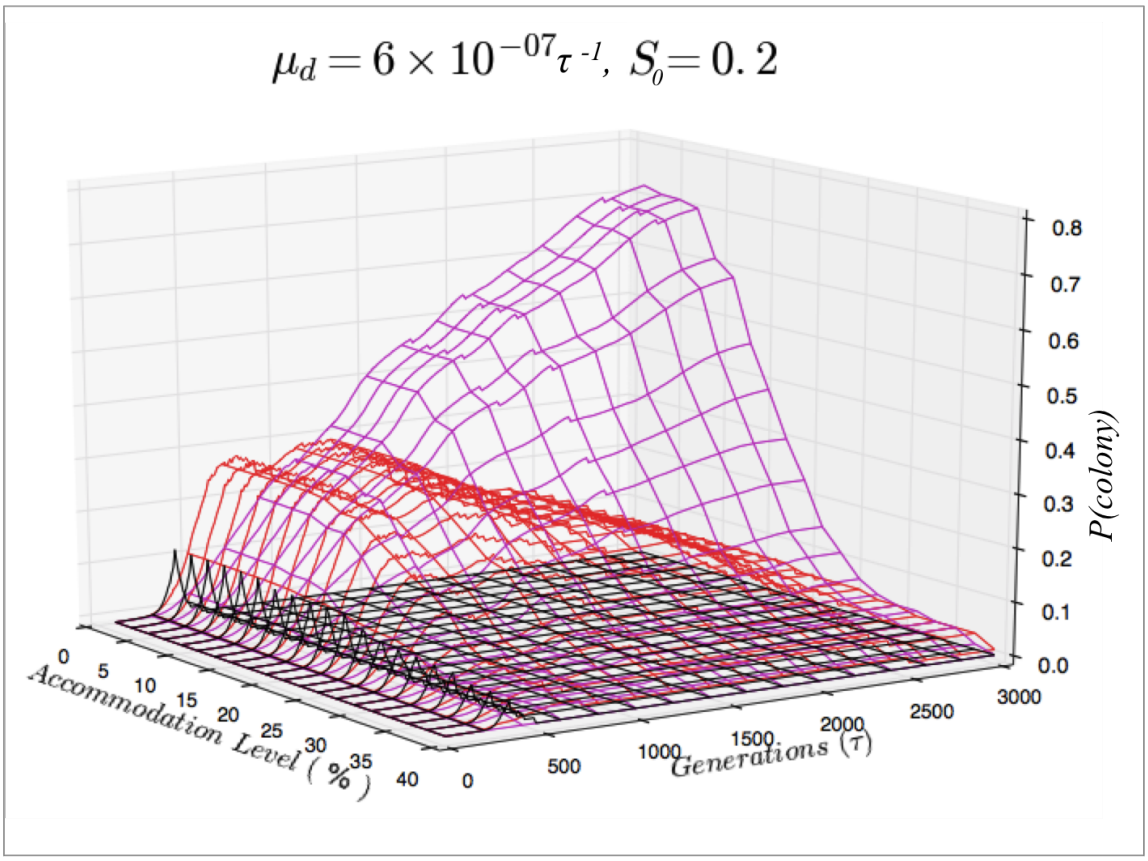


Figure 4.19: Comparison of three different population sizes with the same construct. Here assuming picking only colonies with functional construct (Case 1). **Lilac:** $N = 4 \times 10^5$. **Red:** $N = 4 \times 10^6$. **Black:** $N = 4 \times 10^8$. The vertical z-axis shows the probability of picking a colony with minimum level of accommodation (Acc % given on the x-axis), after τ number of generations (y-axis).

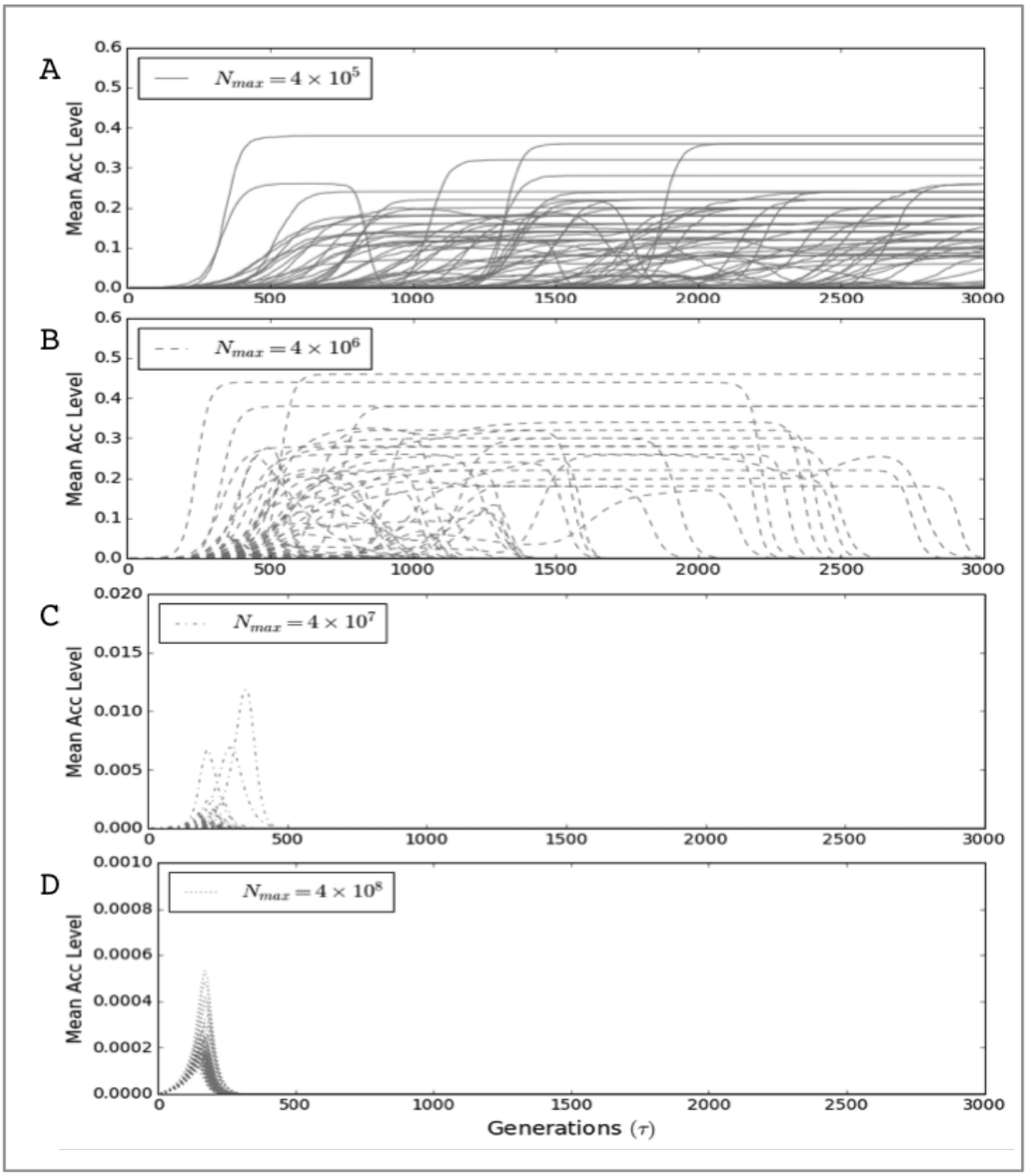


Figure 4.20: The full spread of all samples for a construct with a fitness cost of $S_0 = 0.2$ and destruction rate of $\mu_d = 6 \times 10^{-7}\tau^{-1}$. Each line in the subplots shows the mean accommodation level (s_a/S_0) for all the individuals of a single population. The mean value of all the populations in each subplot are seen in Figure 4.11C. There are 100 lines in each subplot. **A:** $N = 4 \times 10^5$. **B:** $N = 4 \times 10^6$. **C:** $N = 4 \times 10^7$. **D:** $N = 4 \times 10^8$.

4.4 Discussion

An initial purpose of the developed model is to inform experimental parameters which can increase chances of locating and identifying the first instances of accommodating mutations which are increasing the genetic stability of an engineered construct. The current model implementation assumes that internal changes in a host from an engineered construct are having a qualitatively similar effect on the DFE as the external changes from a new environment. It further assumes that the evolutionary benefit of the accommodating mutations are reduced with the same rate as the rate of decline for environmental adaptation during adaptation to a new environment. In order to parameterise the model with regards to the DFE impact from engineered construct a sequence of experimental steps have been identified. The simplest initial experiment would be designed as follows:

1. Insert a construct into a host.
2. Culture a large number of samples of the transformed host (*e.g.* a 96-wellplate).
3. In parallel, culture the same number of samples of the non-transformed host.
4. Record their rate of adaptation and compare the rate between hosts and controls.

If hosts carrying the construct have a higher rate of adaptation (fitness velocity) compared to empty controls, then that would give an initial indication of internal adaptation and construct accommodation. However, the next step to validate the presence of construct accommodation would require testing the rate at which constructs are lost in presumed accommodated hosts, and compare the stability with newly transformed hosts. The DNA sequence of the construct in the hosts with an improved half-life (or reduced rate of construct loss) also needs to be verified to show that the reduced S_0 is not an effect of mutations in the construct. Ideal for making this test would be to use a construct with a colorimetric marker. Easily detectable (or visible) markers would more easily prevent recording fitness increases which are due to construct loss, rather than host adaptation. The frequency and magnitude of accommodation will inform the 'accommodation' parameters in the model (DFE_A and μ_A).

4.4.1 Practical Use of the Model

To make practical use of the model it is necessary to consider the experimental constraints. These considerations are important for enabling both (i) the quantification biological assumptions and (ii) applied use of the model for enhancing construct stability.

For the model implemented it is assumed that (i) non-mutated genotypes, (ii) environmental and (iii) accommodated mutations can not be distinguished from each other visually. In other words, they can not be distinguished from each other in real time or 'online', in order to inform instant decisions (and branching logic) of the laboratory procedure. Environmental plus accommodated mutations can only be distinguished from non-mutated genotypes with competitive growth experiments and environmental mutations can only be distinguished from accommodated mutations via genetic stability tests of the construct; *e.g.* a *half-life* test. As is also the underlying purpose, accommodated mutations will, in a *half-life* test, exhibit a higher genetic stability compared to environmental mutations carrying the same level of evolved overall fitness. However, these practical constrains limit the possible experimental approaches that the model is aiming to inform and predict.

The model aims to find beneficial methods for culturing (and subculturing) a population such that the likelihood of obtaining construct accommodation is increased within a feasible time-frame and a minimal set of repeats and sample replicates.

Experimental parameters which could be predicted, fine-tuned and refined by the model are:

- The culture population size.
- Mutation rates. (*i.e.* benefit or disadvantage of employing hyper-mutators)
- Time points between spreading of individual colonies, picking and subculturing.

4.4.2 An Initial Proposal of Experimental Setup

In order to illustrate in detail the constraints of experimental model assumptions an example method is provided to describe a feasible form of a practical method, highlighting the underlying rationale of the method. This example method is intended to be neither

new nor revolutionary compared to typical evolutionary experiments. But instead, it aims to show a simple example for how the model could be used and to put it into a practical context.

The example method below describes a feasible path for providing both (i) an initial indication for the existence of accommodating mutations as well as (ii) data for the first interpretation of alterations of the DFE as an effect of engineered constructs introduced to organisms. In addition, samples produced during the experiment (if saved) can subsequently be utilised (in a second experiment) for investigating the occurrence as well as distribution of accommodating mutations able to reduce the fitness cost of the engineered construct.

A specific example for illustrating the general experimental method that could be applied in practice is as follows:

1. A large sample of identical engineered host populations providing a colorimetric output is grown in a 96-wellplate format. The same number of non-engineered samples is grown in an identical setup for providing comparison. See Figure 4.21.

2. The two sets of samples are diluted with the same time interval and dilution multiple in order to maintain them at the same age in terms of generations. (*E.g.* 256x dilution every 6 hours to provide 32 generations of growth every 24 hours.)

3. All 96-wellplates are grown inside plate readers in order to provide frequent and continuous optical density recordings for calculation of their specific growth rate. These measurements provide the data for interpreting the fitness velocity of each sample (*i.e.* growth rate changes over time).

4. When a large proportion of the engineered populations 'visually' has lost its colorimetric output then a subset of the populations remaining intact is plated onto solid plates for isolation of individual colonies. The same number of non-engineered reference populations is plated onto solid plates. See Figure 4.22.

5. The same number of colonies is picked from the plates as the original number of populations grown on the 96-well plates. From the engineered plates only colonies

that exhibit the colorimetric output are picked. From the non-engineered plates colonies are picked without any selection criterion.

6. This experiment continues for *e.g.* 2 months ($\sim$ 2,000 generations), during which time the evolutionary growth rate trajectory of the engineered population is compared with non-engineered growth rate trajectory. The growth-rate data can then elucidate the so far non-examined question of whether introducing an engineered construct changes the distribution of fitness effects of a population, importantly in the absence of mutations destructive to the construct by retreat to *wild type*.

7. Then subsequently, a second *half-life* experiment on the population samples obtained by the end of this experiment will be able to show the presence of accommodating mutations. (The presence of accommodating mutations would increase the half-life of the adapted hosts and with intact constructs, compared to samples prior to the subculturing procedure.)

This very first experimental step would neither prove nor disprove the model. However, it could provide an initial proof-of-concept for the existence of accommodating mutations with reduced construct costs in the context of deliberately designed and engineered biology. If populations with a functional construct phenotype by the end of evolutionary experiment have a longer half-life measure compared to the initially engineered hosts at the beginning of the experiment then this provides a first indication that accommodating mutations have occurred. Confirmation that the hosts still harbour a non-destructed construct can be determined by DNA sequencing of the specific construct sequence.

During the course of the growth experiment (and before the half-life tests) indications of accommodating mutations could also be observed. If accommodating mutations are spreading in the sample populations then the rate of construct loss should be higher in the beginning of the experiment compared to at the end of the experiment. For example, if 300 sample populations are used, if 240 of those samples (80%) have lost the engineered phenotype after 3 months of culturing, but only 36 of the remaining 60 samples (60%) have lost the phenotype during a subsequent 3 months of culturing. Already such data would be an initial (hopeful) indication which suggests construct accommodation during

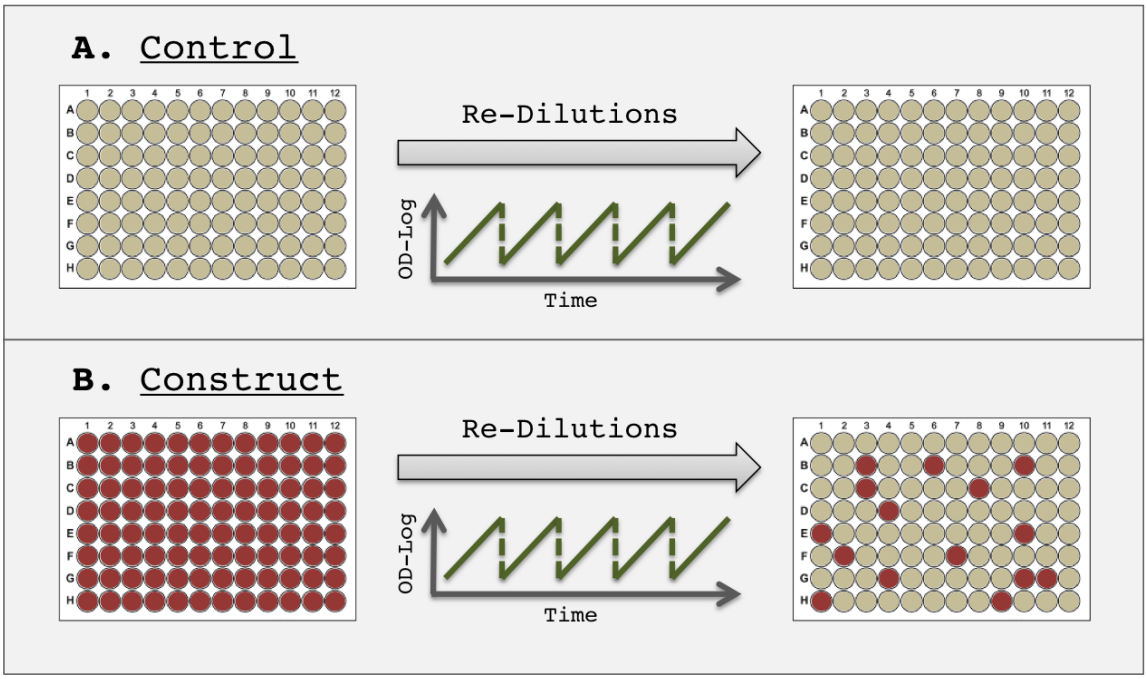


Figure 4.21: After multiple generations, from iterative dilutions of the culture to fresh liquid culture plates, the construct phenotype will be lost from many of the engineered populations. Subsequently, a subset of the populations maintaining the construct are plated for picking colonies to initiate fresh plates. The figure shows 96-platewells used for sample culturing, and each well contains an isolated population sample. Red wells indicate an engineered phenotype. Gray wells indicate populations without the engineered construct. **A**: Control populations without an engineered construct. **B**: Engineered populations. After multiple cycles of growth and re-dilutions a large proportion of the engineered populations will have lost the construct.

the course of the experiment.

For more useful data, in the form of refined model input parameters, as well as validation of the model as a practical tool for enriching accommodating mutation, further experiments and assays are required. Frozen population samples from this initial proof-of-concept experiment can be used for that purpose.

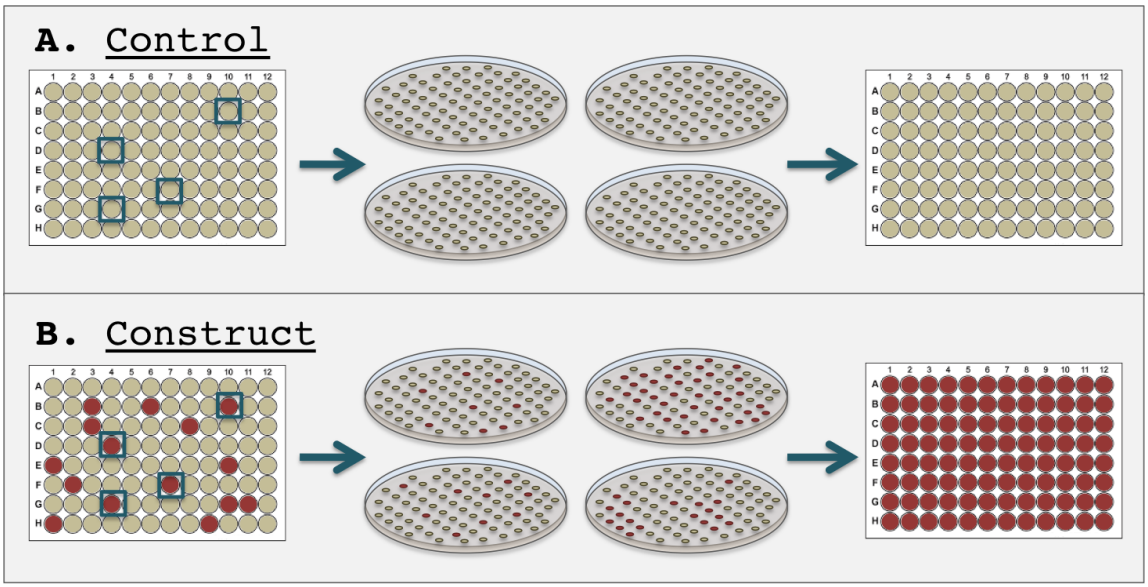


Figure 4.22: A subset of the populations maintaining the construct phenotype, post multiple dilutions (Figure 4.21), is plated onto solid plates (selected populations for plating are shown enclosed in squares). An equal number of non-engineered reference populations is also plated, either they can be randomly selected or else evenly distributed according to their growth rate. Colonies on solid plates maintaining the engineered phenotype are picked to restart fresh plates of engineered populations. An equal number of non-engineered reference colonies are picked, without selection criteria. **A**: Control populations without an engineered construct. **B**: Engineered populations.

4.4.3 Model Validation and Improved Parameterisation

Samples and data from the outlined initial proof-of-concept experiment for accommodating mutations can be used to refine the current model assumptions. Some of the assumptions are easier to inform; such as those directly relating to the construct. Meanwhile, other assumptions are a more difficult to elucidate; such as those relating to the distribution of accommodating effects.

The fitness cost of the construct (S_0) can be recorded from the growth rate of an engineered host compared to the growth rate of a wild type host. The mutational frequency of the construct (μ_d) can be initially estimated from the DNA sequence length of the engineered construct. However, it can also be necessary to adjust the mutational frequency predicted by the sequence length, *e.g.* if identifying in known mutational hotspots by computational sequence screens [92]. In addition, the destructive mutation rate of the construct (μ_d), together with its distribution of fitness recovery (DFE_d) can be simulated from the rate of construct loss in experimental half-life tests together with the determined cost S_0 (Section 0.2.2). Such simulations of DFE_d from half-life data would likely improve upon the current assumption of a uniform distribution of fitness recovery from mutations in the construct sequence. Also, testing more than one construct, and with different construct costs (S_0), could improve the model parameterisation further.

Two assumptions which are of key importance to the model are the DFE_A and μ_A parameters. Both input parameters are currently limited to best guesses. The presence of accommodating mutations in the initial proof-of-concept experiment would enable an improvement of both these parameters in the model (albeit in the connection with the exact construct and host used for the specific experiment). In comparison to the construct destruction parameters, more data is required to improve upon the current assumptions regarding DFE_A and μ_A. With the assumption that accommodating phenotypes were found and isolated within the initial proof-of-concept experiment (Subsection: An Initial Proposal of Experimental Setup) then frozen samples from that experiment can be analysed further. By conducting multiple half-life experiments from samples of different time points of the initial experiment then the rate μ_A as well as magnitude of accommodating

mutations can be informed. Hosts with increased half-lives would indicate the magnitude of accommodating mutations, and μ_A would be indicated by their frequency within the whole population sample. However, depending on the total number of positive samples with accommodating characteristics from the initial experiment it is possible that whole-genome sequencing of the different sample time points is required to enhance the data (by finding the exact number of mutations at each sample time point). The model would then be fitted against the new data in order to identify what input parameters of μ_A and DFE_A are consistent with the experimental output. Improved model parameters for DFE_A and μ_A would be screened against the new data similarly to how the environmental DFE (DFE_n) was fitted to the LTEE in Section 4.2.1 and Figure 4.8. In other words, potential shapes of DFE_A and μ_A would be simulated in the model to find groups of parameter pairs which are consistent with the data output of the experiment, and thus narrowing the assumptions for DFE_A and μ_A.

After fine-tuning and narrowing the initial assumptions of the model parameters from the first experiment (*i.e.* those stated above: μ_d, DFE_d and in particular μ_A and DFE_A) then different culturing scenarios can then be simulated to identify improved experimental conditions for increasing the probability of picking hosts with accommodating effects. The refined model parameters would be used to improve the parameters of the experimental setup to enrich the frequency of accommodating mutations. Experimental parameters which readily can be tuned are (i) the populations sizes and (ii) time points for picking individual samples for assay and sequencing. The basal host mutation rate μ could also be tuned or increased, albeit less easily. Designing and conducting new experiments according to the identified enhanced conditions from the model can then validate the usefulness of the model. The model is validated by demonstrating that the predicted experimental conditions by the model lead to a higher probability of selecting hosts with accommodating mutations, according to the identified sampling times. The improved model parameters from these first experiments can also be saved as sets of enhanced initial guesses for future similar experiments, as well as for applied purposes of accommodating engineered functions of value to practical biotechnology applications.

4.4.4 Tracking Time or Loss of Construct

Instead of pre-determining a set of generations for which one should spread and pick colonies with accommodating mutations, a feasible alternative could be to keep track of the proportion of destructed populations. When destructed constructs has reached a certain proportion of the total then colonies would be picked from populations that have not fixed construct destructed genotypes. The proportion of destructed constructs could either be recored as the ratio of sample populations where most individuals have lost their constructs, or more involved measurements of total proportion of loss within all population samples. Depending on the type of engineered phenotype made, the proportion of samples that have lost their constructs can be made easy to track in real time. For example, a colorimetric output can be recorded optically from visual cues, the simplest tracker output being the ratio of colonies with or without the phenotype.

4.4.5 Different Constructs

As stated previously, the model assumes a qualitatively similar distribution and change of distribution (due to adaptation) of internal 'accommodating' mutations and external environmental mutations with positive fitness effects. However, for simplicity and in the absence of other evidence, the current model also assumes the same magnitude of DFE changes from internal and external environment perturbations. However, depending on the type of construct introduced, various degrees of similarity are expected between external and internal host adaptation. Hosts are assumed to adapt easier to flux imbalances from new metabolic pathways, compared to adapting to raw 'inert' protein expression. Flux imbalances require internal 'rewiring' by the cell [91, 140, 185] whereas inert protein expression is assumed to be limited by growth-laws [162]. However, depending on the amino acid (and/or tRNA) usage by an inert inserted protein it is possible that its mere expression would also trigger to flux-imbalances.

A simple method to test the difference between a metabolic 're-wiring load' and a growth-law limited 'protein expression burden' would be to use a pathway which leads to a visual cue, visible on a colony level. For example the Lycopene pathway [2, 66], which

yields a clear colorimetric output. Meanwhile, fluorescent protein expression can be used for the host with a pure 'protein expression burden'. Then the rate of adaptation between the hosts can be compared. The assumption stated above predicts that load from the new metabolic pathway will lead to a faster rate of adaptation, compared to adaptation to the burden from a high protein expression level. A faster rate of adaptation (or fitness velocity) can indicate a large Υ in Equation (4.17), or otherwise a larger subarea of DFE_A verus DFE_n (Equations 4.11 and 4.9).

4.5 Conclusions and Contributions

Chapter 4 has developed an evolutionary model for identifying patterns and criteria for reducing fitness costs of engineered biology through *internal environment* adaptation of the host genome. The model is developed by assuming a similar process to how multi-resistant superbug pathogens arise, and aims to better formalise and reconstruct this process in order to utilise the process for obtaining longer lasting engineered constructs. The evolutionary behaviour of the model (without a construct) has been parameterised to the data from the LTEE experiment. Initial assumptions have been made for how the presence of engineered constructs alters the distribution of fitness effects. The assumptions state that an engineered constructs alters the DFE of a host in two ways: (i) introducing destructive mutations and (ii) expanding the area of mutations with positive fitness effects in the DFE, which are due to impacts on the host internal environment from the new construct.

When providing the model with the construct parameters of construct mutation rate (μ_d) and a cost of the construct (S_0), the model can be used to identify picking times and culture container volumes for enhancing the probability of finding accommodating mutations. The μ_d of the construct can be estimated from the DNA sequence length of the engineered construct. The S_0 can be recorded from the growth rate of an engineered host compared to the growth rate of a wild type host. However, the current model is limited by the less informed initial assumptions regarding DFE_A and μ_A. In the presented model these variables are assumed to be linked to how hosts adapt to new external environments. These assumption links include (i) the rate of decline in likelihood as well as magnitude for accommodating mutations as adaptation progresses, and (ii) how the magnitude of internal construct cost impacts the initial probability of accommodating mutations to occur.

The presented model aims to guide experimental methods and it provides suggested enhanced conditions for how to evolve host genome architecture with reduced burden for engineered biology. Future experiments can refine the input assumptions of the model, and experimental evidence is also required to validate the usefulness of the model in an applied workflow for construct cost reductions.

5. Chapter 5

5.1 Thesis Summary

This thesis has investigated burden and genetic stability in biology with a number of separate methods. Chapter 1 has developed computer coded and automated robotic protocols for repeatable and accessible testing of biological constructs by different user groups in different geographic locations. The methods developed in Chapter 1 were later applied in Chapter 2 and Chapter 3 for generating experimental data and providing model inputs. Chapter 1 also developed an online platform and user interface for the engineering of biology, the DSB platform. The DSB platform introduces new abstraction layers to Synthetic Biology research and is accessed remotely.

The purpose of the developed abstraction layer is to enable specialisation of skills which will drive the field forward rapidly. Today if one wants to write, test or edit a new software script, this is fast and easy to implement. There is no similar fast and easy way to run and iterate on DNA code for engineering biology. Compared to today, software code was also slow to develop when transported on physical devices (*e.g.* CD or floppy disk), similar to how DNA tubes are sent across the world today. The DSB platform is a proof-of-concept for how DNA code can be transmitted and collaborated on, in an electronic and accessible open format.

Chapter 2 developed models for informing burden parameters during the design phase of engineering biology. The chapter presents a model for informing the burden of an engineered construct depending on its translation efficiency. The model is derived from a previous 'growth law' but predicts a non-linear relationship between burden and the ex-

pression of engineered constructs, for constructs with reduced translational efficiency. This predicted non-linear effect of expression on burden is an addition to previously predicted and experimentally verified linear relationships between burden and engineered expression [162]. Reduced translational efficiency can be caused by 'overinitiation' [37] from designs with excessive RBS strengths, and it was the aim of the model that it can be used to identify such suboptimal designs directly from growth measurements. However, experimental conditions for model verification were difficult, partly because of difficulties when comparing growth cost from different experiments. An additional outcome from Chapter 2 was that the predicted translational initiation rates from the RBS calculator were not very accurate. An outcome which also impacted the ability for designing and comparing intended designs in the experimental part of the chapter.

Motivated by Chapter 2, and the need for better understanding of construct-host interactions during fluctuating growth, Chapter 3 developed a Dynamic Resource Allocation Model (DRAM) for interpreting burden. The model targets and predicts the transitionary cost of an engineered construct during changes in growth. Two new metrics for burden were proposed from fitting the model to data: (i) peak ribosome expression level during growth phase transition and (ii) delay in signal processing for environmental change. The 'rate of ribosome decline' was a further conceptual burden parameter also identified by the DRAM. According to the model fitted to data, constructs reduce the peak level of ribosome expression during transition to high growth rate. They also increase the signalling delay of responding to a more favourable environment. From using a larger number of population samples, which had a random and well distributed spread of growth initiation prior to subculturing, the model could also be fitted to predict the rate of decline for the ribosome fraction following peak expression levels. According to the model prediction, cells harbouring an engineered construct have a more rapid decline of the ribosome fraction when adjusting to less favourable growth conditions [132].

Chapter 4 took a different view from the deterministic burden models of the previous chapters, and developed a model for informing how engineered burden can be reduced via adaptive methods. The underlying model assumptions were developed by assuming a similar process to how multi-resistant 'superbugs' arise. The change in distribution of fitness

effects over time was fitted and parametrised with data from the LTEE experiment. Different engineered constructs and experimental conditions were simulated with the model. The model can be used to predict what experimental procedure is beneficial depending on parameters of the engineered construct. According to the simulated results, a larger culture volume and population size is in many instances better for increasing probability of picking accommodated hosts with increased level of reduced burden.

5.2 Summary of Key Limitations

Currently, the only automated data interpretation and analysis produced by the underlying software code are: (i) OD information, (ii) expression information, (iii) expression per OD, (iv) relative expression compared to negative and positive control samples, and (v) calculation of fit to a particular design function. The only design guidance provided by the platform is annotations and reference links provided by other users, as well as some minor automated guidance, such as where forks have been produced and where they are located. This is clearly a limiting selection of metrics, compared to what could be feasible to implement. The current features, as listed above, are to be seen as 'proofs of concept'. The expansion of the DSB platform would benefit from the focus on the incorporation of more software for data interpretation and design guidance. The current proof-of-concept is also limited to a single organism (*E. coli*), a single engineered output (protein expression) and a single ancillary output (growth).

The Dynamic Resource Allocation Model in Chapter 3 only covers the very initial growth phases in a new more favourable growth environment. In other words, only for growth conditions where growth is limited by the size of the ribosome fraction. The expression of proteins for nutrient import is expected to be growth limiting in the phase for growth deceleration. Expansion of the model into a full growth curve could enable the interpretation of the level of competition between the expression of engineered constructs and the expression of nutrient importing proteins (Φ_M). This correlation could then enable the time specific burden of engineered biology at any point of time for growth. However, it is also possible that signal delays in environmental responses would impact direct corre-

lations between fraction expression levels over longer time periods. Moreover, some of the assumptions of the model are not fully verified, for example that the rate of translation by ribosomes is constant over time.

The model in Chapter 4 informs experimental setups and conditions for adapting hosts to reduced fitness cost from particular constructs. However, some of the input parameters to the model, in particular the parameters regarding different classes of DFEs are poorly characterised and rely on best guesses and other assumptions. This is a clear limitation of the current presented model. The currently less known DFE parameters would be of broad general interest to investigate, aside from the aim of reducing engineered burden. However, the model developed in Chapter 4 will hopefully make it easier to interpret the data output from such future experiments (as outlined within the discussion of Chapter 4), in order to fully characterise the currently less quantified parameters. In addition, simulating different plausible outcomes from the experiments with the model before initiating the experiments could guide best practices of the experiments and identify potential pitfalls. Input parameters which are feasible to supply to the model include different sets of plausible DFEs, evolving properties of DFEs, number of samples, plating times, culture sizes *etc.*

5.3 Future Work

In future work the dynamic model developed in Chapter 3 can be used to investigate more experimental data sources with engineered constructs. For example, it would be interesting to observe how the different burden parameters in the dynamic model are impacted by the different combinations of construct parameters produced in Synthetic Biology experiments, such as plasmid copy number, RBS versions and promoter strengths. Further, it would be interesting to expand the Dynamic Resource Allocation Model into a full growth curve for informing the behaviour of a larger set of protein fractions.

It would also be useful to expand the experimental data in Chapter 2 with similar experiments. These could be conducted with the narrow but specific aim of finding answers to the model of translation efficiency in engineered biology, and how that impacts burden. Two direct alternatives to the available experimental data would be to: (i) de-codon

optimise the open reading frame of the expressed construct and (ii) grow the cells in a medium with poor nutritional quality. Recoding the open reading frame of the construct has the effect of increased stalling by ribosomes during translation. This will amplify the effect of overinitiation from high rates of translation initiation. Codon de-optimisation has previously been used successfully for this purpose during the investigation of burden [78, 37].

Growing the cells in a medium with less nutrients reduces the growth rate. For example by culturing in minimal medium, instead of rich medium. Slower growth means that more data points can be collected during the exponential growth phase. This also means that more accurate data can be produced with less noise. Minimal medium has previously been used for measuring long periods of growth in exponential phase [168].

A third but slightly more challenging alternative would be to conduct experiments with very small cell concentrations through many and frequent dilutions. Even if cell densities will be too low to measure reliably, the densities can be measured at the end of growth. By that time the relative differences between very small growth rate differences will be amplified by the number of dilutions. However, it is crucial that the fastest growing reference strains are never on the border of leaving the exponential phase during the iteration of dilutions. If they do leave the exponential phase briefly, then growth will slow down during the subsequent transition back to exponential growth, as seen by the Dynamic Resource Allocation Model in Chapter 3.

It would be interesting to include the model in Chapter 3 into the stochastic growth simulations with dilutions seen in Chapter 4. This could inform how burden is impacted from multiple cycles of feast and famine; a growth pattern which is generally seen in Synthetic Biology experiments and in many applied environments. The combination of the Dynamic Resource Allocation Model into the simulated stochastic populations might inform how the genetic stability can be improved from simple adjustments of media richness and dilution time periods.

Within Chapter 4 examples have been given for how to conduct experiments to select accommodating mutations within host organisms. When conducting the suggested experiments it would be of useful general interest to employ the strains from the LTEE ex-

periment as host organisms for the different engineered constructs. This would provide an additional level of comparison and verification of correct behaviour of the negative controls. At the same time as testing the effect on adaptation from the inclusions of engineered constructs, it would be very informative to test the rate of adaptation from the LTEE strains but at different population sizes. Ideally, one could take strains from a few different time points of evolution, *e.g.* every 10,000 generation, and then culture them under identical conditions but for a range of different population sizes. The rate of adaptation for the differently pre-adapted strains grown in different volumes of medium would provide truly valuable inputs for informing the DFE shape of the organism, as well as how the shape of the DFE has evolved over time.

For the future improvement of the DSB platform, the key next additions should preferably not be an expansion in the number of protocols it runs. Instead, more rapid improvements could be achieved by the addition of further software capabilities, which can aid in the (i) design and (ii) data analysis. Initial additions to design guidance would benefit from including: (i) automatic annotation suggestions from parsing of external DNA databases, (ii) cross referencing of features from older uploaded sequences, (iii) drag-and-drop of discrete components from previous runs and (iv) integration of DNA sequence analysis tools such as sequence stability, terminator strength, RBS rate of initiation etc.

In addition to design, data analysis is another part of the engineering cycle which can be time consuming if done manually. The DSB platform drastically speeds up data analysis for small set of proof-of-principle metrics, as outlined above in the subsection on current limitations. The number of metrics can be expanded for providing further insight from the data as instantaneously as the currently implemented metrics. To begin with, a useful addition to the automated data analysis provided on the DSB platform would be the Dynamic Resource Allocation Model developed in Chapter 3. Not only does the model infer burden parameters from data, but it also provides an estimated expression level of engineered components over time. An additional automation which could then be added to this is the relative size of this engineered fraction compared to the positive control of pure GFP expression. This will then give the relative distribution of resources from the cells which are directed towards GFP expression instead of other circuit components. For

example, indicating that 30% of a circuit is consumed by the GFP marker expression, and the remaining 70% by other regulatory components. As a practical example, a circuit is designed with three components, all of which are intended to have the same expression level, with the GFP marker being one of them. If the GFP expression level is only indicated at a level of *e.g.* 5% of the aggregated burden, then that gives a clear signal towards expression and burden inefficiencies among the other two circuit components.

At a later stage, the Digital Synthetic Biology Platform could be enhanced by expanding it with more experimental protocols. The DSB Platform could also open up to external makers of tools for data interpretation and design methods. This would have the scope to improve the design experience for biological engineers using the platform while simultaneously providing tool builders with an environment where their design tools can be tested. In other words, tool builders and engineers could provide their development of analysis tools directly to the end users for viewing design outputs. Engineers could then iterate on each others' analysis algorithms in the same way as the platform today makes it possible to iterate on each others' biological designs.

Combining the predictive design models for burden in Chapter 2 together with the analysing burden models for data interpretation in Chapter 3 should hopefully create a strong starting point for engineering and evaluating designs with the least possible burden on their hosts. The evolutionary model from Chapter 4 can be used to interpret the effects of these enhancements during longer time periods, and also inform ideal conditions for adapting the remaining cost towards zero. It is the wish of the author that the combined methods of design, testing and adaptation as highlighted in this thesis will enable the future creation of biological systems with minimal impacts on their hosts, and ever longer lasting functional stability.

Bibliography

[1] Sang Kyun Ahn, Kapil Tahlan, Zhou Yu, and Justin Nodwell. Investigation of transcription repression and small-molecule responsiveness by tetr-like transcription factors using a heterologous escherichia coli-based assay. *Journal of bacteriology*, 189(18):6655–6664, 2007.

[2] Hal Alper, Yong-Su Jin, JF Moxley, and G Stephanopoulos. Identifying gene targets for the metabolic engineering of lycopene biosynthesis in escherichia coli. *Metabolic engineering*, 7(3):155–164, 2005.

[3] Evan Appleton, Curtis Madsen, Nicholas Roehner, and Douglas Densmore. Design automation in synthetic biology. *Cold Spring Harbor perspectives in biology*, page a023978, 2017.

[4] Adam P Arkin. A wise consistency: engineering biology for conformity, reliability, predictability. *Current opinion in chemical biology*, 17(6):893–901, 2013.

[5] Adam P Arkin and Daniel A Fletcher. Fast, cheap and somewhat in control. *Genome Biol*, 7(8):114, 2006.

[6] Kotakonda Arunasri, Mohammed Adil, Pathan Akbar Ali Khan, and Sisinthy Shivaji. Global gene expression analysis of long-term stationary phase effects in e. coli k12 mg1655. *PloS one*, 9(5):e96701, 2014.

[7] Mariette R Atkinson, Michael A Savageau, Jesse T Myers, and Alexander J Ninfa. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in escherichia coli. *Cell*, 113(5):597–607, 2003.

[8] Tove Atlung, Bjarke Bak Christensen, and Flemming G Hansen. Role of the rom protein in copy number control of plasmid pbr322 at different growth rates inescherichia colik-12. *Plasmid*, 41(2):110–119, 1999.

[9] Jean-Christophe Augustin, Laurent Rosso, and Vincent Carlier. A model describing the effect of temperature history on lag time for listeria monocytogenes. *International Journal of Food Microbiology*, 57(3):169–181, 2000.

[10] Simon Ausländer, David Ausländer, Marius Müller, Markus Wieland, and Martin Fussenegger. Programmable single-cell mammalian biocomputers. *Nature*, 487(7405):123–127, 2012.

[11] William Bacchus and Martin Fussenegger. Engineering of synthetic intercellular communication systems. *Metabolic engineering*, 16:33–41, 2013.

[12] Susan F Bailey and Thomas Bataillon. Can the experimental evolution programme help us elucidate the genetic basis of adaptation in nature? *Molecular ecology*, 2015.

[13] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016.

[14] Frederick K Balagaddé, Hao Song, Jun Ozaki, Cynthia H Collins, Matthew Barnet, Frances H Arnold, Stephen R Quake, and Lingchong You. A synthetic escherichia coli predator–prey ecosystem. *Molecular systems biology*, 4(1):187, 2008.

[15] József Baranyi. Stochastic modelling of bacterial lag phase. *International journal of food microbiology*, 73(2):203–206, 2002.

[16] József Baranyi and Carmen Pin. Estimating bacterial growth parameters by means of detection times. *Applied and environmental microbiology*, 65(2):732–736, 1999.

[17] József Baranyi and Terry A Roberts. Mathematics of predictive food microbiology. *International journal of food microbiology*, 26(2):199–218, 1995.

[18] Jeffrey E Barrick, Mark R Kauth, Christopher C Strelioff, and Richard E Lenski. Escherichia coli rpob mutants have increased evolvability in proportion to their fitness defects. *Molecular biology and evolution*, 27(6):1338–1347, 2010.

[19] Thomas Bataillon and Susan F Bailey. Effects of new mutations on fitness: insights from models and data. *Annals of the New York Academy of Sciences*, 1320(1):76–92, 2014.

[20] Maxwell Bates, Aaron J Berliner, Joe Lachoff, Paul R Jaschke, and Eli S Groban. Wet lab accelerator: a web-based application democratizing laboratory automation for synthetic biology. *ACS synthetic biology*, 6(1):167–171, 2016.

[21] F Baty, JP Flandrois, and ML Delignette-Muller. Modeling the lag time of listeria monocytogenes from viable count enumeration and optical density data. *Applied and environmental microbiology*, 68(12):5816–5825, 2002.

[22] Florent Baty and Marie-Laure Delignette-Muller. Estimating the bacterial lag time: which model, which precision? *International journal of food microbiology*, 91(3):261–277, 2004.

[23] Silke Besier, Albrecht Ludwig, Volker Brade, and Thomas A Wichelhaus. Compensatory adaptation to the loss of biological fitness associated with acquisition of fusidic acid resistance in staphylococcus aureus. *Antimicrobial agents and chemotherapy*, 49(4):1426–1431, 2005.

[24] Matthew S Bienick, Katherine W Young, Justin R Klesmith, Emily E Detwiler, Kyle J Tomek, and Timothy A Whitehead. The interrelationship between promoter strength, gene expression, and growth rate. *PLoS ONE*, 9, 2014.

[25] Francisco Bolivar, Raymond L Rodriguez, Patricia J Greene, Mary C Betlach, Herbert L Heyneker, Herbert W Boyer, Jorge H Crosa, and Stanley Falkow. Construction and characterization of new cloning vehicle. ii. a multipurpose cloning system. *Gene*, 2(2):95–113, 1977.

[26] Amin Espah Borujeni, Anirudh S Channarasappa, and Howard M Salis. Translation rate is controlled by coupled trade-offs between site accessibility, selective rna unfolding and sliding at upstream standby sites. *Nucleic acids research*, page gkt1139, 2013.

[27] Evert Bosdriesz, Douwe Molenaar, Bas Teusink, and Frank J Bruggeman. How fast-growing bacteria robustly tune their ribosome concentration to approximate growth-rate maximization. *FEBS Journal*, 282(10):2029–2044, 2015.

[28] RL Buchanan and Miriam L Cygnarowicz. A mathematical approach toward defining and calculating the duration of the lag phase. *Food Microbiology*, 7(3):237–240, 1990.

[29] RL Buchanan and LA Klawitter. Effect of temperature history on the growth of listeria monocytogenes scott a at refrigeration temperatures. *International journal of food microbiology*, 12(2-3):235–245, 1991.

[30] RL Buchanan and Myron Solberg. Interaction of sodium nitrate, oxygen and ph on growth of staphylococcus aureus. *Journal of Food Science*, 37(1):81–85, 1972.

[31] RL Buchanan, RC Whiting, and WC Damert. When is simple good enough: a comparison of the gompertz, baranyi, and three-phase linear models for fitting bacterial growth curves. *Food Microbiology*, 14(4):313–326, 1997.

[32] Barry Canton, Anna Labno, and Drew Endy. Refinement and standardization of synthetic biological parts and devices. *Nature biotechnology*, 26(7):787–793, 2008.

[33] Max Carbonell-Ballestero, Eva Garcia-Ramallo, Raúl Montañez, Carlos Rodriguez-Caso, and Javier Macía. Dealing with the genetic load in bacterial synthetic biology circuits: convergences with the ohm's law. *Nucleic acids research*, 44(1):496–507, 2015.

[34] Stefano Cardinale, Marcin Pawel Joachimiak, and Adam Paul Arkin. Effects of genetic variation on the e. coli host-circuit interface. *Cell reports*, 4(2):231–237, 2013.

[35] Javier Carrera, Guillermo Rodrigo, Vijai Singh, Boris Kirov, and Alfonso Jaramillo. Empirical model and in vivo characterization of the bacterial response to synthetic gene expression show that ribosome allocation limits growth rate. *Biotechnology journal*, 6(7):773–783, 2011.

[36] Arturo Casadevall, Lee M Ellis, Erika W Davies, Margaret McFall-Ngai, and Ferric C Fang. A framework for improving the quality of research in the biological sciences, 2016.

[37] Francesca Ceroni, Rhys Algar, Guy-Bart Stan, and Tom Ellis. Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nature methods*, 12(5):415–418, 2015.

[38] Gianni Cesareni, M Helmer-Citterich, and L Castagnoli. Control of cole1 plasmid replication by antisense rna. *Trends in Genetics*, 7(7):230–235, 1991.

[39] Gianni Cesareni, Mark A Muesing, and Barry Polisky. Control of cole1 dna replication: the rop gene product negatively affects transcription from the replication primer promoter. *Proceedings of the National Academy of Sciences*, 79(20):6313–6317, 1982.

[40] Annie C Chang and Stanley N Cohen. Construction and characterization of amplifiable multicopy dna cloning vehicles derived from the p15a cryptic miniplasmid. *Journal of bacteriology*, 134(3):1141–1156, 1978.

[41] Bor-Sen Chen and Yu-Chao Wang. *Synthetic Gene Network: Modeling, Analysis and Robust Design Methods*. CRC Press, 2014.

[42] Luis-Miguel Chevin. On measuring selection in experimental evolution. *Biology letters*, 7(2):210–213, 2011.

[43] Hsin-Hung Chou, Hsuan-Chao Chiu, Nigel F Delaney, Daniel Segrè, and Christopher J Marx. Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science*, 332(6034):1190–1192, 2011.

[44] L Clarke, J Adams, P Sutton, JW Bainbridge, E Birney, J Calvert, A Collis, R Kitney, P Freemont, P Mason, et al. A synthetic biology roadmap for the uk. *https://connect.innovateuk.org*, 2012.

[45] LJ Clarke and RI Kitney. Synthetic biology in the uk–an outline of plans and progress. *Synthetic and Systems Biotechnology*, 1(4):243–257, 2016.

[46] Ryan E Cobb, Ning Sun, and Huimin Zhao. Directed evolution as a powerful synthetic biology tool. *Methods*, 60(1):81–90, 2013.

[47] Douglas Crockford. The application/json media type for javascript object notation (json). 2006.

[48] Andrew Currin, Neil Swainston, Philip J Day, and Douglas B Kell. Synthetic biology for the directed evolution of protein biocatalysts: navigating sequence space intelligently. *Chemical Society Reviews*, 44(5):1172–1239, 2015.

[49] Xiongfeng Dai, Manlu Zhu, Mya Warren, Rohan Balakrishnan, Vadim Patsalo, Hiroyuki Okano, James R Williamson, Kurt Fredrick, Yi-Ping Wang, and Terence Hwa. Reduction of translating ribosomes enables escherichia coli to maintain elongation rates during slow growth. *Nature Microbiology*, 2:16231, 2016.

[50] Tal Danino, Octavio Mondragón-Palomino, Lev Tsimring, and Jeff Hasty. A synchronized quorum of genetic clocks. *Nature*, 463(7279):326–330, 2010.

[51] Joseph H Davis, Adam J Rubin, and Robert T Sauer. Design, construction and characterization of a set of insulated bacterial promoters. *Nucleic acids research*, page gkq810, 2010.

[52] Hidde De Jong, Delphine Ropers, Johannes Geiselmann, Delphine Ropers, Aline Métris, Aline Métris, Susan George, Delphine Ropers, Nils Giordano, Francis Mairet, et al. Resource reallocation in bacteria by reengineering the gene expression machinery. *Trends in Microbiology*, 2017.

[53] Karen De Roy, Massimo Marzorati, Pieter Van den Abbeele, Tom Van de Wiele, and Nico Boon. Synthetic microbial ecosystems: an exciting tool to understand and apply microbial communities. *Environmental microbiology*, 16(6):1472–1481, 2014.

[54] Erez Dekel and Uri Alon. Optimality and evolutionary tuning of the expression level of a protein. *Nature*, 436(7050):588–592, 2005.

[55] Gloria Del Solar, Rafael Giraldo, María Jesús Ruiz-Echevarría, Manuel Espinosa, and Ramón Díaz-Orejas. Replication and control of circular bacterial plasmids. *Microbiology and molecular biology reviews*, 62(2):434–464, 1998.

[56] Domitilla Del Vecchio, Alexander J Ninfa, and Eduardo D Sontag. Modular cell biology: retroactivity and insulation. *Molecular systems biology*, 4(1):161, 2008.

[57] ML Delignette-Muller. Relation between the generation time and the lag time of bacterial growth kinetics. *International journal of food microbiology*, 43(1):97–104, 1998.

[58] Hengjiang Dong, Lars Nilsson, and Charles G Kurland. Gratuitous overexpression of genes in escherichia coli leads to growth inhibition and ribosome destruction. *Journal of bacteriology*, 177(6):1497–1504, 1995.

[59] John W Drake. A constant rate of spontaneous mutation in dna-based microbes. *Proceedings of the National Academy of Sciences*, 88(16):7160–7164, 1991.

[60] Thomas Egli. Microbial growth and physiology: a call for better craftsmanship. *Frontiers in microbiology*, 6, 2015.

[61] Santiago F Elena and Richard E Lenski. Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. *Nature Reviews Genetics*, 4(6):457–469, 2003.

[62] Tom Ellis, Xiao Wang, and James J Collins. Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nature biotechnology*, 27(5):465–471, 2009.

[63] Michael B Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[64] Drew Endy. Foundations for engineering biology. *Nature*, 438(7067):449–453, 2005.

[65] Adam Eyre-Walker and Peter D Keightley. The distribution of fitness effects of new mutations. *Nature Reviews Genetics*, 8(8):610–618, 2007.

[66] William R Farmer and James C Liao. Improving lycopene production in escherichia coli by engineering metabolic control. *Nature biotechnology*, 18(5):533, 2000.

[67] Ronald Aylmer Fisher. *The genetical theory of natural selection: a complete variorum edition*. Oxford University Press, 1930.

[68] Itzhak Fishov, Arieh Zaritsky, and NB Grover. On microbial states of growth. *Molecular microbiology*, 15(5):789–794, 1995.

[69] Ari E Friedland, Timothy K Lu, Xiao Wang, David Shi, George Church, and James J Collins. Synthetic gene networks that count. *science*, 324(5931):1199–1202, 2009.

[70] Martin Fussenegger. Synthetic biology: Synchronized bacterial clocks. *Nature*, 463(7279):301–302, 2010.

[71] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–342, 2000.

[72] John H Gillespie. Molecular evolution over the mutational landscape. *Evolution*, pages 1116–1129, 1984.

[73] John H Gillespie. *Population genetics: a concise guide*. JHU Press, 2010.

[74] Nils Giordano, Francis Mairet, Jean-Luc Gouzé, Johannes Geiselmann, and Hidde De Jong. Dynamic optimisation of resource allocation in microorganisms. In *21st International Symposium on Mathematical Theory of Networks and Systems (MTNS 2014)*, 2014.

[75] Nils Giordano, Francis Mairet, Jean-Luc Gouzé, Johannes Geiselmann, and Hidde De Jong. Dynamical allocation of cellular resources as an optimal control problem: novel insights into microbial growth strategies. *PLoS Comput Biol*, 12(3):e1004802, 2016.

[76] Benjamin Gompertz. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philosophical transactions of the Royal Society of London*, 115:513–583, 1825.

[77] Benjamin H Good, Igor M Rouzine, Daniel J Balick, Oskar Hallatschek, and Michael M Desai. Distribution of fixed beneficial mutations and the rate of adaptation in asexual populations. *Proceedings of the National Academy of Sciences*, 109(13):4950–4955, 2012.

[78] Thomas E Gorochowski, Irem Avcilar-Kucukgoze, Roel AL Bovenberg, Johannes A Roubos, and Zoya Ignatova. A minimal model of ribosome allocation dynamics captures trade-offs in expression between endogenous and synthetic genes. *ACS synthetic biology*, 5(7):710–720, 2016.

[79] Tanja M Gruber and Carol A Gross. Multiple sigma subunits and the partitioning of bacterial transcription space. *Annual Reviews in Microbiology*, 57(1):441–466, 2003.

[80] Vishal Gupta, Jesús Irimia, Iván Pau, and Alfonso Rodríguez-Patón. Bioblocks: Programming protocols in biology made easier. *ACS synthetic biology*, 6(7):1230–1232, 2017.

[81] Andras Gyorgy, José I Jiménez, John Yazbek, Hsin-Ho Huang, Hattie Chung, Ron Weiss, and Domitilla Del Vecchio. Isocost lines describe the cellular economy of genetic circuits. *Biophysical journal*, 109(3):639–646, 2015.

[82] Daniel L Halligan and Peter D Keightley. Spontaneous mutation accumulation studies in evolutionary genetics. *Annual Review of Ecology, Evolution, and Systematics*, 40:151–172, 2009.

[83] Matthew Hegreness, Noam Shoresh, Daniel Hartl, and Roy Kishony. An equivalence principle for the incorporation of favorable mutations in asexual populations. *Science*, 311(5767):1615–1617, 2006.

[84] Karl Heilbron, Macarena Toll-Riera, Mila Kojadinovic, and R Craig MacLean. Fitness is strongly influenced by rare mutations of large effect in a microbial mutation accumulation experiment. *Genetics*, 197(3):981–990, 2014.

[85] Jorge Pérez Heredia, Barbora Trubenová, Dirk Sudholt, and Tiago Paixão. Selection limits to adaptive walks on correlated landscapes. *Genetics*, pages genetics–116, 2016.

[86] A Herman and G Wegrzyn. Regulation of replication of plasmid pbr322 in amino acid-starved escherichia coli strains. *Molecular and General Genetics MGG*, 243(4):374–378, 1994.

[87] Steven J Hersch, Sara Elgamal, Assaf Katz, Michael Ibba, and William Wiley Navarre. Translation initiation rate determines the impact of ribosome stalling on bacterial protein synthesis. *Journal of Biological Chemistry*, 289(41):28160–28171, 2014.

[88] Ryan T Hietpas, Claudia Bank, Jeffrey D Jensen, and Daniel NA Bolon. Shifting fitness landscapes in response to altered environments. *Evolution*, 67(12):3512–3522, 2013.

[89] Ryan T Hietpas, Jeffrey D Jensen, and Daniel NA Bolon. Experimental illumination of a fitness landscape. *Proceedings of the National Academy of Sciences*, 108(19):7896–7901, 2011.

[90] JA Hudson. Effect of pre-incubation temperature on the lag time of aeromonas hydrophila. *Letters in applied microbiology*, 16(5):274–276, 1993.

[91] Mark Isalan, Caroline Lemerle, Konstantinos Michalodimitrakis, Carsten Horn, Pedro Beltrao, Emanuele Raineri, Mireia Garriga-Canut, and Luis Serrano. Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 452(7189):840, 2008.

[92] Benjamin R Jack, Sean P Leonard, Dennis M Mishler, Brian A Renda, Dacia Leon, Gabriel A Suárez, and Jeffrey E Barrick. Predicting the genetic stability of engineered dna sequences with the efm calculator. *ACS synthetic biology*, 2015.

[93] Arvi Jõers and Tanel Tenson. Growth resumption from stationary phase reveals memory in escherichia coli cultures. *Scientific reports*, 6, 2016.

[94] Moshe Kafri, Eyal Metzl-Raz, Ghil Jona, and Naama Barkai. The cost of protein production. *Cell reports*, 14(1):22–31, 2016.

[95] Stuart Kauffman and Simon Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45, 1987.

[96] Jason R Kelly, Adam J Rubin, Joseph H Davis, Caroline M Ajo-Franklin, John Cumbers, Michael J Czar, Kim de Mora, Aaron L Glieberman, Dileep D Monie, and Drew Endy. Measuring the activity of biobrick promoters using an in vivo reference standard. *Journal of biological engineering*, 3(1):4, 2009.

[97] Christian Kemmer, Marc Gitzinger, Marie Daoud-El Baba, Valentin Djonov, Jörg Stelling, and Martin Fussenegger. Self-sufficient control of urate homeostasis in mice by a synthetic circuit. *Nature biotechnology*, 28(4):355–360, 2010.

[98] Ahmad S Khalil and James J Collins. Synthetic biology: applications come of age. *Nature Reviews Genetics*, 11(5):367–379, 2010.

[99] Amy I Kim, Pallavi Ghosh, Michelle A Aaron, Lori A Bibb, Shruti Jain, and Graham F Hatfull. Mycobacteriophage bxb1 integrates into the mycobacterium smegmatis groel1 gene. *Molecular microbiology*, 50(2):463–473, 2003.

[100] Lance Kizer, Douglas J Pitera, Brian F Pfleger, and Jay D Keasling. Application of functional genomics to pathway optimization for increased isoprenoid production. *Applied and environmental microbiology*, 74(10):3229–3241, 2008.

[101] Stefan Klumpp, Zhongge Zhang, and Terence Hwa. Growth rate-dependent global effects on gene expression in bacteria. *Cell*, 139(7):1366–1375, 2009.

[102] Arthur L Koch and Carol S Deppe. In vivo assay of protein synthesizing capacity of escherichia coli from slowly growing chemostat cultures. *Journal of molecular biology*, 55(3):549–562, 1971.

[103] Sriram Kosuri, Daniel B Goodman, Guillaume Cambray, Vivek K Mutalik, Yuan Gao, Adam P Arkin, Drew Endy, and George M Church. Composability of regulatory sequences controlling transcription and translation in escherichia coli. *Proceedings of the National Academy of Sciences*, 110(34):14024–14029, 2013.

[104] Beat P Kramer, Alessandro Usseglio Viretta, Marie Daoud-El Baba, Dominique Aubel, Wilfried Weber, and Martin Fussenegger. An engineered epigenetic transgene switch in mammalian cells. *Nature biotechnology*, 22(7):867–870, 2004.

[105] Roberta Kwok. Five hard truths for synthetic biology: can engineering approaches tame the complexity of living systems? roberta kwok explores five challenges for the field and how they might be resolved. *Nature*, 463(7279):288–291, 2010.

[106] Heewook Lee, Ellen Popodi, Haixu Tang, and Patricia L Foster. Rate and molecular spectrum of spontaneous mutations in the bacterium escherichia coli as determined by whole-genome sequencing. *Proceedings of the National Academy of Sciences*, 109(41):E2774–E2783, 2012.

[107] Richard E Lenski, Michael J Wiser, Noah Ribeck, Zachary D Blount, Joshua R Nahum, James Jeffrey Morris, Luis Zaman, Caroline B Turner, Brian D Wade, Rohan Maddamsetti, et al. Sustained fitness gains and variability in fitness trajectories in the long-term evolution experiment with escherichia coli. *bioRxiv*, page 027391, 2015.

[108] Joshua Lilly and Manel Camps. Mechanisms of theta plasmid replication. *Microbiology spectrum*, 3(1), 2015.

[109] S Lin-Chao and H Bremer. Effect of the bacterial growth rate on replication control of plasmid pbr322 in escherichia coli. *Molecular and General Genetics MGG*, 203(1):143–149, 1986.

[110] Sue Lin-Chao, Wen-Tsuan Chen, and Ten-Tsao Wong. High copy number of the puc plasmid results from a rom/rop-suppressible point mutation in rna ii. *Molecular microbiology*, 6(22):3385–3393, 1992.

[111] Qijun Liu, Jörg Schumacher, Xinyi Wan, Chunbo Lou, and Baojun Wang. Orthogonality and burdens of heterologous and gate gene circuits in e. coli. *ACS synthetic biology*, 7(2):553–564, 2018.

[112] Wesley Loftie-Eaton, Hirokazu Yano, Stephen Burleigh, Ryan S Simmons, Julie M Hughes, Linda M Rogers, Samuel S Hunter, Matthew L Settles, Larry J Forney, José M Ponciano, et al. Evolutionary paths that expand plasmid host-range: implications for spread of antibiotic resistance. *Molecular biology and evolution*, 33(4):885–897, 2016.

[113] Donald E Low, James D Kellner, and Gerard D Wright. Superbugs: How they evolve and minimize the cost of resistance. *Current infectious disease reports*, 1(5):464–469, 1999.

[114] Michael Lynch. Evolution of the mutation rate. *Trends in Genetics*, 26(8):345–352, 2010.

[115] Daniel Madar, Erez Dekel, Anat Bren, Anat Zimmer, Ziv Porat, and Uri Alon. Promoter activity dynamics in the lag phase of escherichia coli. *BMC systems biology*, 7(1):136, 2013.

[116] Rohan Maddamsetti, Richard E Lenski, and Jeffrey E Barrick. Adaptation, clonal interference, and frequency-dependent interactions in a long-term evolution experiment with escherichia coli. *Genetics*, pages genetics–115, 2015.

[117] Sophie Maisnier-Patin and Dan I Andersson. Adaptation to the deleterious effects of antimicrobial drug resistance mutations by compensatory evolution. *Research in microbiology*, 155(5):360–369, 2004.

[118] Sophie Maisnier-Patin, Otto G Berg, Lars Liljas, and Dan I Andersson. Compensatory adaptation to the deleterious effect of antibiotic resistance in salmonella typhimurium. *Molecular microbiology*, 46(2):355–366, 2002.

[119] Arijit Maitra and Ken A Dill. Bacterial growth laws reflect the evolutionary importance of energy efficiency. *Proceedings of the National Academy of Sciences*, 112(2):406–411, 2015.

[120] Vincent JJ Martin, Douglas J Pitera, Sydnor T Withers, Jack D Newman, and Jay D Keasling. Engineering a mevalonate pathway in escherichia coli for production of terpenoids. *Nature biotechnology*, 21(7):796–802, 2003.

[121] RC McKellar. A heterogeneous population model for the analysis of bacterial growth kinetics. *International Journal of Food Microbiology*, 36(2-3):179–186, 1997.

[122] TA McMeekin, J Olley, DA Ratkowsky, and T Ross. Predictive microbiology: towards the interface and beyond. *International journal of food microbiology*, 73(2):395–407, 2002.

[123] LA Mellefont, T Ross, TA McMeekin, and K Neumeyer. Induction and manipulation of bacterial lag times. In *International Conference on Predictive Modelling in Foods*, volume 1, pages 70–72, 2000.

[124] Gerd HG Moe-Behrens, Rene Davis, and Karmella A Haynes. Preparing synthetic biology for the world. *Frontiers in microbiology*, 4, 2013.

[125] Douwe Molenaar, Rogier van Berlo, Dick de Ridder, and Bas Teusink. Shifts in growth strategies reflect tradeoffs in cellular economics. *Molecular systems biology*, 5(1):323, 2009.

[126] Patricia Molina-Espeja, Javier Viña-Gonzalez, Bernardo J Gomez-Fernandez, Javier Martin-Diaz, Eva Garcia-Ruiz, and Miguel Alcalde. Beyond the outer limits of nature by directed evolution. *Biotechnology advances*, 34(5):754–767, 2016.

[127] Andrew Morris, James D Kellner, and Donald E Low. The superbugs: evolution, dissemination and fitness. *Current opinion in microbiology*, 1(5):524–529, 1998.

[128] Shankar Mukherji and Alexander Van Oudenaarden. Synthetic biology: understanding biological design from synthetic circuits. *Nature Reviews Genetics*, 10(12):859–871, 2009.

[129] Vivek K Mutalik, Joao C Guimaraes, Guillaume Cambray, Colin Lam, Marc Juul Christoffersen, Quynh-Anh Mai, Andrew B Tran, Morgan Paull, Jay D Keasling, Adam P Arkin, et al. Precise and reliable gene expression via standard transcription and translation initiation elements. *Nature methods*, 10(4):354–360, 2013.

[130] Chris J Myers, Nathan A Barker, Kevin R Jones, Hiroyuki Kuwahara, Curtis Madsen, and Nam-Phuong D Nguyen. ibiosim: a tool for the analysis and design of genetic circuits. *Bioinformatics*, 25(21):2848–2849, 2009.

[131] Alec AK Nielsen, Bryan S Der, Jonghyeon Shin, Prashant Vaidyanathan, Vanya Paralanov, Elizabeth A Strychalski, David Ross, Douglas Densmore, and Christopher A Voigt. Genetic circuit design automation. *Science*, 352(6281):aac7341, 2016.

[132] Axel Nyström, Antonis Papachristodoulou, and Andrew Angel. A dynamic model of resource allocation in response to the presence of a synthetic construct. *ACS synthetic biology*, 2018.

[133] Carrie F Olson-Manning, Maggie R Wagner, and Thomas Mitchell-Olds. Adaptive evolution: evaluating empirical support for theoretical predictions. *Nature reviews. Genetics*, 13(12):867, 2012.

[134] H Allen Orr. The distribution of fitness effects among beneficial mutations. *Genetics*, 163(4):1519–1526, 2003.

[135] H Allen Orr. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics*, 6(2):119–127, 2005.

[136] H Allen Orr. Fitness and its role in evolutionary genetics. *Nature Reviews Genetics*, 10(8):531, 2009.

[137] Michael S Packer and David R Liu. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7):379–394, 2015.

[138] Michael Y Pavlov and Måns Ehrenberg. Optimal control of gene expression for fast proteome adaptation to environmental change. *Proceedings of the National Academy of Sciences*, 110(51):20527–20532, 2013.

[139] Micha Peleg and Maria G Corradini. Microbial growth curves: what the models tell us and what they cannot. *Critical reviews in food science and nutrition*, 51(10):917–945, 2011.

[140] J Christian Perez and Eduardo A Groisman. Evolution of transcriptional regulatory circuits in bacteria. *Cell*, 138(2):233–244, 2009.

[141] L Perfeito, A Sousa, T Bataillon, and I Gordo. Rates of fitness decline and rebound suggest pervasive epistasis. *Evolution*, 68(1):150–162, 2014.

[142] Stefano Perni, Peter W Andrew, and Gilbert Shama. Estimating the maximum growth rate from microbial growth curves: definition is everything. *Food microbiology*, 22(6):491–495, 2005.

[143] Carmen Pin and József Baranyi. Single-cell and population lag times as a function of cell age. *Applied and environmental microbiology*, 74(8):2534–2536, 2008.

[144] S John Pirt et al. *Principles of microbe and cell cultivation.* Blackwell Scientific Publications., 1975.

[145] Filip Poschet, Kristel Bernaerts, Annemie H Geeraerd, Nico Scheerlinck, BM Nicolaı, and JF Van Impe. Sensitivity analysis of microbial growth parameter distributions with respect to data quality and quantity by using monte carlo analysis. *Mathematics and Computers in Simulation*, 65(3):231–243, 2004.

[146] Katarzyna Potrykus, Helen Murphy, Nadège Philippe, and Michael Cashel. ppgpp is the major source of growth rate control in e. coli. *Environmental microbiology*, 13(3):563–575, 2011.

[147] Morgan N Price, Kelly M Wetmore, Adam M Deutschbauer, and Adam P Arkin. A comparison of the costs and benefits of bacterial gene expression. *PloS one*, 11(10):e0164314, 2016.

[148] Juan L Ramos, Manuel Martínez-Bueno, Antonio J Molina-Henares, Wilson Terán, Kazuya Watanabe, Xiaodong Zhang, María Trinidad Gallegos, Richard Brennan, and Raquel Tobes. The tetr family of transcriptional repressors. *Microbiology and Molecular Biology Reviews*, 69(2):326–356, 2005.

[149] Brian A Renda, Michael J Hammerling, and Jeffrey E Barrick. Engineering reduced evolutionary potential for synthetic biology. *Molecular BioSystems*, 10(7):1668–1678, 2014.

[150] FJ Richards. A flexible growth function for empirical use. *Journal of experimental Botany*, 10(2):290–301, 1959.

[151] WE Ricker. 11 growth rates and models. 1979.

[152] Keller Rinaudo, Leonidas Bleris, Rohan Maddamsetti, Sairam Subramanian, Ron Weiss, and Yaakov Benenson. A universal rnai-based logic evaluator that operates in mammalian cells. *Nature biotechnology*, 25(7):795–801, 2007.

[153] Tobin P Robinson, Maria J Ocio, Anu Kaloti, and Bernard M Mackey. The effect of the growth environment on the lag phase of listeria monocytogenes. *International Journal of Food Microbiology*, 44(1):83–92, 1998.

[154] Matthew D Rolfe, Christopher J Rice, Sacha Lucchini, Carmen Pin, Arthur Thompson, Andrew DS Cameron, Mark Alston, Michael F Stringer, Roy P Betts, József Baranyi, et al. Lag phase is a distinct growth phase that prepares bacteria for exponential growth and involves transient metal accumulation. *Journal of bacteriology*, 194(3):686–701, 2012.

[155] Howard M Salis, Ethan A Mirsky, and Christopher A Voigt. Automated design of synthetic ribosome binding sites to control protein expression. *Nature biotechnology*, 27(10):946–950, 2009.

[156] Joseph Sambrook, Edward F Fritsch, Tom Maniatis, et al. *Molecular cloning: a laboratory manual.* Number Ed. 2. Cold spring harbor laboratory press, 1989.

[157] A San Millan, R Peña-Miller, M Toll-Riera, ZV Halbert, AR McLean, BS Cooper, and RC MacLean. Positive selection and compensatory adaptation interact to stabilize non-transmissible plasmids. *Nature communications*, 5, 2014.

[158] Alvaro San Millan, Alfonso Santos-Lopez, Rafael Ortega-Huedo, Cristina Bernabe-Balas, Sean P Kennedy, and Bruno Gonzalez-Zorn. Small-plasmid-mediated antibiotic resistance is enhanced by increases in plasmid copy number and bacterial fitness. *Antimicrobial agents and chemotherapy*, 59(6):3335–3341, 2015.

[159] Carlota Diaz Sanchez-Bustamante, Jens M Kelm, Barbara Mitta, and Martin Fussenegger. Heterologous protein production capacity of mammalian cells cultivated as monolayers and microtissues. *Biotechnology and bioengineering*, 93(1):169–180, 2006.

[160] Jon Schnute. A versatile growth model with statistically stable parameters. *Canadian Journal of Fisheries and Aquatic Sciences*, 38(9):1128–1140, 1981.

[161] Stephanie J Schrag, Véronique Perrot, and Bruce R Levin. Adaptation to the fitness costs of antibiotic resistance in escherichia coli. *Proceedings of the Royal Society of London B: Biological Sciences*, 264(1386):1287–1291, 1997.

[162] Matthew Scott, Carl W Gunderson, Eduard M Mateescu, Zhongge Zhang, and Terence Hwa. Interdependence of cell growth and gene expression: origins and consequences. *Science*, 330(6007):1099–1102, 2010.

[163] Matthew Scott and Terence Hwa. Bacterial growth laws and their applications. *Current opinion in biotechnology*, 22(4):559–565, 2011.

[164] Matthew Scott, Stefan Klumpp, Eduard M Mateescu, and Terence Hwa. Emergence of robust growth laws from optimal regulation of ribosome synthesis. *Molecular systems biology*, 10(8):747, 2014.

[165] James Scott-Brown and Antonis Papachristodoulou. Visual representation of experimental protocols. *bioRxiv*, page 226852, 2017.

[166] Gerald Selzer, Tapan Som, Tateo Itoh, and Jun-ichi Tomizawa. The origin of replication of plasmid p15a and comparative studies on the nucleotide sequences around the origin of related plasmids. *Cell*, 32(1):119–129, 1983.

[167] Guennadi Sezonov, Daniele Joseleau-Petit, and Richard D'Ari. Escherichia coli physiology in luria-bertani broth. *Journal of bacteriology*, 189(23):8746–8749, 2007.

[168] Irit Shachrai, Alon Zaslaver, Uri Alon, and Erez Dekel. Cost of unneeded proteins in e. coli is reduced after several generations in exponential growth. *Molecular cell*, 38(5):758–767, 2010.

[169] Umender K Sharma and Dipankar Chatterji. Transcriptional switching in escherichia coli during stress and starvation by modulation of $\sigma 70$ activity. *FEMS microbiology reviews*, 34(5):646–657, 2010.

[170] Tatenda Shopera, Lian He, Tolutola Oyetunde, Yinjie J Tang, and Tae Seok Moon. Decoupling resource-coupled gene expression in living cells. *ACS synthetic biology*, 6(8):1596–1604, 2017.

[171] Sean C Sleight, Bryan A Bartley, Jane A Lieviant, and Herbert M Sauro. Designing and engineering evolutionary robust genetic circuits. *J Biol Eng*, 4(1):1–20, 2010.

[172] Hao Song, Ming-Zhu Ding, Xiao-Qiang Jia, Qian Ma, and Ying-Jin Yuan. Synthetic microbial consortia: from systematic analysis to construction and applications. *Chemical Society Reviews*, 43(20):6954–6981, 2014.

[173] Hans Peter Sørensen and Kim Kusk Mortensen. Advanced genetic strategies for recombinant protein expression in escherichia coli. *Journal of biotechnology*, 115(2):113–128, 2005.

[174] CJ Stannard, AP Williams, and PA Gibbs. Temperature/growth relationships for psychrotrophic food-spoilage bacteria. *Food Microbiology*, 2(2):115–122, 1985.

[175] IAM Swinnen, Kdens Bernaerts, Els JJ Dens, Annemie H Geeraerd, and JF Van Impe. Predictive modelling of the microbial lag phase: a review. *International journal of food microbiology*, 94(2):137–159, 2004.

[176] O Tenaillon. The utility of fisher's geometric model in evolutionary genetics. *Annual Review of Ecology, Evolution, and Systematics*, 45:179–201, 2014.

[177] Olivier Tenaillon, Jeffrey E Barrick, Noah Ribeck, Daniel E Deatherage, Jeffrey L Blanchard, Aurko Dasgupta, Gabriel C Wu, Sébastien Wielgoss, Stéphane Cruveiller, Claudine Médigue, et al. Tempo and mode of genome evolution in a 50,000-generation experiment. *Nature*, 536(7615):165, 2016.

[178] Marcel Tigges, Tatiana T Marquez-Lago, Jörg Stelling, and Martin Fussenegger. A tunable synthetic mammalian oscillator. *Nature*, 457(7227):309–312, 2009.

[179] Matthew F Traxler, Sean M Summers, Huyen-Tran Nguyen, Vineetha M Zacharia, G Aaron Hightower, Joel T Smith, and Tyrrell Conway. The global, ppgpp-mediated

stringent response to amino acid starvation in escherichia coli. *Molecular microbiology*, 68(5):1128–1148, 2008.

[180] Sandra Trindade, Lilia Perfeito, and Isabel Gordo. Rate and effects of spontaneous mutations that affect fitness in mutator escherichia coli. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 365(1544):1177–1186, 2010.

[181] Andrea J Twigg and David Sherratt. Trans-complementable copy-number mutants of plasmid cole1. *Nature*, 1980.

[182] SUZANNE JC van GERWEN and Marcel H Zwietering. Growth and inactivation models to be used in quantitative risk assessments. *Journal of Food Protection*, 61(11):1541–1549, 1998.

[183] Tom Vogwill and R Craig MacLean. The genetic basis of the fitness costs of antimicrobial resistance: a meta-analysis approach. *Evolutionary applications*, 8(3):284–295, 2015.

[184] SJ Walker and JE Jones. Protocols for data generation for predictive modeling. *Journal of Industrial Microbiology*, 12(3-5):273–276, 1993.

[185] Li Wang, Fang-Fang Wang, and Wei Qian. Evolutionary rewiring and reprogramming of bacterial transcription regulation. *Journal of genetics and genomics*, 38(7):279–288, 2011.

[186] Zhijun Wang, Guowei Le, Yonghui Shi, Grzegorz Wegrzyn, and Borys Wrobel. A model for regulation of cole1-like plasmid replication by uncharged trnas in amino acid-starved escherichia coli cells. *Plasmid*, 47(2):69–78, 2002.

[187] Zhijun Wang, Zhenghong Yuan, and Ulrich R Hengge. Processing of plasmid dna with cole1-like replication origin. *Plasmid*, 51(3):149–161, 2004.

[188] Zhijun Wang, Zhenghong Yuan, Li Xiang, Junjie Shao, and Grzegorz Wegrzyn. trna-dependent cleavage of the cole1 plasmid-encoded rna i. *Microbiology*, 152(12):3467–3476, 2006.

[189] Leandro Watanabe, Tramy Nguyen, Michael Zhang, Zach Zundel, Zhen Zhang, Curtis Madsen, Nicholas Roehner, and Chris J Myers. ibiosim 3: A tool for model-based genetic circuit design. *ACS Synthetic Biology*, 2018.

[190] Wilfried Weber and Martin Fussenegger. Inducible product gene expression technology tailored to bioprocess engineering. *Current opinion in biotechnology*, 18(5):399–410, 2007.

[191] Wilfried Weber and Martin Fussenegger. Emerging biomedical applications of synthetic biology. *Nature Reviews Genetics*, 13(1):21–35, 2012.

[192] Wilfried Weber, Ronald Schoenmakers, Bettina Keller, Marc Gitzinger, Thomas Grau, Marie Daoud-El Baba, Peter Sander, and Martin Fussenegger. A synthetic mammalian gene circuit reveals antituberculosis compounds. *Proceedings of the National Academy of Sciences*, 105(29):9994–9998, 2008.

[193] Wilfried Weber, Jörg Stelling, Markus Rimann, Bettina Keller, Marie Daoud-El Baba, Cornelia C Weber, Dominique Aubel, and Martin Fussenegger. A synthetic time-delay circuit in mammalian cells and mice. *Proceedings of the National Academy of Sciences*, 104(8):2643–2648, 2007.

[194] Daniel M Weinreich, Nigel F Delaney, Mark A DePristo, and Daniel L Hartl. Darwinian evolution can follow only very few mutational paths to fitter proteins. *science*, 312(5770):111–114, 2006.

[195] R Weiss. *Cellular Computation and Communications using Engineered Genetic Regulatory Networks. 2001.* PhD thesis, MIT, 2001.

[196] Andrea Y Weiße, Diego A Oyarzún, Vincent Danos, and Peter S Swain. Mechanistic links between cellular trade-offs, gene expression, and growth. *Proceedings of the National Academy of Sciences*, 112(9):E1038–E1047, 2015.

[197] Ellis Whitehead, Fabian Rudolf, Hans-Michael Kaltenbach, and Jörg Stelling. Automated planning enables complex protocols on liquid-handling robots. *ACS synthetic biology*, 7(3):922–932, 2018.

[198] Sébastien Wielgoss, Jeffrey E Barrick, Olivier Tenaillon, Stéphane Cruveiller, Béatrice Chane-Woon-Ming, Claudine Médigue, Richard E Lenski, and Dominique Schneider. Mutation rate inferred from synonymous substitutions in a long-term evolution experiment with escherichia coli. *G3: Genes, Genomes, Genetics*, 1(3):183–186, 2011.

[199] Sébastien Wielgoss, Jeffrey E Barrick, Olivier Tenaillon, Michael J Wiser, W James Dittmar, Stéphane Cruveiller, Béatrice Chane-Woon-Ming, Claudine Médigue, Richard E Lenski, and Dominique Schneider. Mutation rate dynamics in a bacterial population reflect tension between adaptation and genetic load. *Proceedings of the National Academy of Sciences*, 110(1):222–227, 2013.

[200] Michael J Wiser and Richard E Lenski. A comparison of methods to measure fitness in escherichia coli. *PLoS One*, 10(5):e0126210, 2015.

[201] Michael J Wiser, Noah Ribeck, and Richard E Lenski. Long-term dynamics of adaptation in asexual populations. *Science*, 342(6164):1364–1367, 2013.

[202] Sydnor T Withers, Shayin S Gottlieb, Bonny Lieu, Jack D Newman, and Jay D Keasling. Identification of isopentenol biosynthetic genes from bacillus subtilis by a screening method based on isoprenoid precursor toxicity. *Applied and environmental microbiology*, 73(19):6277–6283, 2007.

[203] Dominika M Wloch, Krzysztof Szafraniec, Rhona H Borts, and Ryszard Korona. Direct estimate of the mutation rate and the distribution of fitness effects in the yeast saccharomyces cerevisiae. *Genetics*, 159(2):441–452, 2001.

[204] Sewall Wright. Evolution in mendelian populations. *Genetics*, 16(2):97–159, 1931.

[205] Borys Wróbel and Grzegorz Wegrzyn. Differential amplification efficiency of pmb1 and p15a (cole1-type) replicons in escherichia coli stringent and relaxed strains starved for particular amino acids. *Microbiological research*, 152(3):251–255, 1997.

[206] Zhengyao Xu, Louise Thomas, Ben Davies, Ronald Chalmers, Maggie Smith, and William Brown. Accuracy and efficiency define bxb1 integrase as the best of fifteen

candidate serine recombinases for the integration of dna into the human genome. *BMC biotechnology*, 13(1):87, 2013.

[207] Song Yang, Sean C Sleight, and Herbert M Sauro. Rationally designed bidirectional promoter improves the evolutionary stability of synthetic genetic circuits. *Nucleic acids research*, 41(1):e33–e33, 2012.

[208] Celeste Yanisch-Perron, Jeffrey Vieira, and Joachim Messing. Improved m13 phage cloning vectors and host strains: nucleotide sequences of the m13mpl8 and puc19 vectors. *Gene*, 33(1):103–119, 1985.

[209] Lingchong You, Robert Sidney Cox, Ron Weiss, and Frances H Arnold. Programmed population control by cell–cell communication and regulated killing. *Nature*, 428(6985):868–871, 2004.

[210] MH Zwietering, JC De Wit, HGAM Cuppers, and K Van't Riet. Modeling of bacterial growth with shifts in temperature. *Applied and Environmental Microbiology*, 60(1):204–213, 1994.

[211] MH Zwietering, Il Jongenburger, FM Rombouts, and K Van't Riet. Modeling of the bacterial growth curve. *Applied and environmental microbiology*, 56(6):1875–1881, 1990.

[212] MH Zwietering, FM Rombouts, and K Riet. Comparison of definitions of the lag phase and the exponential phase in bacterial growth. *Journal of applied bacteriology*, 72(2):139–145, 1992.