PROTOTYPING IN *ESCHERICHIA COLI* OF STABLE AND EFFECTIVE METABOLIC PATHWAYS FOR *SYNECHOCYSTIS* SP. PCC 6803

George Taylor

Submitted for the Degree of Doctor of Philosophy

Imperial College London
Department of Life Sciences

October 2018

Declaration

I hereby certify that all of the material in this thesis is my own work, unless otherwise stated. All external contributions and any information derived from other sources has been appropriately acknowledged.

George Taylor

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work

Abstract

Engineering well-functioning metabolic pathways remains a challenge, largely because designing a metabolic pathway-encoding construct to express proteins at intended levels is limited and often leads to an imbalance of the pathway's protein concentration profile and therefore a poor performance. Whilst combinatorial and iterative approaches have proven effective for optimising the pathway balance in model organisms, implementing these approaches has typically proven arduous and time-consuming for many other organisms with phenotypes of industrial relevance. Such organisms are often considered as non-model because they are difficult to engineer, often lacking the tractability, fast-growth rates and high transformation efficiencies of model organisms. This project aims to implement rapid prototyping of metabolic pathway-encoding constructs in the tractable organism, *Escherichia coli*, before transferring the construct for expression in the non-model organism, the cyanobacterium *Synechocystis* sp. PCC 6803. *Synechocystis* is a photoautotroph that offers the potential to sustainably produce products from carbon dioxide. However, its slow growth rate, genetic instability and lack of genetic tools make it difficult to engineer. To test the rapid prototyping of metabolic pathway-encoding constructs, genetic parts and a design framework were required. For this, libraries of synthetic promoters and synthetic RBSs were developed, from which portable promoters and RBSs were identified that offered predictive protein expression between the two organisms. A novel multi-part, modular DNA assembly method was developed, termed Start-Stop Assembly, to serve as a framework. Using these tools, variations of the lycopene pathway were compared using current state-of-the-art and novel rapid prototyping designs. In this instance, rapid prototyping did not allow the performance of a pathway in *E. coli* to be used as a prediction of its performance in *Synechocystis*, and therefore it offered no benefit to the pathways design. Implementing a combinatorial approach did however improve the proportion of effective and stable pathways in *Synechocystis*, which led to lycopene being directly produced from carbon dioxide.

Acknowledgements

I would like to thank my supervisor Dr John Heap for giving me the opportunity to work alongside him on this project. The help and guidance you have offered me over the last four years has been invaluable and I have particularly enjoyed our time spent discussing and brainstorming various ideas. It is clear that you have put a lot of effort and commitment into helping me to develop as a researcher.

A special mention goes to my friends and colleagues, both past and present, at The Imperial College Centre for Synthetic Biology. Dr Ciarán Kelly, Dr Yonek Hleba, Lara Vital Sellés, Dr Rochelle Aw, Dr Linda Dekker and Dr Paweł Mordaka, you have always provided me with great support and advice, both in and out of the lab and your company has been a constant and welcome distraction.

To my family, I would like to thank you all for enduring four years' worth of scientific discussion. I am however slightly concerned that none of you really understand what I do! Your support has been fantastic, and I am so lucky to have such a loving family.

To my wife Lucy, I would like to thank you not only for all of the support that you have given me throughout this PhD, but for your continued patience and also your ability just to listen! I don't think I'd be where I am today without you.

Table of Contents

List of Figures 11
List of Tables 16
List of Code Snippets 16
List of Abbreviations 17
1. CHAPTER ONE. Introduction 21
1.1 Industrial biotechnology 21
1.2 Metabolic engineering 22
1.2.1 The history of metabolic engineering 23
1.2.2 Approaches and applications of metabolic engineering 23
1.2.2.1 Rational approaches to metabolic engineering 24
1.2.2.2 Random approaches to metabolic engineering 28
1.2.2.3 Semi-rational approaches to metabolic engineering 29
1.3 The design of metabolic pathway encoding constructs 30
1.3.1 Controlling gene expression 30
1.3.2 Traditional design of metabolic pathway encoding constructs, the pre-synthetic biology era 34
1.3.3 Synthetic Biology aided design of metabolic pathway encoding constructs 35
1.3.3.1 Emergence of Synthetic Biology 35
1.3.3.2 Enhancing the design of metabolic pathway-encoding constructs with Synthetic Biology 36
1.3.3.2.1 Expression control parts developed 37
1.3.3.2.2 Rational design of MPEC 39
1.3.3.2.3 Combinatorial pathway optimisation 42
1.3.4 DNA assembly 44
1.4 Microbial hosts used in Metabolic Engineering 46
1.4.1 Model organisms 46
1.4.1.1 *Escherichia coli* 46
1.4.1.2 *Saccharomyces cerevisiae* 47
1.4.1.3 The need for an alternative substrate 47
1.4.2 Advantages of using non-model organisms as an alternative host 49
1.4.3 Successes in non-model hosts for metabolic engineering 50
1.5 Photoautotrophic microorganisms as alternative microbial hosts 51
1.5.1.1 The history of Cyanobacteria 54
1.5.1.2 Cyanobacterial phylogeny 54
1.5.1.3 Cyanobacterial metabolism 55
1.5.1.3.1 Photosynthetic machinery 55
1.5.1.3.2 Carbon fixation and Calvin-Benson-Bassham (CBB) cycle 58
1.5.1.3.3 Post CBB cycle metabolism 59

1.5.1.4 Strain of cyanobacteria used in Metabolic Engineering and Synthetic Biology 60
1.5.1.5 Challenges associated with engineering cyanobacteria 62
1.6 Challenges of using non-model organisms in metabolic engineering 63
1.7 Overcoming the challenges 65
1.7.1 Rapid Prototyping 65
1.7.2 Portability 66
1.8 Aims and Objectives 68
2. CHAPTER TWO. Materials and Methods 70
2.1 Materials 70
2.2 Bioinformatics methods 70
2.3 Microbiological materials and methods 70
2.3.1 *E. coli* growth conditions 70
2.3.2 *Synechocystis* growth conditions 71
2.3.3 Preparation of electro-competent cells of *E. coli* 71
2.3.4 Transformation of *E. coli* by electroporation 71
2.3.5 Preparation of chemical-competent cells of *E. coli* 72
2.3.6 Transformation of *E. coli* by Heat-Shock 72
2.3.7 Transformation of *Synechocystis* 72
2.4 Molecular biological materials and methods 73
2.4.1 Extraction and purification of plasmid DNA 73
2.4.2 Spectrophotometric quantitation of DNA 73
2.4.3 Amplification of DNA using PCR 73
2.4.4 Amplification of DNA using Touchdown PCR 73
2.4.5 Amplification of DNA using SOE PCR 73
2.4.6 Restriction endonuclease digestion of DNA 73
2.4.7 Phosphorylation of DNA fragments 74
2.4.8 Oligonucleotide preparation 74
2.4.9 Annealing of oligonucleotide linkers 74
2.4.10 Agarose gel electrophoresis 74
2.4.11 Extraction of DNA from agarose gels and reaction mixtures 74
2.4.12 Ligation of DNA 74
2.4.13 Storing genetic parts 75
2.4.14 Start-Stop Assembly reactions 75
2.4.15 Membrane dialysis of DNA 75
2.4.16 Nucleotide sequencing 75
2.4.17 Coding sequences for carotenoid metabolic pathway libraries 75
2.5 Experimental Assays 76
2.5.1 *E. coli* library characterisation by flow cytometry 76
2.5.2 *Synechocystis* library characterisation by flow cytometry 76

2.5.3 Portability validation characterisation in *E. coli* and *Synechocystis* by flow cytometry 77
2.5.4 Analysis of Level 1 assembly fidelity and bias by flow cytometry 77
2.5.5 Analysis of DNA scars and characterisation of RBS library by flow cytometry .77
2.5.6 Glucuronidase assay in *E. coli* 78
2.5.7 Glucuronidase assay in *Synechocystis* 78
2.5.8 Analysis of phenotypic diversity of carotenoid pathway library clones by image analysis 79
2.5.9 Quantification of lycopene content of *E. coli* - Spectrophotometer 79
2.5.10 Quantification of lycopene content of *E. coli* - LCDAD 80
2.5.11 Quantification of lycopene content of *Synechocystis* - LCDAD 80
2.6 Data Analysis 81
2.6.1 Flow cytometry 81
2.7 Computational Methods 81
2.8 Plasmid Construction 81
2.8.1 Library construction 81
2.8.2 EYFP combinatorial build 82
2.8.3 Different combinatorial reporter libraries 82
2.8.4 Start-Stop Assembly plasmids 83
2.8.4.1 Construction of Level 0 vector pStA0 83
2.8.4.2 Construction of Level 1 vectors 83
2.8.4.3 Construction of Level 2 vectors 84
2.8.4.4 Construction of Level 3 vectors 84
2.8.4.5 Parts cloned into pStA0 85
2.8.4.6 Metabolic pathways 86
2.8.4.7 Other plasmids used in study 86
3. CHAPTER THREE. Generation of a genetic toolkit for the cyanobacterium *Synechocystis* sp. PCC 6803 87
3.1 Introduction 87
3.1.1 Portable expression control parts 87
3.1.2 Expression control parts 89
3.1.3 Aims 92
3.2 Results 93
3.2.1 Characterising expression control parts in *E. coli* and *Synechocystis* 93
3.2.2 Promoter library 94
3.2.2.1 Approach for designing library of promoters 94
3.2.2.2 Identifying clones generated by the Synthetic Promoter Library 96
3.2.2.3 Characterisation of the synthetic promoters in *E. coli* DH10B. 97
3.2.2.4 Characterisation of the synthetic promoters in *Synechocystis* sp. PCC 6803. 98

3.2.2.5 Portability of SPL promoter clones between *E. coli* and *Synechocystis*. 99
3.2.3 Promoter Library 2 – A targeted approach for generating a portable library of promoters 101
3.2.3.1 Promoter Library 2 concept.....101
3.2.3.2 Promoter Library 2 design.....102
3.2.3.3 Identifying promoter clones from Promoter Library 2.....106
3.2.3.4 Characterisation of Promoter Library 2 in *E. coli.* 106
3.2.3.5 Characterisation of Promoter Library 2 in *Synechocystis* sp. PCC 6803.....107
3.2.3.6 Sequence level comparison of the consensus sequence for functional and non-functional clones in *Synechocystis* showed high stringency for the consensus 109
3.2.3.7 Portability of Promoter Library 2 between *E. coli* and *Synechocystis* 109
3.2.3.8 Promoters generated using the SPL approach were combined with those making up Promoter Library 2 in an attempt to increase the number of portable promoters. 110
3.2.4 Ribosome-binding site library.....114
3.2.4.1 Designing a degenerate RBS library by sequence randomisation.....116
3.2.4.2 Comparing the range of expression strengths from each RBS library.....118
3.2.4.3 Identifying RBS clones from RBS library 119
3.2.4.4 Characterisation of the RBS library in *E. coli* DH10B showed a wide range of expression strengths.....120
3.2.4.5 Characterisation of the RBS library of *Synechocystis* sp. PCC 6803 showed a wide range of expression strengths. 121
3.2.4.6 Portability of the RBS library between *E. coli* and *Synechocystis*.....123
3.3 Discussion.....126
3.4 Key Outcomes 130
4. CHAPTER FOUR. Isolation and validation of a subset of portable promoters and RBS. 131
4.1 Introduction 131
4.1.1 Identifying subset combinations in datasets 131
4.1.2 Context dependence.....133
4.1.3 Aims 134
4.2 Results.....135
4.2.1 Designing a computational algorithm for identifying the best portable subset.....135
4.2.1.1 Scoring the portability of a subset.....135
4.2.1.2 Search for subsets within a library.....138
4.2.1.3 Implementing a computational tool for search and scoring portable subsets 140
4.2.2 Detailed explanation of the BIOPORT code 144
4.2.2.1 fFindSubset.....144
4.2.2.2 fShortlister.....146

4.2.2.3 fAllSubsets 147
4.2.2.4 fScore 148
4.2.3 Testing SUBSETTER and BIOPORT 150
4.2.4 Application of SUBSETTER to search for portable subsets in the promoter and RBS libraries 150
4.2.4.1 SUBSETTER analysis of the Portable Subset from Promoter Library 150
4.2.4.2 SUBSETTER analysis of the RBS Library 154
4.2.4.3 Gradient dependence: Do promoter and RBS subsets need to be isolated with the same gradient. 159
4.2.4.4 SUBSETTER analysis of the merged dataset 161
4.2.5 Validation of the portable subset 166
4.2.5.1 EYFP combinatorial build 167
4.2.5.2 Different reporters 170
4.2.5.3 RBS rank orders and curating portable subset 176
4.3 Discussion 179
4.4 Key Outcomes 183
5. CHAPTER FIVE. Combinatorially building multi gene pathways using the constraints of the genetic code – Start-Stop Assembly. 184
5.1 Introduction 184
5.1.1 Aims 188
5.2 Results 189
5.2.1 A strategy for functionally-scarless assembly of expression units 189
5.2.2 Requirements for Level 1 assembly from Level 0 parts 193
5.2.3 Validation of multi-part DNA assembly using 3 bp fusion sites 197
5.2.4 A streamlined assembly hierarchy to minimise the number of destination vectors required for new contexts and organisms 205
5.2.5 Implementation of a complete Start-Stop Assembly system 210
5.2.6 Combinatorial Start-Stop Assembly of metabolic pathways provides exploration of design space, phenotypic diversity and high-performance pathway configuration 213
5.3 Discussion 221
5.4 Key Outcomes 223
6. CHAPTER SIX. Implementing rapid prototyping 224
6.1 Introduction 224
6.1.1 Aims 227
6.2 Results 228
6.2.1 Choosing conventional parts 228
6.2.2 Design strategies for the lycopene pathway 231
6.2.3 Assembling the lycopene pathways 236
6.2.4 Pre-screen of lycopene pathways in *E. coli* 239
6.2.5 Transforming and segregating pathways in *Synechocystis* 242

6.2.6 Characterising lycopene pathway clones in *E. coli* 246
6.2.7 Characterising lycopene pathway clones in *Synechocystis* 247
6.2.8 Portability of the libraries of pathway-encoding constructs between *E. coli* and *Synechocystis* 249
6.3 Discussion 253
6.4 Key Outcomes 258
7. CHAPTER SEVEN. Discussion 259
7.1 Rapid prototyping of metabolic pathway-encoding constructs 259
7.2 Factors that affect protein concentration profiles being preserved 260
7.3 Metabolic constraints on portability 264
7.4 Alternative approaches to overcoming engineering non-model organisms…… 266
7.5 Enhancing the cyanobacterial genetic toolkit 268
7.6 Lycopene production 268
7.7 Conclusion 270
7.8 Future work with respect to portability 271
7.9 Future work with respect to increasing lycopene production 271
7.10 Future directions 272
REFERENCES 274
APPENDIX 313

List of Figures

Figure 1.1.	Central dogma of molecular biology.	31
Figure 1.2.	Optimal balance of pathway proteins sustains optimal pathway performance.	37
Figure 1.3.	The process of refactoring a gene cluster.	41
Figure 1.4.	Combinatorial assemblies provide a sparse sampling of the design space.	43
Figure 1.5.	Scanning electron micrographs of bacteria used in this study	47
Figure 1.6.	Routes for biological productions of chemicals and biofuels.	52
Figure 1.7.	Cyanobacterial metabolism showing CO_2 converted to a host of products.	57
Figure 1.8.	Timelines of iterative approaches in fast and slow-growing organisms.	64
Figure 1.9.	Timelines for optimising a metabolic pathway using iterative and rapid prototyping-based approaches.	66
Figure 3.1.	Illustrative example for the concept of rapid prototyping.	87
Figure 3.2.	Illustrative examples of different performances of expression control parts between organisms.	88
Figure 3.3.	Illustrative example showing expected response of protein concentrations from transferring a metabolic pathway-encoding construct between organisms using current state-of-the-art expression tools.	89
Figure 3.4.	*E. coli-Synechocystis* shuttle vector pATM2 containing an *eyfp* reporter gene.	93
Figure 3.5.	The Synthetic Promoter Library design in the context of pATM2.	95
Figure 3.6.	Screening cascade for promoters and RBS clones.	97
Figure 3.7.	Synthetic Promoter Library characterisation in *E. coli* DH10B.	98
Figure 3.8.	Synthetic Promoter Library characterisation in *Synechocystis* sp. PCC 6803.	99
Figure 3.9.	Portability of synthetic promoters from the SPL between *E. coli* DH10B and *Synechocystis* sp. PCC 6803.	100
Figure 3.10.	Potential subsets of portable promoters within the SPL.	101
Figure 3.11.	Illustration showing the hypothesized outcome of the targeted portable design for Promoter Library 2.	102
Figure 3.12.	The range of promoter strengths generated in each of the four Promoter Library 2 designs.	105
Figure 3.13.	The Design of Promoter Library 2 in the context of pATM2.	106
Figure 3.14.	Promoter Library 2 characterisation in *E. coli* DH10B.	107
Figure 3.15.	Promoter Library 2 characterisation in *Synechocystis* sp. PCC 6803.	108
Figure 3.16.	Portability of promoters from Promoter Library 2 between *E. coli* DH10B and *Synechocystis* sp. PCC 6803.	110

Figure 3.17. Promoters generated using the Synthetic Promoter Library and Promoter Library 2 characterised in *E. coli* DH10B. 111
Figure 3.18. Promoters generated using the Synthetic Promoter Library and Promoter Library 2 characterised in *Synechocystis* sp. PCC 6803. 112
Figure 3.19. Portability of the 92 promoters generated using the Synthetic Promoter Library and Promoter Library 2 compared between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 113
Figure 3.20. Portability of the final decluttered Promoter Library between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. 114
Figure 3.21. RBS library design 1 in the context of pATM2. 117
Figure 3.22. The range of RBS strengths generated in each of the nine RBS library designs. 119
Figure 3.23. RBS library characterisation in *E. coli* DH10B. 120
Figure 3.24. Comparison of the predicted and measured strength of the RBS library in *E. coli*. 121
Figure 3.25. RBS library characterisation in *Synechocystis* sp. PCC 6803. 122
Figure 3.26. Comparison of the predicted and measured strength of the RBS library in *Synechocystis*. 123
Figure 3.27. Portability of RBSs between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. 124
Figure 3.28. Portability of the three different designs used to isolate clones for the RBS library. 125
Figure 3.29. Potential subsets of portable RBSs within the RBS library. 126
Figure 4.1. Relationship between dataset size and number of subset combinations. 132
Figure 4.2. Illustrative example of the portability of a part in different genetic contexts. 134
Figure 4.3. Illustration of an ideal portable subset. 136
Figure 4.4. Desirable properties that compose the scoring function. 137
Figure 4.5. Search function parameters of the algorithm. 140
Figure 4.6. Flow-chart of the SUBSETTER algorithm. 143
Figure 4.7. SUBSETTER search on test dataset. 144
Figure 4.8. Using SUBETTER to identify promoter subsets within the promoter library. 151
Figure 4.9. Zoomed in SUBSETTER searches around local optima in the promoter library. 153
Figure 4.10. Using SUBETTER to identify RBS subsets within the RBS library. 155
Figure 4.11. Zoomed in SUBSETTER searches around local optima in the RBS library. 158
Figure 4.12. Schematic representation of using portable subsets in combinatorial assemblies. 160

Figure 4.13. Host comparison of the merged promoter and RBS library between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. 161
Figure 4.14. Using SUBETTER to identify a portable subset within the merged promoter and RBS library. 163
Figure 4.15. Zoomed in SUBSETTER searches around local optima in the merged promoter and RBS library. 164
Figure 4.16. Portable subset highlighted on the merged promoter and RBS library. 165
Figure 4.17. Combinatorial assembly of the EYFP reporter using the portable subset in the pATM2 shuttle vector. 167
Figure 4.18. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the combinatorial assembly of EYFP. 169
Figure 4.19. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the combinatorial assembly of EYFP. 170
Figure 4.20. Reporters used for assessing the effects that context has on the portability of an RBS. 172
Figure 4.21. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the four different fluorescent reporters. 174
Figure 4.22. Host comparison of the glucuronidase activity between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. 176
Figure 4.23. Host comparison of portable RBSs with different reporters. 178
Figure 4.24. Host comparison of EYFP fluorescence intensities using the curated portable subset between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. 179
Figure 4.25. Relationship between dataset size and the number of scores to be calculated. 181
Figure 5.1. Effect of DNA assembly scars immediately upstream of the start codon on expression of *sfGFP*. 188
Figure 5.2. Schematic overview of Start-Stop Assembly. 192
Figure 5.3. Choice of β fusion site. 193
Figure 5.4. Strategy for PCR-amplification and cloning of genetic parts into Level 0 vector pStA0. 196
Figure 5.5. Promoters and RBSs stored in pStA0. 198
Figure 5.6. Design, generation and characterisation of a synthetic RBS library. 199
Figure 5.7. Assessment of fidelity of multi-part Level 1 assembly. 201
Figure 5.8. Assessment of bias of multi-part Level 1 assembly. 203
Figure 5.9. Flow cytometry data for individual clones used in assessment of fidelity and bias of multi-part Level 1 assembly. 204

Figure 5.10.	Alternative hierarchies achieved by alternative organisation of Level 1 donor and Level 2 acceptor fusion sites.	207
Figure 5.11.	Detail of multi-part hierarchical assembly and fusion sites used in Start-Stop Assembly.	209
Figure 5.12.	Strategy for construction of an alternative Level 2 destination vector.	212
Figure 5.13.	Combinatorial assembly of hybrid astaxanthin pathway library pGT535	216
Figure 5.14.	Combinatorial assembly of astaxanthin pathway using Start-Stop Assembly.	218
Figure 5.15.	Analysis of the four β-carotene pathway libraries.	219
Figure 5.16.	Lycopene content of *E. coli* clones from the astaxanthin library.	220
Figure 6.1.	Conventional Parts selected in *Synechocystis* sp. PCC 6803.	230
Figure 6.2.	Subsets of conventional promoters and RBSs shown in the host comparisons of the SPL and RBS libraries.	231
Figure 6.3.	Lycopene pathway.	232
Figure 6.4.	Design principle for Pathway design 1.	234
Figure 6.5.	Design principles for Pathway design 2 – 5.	235
Figure 6.6.	*E. coli-Synechocystis* shuttle vector pGT270 containing appropriate Level 2 Start-Stop Assembly cassette.	236
Figure 6.7.	Assembly of lycopene pathways used in Pathway designs 1 – 5.	238
Figure 6.8.	Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the conventional parts.	240
Figure 6.9.	Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the full subset of portable parts.	241
Figure 6.10.	Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the curated subset of portable parts.	242
Figure 6.11.	*Synechocystis* transformation outcome.	245
Figure 6.12.	Lycopene concentrations in *E. coli*.	247
Figure 6.13.	Lycopene pathway characterisation in *Synechocystis*.	248
Figure 6.14.	Host comparison of lycopene concentrations for pathways derived from Pathway design 2.	250
Figure 6.15.	Host comparison of lycopene concentrations for pathways derived from Pathway design 3.	251
Figure 6.16.	Host comparison of lycopene concentrations for pathways derived from Pathway design 4.	252
Figure 6.17.	Host comparison of lycopene concentrations for pathways derived from Pathway design 5.	253
Figure 7.1.	Relationship between genetic design and pathway performance.	260
Figure 7.2.	Relationship between promoter strength, RBS strength and the portability of a pathway's performance.	261
Figure 7.3.	Heuristic refinement of a combinatorial search space	270

Figure A1.	Combinatorial assembly of monocistronic β-carotene pathway library pGT531.	345
Figure A2.	Combinatorial assembly of operon-based β-carotene pathway library pGT532.	345
Figure A3.	Combinatorial assembly of hybrid β-carotene pathway library pGT533.	346
Figure A4.	Combinatorial assembly of hybrid β-carotene pathway library pGT534.	346
Figure A5.	Level 1 assembly of an expression unit showing detail of Level 0 promoter part.	347
Figure A6.	Level 1 assembly of an expression unit showing detail of Level 0 RBS part.	347
Figure A7.	Level 1 assembly of an expression unit showing detail of Level 0 CDS part.	347
Figure A8.	Level 1 assembly of an expression unit showing detail of Level 0 terminator part.	348
Figure A9.	Level 1 assembly of an expression unit showing detail of Level 1 vector.	348
Figure A10.	Comparison of codon bias between *E. coli* K-12 and *Synechocystis* sp. PCC 6803	349
Figure A11.	Host comparison of lycopene concentrations for pathways derived from Pathway design 1 – 4.	350
Figure A12.	Using angles or gradients as increments to search datasets	351
Figure A13.	Testing SUBSETTER with Anscombe's Quartet.	352

List of Tables

Table 2.1.	Table of bacterial strains used in this study	70
Table 3.1.	Four designs used for the Promoter Library 2.	103
Table 3.2.	Sequence comparison of the -35 and -10 elements of the five active promoter clones isolated from the Promoter Library 2.	109
Table 3.3.	Nine RBS library designs.	117
Table 4.1.	Table of portable promoters that make up SubsetP3.	154
Table 4.2.	Table of portable RBSs in SubsetR3.	159
Table 4.3.	Promoters and RBSs that make up the portable subset.	166
Table 5.1.	List of core Start-Stop Assembly vectors	195
Table 5.2.	Prefix and suffix sequences.	196
Table 6.1.	Conventional promoters and RBSs.	229
Table A1.	Sequences of promoter clones generated in the Synthetic Promoter Library.	313
Table A2.	Sequences of promoter clones generated in the Promoter Library 2.	314
Table A3.	Sequences of RBS clones generated in the RBS Library.	315
Table A4.	Oligonucleotides used in the study.	317
Table A5.	Table of synthetic DNA sequences used in the study.	328
Table A6.	Alignment of the five strongest promoter clones against non-functional clones that differed by a single nucleotide from Promoter Library 2 against.	333
Table A7.	Spacers implemented as double-stranded linkers.	333
Table A8.	Table of plasmids used in this study.	335

List of Code Snippets

Code snippet 1. Code for Subsetter.py 141

Code snippet 2. fFindSubset function. 146

Code snippet 3. fShortlister function. 147

Code snippet 4. fAllSubsets function. 147

Code snippet 5. fScore function. 149

Algorithm A1. Code for BioPort.py 354

List of Abbreviations

α =	Alpha
β =	Beta
γ =	Gamma
δ =	Delta
ε =	Epsilon
σ =	Sigma
μ =	Micro
μg =	Microgram
μl =	Microlitre
μmol =	Micromole
μM =	Micromolar
Δ*G* =	Free energy change
°C =	Degrees Celsius
2PG =	2-phosphoglycolate
3D =	Three dimensional
3PG =	3-phosphoglycerate
a.u. =	Arbitrary units
ABE =	Acetone−butanol−ethanol
ADP =	Adenosine diphosphate
ALE =	Adaptive laboratory evolution
aTc =	Anhydrotetracyline
ATP =	Adenosine triphosphate
AQUA =	Advanced quick assembly
bp =	Basepair
BCD =	Bicistronic design
BMC =	Bacterial microcompartment
BPG =	1,3-Bisphosphoglycerate
BASIC =	Biopart Assembly Standard for Idempotent Cloning
BNICE =	Biochemical Network Integrated Computational Explorer
CA =	Carbonic anhydrase
cm =	Centimetre
CH_4 =	Methane
CO =	Carbon monoxide
CO_2 =	Carbon dioxide
CAD =	Computer aided design
CAI =	Codon adaptation index
CBB =	Calvin-Benson-Bassham
CCM =	CO_2-concentrating mechanism
CDS =	Coding sequence
cfu =	Colony-forming unit
CHO =	Chinese hamster ovary cells
CoA =	Coenzyme A
CLIVA =	Cross-lapping in vitro assembly
COBRA =	Constraint-based reconstruction and analysis
CPEC =	Circular polymerase extension cloning
CRISPR =	Clustered regularly interspaced short palindromic repeats

DAG =	Diacylglycerol
DCW =	Dry cell weight
DNA =	Deoxyribonucleic acid
DoE =	Design-of-experiments
DBTL cycle =	Design-Build-Test-Learn cycle
DMAPP =	Dimethylallyl pyrophosphate
EMP =	Embden-Meyerhof-Parnas
F6P =	Fructose 6-phosphate
FAD =	Flavin adenine dinucleotide
FBA =	Flux balance analysis
fmol =	Femtomole
FNR =	Ferredoxin-$NADP^+$ reductase
FPP =	Farnesyl pyrophosphate
g =	gram
G3P =	Glyceraldehyde 3-phosphate
GAP =	Glyceraldehyde-3-phosphate
GOI =	Gene of interest
GSM =	Global-scale stoichiometric models
Gya =	Billion years
gRNA =	Guide RNA
gTME =	Global transcription machinery engineering
GGPP =	Geranylgeranyl pyrophosphate
H^+ =	Hydrogen ion
H_2 =	Hydrogen
H_2O =	Water
HCO_3^- =	Bicarbonate
IME =	Inverse metabolic engineering
IPP =	Isopentenyl pyrophosphate
ITS =	Initially transcribed sequence
iFFL =	Incoherent feedforward loop
IPTG =	Isopropyl β-D-1-thiogalactopyranoside
kb =	Kilobase
kV =	Kilovolts
l =	Litre
LB =	Luria Bertani
LHA =	Left homology arm
m =	Metre
ml =	Millilitre
mg =	Milligram
M =	Molar
mM =	Millimolar
MCA =	Metabolic Control Analysis
MDF =	Max-min Driving Force
MEP =	Methylerythritol-phosphate
MFA =	Metabolic flux analysis
min =	Minute
MVA =	Mevalonic acid
MPEC =	Metabolic pathway-encoding construct

mRNA = Messenger RNA
nm = Nanometre
NaCl = Sodium chloride
NADH = Nicotinamide adenine dinucleotide
NADPH = Nicotinamide adenine dinucleotide phosphate
O_2 = Oxygen
OD = Optical density
OEC = Oxygen-evolving centre
ORI = Origin of replication
PC = Plastocyanin
Pi = Inorganic phosphate
PQ = Plastoquinone
PAM = Protospacer adjacent motif
PCR = Polymerase chain reaction
PEP = Phosphoenolpyruvate
PHB = Polyhydroxybutyrate
PMF = Proton-motive force
PLA = Polylactic acid
PSI = Photosystem I
PSII = Photosystem II
PUFA = Polyunsaturated fatty acids
r.p.m = Revolutions per minute
RBS = Ribosome-binding site
RC = Reaction centre
RDS = Ribosome docking site
RedLibs = Reduced Libraries
RHA = Right homology arm
RNA = Ribonucleic acid
RNAP = RNA polymerase
rRNA = Ribosomal RNA
RuBisCO = Ribulose-1,5-bisphosphate carboxylase/oxygenase
Ru5P = Ribulose-5-phosphate
RuBP = Ribulose 1,5-bisphosphate
s = Second
SD = Shine-Dalgarno
sp. = Species
SNP = Single-nucleotide polymorphism
SPL = Synthetic Promoter Library
SCRaMbLE = Synthetic Chromosome Recombination and Modification by LoxP-mediated Evolution
SLIC = Sequence- and ligation-independent cloning
SLiCE = Seamless Ligation Cloning Extract
STAR = Small transcription activating RNA
TAG = Triacylglycerol
TALE = Transcription-activator like effector
TCA = Tricarboxylic acid
T.I.R = Translational initiation rates
tracrRNA = Trans-activating crRNA

tRNA =	Transfer RNA
TSS =	Transcriptional start site
TX / TL =	Transcription-translation
UBER =	Universal Bacterial Expression Resource
USER =	Uracil-Specific Excision Reagent
UTR =	Untranslated region
UV =	Ultraviolet
V =	Volts
Xgal =	5-bromo-4-chloro-3-indolyl-β-galactopyranoside
y =	Year

1. CHAPTER ONE. Introduction

1.1 Industrial biotechnology

Industrial biotechnology, also referred to as white biotechnology, is regarded as the third wave of biotechnology, distinct from the other two applications that focus on medical (red biotechnology) and agricultural (green technology) objectives. Industrial biotechnology focuses on the sustainable production of chemicals, materials and fuels from renewable sources, using living cells and/or their enzymes to do so. The field has and continues to attract attention because of its objectives to reduce energy consumption, greenhouse gas emissions and waste generation, as well as helping to enable the necessary shift from fossil carbon-based to bio-based production of value-added chemicals. These objectives are essential, given the challenges that the Earth and its population are facing due to climate change. Human activities, largely including the release of CO_2 from fossil fuels, have been named as the predominant cause of climate change (1). Greenhouse gas emissions at their current levels are contributing to more intense weather patterns, including droughts and heat waves, as well as rising sea levels (1). The United Nations climate change policy aims to limit global warming to below 2 °C in order to limit consequences, which is arguably not achievable and would not even be enough to prevent further disaster (2).

Considering the stated objectives of industrial biotechnology and also recent advances in metabolic engineering, synthetic biology, systems biology and enzyme engineering; industrial biotechnology has grown exponentially in the last few decades. As such, there are an increasing number of chemicals and materials, such as ethanol (many companies), 1,3-propanediol (DuPont), lactic acid (Cargill), farnesene (Amyris) and isobutanol (Gevo), being microbially produced at an industrial scale. As a result, the industrial biotechnology global market size was estimated at $369.62 billion in 2016 and is predicted to reach a global market size of $727.1 billion by 2025 (3). As the predicted increase in market size suggests, the field of industrial biotechnology is expected to continue to rapidly grow, fueled by: i) the need for highly efficient processes at lower operating and capital expenditures; ii) political and societal demands for sustainable and environmentally friendly industrial production systems; and iii) demand for raw materials and energy, as crude oil reserves continue to deplete (4).

Microorganisms were exploited long before their discovery; to preserve milk, fruits and vegetables, as well as to produce wine, beer, cheeses, bread, pickled foods and

vinegar, with the implementation of yeast fermentation in their production dating back to before 7000 BC in Sumeria and Babylonia (5). Since then microorganisms have continued to be used in the food industry and in more recent years they have been exploited to naturally produce other non-food-based products of interest. The first example of a non-food fermentation process developed at large-scale was using *Clostridium* to produce acetone, butanol and ethanol in World War I (6), where the acetone was used to manufacture cordite, a smokeless alternative to gunpowder. The production of penicillin in *Penicillium chrysogenum* NRRL 1951 soon followed around the time of World War II (1940s) (6). During this time the need for penicillin was so great that it prompted a whole new technology known as “strain improvement” which primarily used random mutagenesis (induced by chemical mutagens or radiation) and screening for improved product production. The follow on from penicillin was the discovery of a wide range of commercially-important primary metabolites being naturally produced by fermentation, such as amino acids in *Corynebacterium* (7), vitamins in fungi such as *Eremothecium ashbyi* and *Ashbya gossypii* (8), organic acids, such as citric acid and lactic acid by *Aspergillus niger* (9) and *Lactobacillus* (10), respectively, and alcohols including ethanol in yeast (11), to name but a few.

Fields associated with industrial biotechnology, such as metabolic engineering, synthetic biology and systems biology, have advanced in recent years and this has expanded the potential of industrial biotechnology, not just to exploit products that are naturally made by microbes, but also to engineer the production of products that aren’t natural to that organism.

1.2 Metabolic engineering

Metabolism refers to the biochemical reactions that occur in a cell or organism. Such biochemical reactions encode functions such as the conversion of carbon substrates to cellular biomass precursors, energy equivalents and biochemical molecules (12). Specific selective pressures from varying environmental conditions have led to the evolution of a great diversity of the cellular metabolisms of different organisms. The differences in cellular metabolism allows certain organisms to perform different functions, including growing in different conditions, such as heterotrophs, autotrophs or photoautotrophs, and producing a plethora of different products, many of which have attractive properties.

Nature's metabolism is however constrained and therefore shaped by chemical and physical forces. These constraints are the stoichiometry, thermodynamics and kinetics of the biochemical reactions, which dictate not only the feasibility of a reaction but also the rate of the reaction (12). Stoichiometry dictates that a reaction can only occur when the total mass of reactants equals the total mass of products. The flux (rate and direction of flow of metabolites) of a reaction can be determined by the thermodynamics and kinetics of each enzyme-catalysed step in the pathway. From a thermodynamic perspective, the free energy change (ΔG) of a reaction can dictate whether a given reaction proceeds in the forward or reverse direction, while the kinetics of an enzyme can affect the rate of the reaction. The kinetics of biochemical reactions can further be constrained by thermodynamic potentials of those reactions (13-15).

1.2.1 The history of metabolic engineering

The field of metabolic engineering first emerged around 30 years ago, following the publication of two seminal papers (16, 17). The field was quickly defined as the "directed improvement of product formation or cellular properties through the modification of specific biochemical reactions or the introduction of new genes with the use of recombinant DNA technology" (18). The new field differentiated itself from other fields such as genetic engineering with its distinct focus on investigating and optimising the properties of integrated metabolic pathways and genetic regulatory networks, which at the time was a step beyond research, which had largely focused on individual genes and enzymes (19). Furthermore, metabolic engineering aimed to design, construct and optimise metabolic pathways that perform in a cost-effective manner (that is optimised titre, rate (productivity) and yield), subsequently resulting in microbes that can be used as biocatalysts for the cost-effective productions of fuels, chemicals and pharmaceuticals, hence the fields strong association with industrial biotechnology. The field of metabolic engineering is therefore not simply made up of just genetic engineering and molecular biology, but also graph theory, chemical reaction engineering, biochemistry and optimisation.

1.2.2 Approaches and applications of metabolic engineering

Most applications of metabolic engineering can be split into one of three approaches: i) rational approaches; ii) random approaches; or iii) semi-rational approaches. In a simple sense, rational approaches rely on prior understanding of the pathways and their associated metabolic fluxes, kinetics and regulations and then use rationally-made decisions to manipulate metabolism shifting flux towards a desired product. In

contrast, random approaches do not rely on *a priori* knowledge of metabolic flux or kinetics, rather they implement random changes to the genotype of an organism and then screen mutants for improved phenotype. Semi-rational approaches introduce random changes to the genotype of an organism at rationally deduced locations. For example, the expression control parts of a gene of interest, or at specific locations in a gene.

1.2.2.1 Rational approaches to metabolic engineering

Over the last 20 - 30 years, most metabolic engineering efforts have used rational approaches to manipulate metabolism. Rational approaches use prior knowledge of every possible pathway route for converting a substrate to a target product to make rational manipulations to the metabolism to improve the productivity of the target product. This might include removing bottlenecks identified in pathways kinetics, removing side reactions that redirect metabolic flux away from the target product or introducing a heterologous enzyme to complete a pathway. To make such decisions, it is necessary to understand metabolic topography, kinetics and regulation of the pathways in question. To determine the aforementioned factors, many tools have been developed since the emergence of metabolic engineering.

To identify, understand and compare metabolic pathways, all possible routes that could be used to produce a target product from a specific substrate should ideally be identified, although this is not always necessary, as rational approaches can be implemented for any given single metabolic pathway. To identify and compare metabolic pathways, many tools have been developed, using biochemical information from a variety of sources to create a list of biochemical reactions that make up the metabolic network of that organism. The first examples of such a tool was developed in the early 1990s and used stoichiometric constraints and a database of biochemical reactions to identify pathway routes from a specified substrate to target product (20). It was successfully used to identify several pathways that bypassed bottlenecks in the biosynthesis of lysine. This type of methodology is under continual development, with many more methods since having been developed, such as the Biochemical Network Integrated Computational Explorer (BNICE) (21), PathPred (22), RetroPath (23) and DESHARKY (24). These methods have successfully designed and evaluated pathways for the production of 3-hydroxypropanoate, an industrial precursor (25) and 1,4-butanediol from different sugar substrates (26).

Having established all of the potential pathway routes in an organism's metabolic network, next the rates of each potential pathway can be analysed to understand the specific flow of metabolites in the metabolic network. This greater understanding of a pathway can be used to identify critical branching points in the pathway (27, 28), to discover unusual pathways in less characterised species (29), to define the maximum theoretical yield for the synthesis of products from complex integrated pathways producing and consuming multiple cofactors and intermediates (30) and to rationally identify targets for genetic manipulation within that pathway. Metabolic flux analysis (MFA) is the principal technique that can be used for analysing the flow of metabolites as it quantifies the intracellular metabolic fluxes. MFA is based on the global stoichiometry of metabolic reactions (global-scale stoichiometric models (GSM)) and the mass balances around intracellular metabolites under the pseudo-steady state assumption (31-33). Two methods have typically been used: ^{13}C-based flux analysis and constraints-based flux analysis. However, an inherent problem with ^{13}C-based flux analysis approaches is the cost and expertise needed for these experimental procedures. As a result, purely computational methods to estimate fluxes under different conditions have more recently been developed.

Flux balance analysis (FBA) is a widely used computational approach developed for assessing the metabolic flux of an organism. Like MFA, FBA employs GSMs and mass balances and unlike MFA, it applies user-based constraints. User constraints are applied to maximise or minimise an objective function, such as maximising metabolite or biomass production, which can be simulated using software such as the constraint-based reconstruction and analysis (COBRA) package under a range of conditions, such as oxygen or carbon dioxide (CO_2) availability (34). FBA is becoming increasing applied in metabolic engineering as more GSMs become available. Currently there are 113 prokaryotic GSMs available (35), including *Bacillus subtilis* (36), *Clostridium acetobutylicum* (37) and *Corynebacterium glutamicum*, and 57 eukaryotic GSMs (35), including *Saccharomyces cerevisiae* (38) and *Arabidposis thaliana* (39). FBA has become an essential tool for increasing target product production in metabolic engineering as the flux distribution predictions can provide information for rationally deduced genetic manipulations, such as identifying knockout targets. The success of this was demonstrated when the lycopene production was increased 37% in a FBA designed knockout strain over its parental strain (40). Some of these methods have been formalised into widely available tools that are easier to implement, such as OptKnock (41) and IdealKnock (42), which allows users to determine a set of targeted knockouts to increase the flux of a desired product.

Such stoichiometric based techniques have been used to develop a rationale for genetic manipulation of an organism. The rationally deduced strategies have most often been based on gene knockouts, overexpression of homologous genes, or the introduction of heterologous genes.

The stoichiometry of an organism's metabolism is significantly simpler to model compared to the kinetics of an organism's metabolism, as it maps whether a reaction can occur or not but offers no detail on the rate of the reaction. As a result, stoichiometric models have been easy to implement in FBA and MFA and have worked well for identifying targets for redirecting metabolic flux as described above. However, one major limitation of stoichiometric models is that it does not take into account kinetics, concentrations and the regulation of enzymatic reactions (43), instead they just account for the presence or absence of an enzyme. In reality, specific enzyme concentration and kinetics are required for many reactions to function appropriately and therefore kinetics and stoichiometry need to be modelled together to provide a fine-level of enzyme rates.

One kinetic based technique is Metabolic Control Analysis (MCA), which is frequently used to mathematically determine the effects that different enzyme levels have on metabolic fluxes and the concentration of metabolites in a steady-state (44-47). MCA predictions have been used in *L. lactis*, *Aspergiullus nidulans, Aspergillus niger, Sacchromyces cerevisiae* and *E. coli* (48). In *E. coli*, it identified *gldA* and *dhaK* as bottlenecks in the production of glycerol (49). However, MCA and other kinetic based models have not widely been used because enzyme kinetic properties are not only laborious to measure, but also differ between organisms and isozymes. As a result, the relevant kinetic information tends not to be available.

Recently, some progress was made that overcomes the complex nature of using the kinetics of enzymatic reactions. Recently the Max-min Driving Force (MDF) methodology was developed, which allows the thermodynamic and stoichiometric landscape of metabolic pathways to be quantitatively analysed (50). This in turn allows specific reactions within the pathway which, due to their thermodynamics have a low thermodynamic driving force, are therefore likely to limit metabolic flux and are likely require a higher enzyme concentration to catalyse forward flux (50). MDF has successfully identified key limiting reactions in central metabolism that have low thermodynamic driving forces and if the enzyme concentration was likely increased, flux would in turn also increase (50).

Using the stoichiometric-based approaches outlined above it has been possible to strategically identify gene knockouts that will increase production of a target product by preventing redirection of metabolic flux to unnecessary byproducts and channeling it to the target product. This was demonstrated in *E. coli* where succinic acid production was increased seven-fold from the wild-type parent strain by knocking out three genes, *ptsG*, *pykF* and *pykA*, which had been rationally identified using FBA as a cause for the redirection of metabolic flux to pyruvate and its downstream metabolites (51).

Targeting the overexpression of a homologous gene has also proven a successful strategy for overcoming rate limiting (kinetic) bottlenecks in a pathway by improving the reactions rate by increasing the enzyme concentration, which in turn improves metabolic flux through the pathway. An example that highlights the importance of integrated pathway analysis is the overproduction of lysine in *C. glutamicum*. *C. glutamicum* naturally produces amino acids such as lysine and has been used on an industrial scale to do so. However, strain improvement was still targeted to improve lysine yield from glucose. Using ^{13}C-based MFA revealed that 90% of lysine flux was directed through the pyruvate carboxylase (*pyc*) and highlighted it as a potential bottleneck and therefore a sensible target for engineering (52). The end result was rational overexpression of *pyc* and aspartokinase (*ask*), the downstream gene of *pyc*, which yielded an increase in lysine production by 150% (53).

Rationally-identified heterologous genes can be introduced to improve production of a product that a host naturally produces or to allow the biosynthesis of a non-native product. For example, by using GSMs to analyse metabolic flux in *S. cerevisiae*, ethanol production was improved. Metabolic analysis showed that introducing *gapN* (in this case from *Streptococcus mutans*) would decrease glycerol formation, which would in turn increase the availability of NADH for ethanol production (54). In many cases, the introduction of a single heterologous gene is not enough to produce the target product and more genes have to be expressed. For example, taxadiene is a non-native product to *E. coli*, however *E. coli* was engineered to produce it by introducing two heterologous genes, which converted the native isoprenoid precursors into the non-native taxadiene. In this instance, four homologous genes from the MEP were also overexpressed to increase the flux to the isoprenoid precursors (55). Often a combination of rational approaches are used together. For example, an optimisation of the production of 2-methyl-1-butanol was achieved in *E. coli* by overexpressing the homologous threonine biosynthesis and heterologous isoleucine biosynthesis pathways, while also ‘knocking-out’ competing pathways (56).

1.2.2.2 Random approaches to metabolic engineering

While random approaches differ to rational and semi-rational approaches in their implementation, they both share the same goal in metabolic engineering, which is to improve the production of a target product. However, unlike rational and semi-rational approaches, random approaches require no *a priori* knowledge of metabolic pathways and their global topography. Instead, they implement changes to the genotype of an organism that are random and then screen many strains for the desired or improved phenotype. Random approaches have the advantage of improving a strain's phenotype in nonintuitive ways. Many of the techniques that are now implemented in metabolic engineering have long been used in other industrial biotechnological applications pre-dating metabolic engineering. Here, some of the random approaches that have been used in metabolic engineering are introduced.

Random mutagenesis has been extensively used for a long time to produce industrially applicable strains. The approach involves exposing microorganisms to mutagens, such as UV light or chemical mutagens, to increase the frequency of mutations. The resultant strain library is then screened for phenotypes with improved production of the target product. Most mutations will be irrelevant to the metabolism of interest. Rarely a mutation will have a beneficial effect. This approach was used to increase the production of penicillin in *Penicillium chrysogenum* and resulted in a 4,000-fold increase (50 gl^{-1}) over the parental strain (57). Historically the mechanism for such improvements would not have been understood, but since the development of high-throughput DNA sequencing, it has become feasible to sequence the genome of improved strains, identify mutations and potentially determine the genetic basis for the improvement. Following which, inverse metabolic engineering (IME) was developed in the early 2000s, which combines both random and rational approaches. IME uses random mutagenesis to generate and isolate improved strains, it then goes one step further by identifying the specific mutations accountable for the improved phenotype and then rationally reconstructing the strain (58). Illustrative examples include, improving the production of ethanol by 70% and xylose uptake by 100% in *S. cerevisiae* (59), and heterologous protein production in *E. coli* (60).

Adaptive laboratory evolution (ALE) emerged two decades ago as a valuable method to develop and optimise strains and since has become a widely used tool in metabolic engineering, improving yields and reducing costs in industrial settings. Metabolic engineering applications have been able to take advantage of the fast adaptation of microbes towards changing environments (61), where spontaneous mutations and

genomic rearrangements occur providing a fitness advantage under particular selection pressures within the population. ALE has predominately been used in fast-growing tractable organisms such as *E. coli* and *S. cerevisiae*; notable examples in *E. coli* include improving ethanol tolerance (62) and the co-utilisation of xylose and glucose (63).

1.2.2.3 Semi-rational approaches to metabolic engineering

As described earlier, semi-rational approaches introduce random mutations to the genotype of an organism but at specific locations in the genome or pathway, hence their semi-rational nature. Here, some of the semi-rational approaches that have been used in metabolic engineering are introduced.

Global transcription machinery engineering (gTME) introduces mutations to the DNA-binding regions of the transcription factor via error-prone PCR. Resultant mutations can affect the binding affinities between transcription factors and promoters, which in turn affects 'global' expression profiles in the cell. It was first used to improve *S. cerevisiae* tolerance to ethanol (64) and has since been used to improve many other phenotypes in many microorganisms (65).

COMPACTER is another technique that falls into the category of semi-rational approaches. While its pathway design starts simple, assembling each gene in the pathway under an individual promoter and terminator, it then uses nucleotide analog mutagenesis to introduce mutations to the promoters. High throughput screening can then be used to identify pathways that have an optimal-balance of expression levels (66).

SCRaMbLE is a semi-rational technique that rearranges and deletes genes by using symmetrical loxP recombination sites in the 3′ untranslated regions (3′-UTR) of all non-essential genes (67) that have been added to the synthetic chromosomes of *S. cerevisiae* (68). While the reorganisation is random, it requires the concerted effort to rationally place loxP sites around non-essential genes. A recent study generated used SCRaMbLE based combinatorial genomic rearrangement of the synthetic chromosome V to generate strains with improved violacein and penicillin production and xylose utilisation (69).

Genetic rewiring is a strategy for introducing novel interactions into a transcriptional regulatory network (70). This is accomplished by creating non-native combinations between the promoters and coding sequences of transcriptional regulators in a cell and exposing it to selection pressures. This synthetic construct effectively 'rewires' the regulatory architecture of the strain creating new routes through which regulatory information can flow (70). Such synthetic network architectures have successfully improved *E. coli's* capacity to survive at greater temperatures or improved longevity in the stationary phase (70), as well as improving production of products such as lycopene in yeast (71).

Semi-rational approaches can also be used on a single-gene scale rather than a global-scale, which is typically implemented using mutagenesis and directed evolution of existing enzymes to create new functionality or alter specificity. This approach was used to engineer a novel pathway that produces polylactic acid (PLA), a biodegradable polymer from lactate, to produce PLA. D-lactate would first need to be converted to lactyl-CoA and then to PLA, however no enzymes existed in nature that could perform such reactions. Similarly performing enzymes were engineered through random mutagenesis, site-directed mutagenesis and saturation mutagenesis to encode the novel reactions (72).

Synthetic biology has also developed new genome engineering technologies that have been applied in metabolic engineering, such as the bacterial CRISPR–Cas9, which has been used in bacteria and yeast (73-78). A trans-activating crRNA (tracrRNA): crRNA duplex (or a chimeric guide RNA (gRNA)) directs the Cas9 protein to cleave a target DNA sequence with a required protospacer adjacent motif (PAM). This technology has been applied for metabolic engineering in *Saccharomyces cerevisiae*, in which all combinations of five gene disruptions were searched for improved mevalonate production (79). In *Escherichia coli*, CRISPR–Cas9 has been used to integrate the β-carotene synthetic pathway into the genome and to optimize the methylerythritol-phosphate (MEP) pathway and central metabolic pathways for β-carotene overproduction (80).

1.3 The design of metabolic pathway encoding constructs

1.3.1 Controlling gene expression

The development of effective metabolic pathway-encoding constructs is a key feature of metabolic engineering that has allowed not only improved production of natural

products, but also the production of non-native products in a given organism. The expression of homologous or heterologous genes in any given organism depends on many factors, including DNA replication, transcription initiation, transcription termination, RNA replication, translation initiation, translation elongation and mRNA and protein degradation (Figure 1.1). Some of these factors can be controlled by expression control parts. In the case of prokaryotes and eukaryotes, this has typically involved plasmid copy numbers, promoters, transcriptional terminators and coding sequences (CDSs). Prokaryotes can additionally use ribosome-binding sites (RBSs). To date, metabolic engineering has predominantly used promoters and RBSs to express the proteins encoded by metabolic pathway-encoding constructs.

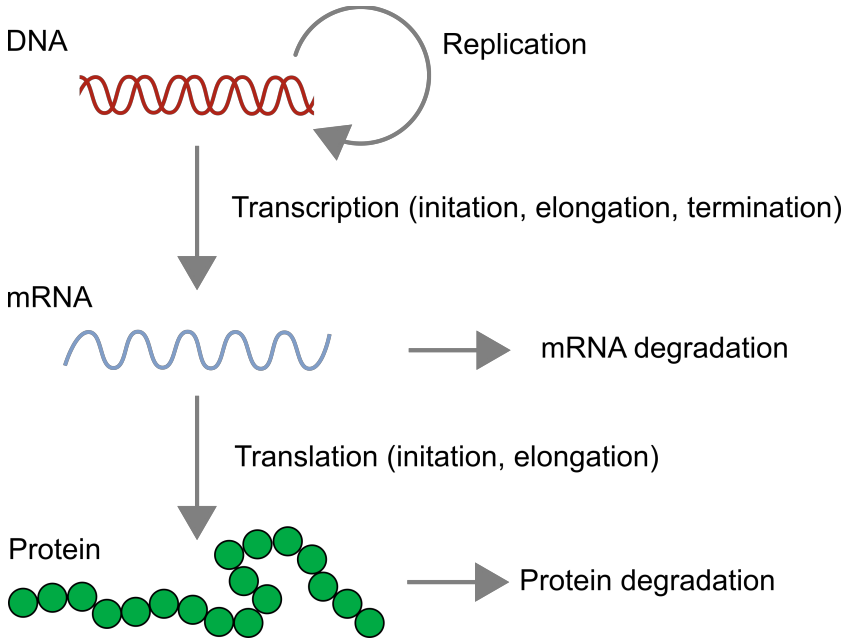


Figure 1.1. Central dogma of molecular biology. Protein concentration is affected by many factors. DNA replication allows more DNA to be generated. Transcription initiation, elongation, termination regulate the transcription of DNA to mRNA. Translation initiation and elongation control the translation of mRNA to protein. mRNA and proteins can degrade over time.

Transcription in prokaryotes is completely dependent on the multi-subunit RNA polymerase (RNAP). To initiate transcription of a certain gene, the RNAP must first interact with a sigma factor to form the RNAP holoenzyme. The sigma factor ensures that promoter DNA sequences are specifically recognised and therefore the RNAP holoenzyme is correctly positioned at the target promoter (81). As a result, promoters control transcription of all genes. Transcription initiation is predominately dependent of the RNAP holoenzyme, specifically the sigma factor, recognising the promoter sequence. For this there are two principal elements, the -10 and the -35 hexamers,

which are recognised by region 2.4 and 4.2 specifically of the sigma factor subunit (81), however other regions of the promoter sequence contribute to the RNAP holoenzyme binding such as the extended -10 element and the UP element (81). Once the RNAP holoenzyme is tightly bound to the promoter sequence, transcription initiates. Promoters control the expression of every gene, hence their frequent used in metabolic pathway-encoding constructs. Furthermore, by varying the sequence of the promoter, it is possible to change the DNA-binding affinity of the sigma factor subunit of the RNAP holoenzyme to the promoter DNA sequence, which in turn changes the rate of transcription initiation (82).

Following transcription initiation and elongation, transcriptional termination can take place, in prokaryotes and this can be controlled by transcriptional terminators. There are two types of transcriptional terminators: i) Rho-dependent terminators, whereby the homohexameric protein Rho binds to a Rho-utilisation site (*rut*) on the mRNA transcript and then the Rho protein translocates along the mRNA transcript using energy created by its ATPase during the Rho-mRNA binding. Once Rho encounters a polymerase pause site, it uses its helicase activity to dissociate the RNA:DNA hybrid (83); and ii) Rho-independent terminators; these termination sites contain two key features, a GC-rich self-complementary region with a few intervening bases and secondly, a series of U residues downstream of the self-complementary region, both of which are transcribed to the mRNA. When the self-complementary sequence is transcribed it forms a hairpin loop secondary structure, which interacts with the RNAP, causing it to pause. The pause most likely induces a conformational change in the RNAP, which along with the extremely unstable binding between the U residues of the mRNA transcript and template DNA cause the shearing between the two and therefore the release of the mRNA transcript (12). Transcriptional terminators are therefore popular expression control parts, which allow greater independence of protein concentration. Furthermore, they can prevent the expression of adjacent downstream genes, or be placed upstream of a gene to prevent any background transcription from upstream genes (84).

Translation initiation is one of the major steps in not only translation but also gene expression, as it can vary protein concentrations by several orders of magnitude (85). To initiate translation, a ribosome docks with the mRNA transcript, where the 16S ribosomal RNA (rRNA) within the 30S subunit of the ribosome binds to a specific sequence in the 5'-untranslated region (UTR), known as the ribosome-binding site (RBS). Within the RBS, there is a short sequence, termed the Shine-Dalgarno (SD)

sequence that is complementary to the 3' end of the 16S rRNA subunit (86). As mentioned above, variation in translation initiation rates can cause massive differences to protein concentrations. There are three main factors that can affect the rate of translation initiation: i) secondary structures from the global folding of transcribed mRNAs can impede the binding between the mRNA transcript and the ribosome during translation initiation (87); ii) regional folding of nucleotides in the RBS region of the mRNA transcript, whereby the sequence upstream of the start codon, of approximately 30 base pairs (ribosome docking site (RDS)) must be unfolded and therefore exposed to allow the ribosome recognition sequence to bind; and iii) efficiency of the ribosome binding to the RBS, as determined by the binding affinities between the SD sequence and the 16S rRNA anti-SD sequence, which are affected by the RBS sequence itself as well as the spacing between the SD sequence and the start codon (88). It is as a result of this last factor that RBSs can be used to vary protein expression, hence their frequent use in designing metabolic pathway-encoding constructs.

Translation elongation follows translation initiation and encodes the synthesis of proteins from the mRNA transcript. It has been shown that codon usage can vary protein concentration by at least 40-fold (89). Variation in the rate of translation elongation is predominantly caused by codon bias. There are 61 codons in the genetic code that encode 20 amino acids and three codons that encode stop signals. 18 of the 20 amino acids are encoded by multiple synonymous codons, however in the mid 1980s the comparative sequence analysis of synonymous codons in genes of different organisms revealed that each organism preferred a different set of codons to the others, also known as codon bias (90). Furthermore, not only does the codon bias vary between organisms but also for different genes within one genome (91). Therefore, when more frequently used codons are used and recognised by more abundant tRNAs, this will result in faster translation elongation rates. This was demonstrated when coding sequences were recoded to use more frequent codons, which at least in most cases, resulted in increased heterologous gene expression (89, 92). With different organisms having different codon biases, heterologous genes are often recoded before their incorporation into metabolic pathway-encoding constructs. They are recoded as a means to maximise the translation elongation rate of the gene in the new host. This is often carried out using metrics such as the codon adaptation index (CAI), which is a measure of the frequency of optimal codons in a given organism based on the codon usage frequency in a reference set of highly expressed genes (93).

1.3.2 Traditional design of metabolic pathway encoding constructs, the pre-synthetic biology era

While the rationale approaches described in section **1.2.2.1** have often proved a strong rational for a specific gene(s) to be targeted, the level at which those genes should be expressed at has traditionally been an "all or nothing" approach. In this approach genes have typically been knocked out altogether or strongly over-expressed, rather than the increasingly more common accepted approach of fine-tuning expression levels (introduced in section **1.3.3.2**).

The most commonly-used approach to knockout a target gene involves disrupting it with an antibiotic marker (51, 56). The most frequently used design to overexpress a gene involves replacing the native promoter of a heterologous gene with a promoter (constitutive or inducible) specific to the host organism in which it is being expressed. Rather than using a defined RBS, the target gene was typically amplified with the DNA sequence upstream of the start codon, which was presumed to contain the RBS. When a pathway required the expression of multiple genes, they were typically arranged as an operon, in which multiple genes were expressed by a single promoter (94). With the entire pathway typically amplified as one DNA fragment and cloned downstream of a single promoter native to the host organism, as before, this design does not use defined RBSs, but rather assumes the RBS for each gene falls within the amplified DNA fragment. Amplifying the metabolic pathway as one fragment was most likely the most suitable approach prior to the recent and rapid advance in DNA assembly methods and DNA synthesis. The metabolic pathways are typically expressed on a plasmid, which provides a further expression control part, plasmid copy number (high-, medium-, low-copy number), that could be used to alter gene expression (94).

In this era of metabolic engineering, the organisms *Escherichia coli* (*E. coli*) and *Saccharomyces cerevisiae* (*S. cerevisiae*) were predominately used (as further discussed in section **1.4.1**). In *E. coli*, the native inducible promoters P_{lac} (95), P_{lacUV5} (96), P_{BAD} (97), P_{trp} (98) and the hybrid promoter P_{trc} were most frequently used to express homo- or heterologous genes. Some notable examples include using P_{lac} to overexpress isopentenyl diphosphate isomerase and the carotenoid biosynthetic pathway (six genes), which produced a range of carotenoids, including lycopene, β-carotene and zeaxanthin (99). In this carotenoid pathway, the seven genes were split into three operons on three medium-copy plasmids (p15A) (99). P_{lacUV5} has been used to express operons on low-copy plasmids that encode metabolic pathways that produce aromatic compounds such as tryptophan (100, 101). P_{trc} was used in the early

design of many operon-based metabolic pathway-encoding constructs to overexpression genes, such as *dhaB* and *dhaT* to produce 1,3-propanediol in *E. coli* (102) and *gldA* and *dhaB* to produce 1,2-propanediol (103).

In *S. cerevisiae*, strong native promoters such as the constitutive promoter P_{ADH1} has been used to express heterologous genes, such as α-amylase from *Bacillus amyloliquefaciens* (104), endo-β-1,3-1,4-glucanase from *B. subtilis* (105) and β-galactosidase gene of *Aspergillus niger* (106). Other native promoters include P_{GAL7}, P_{PGK}, P_{GDP}, which were used along with P_{ADH1} to express *crtB*, *crtE*, *crtI* and *crtY*, respectively to produce carotenoids, such as lycopene and β-carotene (107).

As highlighted above, the majority of metabolic pathways are still constructed *ad hoc* and require significant amounts of trial and error, substituting expression control parts to ascertain some level of pathway performance. On the turn of the millennium, the field of synthetic biology emerged and further developed the way in which metabolic pathway-encoding constructs were designed. Synthetic biology introduced a more refined outlook on metabolic pathway design, in which pathway components weren't simply deleted or overexpressed, but were fine-tuned to an optimal expression level.

1.3.3 Synthetic Biology aided design of metabolic pathway encoding constructs

1.3.3.1 Emergence of Synthetic Biology

Synthetic biology is a new field of research that emerged in the early 2000s with the goal of rationally redesigning existing biological systems and engineering novel biological systems to generate new chemicals, improve human health and address environmental issues. The field builds on advances in molecular biology, cell biology, systems biology, metabolic engineering and genetic engineering but aims to frame this knowledge within an engineering-inspired approach (108). The engineering-inspired approaches have led to key goals in synthetic biology, such as the modularisation of biological systems and standardisation and abstraction of expression control parts, which could be used to model, tune, rearrange and assemble novel or redesigned biological systems that can perform specific functions (109). As a result, synthetic biology research has become increasingly applied in metabolic engineering for improving the design and performance of metabolic pathways as well as in other fields, such as therapeutics and materials (108, 109).

Initially, synthetic biology focused on generating standardised expression control parts that were then used to express genetic systems, such as metabolic pathways (110) and to generate devices, such as genetic switches, cascades, pulse generators and oscillators (108, 109), which are able to control specified behaviours, including regulation of gene expression, protein function and metabolism. Having developed basic expression control parts and devices, synthetic biology is moving forward and focusing on integrating such genetic tools in larger biological systems. However, the incorporation of these genetic tools into larger biological systems remains a challenge as complete abstraction of genetic parts is yet to be achieved.

1.3.3.2 Enhancing the design of metabolic pathway-encoding constructs with Synthetic Biology

Synthetic biology approaches have been applied to metabolic engineering in order to overcome the “all or nothing” approach to gene deletion or overexpression, with fine-tuning of protein expression levels in a metabolic pathway to obtain optimal pathway performance. The optimal performance of a metabolic pathway generally requires the intricate balance of protein expression levels (111). Natural metabolic pathway’s and the sequences encoding them, are optimised by natural selection to efficiently produce the intended product without any deleterious effects on the cell (112). However, heterologous or synthetic metabolic pathways have not been optimised by natural selection and can therefore often be non-optimal and result in one or more of the following issues: i) poor flux through the pathway to the intended product; ii) the formation of by-products; iii) the accumulation of intermediates (which may be toxic); and iv) excessive enzyme overexpression (113, 114) (Figure 1.2), which can all contribute to deleterious effects on the cell (113, 115-118). These issues largely depend on the profile of absolute and relative concentrations of enzymes in the pathway. Synthetic biologists have devised a number of design approaches that can be implemented to optimise a pathway’s enzyme concentration profile, some of which are discussed below.

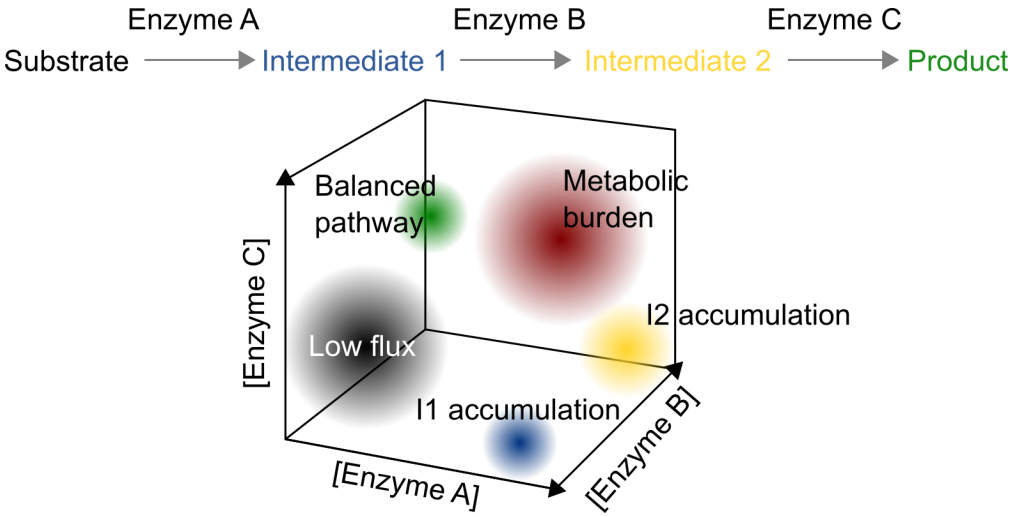


Figure 1.2. Optimal balance of pathway proteins sustains optimal pathway performance. The optimal performance of this three-enzyme pathway requires specific expression of each gene (shown in green). Small deviations in the expression level of one or more of the enzymes can result in poor pathway performance, such as low flux (shown in black), and accumulations of intermediates (shown in blue and yellow), and/or metabolic burden (shown in red). Figure Adapted from Jeschek *et al.* (119).

1.3.3.2.1 Expression control parts developed

Prior to the emergence of synthetic biology, only a few expression control parts were available and even fewer had been characterised (as described in section **1.3.2**). Those that were available were predominantly used in organisms such as *E. coli* and *S. cerevisiae*. Furthermore, the expression control parts were not suitable for fine-tuning protein expression levels. As such, following the emergence of synthetic biology, a huge effort was made to increase the number of available well characterised expression control parts that would be suitable for fine-tuning. Initially, the vast majority of effort was spent improving the genetic toolboxes for *E. coli* and *S. cerevisiae*, but in more recent years attention has turned to developing the genetic toolbox for alternative organisms.

To allow the expression control parts to be used to fine-tune protein concentrations on a transcriptional level, libraries of constitutive synthetic promoters were developed (40, 82, 120-122) that span a wide range of steps, increasing over small and even increments. Such promoter libraries were traditionally generated by ‘up’ and ‘down’ mutations and in more recent years they have been generated using random and semi-rational sequence randomisation. To allow transcriptional expression to be switched on and tuned using inducer concentration, rather than the continual expression that constitutive promoters provide, inducible promoters have been developed (123-126).

Most of the inducible promoters used have been taken from *E. coli* regulatory expression systems such as P_{rhaBAD}, P_{tet}, P_{lac} and P_{araBAD} and more recently, libraries of inducible promoters have been generated using mutagenesis and then screened for improved properties (125, 126). Ideal properties of an inducible promoter system include: i) not "leaking" in the absence of an inducer; ii) non-toxic, non-metabolisable and readily available inducer molecules that are stable under experimental conditions; and iii) inducer molecules providing a linear response in expression to inducer concentration, allowing fine-tuning of induction over a wide range. Transcriptional terminators are now abundant for *E. coli*, following an exhaustive characterisation of 582 natural and synthetic terminators (127). This allows genes in a pathway to be configured as monocistronic expression units, which have advantages over operons, particularly as they allow greater independence of expression units and more fine-grained control of expression, as both promoters and RBSs can be varied for each unit.

Tools have also been developed that can be used to fine-tune protein concentrations on a translational level. The most notable is the RBS calculator, an *in silico* tool that can be used to predict the translational initiation rates (T.I.R) for a protein with any RBS sequence and rationally forward-engineer an RBS sequence to give a desired protein concentration (85). The RBS calculator uses a statistical thermodynamic model to determine how well a specific ribosome will bind to specific mRNA sequences. The model considers key molecular interactions that affect the rate of translation initiation and then uses these to calculate the Gibbs free energy of the ribosome binding, which is subsequently linked to the T.I.R of a CDS's proteins. To design an RBS sequence with a user-defined T.I.R, the forward engineering function of the RBS calculator uses a stochastic optimisation method in combination with the thermodynamic model. A recent addition to the RBS calculator is the RBS library function that now allows it to generate a library of synthetic RBSs which equally spans between a user-defined range of protein concentrations. In addition to the RBS calculator, two more equivalent tools have been developed, the UTR designer (128) and the RBS designer (129). With the decreasing cost of *de novo* DNA synthesis (130), it has been possible to codon optimise heterologous genes (with respect to the host organism), with many automated codon optimisation algorithms having been developed, such as OPTIMIZER (131). The most common approach to codon optimisation involves maximising a gene's CAI to match that of the expression host, which has become increasing possible with the number of sequenced genomes now publicly available at the National Center for

Biotechnology Information. Furthermore, codon optimisation allows other features, such as restriction sites that might be needed for DNA assembly to be removed.

On a post-translational level, protein degradation can be controlled by tagging proteins with degradation targeting peptides that impart different degradation dynamics (132).

1.3.3.2.2 Rational design of MPEC

Despite the understanding that enzyme concentration profiles within a metabolic pathway can be very sensitive to changes in expression levels, metabolic pathway-encoding constructs are still frequently designed to simply over-express genes in a pathway using either inducible promoters (133-135) or strong constitutive promoters and/or strong RBSs (136, 137). In general, these designs have led to broken or poorly performing pathways (53, 116). One approach that synthetic biologists have used to optimise the performance of the pathway uses iterative cycles to rationally substitute expression control parts in metabolic pathway-encoding constructs, after which the new construct is characterised and analysed. This approach mirrors the Design-Build-Test-Learn (DBTL) cycle that is traditionally used in computational and engineering sciences (138). In each iteration of the DBTL cycle, expression control parts are rationally substituted for alternatives in an attempt to express the enzymes closer to their optimal expression level. However, with the limited ability to predict the expression level that an enzyme will be expressed at, this is largely a trial and error process. It is challenging to express proteins at specific concentrations because: i) the rate of protein production (mol second^{-1}) for any given expression control parts is not known, rather the strength is given as a relative expression unit; ii) no accurate rate for protein production of any given CDS is available; iii) even if an accurate rate of protein production for expression control part was available, the rates will change when used in different genetic contexts, due to context dependence (further discussed in section **4.1**); and iv) the right theoretical framework is not available to combine all of the above.

A more recent study described a new rational design strategy that reduces the cellular burden. Burden is a phenomenon used to describe the deleterious effects that over-expressing native and/or heterologous genes have on the cell. These effects include slower growth rates and lower protein yields, which can result from the unnatural and extra load that expression of these genes can have on the cell (117). It was shown that the depletion of translational and transcriptional resources (e.g. ribosomes and RNA

polymerases, respectively) are the major causes of burden, but careful choice of promoters and RBSs can be used to reduce the metabolic burden (114). This work described and demonstrated that the novel design strategy could be used to decrease burden, achieved by using a strong promoter and weak RBS, in an attempt to avoid 'overinitiation' of translation, and developed a capacity monitor that can be used to measure the metabolic burden of a pathway (114). This work has since been further developed into a cell-free prediction tool that can analyse and use the cost of gene expression in a pathway to predict the metabolic burden that the pathway would have on the cell, as was demonstrated with the luciferase and β-carotene pathways (139).

Another approach to designing a metabolic pathway-encoding construct is called refactoring. The term "refactor" originated in software development, whereby a code underlying a program is rewritten to achieve a given goal, such as stability, without changing its function (140). This approach has since been used by synthetic biologists to 'clean up' a natural genetic system (141, 142). By clean up, the goal of refactoring is to create a completely defined and modular system, which prevents adjacent elements from having undesired effects on one another, which in turn allows substitution of expression control parts to be used to modify the expression of only the intended individual parts. This is achieved by systematically removing all of the native regulation of the system and replacing it with well-characterised synthetic expression control parts (142). The end result is a genetic system with a different DNA sequence but one that encodes the same function. This approach was first applied in biology to simplify a T7 phage genome by redesigning known genetic elements to be individually changeable by restriction digest (141).

In one example, the *nif* gene cluster was refactored (142). In this case, refactoring started with removing the systems complexity and native regulation, such as eliminating all of the noncoding DNA and regulatory genes, leaving only the essential genes. To remove internal regulation within the essential genes, such as operators, promoters, mRNA secondary structure, pause sites, methylation sites and codon regulation, they were recoded to produce the same protein but from a DNA sequence that differed to the wild-type sequence. The redundancy of the genetic code makes this possible, as there are many alternative coding sequences that could encode a given protein. Having recoded the essential genes to the genetic system, they were organised into well-defined and modular systems, using well characterised synthetic promoters, RBSs and terminators to assemble each gene in the system. The end result was a refactored genetic system that encodes the same function as the native system,

but rather than using its native regulation, it uses well-characterised synthetic expression control parts. A refactored system enables simplified “large-scale” part substitution, which is essential for tuning a genetic systems performance or for transferring to a different species, where expression control parts specific to that organism are required, which is further described in section **1.6**.

Since refactoring the T7 phage genome, refactoring has been applied to a number of genetic systems, such as the *Klebsiella oxytoca*’s nitrogen fixation (*nif*) gene cluster that was refactored, simplifying the native gene cluster from its 20 genes encoded in seven operons to 16 genes encoded in four operons that are controlled by a single controller construct (Figure 1.3) (142). Following this work, the silent spectinabilin gene cluster in *Streptomyces orinoci* (143), the six-gene photosystem II core in the chloroplast in *Chlamydomonas reinhardtii* (144), the Embden-Meyerhof-Parnas (EMP) pathway in *E. coli* (145) and the pathways for light sensing proteins in *E. coli* (146) were all refactored.

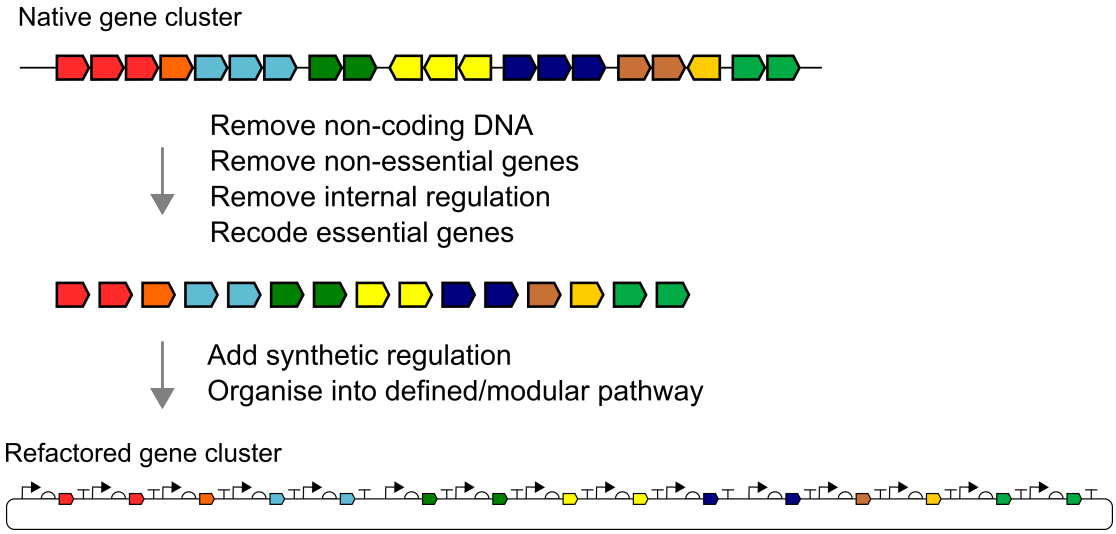


Figure 1.3. The process of refactoring a gene cluster. The illustrative ‘native’ gene cluster is composed of 20 genes. First, non-essential genes, internal regulation and non-coding DNA are removed, and the essential genes are recoded, this process removes native complexity. The recoded essential genes are then organised into defined and modular pathways with synthetic expression control parts to regulate their expression.

Furthermore, tools have been developed to help re-design genetic systems, such as Genome Calligrapher, which can be used to refactor bacterial chromosomes for *de novo* DNA synthesis (147), as was demonstrated with a 168 kb of essential cell-cycle genes for *Caulobacter crescentus*. Plug-and-Play frameworks have also been

developed for pathway refactoring in *E. coli* and *S. cerevisiae* (148), which were used to refactor carotenoid biosynthesis pathways for both organisms.

Even if an optimal enzyme concentration profile for a particular pathway in a given host organism has been established, whether it's by metabolic modelling methods or *in vivo* analysis such as the tolerance analysis carried out for the genes in the *nif* gene cluster (142), it remains challenging to rationally design an individual DNA sequence to express a protein of interest to a specific target concentration (142), for reasons which have already been specified. This is a problem which increases in difficulty with increasing numbers of proteins in a multi-protein expression construct (142, 149). As a result, when rationally designing a metabolic pathway-encoding construct many iterations of the DBTL cycle are often required.

1.3.3.2.3 Combinatorial pathway optimisation

The iterative rational design approaches described above design constructs using individual expression control parts for each gene in a pathway. Such designs typically require multiple iterations of the DBTL cycle to optimise the pathways performance, by substituting parts until pathway proteins are expressed at, or near to, an optimal concentration. An alternative to this approach is combinatorial pathway implementation and optimisation, which generates combinatorial libraries of metabolic pathways, rather than an individual metabolic pathway-encoding construct. Pathway libraries are created by varying, independently or partially-independently, the expression of each protein in a pathway, by i) the choice of CDS that encodes the proteins; and ii) using mixtures of expression control parts instead of individual expression control parts (149-151). The design of the mixtures (of expression control parts), in terms of the number of parts and the distribution of their expression levels, will determine the coverage and distribution of the design space for the given protein, or multidimensional design space when multiple proteins in a pathway are being combinatorially optimised in parallel. The total design space can be very large for any given pathway. A combinatorial library is typically a sparse sampling of the total design space; the distribution of which is determined by the choices of mixtures of expression control parts (Figure 1.4). Combinatorial pathway optimisation provides a tunable search strategy effective for large design spaces, which is primarily limited by the throughput of screening once host cells are transformed with the library. Once functional metabolic pathway-encoding construct designs have been identified, the pathways performance can be further improved by refining the search space around

local optima. Refining the search space can be rationally applied by using more fine-grained rounds of combinatorial searches around local optima (119), and/or by manually-designed substitutions of specific expression control parts.

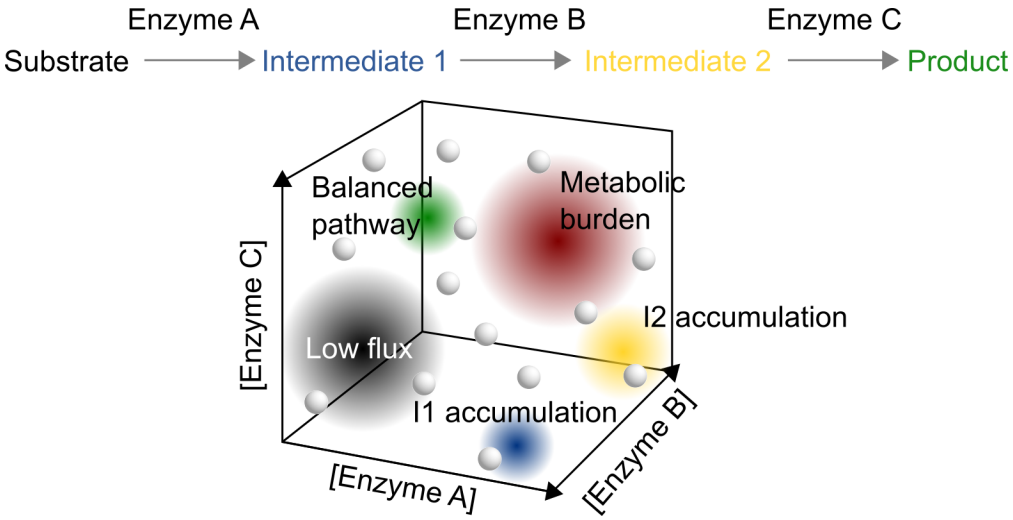


Figure 1.4. Combinatorial assemblies provide a sparse sampling of the design space. Combinatorial assembly simultaneously varies the expression levels of enzymes, typically by varying promoter and RBS strengths. The number and distribution of parts used determines the sampling of a design space. Typically, combinatorial assemblies offer a sparse sampling of the design space.

Alternatively, combinatorial libraries can be semi-rationally designed in the first place to reduce the combinatorial explosion of experimental effort when it comes to screening large pathway library, such as RedLibs (Reduced Libraries), an algorithm that rationally designs smart combinatorial libraries, which was shown to improve violacein production in *E. coli* (119). Another example that was recently published, used the Design-of-experiments (DoE) methodology to guide the iterative design of combinatorial libraries. The relevant information was obtained while sampling a small proportion of the pathway library (152). Over the course of three iterations of DoE guided combinatorial pathway design, the production of itaconic acid was increased 4-fold in *S. cerevisiae* (152).

Combinatorial pathway optimisation is becoming an increasingly popular strategy for improving product titres in metabolic engineering. To date, it has been predominately used to optimise pathways in *E. coli* and *S. cerevisiae*. The combinatorial pathway optimisation of xylose utilisation and ethanol production in *S. cerevisiae* (153) and pinocembrin production in *E. coli* (154) were improved by combinatorially varying the homologous genes that encode enzymes from various organisms. Varying the protein concentrations in a pathway by using mixtures of expression control parts (promoters,

RBSs, terminators, copy number) have also been used to improve the production of products, such as violacein (119, 137, 155), lycopene (150), neurosporene (156), taxadiene (55), fatty acids (157) and nitrogen fixation (149) in *E. coli*, and ethanol production in *S. cerevisiae* (66).

Like the semi-rational approaches used in metabolic engineering (as described in section **1.2.2.2**), the combinatorial pathway optimisation of metabolic pathway-encoding constructs requires little *a priori* knowledge of suitable enzyme concentrations required for a well-functioning pathway. Nor does it require an understanding of the resultant protein concentrations that will be generated from the combinations of expression control parts and CDSs. If *a priori* knowledge of the suitable expression levels of given enzymes is available, it can be used to refine combinatorial optimisation around specific expression levels. To implement combinatorial pathway optimisation, this requires: i) knowledge of the target metabolic pathway and the genes that encode it; ii) suitable expression control parts that span an appropriate range of strengths; and iii) a suitable modular multi-part DNA assembly method.

1.3.4 DNA assembly

DNA assembly is a key part of constructing metabolic pathway-encoding constructs, as well as other genetic expression systems and even whole chromosomes (158). For over 40 years, type II restriction enzymes and DNA ligases have been used to “cut and paste” DNA fragments together and have been the predominant method used for molecular cloning (159). Briefly, type II restriction enzymes are first used to digest DNA fragments, which generates sticky ends and then in a separate second reaction DNA ligase is used to assemble together DNA fragments with compatible sticky ends.

Since then, the field of DNA assembly has rapidly advanced, with many different assembly methods having been developed. This has largely been prompted by the necessity to assemble metabolic pathway-encoding constructs that are composed of many different components, such as those previously described. Advantageously, many of these methods have been developed, so unlike the original “cut and paste” type II restriction enzymes approach, multiple DNA fragments can be assembled in a single step, which are sometimes referred to as a ‘one-pot’ reaction (160). As a result, building a bacterial plasmid, which in the 1970s used to be an endeavour worthy of a

multi-author high-profile publication (161), today is routinely performed by undergraduate students (162).

These multi-part DNA assembly methods can mostly be categorised into one of two types: i) modular approaches, in which a framework of specified steps, rules and design constraints including pre-defined module formats, allows highly efficient multi-part assembly of individual constructs or designed combinatorial mixtures of constructs. However, these benefits come at the cost of fixed 'scar' sequences at junctions between parts in the assembled construct and a substantial upfront investment of effort and resources to obtain a required set of suitable assembly vectors, and to prepare sequences of interest in the appropriate format. ii) Bespoke approaches, which are flexible, require little upfront effort to establish or plan future steps, and typically involve an *ad hoc* approach to the design of cloning strategies which is intuitively more similar to conventional cloning approaches. However, these non-modular approaches depend on sequence overlaps to direct assembly, so they have a greater need for custom oligonucleotide primers (typically one pair per junction between parts) and sequence verification (due to PCR steps) and are less suitable for combinatorial assembly due to lower efficiency (meaning fewer clones are obtained following transformation so smaller libraries are generated) and greater potential for bias particularly due to repetitive sequences.

Modular multi-part DNA assembly methods include Golden Gate assembly (163-165) (and variants (137, 165-174)) which is further described in section **5.1**, BASIC (175, 176), BioBrick assembly (177) (and variants such as BglBrick (178, 179)) and Gateway cloning (180-182).

Bespoke multi-part DNA assembly methods include Gibson Assembly (183), AQUA cloning (184), Twin Primer Assembly (185), ligase cycling reaction (186), SLIC (187), SLiCE (188), overlap extension PCR (189), USER (190), CLIVA (191) and CPEC (192). Multi-part DNA assembly has quickly become important in metabolic engineering and synthetic biology, enabling an increase in the scale, scope and speed of studies (160).

1.4 Microbial hosts used in Metabolic Engineering

1.4.1 Model organisms

The majority of metabolic engineering applications have to date made use of a small number of organisms, with the principal prokaryotic organism being *E. coli* and the predominant eukaryote being *S. cerevisiae*, a direct result of the highly tractable nature of both of these organisms. *E. coli* and *S. cerevisiae* are easily cultured in the laboratory, both have rapid growth rates and both have a wide range of molecular biology techniques and synthetic biology tools available for their genetic manipulation. Furthermore, their genomes were among those first sequenced and publicly available, which enabled development of bioinformatic tools to complement the research. Moreover, both of these organisms have been used for fundamental biological studies, which has led to a wealth of knowledge about their physiology, metabolism, and genetics, which further aided their development as useful organisms. As a result, both *E. coli* and *S. cerevisiae* are often, and hereafter in this project, referred to as model organisms, and have been widely used and engineered to produce a whole range of products, which are further outlined below.

1.4.1.1 Escherichia coli

E. coli is a gram-negative, facultatively anaerobic bacterium (Figure 1.5A) originally discovered in the human colon in 1885 by German bacteriologist Theodor Escherich (193). Since then, *E. coli* has become the most extensively studied, developed and well characterised organism on earth. *E. coli*, the 'workhorse' of most molecular and synthetic-biology based laboratories, when grown under anaerobic fermentation conditions, naturally produces acetate, ethanol, lactate, formate and succinate (194). *E. coli* has been engineered to improve the production of not only these products (195), but also a wide range of many others not produced naturally, which include: amino acids at an industrial scale (196), a process traditionally carried out in the natural producer *Corynebacterium glutamicum*; producing similar levels of n-butanol to *Clostridia* (197, 198); to producing many other products, such as isobutanol (199), isoprene (200), short-chain fatty acids (201), hydrocarbons (201), biomaterials (202), carotenoids (115, 150, 203-207) and many more (208). *E. coli* are heterotrophic bacteria and therefore cannot produce their own energy; instead they obtain free energy through the oxidation of sugar-based feedstocks, most typically glucose, to produce these products (209).

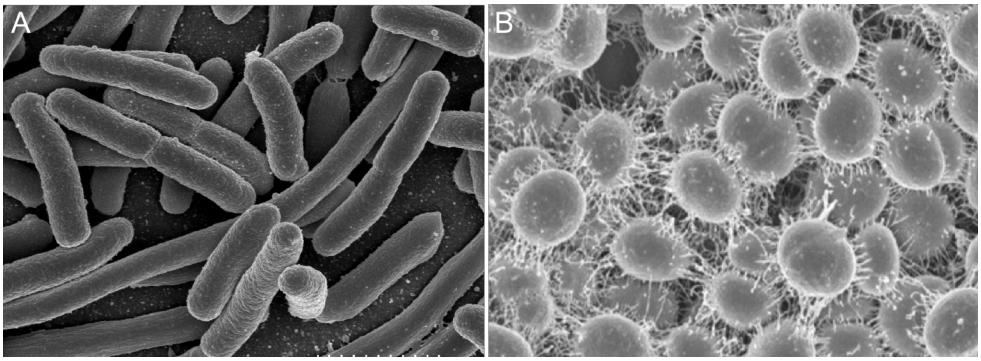


Figure 1.5. Scanning electron micrographs of bacteria used in this study. (A) wild-type *Escherichia coli* (Image credit: National Institute of Allergy and Infectious Diseases (NIAID), Flickr). (B) wild-type *Synechocystis* sp. PCC 6803 (Image credit: Poliquin *et al*. (210).

1.4.1.2 Saccharomyces cerevisiae

S. cerevisiae, a species of yeast, commonly known as baker's yeast and brewer's yeast because of its use in baking and brewing, respectively. *S. cerevisiae* has been extensively used in the food industry as baker's and brewer's yeast due to its combined ability to produce a high yield of ethanol (homo-ethanol fermentation (211)), its high tolerance to ethanol (in traditionally-improved strains (211, 212)) and its palatable taste. Another common application of *S. cerevisiae* has been to commercially produce bioethanol (213), an increasingly used biofuel (214). As a result, *S. cerevisiae* has been one of the most extensively studied eukaryotic organisms. In more recent years attention has turned from producing ethanol to engineering the production of a wide range of products, including alcohols, such as n-butanol (215) and isobutanol (216), organic acids, such as lactic acid (217), succinic acid (precursor for polyesters and alkydresins) (218) and *cis*,*cis*-muconic acid (building block in chemical industry) (219, 220) and terpenes (201). One of the most notable product is artemisinic acid, a precursor of the critically-important antimalarial agent, artemisinin (221). *S. cerevisiae* is a heterotrophic organism, which therefore, like *E. coli*, uses sugar-based feedstocks to grow on and to subsequently produce these products from.

1.4.1.3 The need for an alternative substrate

With the ever-dwindling supply of fossil fuels, associated pollution and the impact of climate change on the planet, a renewable and sustainable substrate for producing fuels and chemicals is desperately needed to reduce carbon emissions and fossil fuel consumption. As such, in a bid to overcome the issues surrounding climate change, in 2015 the Paris Agreement was adopted under the United Nations Framework

Convention on Climate Change, with the aim of limiting global warming to well below 2 °C (2).

To date, the vast majority of commercially available chemicals that have been microbially produced are from sugar feedstocks, which are derived from edible crops such as corn and sugarcane (222). With Earth's population continuing to increase, our dependency on edible crops will also increase. As such, obtaining sugars from edible crops introduces competition with our own food supply and is therefore not the most viable solution. Rising food prices have already been attributed to 'first-generation' biofuels that are derived from edible crops (223). Promising alternatives include using renewable substrates such as lignocellulosic biomass and single carbon (C1) gases (methane (CH_4), carbon monoxide (CO) and CO_2), all of which avoid competition with our own food supply.

Lignocellulosic biomass and C1 gases are promising renewable and sustainable substrates. However, while lignocellulose is cheap and abundant, its resistance to breakdown is making the development of an economic process challenging, as expensive and highly energy intensive pretreatment processes are required (224). Using C1 gases, such as CO, CO_2 and CH_4, as a biological substrate means the carbon is captured before incorporation into the lignocellulose. C1 gases are an abundant, low cost waste product produced by a wide range of industrial (e.g., steel manufacture and power generation) and renewable processes (e.g., anaerobic digestion and agricultural waste). With fossil fuels contributing, the total global carbon emission was equivalent to approximately 9.7 gigatons in 2014 (225).

Furthermore, the production of low value products, such as solvents and transportation fuels derived from sugar-based feedstocks, needs to be much more efficient, as the high cost of sugar-based feedstocks can make up half of the total production cost, resulting in smaller profit margins (226). For this reason, a cheaper feedstock such as CO_2 and/or CH_4 would be more attractive (227). While high value products, such as pharmaceuticals, amino acids, vitamins and flavours, have a bigger profit margin when produced from sugar feedstocks, a cheaper feedstock would still be preferable.

Engineering model organisms to utilise C1 gases has proven enormously complicated due to the complexity of the engineering the biological systems necessary for their metabolism. As a result, the vast majority of the aforementioned products (sections **1.4.1.1** and **1.4.1.2**) were produced from glucose, an expensive simple sugar that

competes with our food supply. There has been some success engineering model organisms to grow on alternative substrates, for example, *S. cerevisiae* and *E. coli* have been engineered to grow on a range of sugars including, xylose, arabinose, galactose and glucose (217, 228, 229), which are the basic components of lignocellulose. To produce products of interest however, in these cases, the lignocellulose would still require the highly energy intensive pretreatment to break it down into the simpler components mentioned above (224). More notably, CO_2 fixation has recently been demonstrated in *E. coli*, where the cyanobacterial Calvin-Benson-Bassham cycle and carbon concentrating mechanism were engineered and provided a CO_2-fixation rate comparable to that of the photoautotrophic cyanobacteria and algae (230). In a second study, the expression of a well formed and functional α-carboxysome engineered in *E. coli* provides further building blocks for heterotrophic CO_2 fixation (231). It is also worth stating that *E. coli* naturally grows on glycerol, which is a cheap waste product of biodiesel production (232).

1.4.2 Advantages of using non-model organisms as an alternative host

While the model organisms *E. coli* and *S. cerevisiae* have been used mostly as the hosts for metabolic engineering applications, many other diverse organisms that are have also attracted attention as potential alternatives. This is primarily because these organisms often have phenotypes that are of industrial relevance, such as: i) growing in specific environments, for example thermophiles, anaerobes or halophiles (233-235), which are beneficial for reducing mesophilic contamination, increasing enzymatic rates to increase productivities and can simplify recovery of volatile products; ii) using a range of different substrates, such as CO, CO_2, CH_4, syngas (CO and H_2), lignocellulose and its component sugars, and inorganic energy sources, which are renewable and sustainable (236-238); or iii) producing and tolerating large amounts of desired chemical products, such as solventogenic bacteria (6, 7). However, these organisms are often harder to work with compared to model organisms, as they often lack the tractability, fast growth-rates and/or genetic tools of *E. coli* and *S. cerevisiae*; and as such they are hereafter referred to in this project as non-model organisms.

The value of non-model organisms could then be further improved by introducing heterologous or synthetic metabolic pathways, which allow them to produce non-native products or improve the production of products that they naturally produce. Considering the few examples of engineering model organisms to grow on renewable and sustainable substrates, it is more likely that using a non-model organism that has

evolved through natural selection to grow on such a substrate, is a better solution. As a result, non-model organisms have been used to produce a wide range of natural and non-native products, but using non-model organisms does not come without challenges, which are discussed in section **1.6**.

1.4.3 Successes in non-model hosts for metabolic engineering

Non-model organisms have already been used to produce a whole host of products, many of which are naturally produced by the organisms. For example the industrial production of acetone, n-butanol and ethanol has long been performed by the acetone−butanol−ethanol (ABE) fermentation process with anaerobic *Clostridium* species (239, 240), as well as the production of the amino acids in *Corynebacterium glutamicum* (7), both of which have been engineered to improve pathway efficiencies (241, 242). The halophile *Halomonas* can naturally produce large amounts of polyhydroxyalkanoates, (243), such as polyhydroxybutyrate an environmental-friendly polyester and the organism has also been engineered to enhance its production (244-246). The facultative chemolithoautotroph, *Cupriavidus necator*, is able to naturally produce huge amounts of PHB (up to 70% total cell weight) by redirecting carbon flux when subjected to nutrient limiting conditions (247). The ability of this organism to redirect so much of its carbon to PHB has made it an attractive non-model host for non-native chemical production, based on the assumption that the non-native products could replace PHB as the carbon sink. To date, *C. necator* has been engineered to produce non-native products, for example 4.4 mg l^{-1} of alkanes and alkenes; the predominant components of petrol, diesel and jet fuel, were produced from C1 gases (248); and alcohols, such as 250 mg l^{-1} of isopropanol were produced under auxotrophic growth conditions (249).

As well as engineering *C. necator* to produce non-native products, many other non-model organisms have also been engineered to do just that. Methanotrophs have been engineered to produce products, such as carotenoids (250), propylene (251), isobutanol (252) and farnesene (253) directly from methane. *Bacillus subtilis* has been engineered to produce products such as ethanol (254), riboflavins (255) and carotenoids (256). Non-model yeast, such as *Yarrowia lipolytica* has been engineered to produce a whole range of chemicals and fuels from a range of substrates such as raw starch and lignocellulosic materials (257-260). Mammalian cells in particular Chinese hamster ovary (CHO) cells have been engineered to produce a range of commercially available biopharmaceuticals such as Humira, Enbrel and Remicade

(261). The thermophile *Geobacillus* has recently been engineered to produce ethanol at high yields at elevated temperatures by using cellulose derived sugars (262). Photoautotrophs have also been engineered to produce a range of products directly from CO_2, this is described in detail in section **1.5**.

1.5 Photoautotrophic microorganisms as alternative microbial hosts

Photoautotrophs, such as algae and cyanobacteria, pose as promising candidates for the renewable and sustainable production of chemicals and biofuels. Their photosynthetic nature gives them a unique ability to directly convert atmospheric CO_2 into chemicals and biofuels (Figure 1.6), using light and CO_2 as their sole energy and carbon source, respectively (263). CO_2 is a greenhouse gas that is rapidly increasing in atmospheric concentration (264) and therefore serves as an abundant and renewable carbon source; and light is a clean, abundant and renewable energy source. A distinguishing feature that sets photoautotrophs apart from other C1 utilising organisms is their ability to utilise light as their sole energy source. In contrast, other organisms typically rely on obtaining their energy from the oxidation of C1 substrates, such as methane, methanol and CO (236). CO_2 however, is the most oxidised form of carbon and therefore cannot serve as an energy source.

While the photosynthetic nature of plants has already been exploited to renewably produce bioethanol, this process competes with cropland and therefore our food supply (265). The large-scale culturing of algae and cyanobacteria would not require arable land, which therefore does not create competition with our own food supply, which avoids food price increases that would occur from growing crop-based microbial industries (223). Furthermore, algae and cyanobacteria grow faster and are more genetically amenable than plants (266).

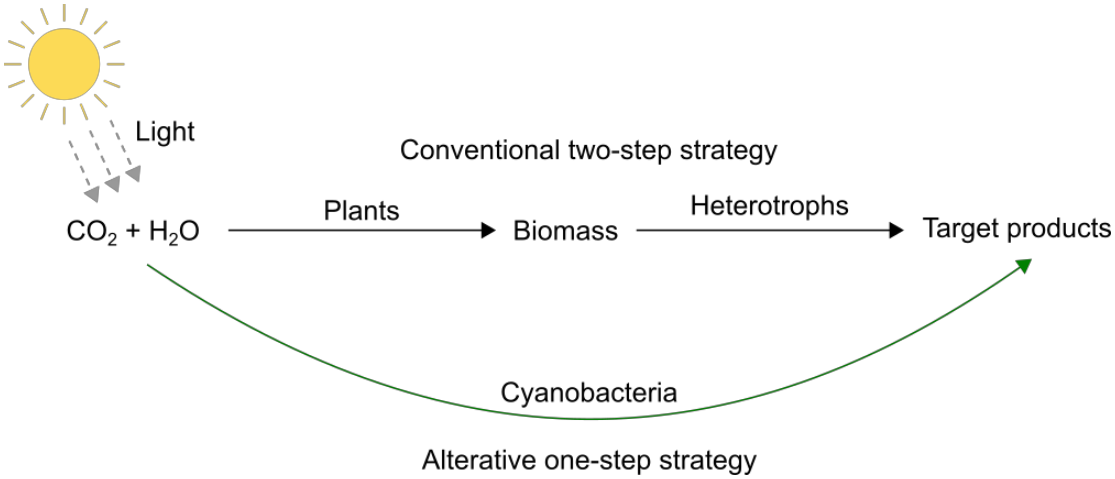


Figure 1.6. Routes for biological productions of chemicals and biofuels. Arrows indicate the carbon and energy flow from substrate to product and the organisms that utilise those carbon and energy sources.

Algae and cyanobacteria, like many non-model organisms discussed in section **1.4.3**, naturally produce many products of interest that are contained in the biomass or reside inside the cell. In algae, some of the products have been produced on a large-commercial scale around the world since the 1960s, such as β-carotene, astaxanthin, polysaccharides and polyunsaturated fatty acids (PUFA), including diacylglycerol (DAG), triacylglycerol (TAG), which are used in the production of biodiesel (267, 268). In cyanobacteria, while their natural products haven't been produced at an equivalent large-commercial scale that some of the algal products have been produced at, they do produce products such as alkanes (269), poly-3-hydroxybutyrate (270, 271), polysaccharides and many isoprenoids derivatives, like carotenoids. Recently, alkanes and alkenes have attracted a considerable amount of attention as a direct drop-in replacement for a major fraction of the current transportation fuels (272).

As the threat of climate change continues to rise, so has the need for the sustainable production of products. This has led to an increase in interest in algae and cyanobacteria, due to their ability to sustainably produce products of interest. The increased interest in these organisms has resulted in the development of many new genetic engineering approaches and synthetic biology tools and techniques. There are however no examples to date whereby engineered photosynthetic microorganisms have been used to produce chemicals or fuels on a commercial scale. There are challenges, as discussed in section **1.5.1.5**, which need to be overcome in order to make this possible. Despite this, there has been a lot of success in demonstrating that non-native products can be produced, primarily in cyanobacteria.

To date, genetic engineering approaches and genetic tools have been developed for only a few algae, most of which have been for the model microalga *Chlamydomonas reinhardtii* (273). So far, a variety of promoters have been developed, for example metal, nutrient and light inducible promoters, as well as native and synthetic constitutive promoters (273), however few other well characterised expression control parts have been developed. Compared to cyanobacteria, algae have proven to be far less amenable to genetic manipulations and therefore few non-native products have been produced. However, the non-native diterpenoids, including casbene, taxadiene, and 13R(+) manoyl oxide were recently produced by expressing diterpene synthases and certain enzymes from the MEP pathway under the native *C. reinhardtii* constitutive promoter HSP70-RBCS-i1 (274). Rather than engineering algae to produce non-native products, research has largely been directed at improving the cost-effective production of its naturally produced biofuels (biodiesel, bioethanol and/or drop-in replacements alkanes and alkenes), focusing on overexpression of acetyl-CoA carboxylase, which is involved in fatty acid synthesis and knocking out genes involved in fatty acid oxidation (275-277). It has also been demonstrated that algal biomass can be directly used as a substrate for the production of biofuels and potentially other chemicals, as demonstrated in *Spirulina* (278), by exploiting the high starch content (10 to 50%) in their biomass (279).

Compared to algae, cyanobacteria have proven much more amenable to genetic manipulation and as a result, more genetic tools have been developed and pathways have been engineered. To date, heterologous biosynthetic pathways have been engineered in cyanobacteria to produce a wide range of products, from bulk chemicals and biofuels, to high-value compounds (280, 281). Products include alcohols and diols, such as ethanol (282-286), butanol (287-289), isobutanol (290-292), 2-methyl-1-butanol (293), 2,3-Butanediol (294-298), 1,2-Propandiol (299) and 1-octanol (300); organic acids, such as L-Lactic acid (301-303), caffeic acid (304) and D-Lactic acid (305, 306); terpenoids, such as manoyl oxide (307), isoprene (308), geranyllinalool (309), limonene (310-312), α-bisabolene (311), β-Phellandrene (313, 314), β-Caryophyllene (315) and squalene (316); sugars such as sucrose (317, 318), glucose (306) and fructose (306); and many other chemicals, such as isobutyraldehyde (291), glycerol (319), D-mannitol (320), acetone (321), fatty acids (322, 323), fatty acid ethyl esters (324) and ethylene (325-327). Furthermore, cyanobacteria have been engineered to utilise sugars, such as xylose (a monomer of hemicellulose) (328) and glycerol (waste product of biodiesel production) (329).

Of these products there are seven examples with titres that are over 1 gl^{-1}, the most notable products include ethanol (284), sucrose (318) and butanediol (297), which have all been produced at over 2 gl^{-1}, with more than 50% of fixed carbon being redirected towards product formation in these three instances (263). However typically, the products have been produced at significantly lower titres. Therefore, for now they have served as more of a proof of principle in cyanobacteria rather than a commercially feasible process, but do highlight the potential of algae and cyanobacteria. Cyanobacteria are an attractive alternative over algae in industrial biotechnological applications owing to their increased growth rate and their ease of genetic manipulation (266).

1.5.1.1 The history of Cyanobacteria

Fossil records indicate that cyanobacteria, a group of Gram-negative prokaryotes, have existed for at least 2.5 billion years (Gya) (330), possibly as far back as 3.5 Gya, and their photosynthetic activity triggered one of the most dramatic events during the evolution of our planet; the oxidation of the Earth's atmosphere (331). Cyanobacteria are widely distributed throughout Earth's aquatic and terrestrial environments, including extreme habitats, such as hot springs, deserts and polar regions (332). Moreover, it is generally accepted that the chloroplasts of plants and algae are derived from a cyanobacterial ancestor (333), implicating them in eukaryotic evolution.

The photosynthetic nature of cyanobacteria have made them one of the most important and globally prolific primary producers throughout much of our planet's history (334, 335). Adding to this, many cyanobacteria (diazotrophic cyanobacteria) are also capable of fixing nitrogen and play a key role in the nitrogen cycle (330, 332). They play a huge role in the global nitrogen budget, with the cyanobacterial genus *Trichodesmium* contributing to approximately 42% of the total global nitrogen fixation of 240 Tg N_2 y^{-1} (336). They also play a huge role in the global food industry, with the rice paddies in Asia, which feed over half of the world's population, depending on cyanobacterial nitrogen fixation (337).

1.5.1.2 Cyanobacterial phylogeny

As a phylum of bacteria, cyanobacteria are monophyletic, yet exhibit a diverse range of morphologies, which has resulted in their different morphologies being traditionally categorised into one of five groups (338). Group I and II cyanobacteria are unicellular coccoids; Group I cells divide by binary fission (e.g., *Synechocystis* sp. PCC 6803),

while Group II cells undergo multiple fission to produce small, easily dispersed cells called baeocytes. Group III-V cyanobacteria consist of filamentous cells that vary in complexity, Group III for example, only have vegetative cells, but Group IV and V consist of vegetative cells that can differentiate into morphologically and structurally distinct heterocysts (339) or akinetes (340). Furthermore, Group V cyanobacteria are able to develop complicated branching patterns with their filaments.

More recent analysis of the cyanobacterial phylogenies (341, 342) has supported some of the factors used in the traditional categorisation, such as the cell differentiation and multiple fission, but not all of them. For example, unicellular and simple filamentous cyanobacteria have not been considered to constitute monophyletic groupings (341, 342) and Group V cyanobacteria are now considered a clade nested within Group IV (341).

1.5.1.3 Cyanobacterial metabolism

1.5.1.3.1 Photosynthetic machinery

Cyanobacteria have a unique metabolism, in that they are the only known prokaryotes capable of oxygenic photosynthesis, which uses light energy to synthesize ATP and NADPH, in the production of oxygen from water (343). The catalysis of cyanobacterial (and algae and plant) oxygenic photosynthesis utilises four defined membrane complexes (344); photosystem I (PSI), photosystem II (PSII), cytochrome b_6f complex and F-ATPase. The F-ATPase produces ATP at the expense of the proton-motive force (PMF) that is generated by the light reactions in PSI and PSII, which is mediated by the cytochrome b_6f complex (Figure 1.7) (344).

Chlorophyll and other pigments harvest light energy and transfer it to the PSI and PSII reaction centres (RCs), which initiates the linear transfer of an electron across the membrane through a chain of cofactors (344). The electron is initially donated by the oxidation of H_2O to O_2 and $4H^+$ (water splitting), which occurs at the oxygen-evolving centre (OEC) in PSII (344). Upon excitement of the PSII RC an electron is transferred through a chain of electron carriers to the plastoquinone (PQ) pool and then shuttled to the cytochrome b_6f complex, which in turn, transfers electrons to PSI via plastocyanin (PC) (344). The PSI RC is excited by light energy which ejects the electron, sending it through another chain of electron carriers until it reaches ferredoxin-$NADP^+$ reductase (FNR), where it reduces $NADP^+$ to NADPH (344), the predominant cyanobacterial reducing agent. This linear electron transport leads to a

proton gradient across the thylakoid membrane, which is then subsequently used for ATP synthesis by the F-ATPase (12). Not all electron transport is linear, in photosynthetic purple bacteria, electrons follow a circular path where they return to the cytochrome b_6f complex (12). The light energy derived ATP and NADPH can then be used to synthesize carbohydrates and other products from CO_2.

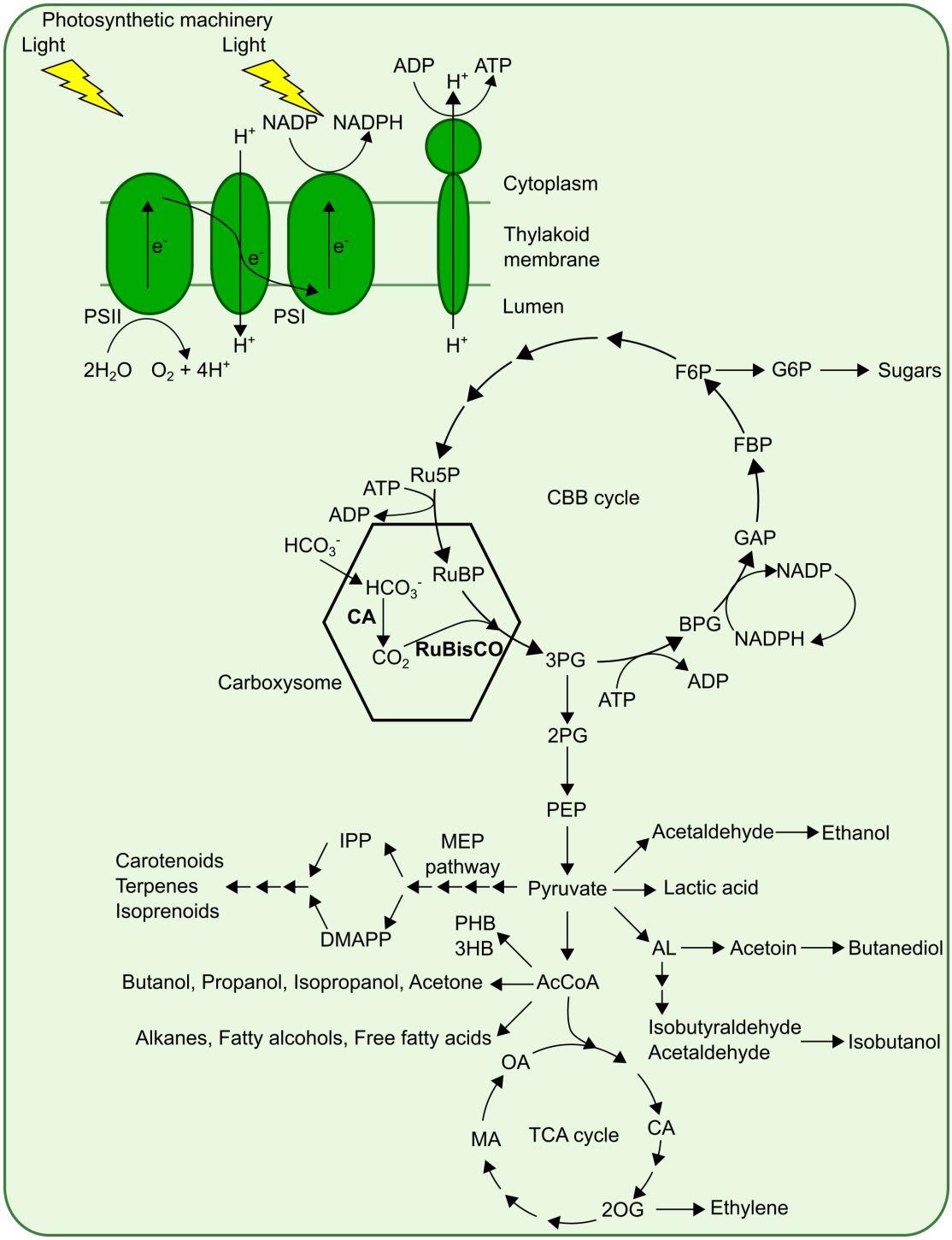


Figure 1.7. Cyanobacterial metabolism showing CO_2 converted to a host of products. The photosynthetic machinery uses light energy, fixed by photosystem I (PSI) and photosystem II (PSII) to split water and generate NADPH and ATP. Within the carboxysome CO_2 is generated by the carbonic anhydrase enzyme (CA) from bicarbonate (HCO_3^-), CO_2 and Ribulose 1,5-bisphosphate (RuBP) are the converted to 3PG by the enzyme Ribulose-1,5-bisphosphate carboxylase/oxygenase (RuBisCO). 3PG can then continue through the Calvin-Benson-Bassham (CBB) cycle and used to produce simple sugars, such as glucose, fructose and sucrose from fructose 6-phosphate (F6P) or regenerate more RuBP, or 3PG can enter glycolysis and be converted to pyruvate, via 2-phosphoglycolate (2PG) and phosphoenolpyruvate (PEP). Pyruvate can then be converted to a whole host of different products.

1.5.1.3.2 Carbon fixation and Calvin-Benson-Bassham (CBB) cycle

Efficient carbon fixation of CO_2 in cyanobacteria is dependent on their CO_2-concentrating mechanism (CCM). The CCM has evolved to enhance the chemical conditions that surround the primary CO_2 fixing enzyme d-ribulose 1,5-bisphosphate carboxylase/oxygenase (RuBisCO), which is not only capable of the carboxylation of CO_2 but of the oxygenation of O_2 (345). The cyanobacterial RuBisCO typically has a low catalytic rate (345) and a low affinity for CO_2 (346) and as a result, preferentially catalyses the addition of O_2 to ribulose-1,5-bisposphate (RuBP) rather than CO_2. As such, carbon fixation is a slow and a kinetically challenging reaction. To make this reaction more kinetically favourable, it is essential to maintain an enriched CO_2 level and/or decreased O_2 level. The cyanobacterial CCM has evolved over many years for exactly this function. The CCM is encapsulated within a proteinaceous bacterial microcompartment (BMC) known as a carboxysome (345). Within the carboxysome is the CO_2 fixing enzyme RuBisCO and a carbonic anhydrase enzyme that catalyses the production of CO_2 from the cytoplasmic bicarbonate (HCO_3^-) pool, enriching CO_2 levels within the BMC, close to RuBisCO (345). Two types of carboxysome exist, α- and β-carboxysome, both of which have selectively permeable membranes that prevents the efflux of CO_2 and the influx of O_2, buts allows transfer of RuBP, 3-phosphoglycerate (3PG) and HCO_3^- between the carboxysome and cytoplasm (345). CO_2 produced by the carbonic anhydrase is then fixed into the 3C sugar 3PG from the 5C RuBP (12) in the first reaction of the CBB cycle (Figure 1.7). 3PG is then able to transfer back to the cytoplasmic pool for the rest of the CBB cycle reactions.

After the carbon fixation, 3PG is reduced to glyceraldehyde-3-phosphate (GAP) in a two-step process via the intermediate 1,3-Bisphosphoglycerate (BPG) using the enzymes phosphoglycerate kinase and glyceraldehyde-dehydrogenase, which require the ATP and NADPH from the oxygenic photosynthesis light reactions, respectively (12). GAP can then either be used to synthesize carbohydrate-based products or it can be recycled back to RuBP via intermediates including fructose-1,6-bisphosphate (FBP), fructose 6-phosphate (F6P) and Ribulose-5-phosphate (Ru5P) in a complex ten-enzyme pathway, again requiring ATP (Figure 1.7) (12).

2PG along with 3PG are products of RuBisCO reacting with O_2. 2PG is toxic to cyanobacteria because it inhibits the CBB enzyme triosephosphate isomerase (347) and therefore it needs to be rapidly metabolised. In cyanobacteria, this is achieved by employing three different pathways: i) the plant-derived C2 cycle; ii) the bacterial-like

glycerate pathway; iii) and the decarboxylating pathway (348), which metabolises 2PG into the non-toxic products ammonium, glycine and CO_2, respectively.

1.5.1.3.3 Post CBB cycle metabolism

In strains of cyanobacteria that are growing photoautotrophically, only the final three steps of glycolysis are required. 3PG from the CBB cycle enters glycolysis and is subsequently converted to 2PG, then phosphoenolpyruvate (PEP) and finally pyruvate (12). Glucose utilising strains of cyanobacteria convert the C6 sugar to two C3 units, pyruvate, through the ten-reaction pathway of glycolysis (12). This pathway releases free energy that is used to synthesize ATP from ADP and Pi. In the first five reactions, two molecules of ATP are consumed to phosphorylate glucose and cleave the C6 molecule into two GAP molecules (12). In the following five reactions, two molecules of GAP are converted to pyruvate, during which four units of ATP are also generated, resulting in a net gain of two ATP units (12).

Pyruvate is a key central metabolite in cyanobacteria metabolism (Figure 1.7), which is essential for amino acid synthesis and growth (349). It can also serve as a key intermediate to the production of many natural and non-native products of interest (Figure 1.7). For example, pyruvate can serve as the precursor to the MEP pathway, which generates isopentenyl pyrophosphate (IPP) and dimethylallyl pyrophosphate (DMAPP), the precursors to terpenes and isoprenoids (263), or pyruvate can serve as a precursor to ethanol, lactic acid and other products of interest (263). Alternatively, it can be converted to acetyl-CoA and enter to TCA cycle (12) or used to generate fatty acids, 3HB-based products and some alcohols (263).

In cyanobacteria the most widely used intermediary metabolite in metabolic engineering is pyruvate. Other intermediary metabolites include acetyl-CoA, F6P, IPP and DMAPP, although these are more distant from the central metabolism, with a lower intracellular concentration; and as such, typically result in significantly lower titres compared to the product titres made via pyruvate (350). Products from pyruvate have resulted in the highest titres, such as of ethanol (284), 2,3-butanediol (297) and lactic acid (302), which have all been over 1.84 g l^{-1}.

1.5.1.3.4 Strain of cyanobacteria used in Metabolic Engineering and Synthetic Biology

Synechocystis sp. PCC 6803, *Synechococcus elongatus* PCC 7942, *Synechococcus* sp. PCC 7002 and the recently identified *Synechococcus elongatus* UTEX 2973 are the most frequently used strains of cyanobacteria applied in metabolic engineering and the sustainable production of chemicals and biofuels. All of these strains have sequence genomes, established culturing methods and at least basic metabolic engineering and synthetic biology tools (266, 351-353).

Synechocystis sp. PCC 6803 is a freshwater unicellular cyanobacterium (Figure 1.5B) that has most likely been the most extensively studied and engineered strain of cyanobacteria. It has been used as the model cyanobacteria for fundamental studies of important processes such as photosynthesis (354-359), circadian rhythms (360) and carbon-concentrating mechanisms (345, 361-363). Glucose tolerant strains have been generated, having originally been identified after a spontaneous mutation occurred when grown in the presence of glucose (364); so *Synechocystis* sp. PCC 6803 can either be grown under photoautotrophic conditions in the presence of CO_2, in mixotrophic conditions in the presence of CO_2 and glucose or in heterotrophic conditions in the presence of just glucose (365). The large amount of engineering that has been performed in *Synechocystis* sp. PCC 6803 is most likely accountable to it being the third prokaryote and the first photosynthetic organism to have had its genome completely sequenced (366) and also due to its naturally transformable nature with heterologous genes (367).

Synechococcus elongatus PCC 7942 is a freshwater unicellular cyanobacterium and like *Synechocystis* sp. PCC 6803, it is one of the more well studied and engineered strains of cyanobacteria (345, 360-362, 368, 369), which is most likely in part due to its naturally transformable nature (370).

Synechococcus sp. PCC 7002 is a unicellular marine species of cyanobacteria, which can grow in a variety of salt, temperature and light conditions (371, 372). This makes the strain more suitable to growing in seawater, which is abundantly available. *Synechococcus* sp. PCC 7002 is naturally transformable (373), but has not been applied in metabolic engineering anyway near as much as the two freshwater strains described above.

There appears to be a growing interest in fast growing cyanobacteria. Recently, the cyanobacterium *Synechococcus elongatus* UTEX 2973 was identified (353), which displayed substantially faster growth rates compared to any of the commonly used cyanobacterium's (353). This fast-growing strain is not naturally competent, like the other three strains described, however it was demonstrated that the strain could be easier transformed using conjugation, to introduce heterologous genes or to knockout native ones (353). It has recently been revealed that *Synechococcus elongatus* UTEX 2973 is a very close relative of *Synechococcus elongatus* PCC 7942, identifying only 55 single-nucleotide polymorphisms (SNPs) and insertion-deletion differences between the two genomes (374). Even more recently, new isolates such as *Synechococcus* sp. PCC 11801 have been identified and reported at recent conferences (Metabolic Engineering 12, Munich).

These four strains of cyanobacteria are the most used strains in metabolic engineering and have each been used to produce products of interest as described in section **1.5**. The four cyanobacteria strains offer different advantages over one another, for example, they all differ in growth rates, number of chromosomes, salt tolerance and genetic tool availability. Cyanobacteria has a relatively slow growth rate compared to popular host strains such as *E. coli*. For example, even when these four strains of cyanobacteria were grown under an enriched CO_2 atmosphere (3%), *Synechocystis* sp. PCC 6803 had a doubling time of 6.6 hours when grown at 300 µmol photons m^{-2} s^{-1}, *Synechococcus elongatus* PCC 7942 had a doubling time of 5.0 hours when grown at 300 µmol photons m^{-2} s^{-1}, *Synechococcus* sp. PCC 7002 had a doubling time of 4.1 hours when grown at 500 µmol photons m^{-2} s^{-1} and the fast-growing *Synechococcus elongatus* UTEX 2973 had a doubling time of 2.1 hours at 500 µmol photons m^{-2} s^{-1} (375). While the doubling times of these four strains were determined under different light intensities and are therefore not directly comparable with one another, the different doubling times suitably demonstrates their slower growth rate compared to that of model organisms, even when they are grown in enriched environments. Cyanobacteria often have multiple copies of their chromosomes (polyploidy) and like with the growth rates, the number of chromosomes varies between strains. For example, *Synechocystis* sp. PCC 6803 has approximately 220 copies when in an exponential phase of growth, whereas *Synechococcus elongatus* PCC 7942 has only four copies (376).

1.5.1.4 Challenges associated with engineering cyanobacteria

There are challenges associated with cyanobacteria, which complicate their ability to be easily engineered. The most notable challenge is their slow-growth rates (375), which slows down the iterations of the DBTL cycle (one cycle takes 1 – 2 months). Certain strains of cyanobacteria are polyploid (376, 377), which makes the isolation of segregated recombinant strains slow and laborious, as many subcultures of transformants colonies under selective pressure are required to obtain full segregation of the insert in each copy of the chromosome. Cyanobacteria also suffer from genetic instability, which up until recently has been somewhat ignored, most probably due to its association with negative results. Cyanobacteria seem to lose function of heterologous DNA over time, particularly in metabolic engineering applications (118). As a result, the production of a given product unexpectedly drops, as a result of one or a few nucleotide mutations in the pathway, or by a complete loss of varyingly sized fragments of the pathway. Both cause a loss of gene expression and therefore pathway function. In recent years genetic instability has been increasingly reported in metabolic engineering applications of cyanobacteria. The ethylene forming enzyme (*efe*) was introduced to *Synechococcus elongatus* PCC 7942 and after applying selective pressures, strains that survived were able to produce ethylene. However, over repeated subcultures the strain lost its ability to produce ethylene due to an unexpected truncation of the *efe* gene (327). Furthermore, the same truncation of the gene was reproducible, occurring in multiple transformant cells. Interestingly, when *efe* was expressed in *Synechocystis* sp. PCC 6803, no genetic instability was observed over six months (378).

Following the first report, genetic instability has been observed in *Synechococcus* sp. PCC 7002, when engineered to produce mannitol, in which a single-base frame-shift was observed that consequently stopped mannitol production (320). Genetic instability was also been observed in *Synechocystis* sp. PCC 6803 when it was engineered to produce lactic acid, but production unexpectedly stopped (237, 379). This was a result of spontaneous mutations in the pathway genes, whereby in one instance, a duplication event introduced a premature stop codon in one of the pathway genes (237). Isopropanol production was repeatedly halted by the same mutation in a pathway gene introduce to *Synechococcus* sp. PCC 7942 (380). A recent study found that fatty alcohol production in *Synechocystis* sp. PCC 6803 dropped by 50% over the course of two subcultures. After four subcultures, the strain had completely lost the ability to produce fatty alcohols (300). The study suggested that the loss of function was caused by a genetic rearrangement in the gene *tes3*, one of the pathway genes.

In *E. coli*, similar incidences of genetic instability have been overcome by deleting *recA* (381), however repeated attempts in cyanobacteria failed to knock out the gene unless under initially dark and then low-light conditions (382). It was concluded that RecA is essential for photoautotrophic growth where UV-light induced DNA damage was frequent.

In addition to challenges introduced above, the genetic toolbox for cyanobacterial strains lags far behind that of model organisms like *E. coli* and *S. cerevisiae*, but in recent years a lot of time has been spent developing the necessary tools to engineer and fine-tune the performance of metabolic pathways. This has involved characterising native and synthetic constitutive promoters (352, 383-385), inducible promoters (124), native and synthetic RBSs (352, 385) and transcriptional terminators (385). However, while these tools have been developed for cyanobacteria, they are typically only characterised in one strain. These tools are further discussed in section **3.1**. and in a recent detailed review (386).

1.6 Challenges of using non-model organisms in metabolic engineering

The challenges encountered when engineering non-model organisms can simply be explained by the approach used to design the metabolic pathway-encoding construct. As described in section **1.3.3.2.2**, our ability to rationally design metabolic pathway-encoding constructs that perform well is still limited and therefore many iterations of the DBTL cycle are required to optimise the pathways performance. In a fast-growing model organism, this isn't an issue as one iteration of the DBTL cycle only takes one to two weeks and therefore multiple iterations is a manageable timeline for a project. However, non-model organisms, such as cyanobacteria (as described in section **1.5.1.4**), often have slow growth rate, which means one iteration of the DBTL cycle takes a considerably longer time (e.g., 1 – 2 months in cyanobacteria). The issue then becomes amplified when multiple iterations of the DBTL cycle are required and a projects timeline quickly becomes unachievable (Figure 1.8).

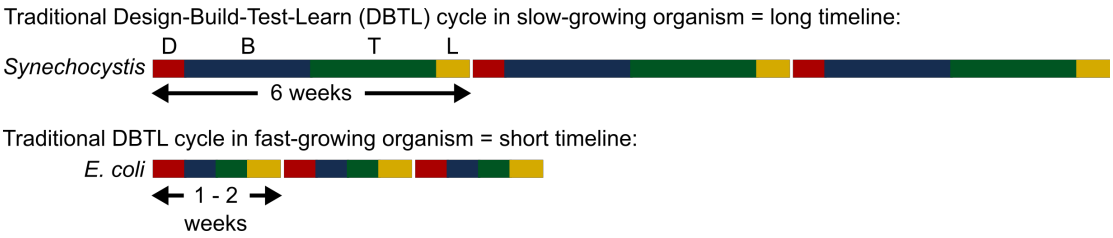


Figure 1.8. Timelines of iterative approaches in fast and slow-growing organisms. With one iteration of the Design-Build-Test-Learn (DBTL) cycle taking longer in slow-growing organism (approximately 6 weeks per iteration) compared to a fast-growing organism (approximately 1 – 2 weeks per iteration). As a result, iterative pathway optimisation can take considerably longer in slow-growing organisms compared to fast-growing organisms.

Rather than using an iterative rational approach to designing metabolic pathways, a combinatorial approach could be used. Unlike in model organisms where the necessary design approaches, such as pathway assembly and pathway clone selection, have been developed that allow combinatorial approaches to be implemented (as described in section **1.3.3.2.3**); no such design approaches have been developed for non-model organisms. Furthermore, when a combinatorial approach is used the organism being engineered typically needs to have a high transformation efficiency. A high transformation efficiency ensures that the largest possible sample of the pathway library is transformed into the organism, which is particularly important when a highly sensitive pathway or system is being engineered, such as the refactored nitrogenase complex (142, 149). The extensive work carried out on model organisms such as *E. coli*, has led to highly-efficient transformation methods (>1 x 10^{10} cfu μg^{-1}) being developed, whereas in non-model organisms, highly-efficient transformation methods have not typically been developed, often with only a few transformant colonies being recovered. Some transformation methods for non-model organisms have been developed with acceptable efficiencies, such as for methanotrophs, where efficiencies vary between $8x10^3$ and $5x10^5$ cfu μg^{-1} (387). However, these efficiencies are still a lot lower than those developed for model organisms.

The last challenge applies to the optimisation of metabolic pathway-encoding constructs when using both iterative-rational and combinatorial approaches. The genetic toolkit of the vast majority of non-model organisms lags far behind that of model organisms. As a result, fewer promoters, RBSs and other expression control parts are available for rational designs, iterative substitutions and for generating combinatorial mixtures of parts.

For non-model organisms to be fully exploited, these challenges will have to be overcome so that iterative and combinatorial approaches can be fully utilised to optimise the performance of metabolic pathways.

1.7 Overcoming the challenges

To overcome the issues surrounding slow-growing organisms or poorly transformable organisms, two obvious solutions would be to use a fast-growing strain of the given organism, for example, using *Synechococcus elongatus* UTEX 2973 rather than *Synechococcus elongatus* PCC 7942, or to develop better transformation methods to improve efficiencies, for example, this was developed for *Clostridium saccharoperbutylacetonicum* strain N1-4 and allowed engineering that resulted in an 18% increase in butanol titre (388). However, unlike cyanobacteria, not all non-model organisms have faster-growing strains and developing transformation methods that will reach the efficiencies needed for large combinatorial pathway libraries could quite conceivably not be possible. Considering this, it would be useful if a pathway could be assembled, tested and optimised in a more tractable organism that doesn't possess the challenges described above and in which iterative and combinatorial approaches can be implemented. The optimised pathway can then be transferred to another less-tractable organism, with the confidence that it would work in the second organism.

1.7.1 Rapid Prototyping

Rapid prototyping is a concept that has been largely used in the development of commercial products since the late 1980s (389). Rapid prototyping involves quickly making working scale prototypes of a product in a short time. Typically, the prototype product is first made as a model using a three-dimensional computer aided design (CAD), which allows it to be visually assessed. It is then constructed using either 3D printing or additive manufacturing to allow it to be tested (389). Rapid prototyping has proven popular in a wide range of industrial design processes, right from the aviation sector to sculpture design (389). This is primarily because rapid prototyping decreases the development time and cost of the product by allowing problems to be identified and the necessary corrections to be made early on in the development process.

Rapid prototyping has hardly been discussed in biology, particularly *in vivo* biology (Dr John Heap, pers. comm), which in turn has led to the concept being explored in this project. Recently, the concept of *in vitro* prototyping has been proposed, whereby cell-free transcription-translation (TX / TL) systems are used to test and optimise the

functionality and performance of a system before it is expressed *in vivo* (390-393). Such *in vitro* prototyping strategies have been used to optimise the production *n*-butanol and polyhydroxyalkanoates bioplastics in *E. coli* (390, 391).

In this project, the concept of rapid prototyping is taken and applied to an *in vivo* optimisation process for non-model organisms, whereby the optimisation of metabolic pathways is carried out in fast-growing model organisms that have high transformation efficiencies, such as *E. coli*, making it easier and faster to implement combinatorial and iterative approaches. Once the pathway is optimised in the model organism, it can be transferred to the non-model organism for pathway expression (Figure 1.9).

Implementing rapid prototyping for *in vivo* optimisation of non-model organisms aims to overcome the long timelines that traditional iterative optimisation processes have had in slow-growing organisms (Figure 1.9). It also aims to overcome the poor pathway library quality of combinatorial approaches in non-model organisms with poor transformation efficiencies.

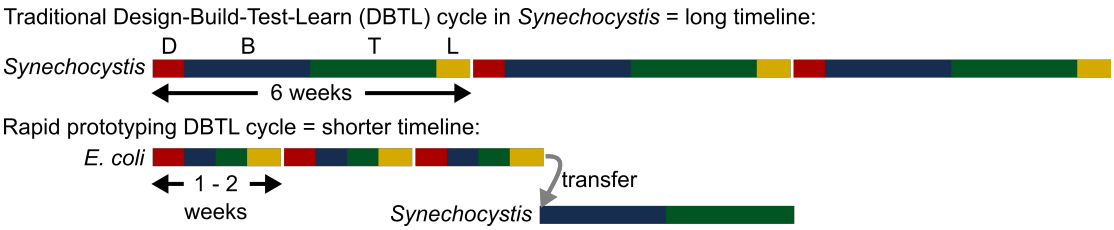


Figure 1.9. Timelines for optimising a metabolic pathway using iterative and rapid prototyping-based approaches. One iteration of the Design-Build-Test-Learn (DBTL) cycle in a slow-growing organism, such as *Synechocystis*, takes approximately six weeks, therefore three iterations of the DBTL takes 18 weeks. One iteration of the DBTL cycle in a fast-growing organism, such as *E. coli*, takes approximately 1 – 2 weeks, therefore three iterations of the DBTL cycle takes between 3 – 6 weeks. If a metabolic pathway was optimised by rapid prototyping, 3 – 6 weeks of optimising before transferring to *Synechocystis* for expression is considerably less time than 18 weeks.

1.7.2 Portability

The effectiveness of prototyping would depend on whether the performance of a multi-protein system in one organism is predictive of the performance of that system in another organism, and if so, how accurate that prediction is. The performance of a metabolic pathway is dependent on expression control parts such as promoters, RBSs, CDSs and terminators (as described in section **1.3**). Engineering a predictive performance between the non-model and model organism (for rapid prototyping) could

be achieved by using expression control parts that perform the same between the two organisms and are therefore 'portable' between the organisms. To engineer the predictive performance that rapid prototyping relies on, expression control parts, which are portable between the two organisms were targeted in this project.

Typically, when a metabolic pathway is transferred to a different organism, the activity of the expression control parts are effected, often breaking or changing the behaviour of the parts and therefore the pathway's performance (394). For example, when the refactored *nif* cluster was transferred from *K. oxytoca* to *E. coli*, its activity decreased by 5% (149). A considerable amount of effort, typically expression control part substitution, is then required to optimise the pathway balance of the metabolic pathway in the new host (395-397). Such incompatibilities between species are most likely caused by the differences in their gene expression machinery (398-402). The different species specific gene expression machinery has resulted in expression control parts that have been tailor-made for each organism (352, 353, 403, 404).

To allow rapid prototyping, expression control parts are needed that are portable between two organisms, whereby transferring the parts between the two organisms won't result in changed pathway performance because the expression control parts activity have changed. Rather, the expression control parts will perform similarly between the two hosts, so the pathways performance in the model organism can be used as a prediction of its performance in the non-model organism.

To date, very little work has been done comparing the activity of expression control parts between different organisms and there has been even less identification and development of portable parts. To the best of my knowledge, there are only two studies that have reported the characterisation of expression control parts between two organisms. First, the performance of a 20-member promoter library, generated for the thermophile *Geobacillus thermoglucosidasius*, was compared between the thermophile and *E. coli*, identifying no correlation of performance between the two organisms (R^2 = 0.04) (235). Second, a synthetic promoter library generated for *Synechococcus* sp. PCC 7002 was characterised in both the cyanobacterium and *E. coli*, which like the previous example, also showed no correlation between the organisms (352).

Efforts to identify or develop portable expression control parts are also very limited. For a long time, phage polymerases have been used to control orthogonal transcription

in many different organisms, but these projects have typically been species specific and have not considered cross-species expression levels. The most notable example is the T7 RNA polymerase, which has successfully been used along with T7 promoters in a variety of hosts, including Gram-negative and -positive bacteria, plant chloroplasts and mammalian cells (142, 149, 405-407).

More recently, a study generated a 13-member synthetic promoter library for *Clostridia* and found it to be highly portable between different species, such as *C. acetobutylicum* ATCC 824 and *C. sporogenes* NCIMB 10696 (R^2 = 0.897), two distantly related strains (408). To date, there are two examples of cross-species portable expression control parts, the first of which uses a viral RNA polymerase that was orthogonal to the three host species, *E. coli*, *Pseudomonas putida* and *B. subtilis*, to create a portable expression system, termed UBER, which proved highly portable between the three organisms (409). The second approach uses an orthogonal and tightly regulated inducible T7-like promoter that performs with a strong correlation (R^2 > 0.94) between *E. coli* and *Halomonas* (246).

1.8 Aims and Objectives

The aim of this PhD project was to develop expression control parts that would allow implementation of rapid prototyping. With these expression control parts, it would be possible to try a first attempt at rapid prototyping and then to evaluate whether it is better than the current state-of-the-art design principles that are used to construct metabolic pathway-encoding constructs in non-model organisms. Here, the non-model organism *Synechocystis* sp. PCC 6803 was used as the first target host organism for rapid prototyping. To attempt rapid prototyping, *Synechocystis* sp. PCC 6803 was paired with the fast-growing model-organism *Escherichia coli*. *Synechocystis* sp. PCC 6803 was chosen as the first target host organism, because its photoautotrophic nature makes it a good candidate for the renewable production of products of interest, however it is a slow-growing organism that has proven hard to engineer.

- Firstly, as described in Chapter Three, the project set out to identify expression control parts that were portable between *E. coli* and *Synechocystis*. Libraries of synthetic promoters and RBSs that were functional in both *E. coli* and *Synechocystis*, were generated, which enabled the relationship of the performance of the libraries to be assessed between the two organisms.

- Secondly, as described in Chapter Four, given what was found in Chapter Three, a subset of portable promoters and RBSs was identified and validated from the libraries that were generated in Chapter Three. To achieve this, a computational tool that identified portable subsets of promoters and RBSs from the libraries was developed. The identified subsets were then characterised in different genetic contexts to further assess their portable performance.
- Thirdly, as described in Chapter Five, a functionally-scarless and modular multi-part DNA assembly method, termed Start-Stop Assembly, was developed, which could be used for the efficient combinatorial assemblies of metabolic pathways while imposing minimal constraints on their design. This was demonstrated by using Start-Stop Assembly for the combinatorial assembly of metabolic pathway-encoding constructs.
- Lastly, as described in Chapter Six, rapid prototyping in *Synechocystis* was tested and compared to the current state-of-the-art design strategies typically used for engineering non-model organisms.

2. CHAPTER TWO. Materials and Methods

2.1 Materials

Except where otherwise specified, bacterial growth media, chemicals and biochemicals were supplied by Sigma-Aldrich or Fisher Scientific, enzymes for molecular biology and their buffers were supplied by New England Biolabs (NEB).

Table 2.1. Table of bacterial strains used in this study

Strain	Genotype	Source
Escherichia coli DH10B	Δ(ara-leu) 7697 araD139 fhuA ΔlacX74 galK16 galE15 e14- φ80dlacZΔM15 recA1 relA1 endA1 nupG rpsL (StrR) rph spoT1 Δ(mrr-hsdRMS-mcrBC)	Invitrogen
Synechocystis sp. PCC 6803	Wild-type, Glucose tolerance	Nixon Lab (Imperial College London)

2.2 Bioinformatics methods

SnapGene (Chicago, USA) was used for all plasmid designs, oligonucleotide designs, sequence alignments and sequence data handling.

2.3 Microbiological materials and methods

2.3.1 *E. coli* growth conditions

Escherichia coli strain DH10B (Table 2.1) was used for all DNA assembly, all other cloning, and all other experiments described in this project. *E. coli* DH10B cells were routinely grown in LB medium (tryptone 10 g l^{-1}, yeast extract 5 g l^{-1} and NaCl 5 g l^{-1}) at 37 °C, with shaking at 225 r.p.m, or on LB agar plates (containing 15 g l-1 bacteriological agar) at 37 °C. LB was supplemented with 5-bromo-4-chloro-3-indolyl-β-galactopyranoside (Xgal, 40 μg ml^{-1}), Isopropyl β-D-1-thiogalactopyranoside (IPTG, 0.1 mM), L-rhamanose (0.6 mg) and either ampicillin (100 μg ml^{-1}), tetracycline (10 μg ml^{-1}), kanamycin (50 μg ml^{-1}) or chloramphenicol (25 μg ml^{-1}) as appropriate.

In order to characterise fluorescence and enzymatic reporters, purified plasmids were transformed via heat shock into chemically competent *E. coli*. Cultures were grown from single colonies in 200 μl LB (50 μg ml^{-1} kanamycin) overnight (16 h) in plates of 96 U-shaped 1.2 ml wells covered with sterile breathable sealing film (Breathe Easy) at 37 °C with shaking at 700 r.p.m on a Multitron shaker (Infors-HT). Overnight cultures

were diluted 1:1000 into a final volume of 200 µl fresh LB medium (50 µg ml^{-1} kanamycin) in deep-well plates and grown for 4 h at 37 °C with shaking at 700 r.p.m.

2.3.2 *Synechocystis* growth conditions

Synechocystis sp. PCC 6803 (the glucose-tolerant derivative of the wild type, Table 2.1) was used for all cyanobacterial experiments described in this project. *Synechocystis* sp. PCC 6803 was routinely grown in TES-buffered (pH 8.2) BG11 medium (410) at 30 °C with agitation at 150 r.p.m and in constant white light at 50 µmol m^{-2} s^{-1}. BG11 was supplemented with kanamycin (30 µg ml^{-1}) as appropriate.

In order to characterise fluorescence and enzymatic reporters, purified plasmids were transformed into *Synechocystis* and subcultured to complete segregation. Cultures were cultivated in 5 ml BG11 medium from a loopful of transformed cells and incubated in 25 ml vented culture flasks under phototrophic conditions, 30 °C, 150 r.p.m, 50 µmol photons m^{-2} s^{-1} light and incubated for 48 h. Following incubation, cultures were diluted to an OD750 of 0.1 and incubated for a further 24 hours.

2.3.3 Preparation of electro-competent cells of *E. coli*

200 ml of fresh LB broth were inoculated with a 1:100 volume of fresh overnight culture. Cells were grown at 37 °C with vigorous shaking to an OD600 of 0.5 – 1.0 (an OD600 indicative of rapid growth). To harvest, the cells were separated into 50 ml centrifuge tubes, chilled on ice for 30 minutes and then centrifuged at 4000 r.p.m for 15 minutes at 4 °C. Pellets were gently resuspended in one culture volume of sterile deionised water and centrifuged at 4000 r.p.m for 15 minutes at 4 °C to wash. Pellets were gently resuspended in 1 ml of sterile 10 % glycerol, pooled, and then centrifuged as before. Finally, the pellet was resuspended to a volume of 0.5 ml in 10 % glycerol. The cells were stored at -80 °C in 55 µl aliquots.

2.3.4 Transformation of *E. coli* by electroporation

Electroporation was used for transforming *E. coli* with libraries and general cloning. For each electroporation, one 55 µl aliquot of electro-competent *E. coli* was thawed on ice and then DNA was added to the cells and stirred gently using the pipette tip. The mixture was gently pipetted into a cold 0.1 cm gap electroporation cuvette. A pulse was applied across the cuvette using a Bio-Rad MicroPulser electroporator set to programme ‘Ec1’ (1.8 kV voltage; capacitance and resistance not stated by this

apparatus or the manufacturer's instructions). Immediately following the pulse 250 µl of LB medium was added to the cuvette. The entire contents of the cuvette were transferred to a 1.5 ml microtube and incubated for 1 hour at 37 °C with 225 r.p.m shaking to allow expression of antibiotic resistance genes. Cells were then plated onto appropriate selective media using suitable dilutions.

2.3.5 Preparation of chemical-competent cells of *E. coli*

5 ml of fresh LB broth were inoculated with a 1:100 volume of fresh overnight culture. Cells were grown at 37 °C with vigorous shaking to an OD600 of 0.3 – 0.5. To harvest, the cells were chilled on ice for 30 minutes and then centrifuged at 4000 r.p.m for 10 minutes at 4 °C. Pellets were gently resuspended in 0.5 ml cold TSB (LB supplemented with 0.05 M MgSO4, 5 % DMSO, 0.5 g PEG 6000) and incubated on ice for 10-30 minutes. Cells were stored at -80 °C in 100 µl aliquots.

2.3.6 Transformation of *E. coli* by Heat-Shock

Heat-shock transformations were used for transforming *E. coli* with purified plasmids. For each transformation, one 100 µl aliquot of chemically-competent *E. coli* cells was thawed on ice, after which DNA was added and the cells mixed gently with a pipette tip. The mixture was incubated on ice for 30 minutes, after which the mixture was incubated for 90 s at 42 °C and returned to the ice for 2 min. 1 ml of LB was added and the culture incubated for 1 h at 37 °C at 225 r.p.m. Following incubation, the cells were pelleted at 3000 g for 5 minutes and then resuspended in 100 µl of LB. Cells were then plated onto appropriate selective media using suitable dilutions.

2.3.7 Transformation of *Synechocystis*

Wild-type *Synechocystis* cells were cultured in BG11 to an optical density (measured at 750 nm) of 0.5 and 4 ml harvested by centrifugation at 3200g for 15 min. Pellets were resuspended in 100 µL BG11, 100 ng of plasmid DNA was added and the mixture was incubated at 100 µmol m^{-2} s^{-1} white light for 60 min. Cells were spotted onto BG11 agar plates and incubated at 50 µmol m^{-2} s^{-1} white light for 24 h at 30 °C. Cells were collected and transferred onto BG11 agar plates supplemented with 30 µg ml^{-1} kanamycin. When single colonies appeared after approximately 7 days, transformants were segregated through passaging on selective plates approximately weekly, and full segregation after approximately 3 weeks was confirmed by PCR.

2.4 Molecular biological materials and methods

2.4.1 Extraction and purification of plasmid DNA

Plasmid DNA was extracted and purified from *E. coli* using the Qiagen QIAprep Spin Miniprep kit in accordance with the manufacturer's instructions.

2.4.2 Spectrophotometric quantitation of DNA

DNA quantification from miniprepped or extracted samples was performed using the NanoDrop 1000 Spectrophotometer in accordance with the manufacturer's instructions.

2.4.3 Amplification of DNA using PCR

DNA sequences were routinely PCR amplified using Phusion High-Fidelity DNA polymerase with Phusion HF-buffer in accordance with the manufacturer's instructions. When required, PCR reactions were supplemented with betaine (1 M). For PCR screening, Taq DNA polymerase with Standard Taq Buffer was used in accordance with the manufacturer's instructions.

2.4.4 Amplification of DNA using Touchdown PCR

When touchdown PCR reactions were used, the annealing step of the PCR reaction followed these conditions: 2 cycles at annealing temperature calculated by NEB Tm Calculator (web tool), followed by a variable number of cycles from starting annealing temperature down to 50 °C decreasing by 0.5 °C increments in each cycle and finally a variable number of cycles at 50 °C until 35 cycles had been complete.

2.4.5 Amplification of DNA using SOE PCR

When required, DNA fragments were spliced together using splicing by overlap extension (SOE) PCR with appropriate primers (189).

2.4.6 Restriction endonuclease digestion of DNA

For both cloning and diagnostic screening purposes, DNA was digested with restriction endonucleases according to the manufacturer's instructions (NEB), typically at 37 °C for 3 hours. To digest parental plasmid following PCR reactions 1 µl of *DpnI* was added to a 25 µl PCR mixture and incubated at 37 °C for 1 hour.

2.4.7 Phosphorylation of DNA fragments

Where appropriate, T4 polynucleotide kinase was used in accordance with the manufacturer's instructions to introduce 5′ phosphate groups to linear DNA fragments.

2.4.8 Oligonucleotide preparation

All oligonucleotides were synthesised by Integrated DNA Technologies (IDT) and dissolved in deionised water at a concentration of 100 μM. Oligonucleotides used as PCR primers were diluted and used at 10 μM. A list of primers use in this study is shown in Table A4.

2.4.9 Annealing of oligonucleotide linkers

Double-stranded linkers containing spacers were generated by annealing complementary pairs of oligonucleotides (Table A7). Pairs of oligonucleotides were mixed together to a final concentration of 10 μM. Oligonucleotide mixtures were heated to 95 °C for 5 mins, and then allowed to cool slowly to room temperature. These annealed linkers were used directly as parts in Level 1 assembly reactions at a concentration of 40 fmol.

2.4.10 Agarose gel electrophoresis

DNA fragments were separated by electrophoresis at 80 - 100 V for 45 - 60 minutes through 0.8 - 1 % agarose gels stained with SYBR safe (Invitrogen, Paisley, UK) and DNA was visualised under ultraviolet light.

2.4.11 Extraction of DNA from agarose gels and reaction mixtures

DNA fragments were extracted from excised agarose gel pieces using the Qiagen QIAquick Gel Extraction Kit in accordance with the manufacturer's instructions, or from PCR, endonuclease digestion and other reactions using the Qiagen QIAquick PCR Purification Kit, in accordance with the manufacturer's instructions.

2.4.12 Ligation of DNA

DNA fragments were ligated using T4 DNA Ligase in accordance with the manufacturer's instructions. The molar ratio of plasmid to insert was typically 3:1, respectively, for cohesive end ligation. The ligation mixture was incubated for 2 hour at room temperature or overnight at 4 °C. In any instance when a linear DNA fragment

was ligated, 1.5 µl of T4 DNA Ligase and 3 µl of T4 DNA Ligase buffer was added to 30 µl of purified DNA fragment and incubated overnight on a 4 °C, to room temperature gradient.

2.4.13 Storing genetic parts

Genetic parts were cloned in Level 0 vector pStA0 using one-pot restriction-ligation assembly reactions (as described below in section **2.4.14** and **2.8.4.5**) of inserts obtained by PCR amplification or DNA synthesis, or by inverse PCR using pStA0 as template and primers incorporating the part sequence. Spacer parts were generated as double-stranded oligonucleotide linkers by oligo annealing (as described in section **2.4.8**) and were not cloned for storage.

2.4.14 Start-Stop Assembly reactions

Start-Stop Assembly reactions contained 20 fmol of destination vector plasmid DNA, 40 fmol of each insert (plasmid DNA or annealed oligonucleotides), T4 DNA Ligase buffer, 400 units of T4 DNA Ligase and 10 units of the appropriate restriction endonuclease (SapI, BsaI or BbsI) in a total reaction volume of 20 µl. Reactions were incubated using a thermocycler for 30 two-step cycles of 37 °C for 5 minutes then 16 °C for 5 minutes, before a single final denaturation step at 65 °C for 20 minutes.

2.4.15 Membrane dialysis of DNA

The products of Start-Stop Assembly reactions and ligation reactions were dialysed through a Millipore 0.025 µm nitrocellulose membrane over deionised water for 60 minutes prior to transformation.

2.4.16 Nucleotide sequencing

Nucleotide sequencing was carried out by Source Bioscience (Cambridge, UK) using Sanger DNA sequencing.

2.4.17 Coding sequences for carotenoid metabolic pathway libraries

The eight CDSs used in the assembly of carotenoid metabolic pathway libraries were *dxs* (encoding 1-deoxyxylulose-5-phosphate synthase; WP_074468184) from *E. coli* MG1655, *crtE* (encoding geranylgeranyl pyrophosphate synthase; WP_010888034) from *Deinococcus radiodurans* R1, *crtB* (encoding phytoene synthase;

WP_010887508) from *D. radiodurans* R1, *crtI* (encoding phytoene desaturase; WP_010887507) from *D. radiodurans* R1, *idi* (encoding isopentenyl pyrophosphate isomerase; AAC32208) from *Haematococcus pluvialis*, *lcyB* (encoding beta-lycopene cyclase; ABR57232) from *Solanum lycopersicum*, *crtZ* (encoding β-carotene hydroxylase; WP_072137426) from *Pantoea ananatis* and *crtW* (encoding β-carotene ketolase; BAB74888) from *Nostoc sphaeroides* PCC 7120. Each CDS was codon-optimised for expression in *E. coli* using the OPTIMIZER guided random methodology (131). Iterations of the optimisation were repeated until each CDS had a codon adaptation index (CAI) score >0.65. Each CDS was synthesised as a linear gBlock DNA fragment (Integrated DNA Technologies).

2.5 Experimental Assays

2.5.1 *E. coli* library characterisation by flow cytometry

E. coli was grown, as described in section **2.5.1.** Briefly, *E. coli* DH10B cells were transformed with purified plasmids. Isolated transformant clones were used to inoculate 200 µl LB broth supplemented with the appropriate antibiotic, which was cultured overnight (16 h) at 37 °C with shaking at 700 r.p.m. Overnight cultures were subcultured by 1:1000 dilution into 200 µl ml fresh LB medium with the appropriate antibiotic and grown for 4 h at 37 °C with shaking at 700 r.p.m. Cultures were diluted 1:50 in filtered PBS and were immediately subjected to flow cytometer analysis. A two-laser FACScan flow cytometer (Becton-Dickinson) equipped with an automated multiwell sampler (AMS) was used for characterisation. The sample was analysed at a data rate of 1000 – 2000 events sec^{-1} for 30 sec. A 488 nm argon ion laser was used for excitation of the EYFP reporter detecting through a 530 nm band-pass filter (FL1) with a gain of 900 V. All part characterisation assays were conducted in quintet.

2.5.2 *Synechocystis* library characterisation by flow cytometry

Synechocystis was grown as described in section **2.5.2**. Briefly, purified plasmids were transformed into *Synechocystis* and subcultured to complete segregation. Cultures were cultivated in 5 ml BG11 medium from a loopful of transformed cells and incubated in 25 ml vented culture flasks under phototrophic conditions, 30 °C, 150 r.p.m, 50 µmol photons m^{-2} s^{-1} light and incubated for 48 h. Following incubation, cultures were diluted to an OD750 of 0.1 and incubated for a further 24 hours. For flow cytometer analysis, a two-laser FACScan flow cytometer (Becton-Dickinson) equipped with an automated multiwell sampler (AMS) was used. *Synechocystis* cultures were diluted 1:50 in filtered

PBS in a 96-well clear plate (Costar) and immediately analysed at a data rate of 1000 – 1500 events s^{-1} for 45 sec. A 488 nm argon ion laser was used for excitation of the EYFP reporter detecting through a 530 nm band-pass filter (FL1) with a gain of 700 V. All part characterisation assays were conducted in triplicate.

2.5.3 Portability validation characterisation in *E. coli* and *Synechocystis* by flow cytometry

E. coli and *Synechocystis* were grown as described in section **2.3.1** and **2.3.2.** Cultures were diluted 1:50 in filtered PBS and immediately subjected to flow cytometer analysis. EYFP fluorescence was measured using an Attune NxT flow cytometer (Invitrogen). The voltage gains for each detector were set to ensure all constructs fell within the dynamic range: FSC, 440 V; SSC 440 V; BL1, 500 V. 10,000 events gated by FSC-H and SSC-H that represent the *E. coli* population were collected at 100 µl min^{-1} sample flow rate.

2.5.4 Analysis of Level 1 assembly fidelity and bias by flow cytometry

E. coli DH10B cells were transformed with Start-Stop Assembly reactions. Isolated transformant clones or pools of transformant clones were used to inoculate 5 ml LB broth supplemented with the appropriate antibiotic, which was cultured overnight (16 h) at 37 °C with shaking at 225 r.p.m. For flow cytometry analysis, overnight cultures were subcultured by 1:1000 dilution into 5 ml fresh LB medium with the appropriate antibiotic and grown for 6 h at 37 °C with shaking at 225 r.p.m. Cultures were diluted 1:50 in filtered PBS and immediately subjected to flow cytometer analysis. Fluorescence was measured using an Attune NxT flow cytometer (Invitrogen). The voltage gains for each detector were set to ensure all constructs fell within the dynamic range: FSC, 440 V; SSC, 440 V; BL1, 500 V. 10,000 events gated by forward scatter height (FSC-H) and side scatter height (SSC-H) that represent the *E. coli* cell population were collected at 100 µl min^{-1} sample flow rate. Flow cytometry data were analysed using FlowJo (http://flowjo.com). Cells were gated by FSC-H and SSC-H and the geometric mean from the BL1 detector was exported as the fluorescence value.

2.5.5 Analysis of DNA scars and characterisation of RBS library by flow cytometry

E. coli DH10B cells were transformed with the appropriate plasmids and individual transformant colonies were used to inoculate 200 µl LB supplemented with ampicillin,

which was incubated overnight (16 h) in plates of 96 U-shaped 1.2 ml wells covered with sterile breathable sealing film (Breathe Easy) at 37 °C with shaking at 700 r.p.m on a Multitron shaker (Infors-HT). For flow cytometer assays, overnight cultures were subcultured 1:1000 into 200 µl fresh LB medium that contained ampicillin and the *rhaBAD* promoter inducer L-rhamnose (0.6 mg ml^{-1}), in deep-well plates and grown for 6 h at 37 °C with shaking at 700 r.p.m. Cultures were diluted 1:50 in filtered PBS and immediately subjected to flow cytometer analysis. GFP fluorescence was measured using an Attune NxT flow cytometer (Invitrogen) as described in section **2.5.4.** Analysis of Level 1 assembly fidelity and bias by flow cytometry.

2.5.6 Glucuronidase assay in *E. coli*

E. coli strains transformed with gusA expression plasmids were grown as described in **2.3.1**. Samples (1 ml) were harvested by centrifugation and pellets were frozen at −20 °C. Before testing, pellets were resuspended in 0.8 mL of buffer Z (60 mM $Na_2HPO_4 \cdot 7H_2O$, 40 mM $NaH_2PO_4 \cdot H_2O$, 10 mM KCl, 1 mM $MgSO_4 \cdot 7H_2O$, pH adjusted to 7.0, and 50 mM 2-mercaptoethanol added freshly). 0.2 ml of each sample was used for OD600 nm measurement. To the remaining sample (0.6 ml), toluene (6 µl) was added, tubes were vortexed for 1 min and incubated on ice for 10 min. Tubes were transferred into a 37 °C heating block and preincubated for 30 min with caps open. Reactions were started by addition of 120 µl of 6 mM p-nitrophenyl-β-D-glucuronide (Merck Millipore) solution in buffer Z. After incubation at 37 °C (5−30 min), reactions were stopped by addition of 1 M Na_2CO_3 (300 µl) and reaction time was recorded. Cell debris was removed by centrifugation at 10,000g for 10 min. Supernatants were transferred into polystyrene spectrophotometer cuvettes and absorbance at 405 nm was determined (Eppendorf BioSpectrometer kinetic). Relative glucuronidase activity was calculated by dividing the absorbance at 405 nm by sample OD600 nm and incubation time.

2.5.7 Glucuronidase assay in *Synechocystis*

Synechocystis strains transformed with gusA expression plasmids were grown as described in **2.3.2**. Samples (1 ml) were harvested by centrifugation and pellets were frozen at −20 °C. Before testing, pellets were resuspended in 0.8 mL of buffer Z (60 mM $Na_2HPO_4 \cdot 7H_2O$, 40 mM $NaH_2PO_4 \cdot H_2O$, 10 mM KCl, 1 mM $MgSO_4 \cdot 7H_2O$, pH adjusted to 7.0, and 50 mM 2-mercaptoethanol added freshly). 0.2 ml of each sample was used for OD750 nm measurement. The remaining sample (0.6 ml) was subject to bead beating for 10 minutes at full speed using a Vortex-Genie 2 fit with the Vortex

Adapter for Genie 2 Vortex Mixer to completely break the cells. The samples were then centrifuged at 10,000g for 10 min and the supernatants (0.5 ml) transferred to a fresh 1.5 ml microtube and incubated on ice for 10 min. Tubes were transferred into a 37 °C heating block and preincubated for 30 min with caps open. Reactions were started by addition of 120 µl of 6 mM p-nitrophenyl-β-D-glucuronide (Merck Millipore) solution in buffer Z. After incubation at 37 °C (5−30 min), reactions were stopped by addition of 1 M Na_2CO_3 (300 µl) and reaction time was recorded. Cell debris was removed by centrifugation at 10,000g for 10 min. Supernatants were transferred into polystyrene spectrophotometer cuvettes and absorbance at 405 nm was determined (Eppendorf BioSpectrometer kinetic). Relative glucuronidase activity was calculated by dividing the absorbance at 405 nm by sample OD750 nm and incubation time.

2.5.8 Analysis of phenotypic diversity of carotenoid pathway library clones by image analysis

E. coli DH10B cells were transformed with 5 µl of Start-Stop Assembly reaction product, allowed to recover for one hour, plated and grown at 30 °C for 48 h. Transformation plates were photographed under consistent conditions using a Canon 60D camera. The images were analysed using OpenCFU v.3.8 beta (411), and colonies were detected using an inverted threshold of 3 and a minimum radius of 10 pixels and a maximum radius of 500 pixels. The area and red, green, blue colours of each colony were determined.

2.5.9 Quantification of lycopene content of *E. coli* - Spectrophotometer

E. coli was cultured from a single colony in 500 µl LB supplemented with the appropriate antibiotics in a 96 U-shaped 1.2 ml well-plate covered with sterile breathable sealing film (Breathe Easy) and grown at 30 °C at 700 r.p.m for 48 h in a Multitron shaker (Infors-HT). Following incubation, the liquid cultures were centrifuged at 4000 r.p.m for 15 minutes before removing the supernatant and adding 100 ul of methanol and 200 ul of hexane to the bacterial cell pellets. After vortexing for 2 minutes, 100 ul of deionised water was added before vortexing the mixture again for 2 minutes. The solution underwent centrifugation at 4000 r.p.m for 20 minutes and the top hexane layer was transferred to a shallow 96-well plate and absorbance at 530 nm was determined (BMG LABTECH POLARstar Omega).

2.5.10 Quantification of lycopene content of *E. coli* - LCDAD

E. coli was cultured in tubes containing 5 ml LB supplemented with the appropriate antibiotics and grown at 37 °C at 225 r.p.m for 24 h. To obtain the dry cell weight (DCW), after 24 h cultivation 2 ml of liquid culture was centrifuged at 14,000 r.p.m for 5 minutes to remove supernatant, the bacterial cell pellet was washed in deionised water and then centrifuged again and dried at 70 °C until a constant weight was obtained. To obtain lycopene concentrations, following 24 hours incubation 2 ml of liquid culture was centrifuged at 14,000 r.p.m for 5 minutes to remove supernatant, the bacterial cell pellet was washed in deionised water and then centrifuged as before. The bacterial cell pellet was resuspended in 1 ml acetone and incubated at 55 °C for 15 minutes to extract lycopene. The supernatant was obtained by filtration through a 0.22 µm pore-size nylon membrane for LC-DAD analysis.

Lycopene was detected and measured using an Agilent LC system with UV/Vis diode array detector. Absorbance at 450 nm and 471 nm were monitored and the peak area corresponding to each component integrated to provide a measure of abundance. The LC column used was an Acquity UPLC Peptide BEH C18 column (2.1 x 100mm, 1.7 µm, 300Å, Waters). LC buffers were 50% of methanol in water (A) and 25% of ethyl acetate in acetonitrile (B). All the solvents used were HPLC grade. The LC method was 6.5 minutes in total with 1.5 minutes of post run time (0-1 min: 30% A, 70% B; 1-6 min: 0.1% A, 99.9% B; 6-6.5 min: 30% A, 70% B; at a flow rate of 0.3 ml/min). The injection volume for the samples was 0.5 µl. Commercially available lycopene (Sigma-Aldrich) was dissolved in acetone as a standard and a standard curve was generated.

2.5.11 Quantification of lycopene content of *Synechocystis* - LCDAD

In order to quantify carotenoid content, purified plasmids were transformed into *Synechocystis* and subcultured to complete segregation. Cultures were cultivated in 5 ml BG11 medium, supplemented with kanamycin (30 µg ml^{-1}), from a loopful of transformed cells and incubated in 25 ml vented culture flasks under phototrophic conditions, 30 °C, 150 r.p.m, 50 µmol photons m^{-2} s^{-1} light and incubated for 48 h. Following incubation, cultures were diluted to an OD750 of 0.1 and incubated for a further 2 weeks. To obtain the dry cell weight (DCW), after 2 weeks cultivation 2 ml of liquid culture was centrifuged at 14,000 r.p.m for 5 minutes to remove supernatant, the bacterial cell pellet was washed in deionised water and then centrifuged again and dried at 70 °C until a constant weight was obtained. To obtain lycopene concentrations, following 2 weeks incubation 2 ml of liquid culture was centrifuged at 14,000 r.p.m for

5 minutes to remove supernatant, the bacterial cell pellet was washed in deionised water and then centrifuged as before. The bacterial cell pellet was resuspended in 1 ml acetone and incubated at 55 °C for 15 minutes to extract lycopene. The supernatant was obtained by filtration through a 0.22 µm pore-size nylon membrane for LC-DAD analysis. Lycopene was detected and measured using an Agilent LC system with UV/Vis diode array detector as described in section **2.7.10** Quantification of lycopene content of *E. coli* - LCDAD.

2.6 Data Analysis

2.6.1 Flow cytometry

Flow cytometry data was analysed using FlowJo (http://flowjo.com). Cells were gated by FSC-H and SSC-H and the geometric mean from the BL1 detector was exported as the EYFP and GFP fluorescence.

2.7 Computational Methods

SUBSETTER was developed in the TextWrangler (Bare Bones Software) using Python as the programming language. For the detailed scrip please refer to Algorithm A1.

2.8 Plasmid Construction

Plasmids, oligonucleotides and synthetic DNA used in this study are shown in Table A8, Table A4 and Table A5, respectively. All oligonucleotides and gBlock linear DNA fragments were synthesised by Integrated DNA Technologies Inc. All Start-Stop Assembly vector sequences have been deposited in GenBank (accession numbers shown in Table 5.1).

2.8.1 Library construction

The SPL, Promoter Library 2 and RBS Library were generated by inverse PCR with 5′-phosphorylated degenerated primers using pATM2, SPLc192 and SPLc3, respectively as templates. After the PCR, *DpnI* was added to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. The SPL was generated using the degenerate primers oligoATM3 and oligoATM5. Four Promoter Library 2 libraries were generated using the degenerate primers oligoGT22, oligoGT23, oligoGT24, oligoGT25 and oligoGT26. Nine RBS libraries were generated using the degenerate primers oligoATM51 – 60.

2.8.2 EYFP combinatorial build

pGT16 – 51 were constructed by inverse PCR with 5′-phosphorylated primers (oligoGT114 – 125) using pATM2 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.3 Different combinatorial reporter libraries

To construct pGT11, a synthetic gBlock DNA fragment (dnaGT2) encoding *efyp* version 3 coding sequence was digested with *PstI* and *BamHI* and ligated to vector pATM2, which had been cut with the same enzymes. pGT52 – 56 were constructed by inverse PCR with 5′-phosphorylated primers (oligoGT128 – 133) using pGT11 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

pGT12 was constructed by inverse PCR with 5′-phosphorylated primers (oligoGT126 and 127) using pGT30 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. pGT57 – 61 were constructed by inverse PCR with with 5′-phosphorylated primers (oligoGT128 – 133) using pGT12 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

PGT62 – 67 were constructed by inverse PCR with 5′-phosphorylated primers (oligoGT133 – 139) using pATM3 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

To construct pGT13, the *GusA* coding sequence of plasmid pPM6 (408) was PCR-amplified (oligoGT142 and 143) and the vector backbone of pGT30 was PCR amplified (oligoGT140 and 141). The resultant DNA fragments were ligated together by blunt-end ligation. pGT68 – 72 were constructed by inverse PCR with 5′-phosphorylated primers (oligoGT133 and 144 – 148) using pGT13 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.4 Start-Stop Assembly plasmids

2.8.4.1 *Construction of Level 0 vector pStA0*

Level 0 vector pStA0 was derived from pUC19. To silence and remove relevant type IIS restriction sites, pGT421 was generated. To generate pGT421 the backbone was PCR-amplified in two fragments (oligoGT249 – oligoGT252) using pUC19 as a template and assembled together by SOE-PCR. The backbone was assembled with a synthetic gBlock DNA fragment (dnaGT6) by blunt-end ligation. To generate pStA0 the backbone sequence of pGT421 was PCR-amplified (oligoGT253 and oligoGT254) and blunt-end ligated with a *lacZα* fragment PCR-amplified (oligoGT255 and oligoGT256) from pUC19.

2.8.4.2 *Construction of Level 1 vectors*

The nine Level 1 vectors (Table A8) were derived from pACYC184. To silence and remove relevant type IIS restriction sites from the pACYC184 backbone, pGT422 was generated. To generate pGT422 the backbone was PCR-amplified in two fragments (oligoGT257 – oligoGT260) using pACYC184 as a template and assembled together by SOE-PCR. The vector backbone was then blunt-end ligated with a *lacZα* fragment PCR-amplified (oligoGT261 and oligoGT262) from pUC19.

To generate pStA1AZ the *lacZα* fragment was PCR-amplified (oligoGT356 and oligoGT357) and the vector backbone were PCR-amplified (oligoGT266 and oligoGT265) using pGT422 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. To generate pStA1BC the *lacZα* fragment was PCR-amplified (oligoGT363 and oligoGT366) and the vector backbone were PCR-amplified (oligoGT268 and oligoGT270) using pGT422 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. To generate pStA1CD the *lacZα* fragment was PCR-amplified (oligoGT364 and oligoGT367) and the vector backbone were PCR-amplified (oligoGT268 and oligoGT273) using pGT422 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. To generate pStA1DE the *lacZα* fragment was PCR-amplified (oligoGT365 and oligoGT368) and the vector backbone were PCR-amplified (oligoGT268 and oligoGT276) using pGT422 as a template and assembled together by SOE-PCR,

followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

Plasmids pStA1AB, pStA1BZ, pStA1CZ, pStA1DZ and pStA1EZ were generated by inverse PCR with 5′-phosphorylated primers (oligoGT268, 279, 358 – 362) and pStA1AZ as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.4.3 *Construction of Level 2 vectors*

The three Level 2 vectors (Table A8) were derived from pACYC177. To silence and remove any relevant type IIS restriction sites from the pACYC177 backbone, pGT410 was generated. To generate pGT410 the vector backbone from pACYC177 was PCR-amplified (oligoGT482 and oligoGT483) and blunt-end ligated with a *lacZα* fragment PCR-amplified from pStA0 (oligoGT261 and oligoGT262). To generate pGT411 the *lacZα* fragment was PCR-amplified (oligoGT377 and oligoGT378) and the vector backbone were PCR-amplified (oligoGT289 and oligoGT265) using pGT410 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. To generate pGT412 the *lacZα* fragment was PCR-amplified (oligoGT379 and oligoGT381) and the vector backbone was PCR-amplified (oligoGT380 and oligoGT265) using pGT410 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. To pGT413 the *lacZα* fragment was PCR-amplified (oligoGT382 and oligoGT384) and the vector backbone was PCR-amplified (oligoGT265 and oligoGT383) using pGT410 as a template and assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

Plasmids pStA212, pStA223 and pStA234 were generated by inverse PCR with 5′-phosphorylated primers (oligoGT569 and oligoGT570) using pGT411, pGT412 and pGT413 as templates, respectively, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.4.4 *Construction of Level 3 vectors*

The Level 3 vectors (Table A8) were derived from pACYC184. To remove any relevant type IIS restriction sites from the pACYC184 backbone, pGT414 was generated. To

generate pGT414 the vector backbone from pACYC184 was PCR-amplified (oligoGT297 and oligoGT298) and blunt-end ligated with a *lacZα* fragment PCR-amplified from pStA0 (oligoGT261 and oligoGT262).

To generate pStA313, the *lacZα* fragment was PCR-amplified (oligoGT385 and oligoGT386) and the vector backbone was PCR-amplified (oligoGT265 and oligoGT299) using pGT414 as a template. The two DNA fragments were assembled together by SOE-PCR, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

The plasmid pStA314 was generated by inverse PCR with 5′-phosphorylated primers (oligoGT265 and oligoGT387) and pStA313 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.4.5 *Parts cloned into pStA0*

To generate pStA0::*eyfp* the *eyfp* coding sequence was PCR-amplified from the reporter plasmid pATM2 (124) (oligoGT614 and oligoGT615). The gel-purified PCR product was directly used in a Level 0 Start-Stop Assembly reaction with pStA0. The Level 0 reaction product was then used to transform *E. coli* and the plasmids of isolated transformant colonies were purified and sequence verified.

To generate the eight plasmids pStA0::*crtI*, pStA0::*crtB*, pStA0::*crtE*, pStA0::*dxs*, pStA0::*crtW*, pStA0::*crtZ*, pStA0::*lcyB* and pStA0::*idi* codon-optimised CDSs (Table A8) were synthesised as linear gBlock DNA fragments (Integrated DNA Technologies) and cloned into pStA0. Linear gBlock DNA fragments were used directly in Level 0 Start-Stop Assembly reactions with pStA0. The Level 0 reaction product was then used to transform *E. coli* and the plasmids of transformant colonies were purified and sequence verified.

The promoters, RBSs and terminators were cloned into pStA0 by inverse PCR with 5′-phosphorylated primers (oligoGT317 – 336, 436 – 447, 503 – 505, 507 – 510, 512 – 517, 568 and 580 – 581) and pStA0 as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

2.8.4.6 *Metabolic pathways*

Lycopene, β-carotene and astaxanthin encoding pathways were assembled using Start-Stop Assembly as described in the sections **5.2.6** and **6.2.3**.

2.8.4.7 *Other plasmids used in study*

To generate pGT270, the pSHUTTLE2 backbone was PCR-amplified (oligoGT373 and oligoGT374) and assembled with the Level 2 Start Stop Assembly cassette that had been PCR-amplified from pStA212 (oligoGT375 and oligoGT376) by SOE-PCR. The resultant PCR reaction was treated with *DpnI* to digest the template, then circularisation by ligation with T4 ligase.

The construct pGT274 was generated by inverse PCR with 5′-phosphorylated primers (oligoGT468 and oligoGT472) using pCK302 (123) as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. The construct pGT550 was generated by inverse PCR with 5′-phosphorylated primers (oligoGT660 and oligoGT657) using pCK302 (123) as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. The construct pGT551 was generated by inverse PCR with 5′-phosphorylated primers (oligoGT660 and oligoGT658) using pCK302 (123) as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase. The construct pGT552 was generated by inverse PCR with 5′-phosphorylated primers (oligoGT660 and oligoGT659) using pCK302 (123) as a template, followed by addition of *DpnI* to the PCR reaction to digest the template, then circularisation by ligation with T4 ligase.

3. CHAPTER THREE. Generation of a genetic toolkit for the cyanobacterium *Synechocystis* sp. PCC 6803.

3.1 Introduction

3.1.1 Portable expression control parts

To apply rapid prototyping in the non-model organism *Synechocystis*, a metabolic pathway would first be assembled, tested and optimised in the model organism *E. coli*. Following this, the metabolic pathway-encoding construct would be transferred to *Synechocystis* where, in principle it would perform the same, so that no further optimisation is required in the non-model organism (Figure 3.1). For this to work, each gene in the metabolic pathway-encoding construct would have to express proteins at a similar concentration in each organism (Figure 3.1). In terms of rapid prototyping, this would ensure the proteins were still expressed at the optimal level and that the ratio of proteins within the pathway were maintained in *Synechocystis*, after the *E. coli* optimisation.

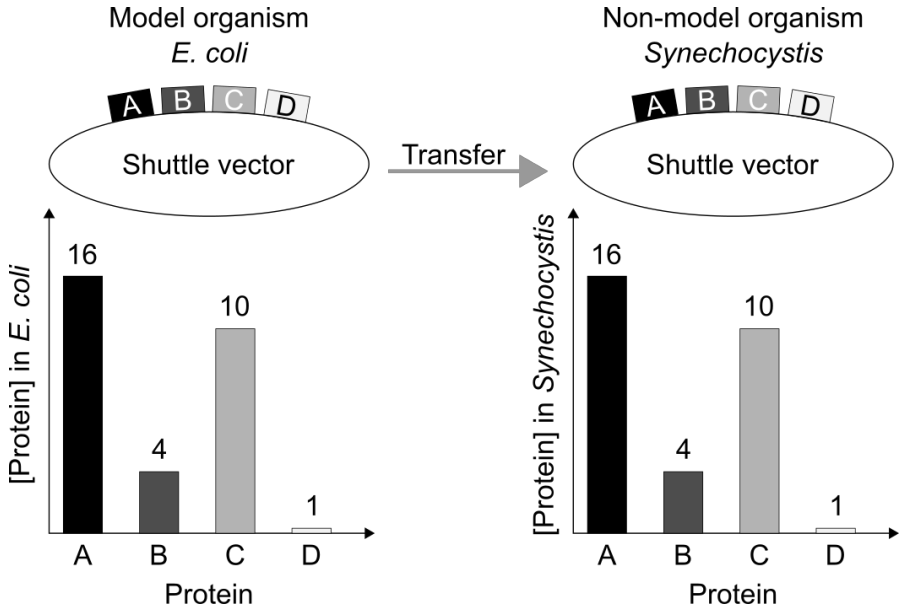


Figure 3.1. Illustrative example for the concept of rapid prototyping. This figure shows that protein concentrations are preserved when a metabolic pathway-encoding construct is transferred between organisms. To allow apply rapid prototyping similar protein concentrations must be expressed by the same metabolic pathway-encoding construct in the model and non-model organism.

To allow protein concentrations to be preserved between the two organisms, expression control parts would be required that performed the same or similarly between the non-model and model organism (Figure 3.2c) and which would therefore be portable between the two organisms. Furthermore, the portability of the expression

control parts would need to behave similarly between the two organisms when composed in expression units and metabolic pathway-encoding constructs, where the expression control parts would be subject to different genetic contexts. In other words, the effects that genetic contexts have on the expression control parts would need to be similar in both the non-model and model organism. Portable expression control parts that perform the same between the pair of organisms would ideally express proteins at the same concentration between the two organisms, but in reality, similar relative concentrations might be more realistic. Therefore, rapid prototyping would rely on portable expression control parts (hereafter referred to as portable parts), which would allow the performance of a metabolic pathway in *E. coli* to be used as a prediction of the pathway's performance in *Synechocystis*.

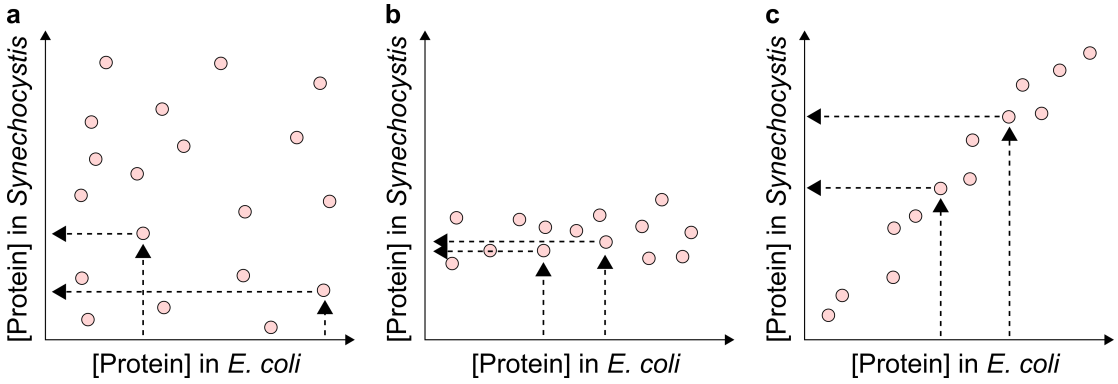


Figure 3.2. Illustrative examples of different performances of expression control parts between organisms. Expression control parts that do not result in similar protein concentrations between *E. coli* and *Synechocystis* would not serve well as portable parts (panel a and b). Portable expression control parts need to result in similar protein concentrations between *E. coli* and *Synechocystis* (panel c).

In reality, if the state-of-the-art expression control parts available at the start of this project, including the strong native cyanobacterial promoters, P_{psbA2}, P_{rbcL}, $P_{sll1220}$ and P_{cpcB} (282, 328, 384, 385, 412-416), were used to assemble a metabolic pathway-encoding construct, the activity of these expression control parts would typically change when the construct was transferred from one organism, in this case *E. coli*, to the other, in this case *Synechocystis* (Figure 3.2a and b). This in turn would lead to different concentrations of the pathway proteins (Figure 3.3) and could result in a change in the pathways performance. For example, when a modified CoA-dependent 1-butanol production pathway was transferred from *Clostridium* to *Synechococcus elongatus* PCC 7942, the resultant strain did not produce 1-butanol (287), however this was as a result of the pathways sensitivity to oxygen, as the pathway was derived from a strict anaerobe. This was also the case when the refactored *nif* gene cluster was transferred from *Klebsiella oxytoca* to *E. coli*, where there was a decrease from

7% to 2% of the wild-types activity (149). Therefore, with these state-of-the-art parts, it would not be possible to optimise the performance of a metabolic pathway in a model organism before transferring it to the non-model organism and to expect it to perform the same.

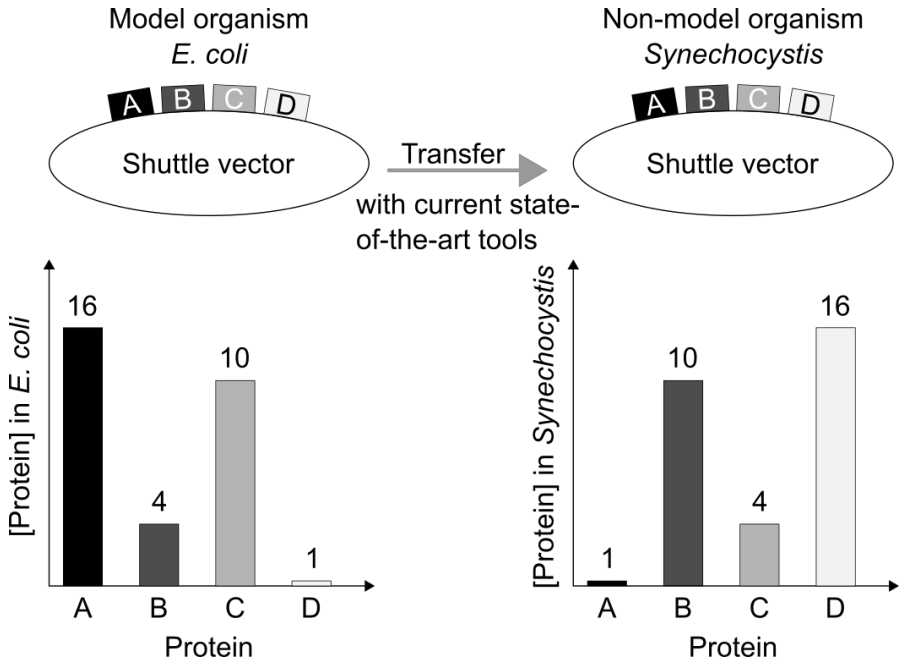


Figure 3.3. Illustrative example showing expected response of protein concentrations from transferring a metabolic pathway-encoding construct between organisms using current state-of-the-art expression tools. The activity of expression control parts is altered when used in a different organism, therefore they would not preserve protein concentrations in a given metabolic pathway-encoding construct when transferred to a different organism.

3.1.2 Expression control parts

At the start of the project, no portable expression control parts had been developed for any organisms and the cyanobacterial toolkit was limited to native promoters, such as the strong constitutive promoters including P_{psbA2}, P_{rbcL}, $P_{sll1220}$ and P_{cpcB} (282, 328, 384, 385, 412-416). These strong promoters offer little ability to fine-tune protein concentrations and some, such as $P_{sll1220}$, are not functional in *E. coli*. Therefore the state-of-the-art tools that were available at the start of this project were poor candidates for portable parts.

Since the start of this project, two cross-species portable expression parts have been developed. The first used an orthogonal viral T7 RNA polymerase, which avoids interaction with the native regulation and function of host cell, to control protein expression in *E. coli*, *Pseudomonas putida* and *B. subtilis*. The portable expression

system, termed UBER, is functional in the three organisms, however it does not preserve protein concentrations between the three organisms and therefore would not be suitable for rapid prototyping (409). The second portable expression part developed uses an orthogonal T7-like RNA polymerase and promoter, derived from Morganella phage MmP1, which allows metabolic pathways to be transferred between *E. coli* and *Halomonas*, while preserving the metabolic pathways performance (R^2 > 0.94) (246).

Since the start of this project, a considerable amount of effort has been put into developing the cyanobacterial toolkit. More strong constitutive promoters have been developed (328, 384, 385, 415), however they are not ideal for fine-tuning protein expression levels. Two constitutive promoter libraries have been reported, one offering a 3-log dynamic range (352) that was based on P_{cpcB} and the other composed of 17 native and chimeric promoters focused on generating stronger rather than weaker promoters (383). Whilst this is a step in the right direction for developing cyanobacterial expression control parts, these parts are still not ideal for fine-tuning metabolic pathway-encoding constructs. This is because the libraries do not offer a wide enough range of strengths that increase by small increments, which is evident when compared to those that have developed for model organisms such as *E. coli* (82).

The P_{rhaBAD} inducible promoter system was recently demonstrated as a non-toxic, photostable, non-metabolisable inducer, with a linear response to inducer concentrations and no basal transcription in the absence of inducer in *Synechocystis* (124). Prior to that, the available inducible promoter systems described in cyanobacteria were far from ideal. For example, the metal-ion inducible promoters described (417-421) require a toxic level of inducer concentration for detectable and unimodal expression (418). Furthermore, many of the metals are found in the standard cyanobacterial growth media (410) resulting in low levels of induction. The anhydrotetracyline (aTc)-inducible promoter system has shown good dynamic range matched with low basal transcription in the absence of inducer (422, 423), however aTc is extremely sensitive to light and therefore a poor choice of inducer for a photoautotrophic organism. Lastly, isopropyl β-D-1thiogalactopyranoside (ITPG), the non-metabolisable analogue of lactose, has been used as an inducer to *lac*-based promoters in cyanobacteria, but basal transcription in the absence of inducer was observed (287, 295, 296, 306, 352, 424, 425).

While RBSs have been extensively used to control protein levels in model organisms (426-429), surprisingly even fewer RBSs have been characterised, compared to

promoters, in cyanobacteria. Prior to this project, the native RBSs for the *psbA2* and *rbcL* genes (417) were the only characterised RBSs in cyanobacteria. Since this project has started, three RBS libraries have been described, one composed of 20 native RBSs (385), another by making rational mutations to the RBS sequence of a synthetic RBS that was generated using the RBS Calculator (383) and lastly, a library of eight RBSs generated using the RBS Calculator (352).

At the start of this project few transcriptional terminators had been utilised in cyanobacteria, only the RuBisCO terminator (430) and strong *E. coli* terminators (291, 431). Furthermore, since the start of the project, ten native cyanobacterial terminators have been characterised (385), further increasing the number of available terminators.

To date, shuttle vectors have been developed and used for engineering cyanobacteria. Using a shuttle vector allows metabolic pathway-encoding constructs to be assembled in fast-growing and tractable model organisms, typically *E. coli*, before they are transferred to cyanobacteria. Currently there are two types of shuttle vector that have been developed for cyanobacteria; integrative and replicative. Integrative plasmids use homologous recombination to integrate the genetic construct into the cyanobacteria genomes (266, 370). Replicative plasmids replicate in the cyanobacteria. There are two types of replicative plasmids: those that use replicons from broad-host range plasmids (412), such as the RSF1010 plasmids that can be used in a wide range of Gram negative organisms, including *E. coli* and many strains of cyanobacteria (432, 433); and those which are derived from endogenous plasmids from cyanobacteria (282, 434). While replicative plasmids can be quicker to introduce to cyanobacteria, they can exhibit some instability (435, 436).

When using integrative shuttle vectors in cyanobacteria, it is preferable that a neutral site is used for the integration. A neutral site, which is used in cyanobacteria, but not broadly in other organisms, is one that can be disrupted by integration of ectopic DNA without affecting cellular viability or without causing any distinguishable phenotype. At the start of this project, a limited number of neutral sites had been described (364, 437-439), however since then, a number of neutral sites in cyanobacteria genomes (440-442) have been reported.

Obtaining an optimal performance of a metabolic pathway typically requires the intricate balance of protein concentrations, which can often be achieved by fine-tuning their protein expression levels (111). Cyanobacteria lack the genetic tools required for

such engineering and fine-tuning of protein expression levels and therefore it is essential to generate expression control parts for cyanobacteria that will allow heterologous and synthetic metabolic pathways to be fine-tuned. Furthermore, with rapid prototyping in mind, portable parts must be developed that allow the engineering and fine-tuning of metabolic pathways in both *E. coli* and *Synechocystis*. Therefore, firstly, considering the lack of available cyanobacterial expression control parts, particularly at the start of this project, and secondly, rapid prototyping's heavy reliance on the portable parts, this project was started by generating libraries of promoters and RBSs and investigating how they performed between *E. coli* and *Synechocystis*.

3.1.3 Aims

The aim of this chapter is to develop the portable expression control parts that are needed to implement rapid prototyping in *Synechocystis* sp. PCC 6803 using *Escherichia coli* as the fast-growing model organism. With no prior understanding on how to generate portable parts for these organisms, or at the start of this project, for any organisms, libraries of constitutive promoters and ribosome-binding sites (RBSs) were generated. In an attempt to identify portable parts and understand how these parts perform between these two organisms, their performance was compared between the two organisms. This approach also had a secondary advantage; to fill the deficit of cyanobacterial expression control parts that were lacking at the start of this project. The aims of this chapter can be simplified into three points:

- To develop a library of constitutive promoters that offers a wide range of strengths for controlling transcriptional initiation in *E. coli* and *Synechocystis*.
- To develop a library of ribosome-binding sites that offer a wide range of strengths for controlling translational initiation in *E. coli* and *Synechocystis*.
- To compare the relationship in their performance between the two organisms, to assess their potential as portable parts for rapid prototyping.

Declaration – Antonio Torres Méndez an MRes student helped at the start of this project by constructing the reporter plasmid, pATM2 and also helped in the design and construction of the SPL promoter library.

3.2 Results

3.2.1 Characterising expression control parts in *E. coli* and *Synechocystis*

To allow the characterisation of promoters and RBSs in both *E. coli* and *Synechocystis*, the reporter-shuttle plasmid, pATM2 (Figure 3.4), was used. pATM2 had previously been constructed in the Heap lab and used to characterise the inducible P_{rhaBAD} promoter system in *Synechocystis* sp. PCC 6803 (hereafter referred to as *Synechocystis*) (124). The plasmid was designed to replicate in *E. coli* using the low-copy origin of replication (ORI), p15A, which generates approximately 15-20 copies of the plasmid per cell (161). To integrate into *Synechocystis*' genome the shuttle vector contains two homology arms that integrate at the *ssl0410* locus, a previously described neutral site adjacent to the *ndhB* locus (439). Longer homology arms compared to those typically reported (approximately 500 base pairs (440)) were used, which were one kilobase (kb) in size, to provide more efficient homology when transforming larger constructs. The pATM2 vector contains a kanamycin resistance gene that encodes an aminoglycoside phosphotransferase, the expression of which provides resistance to kanamycin in both *E. coli* and *Synechocystis*. pATM2 contains the fluorescent reporter enhanced yellow fluorescent protein (EYFP; BioBrick repository Part BBa_E0030), which could be used to characterise expression control parts by using its fluorescence as an indirect measure of part strength in both cyanobacteria (124, 352, 385, 412, 416, 422, 441, 443) and *E. coli* (352). EYFP was chosen because it displayed a better signal/noise ratio in *Synechocystis* compared to the commonly used sfGFP (unpublished data, John Heap, pers comm).

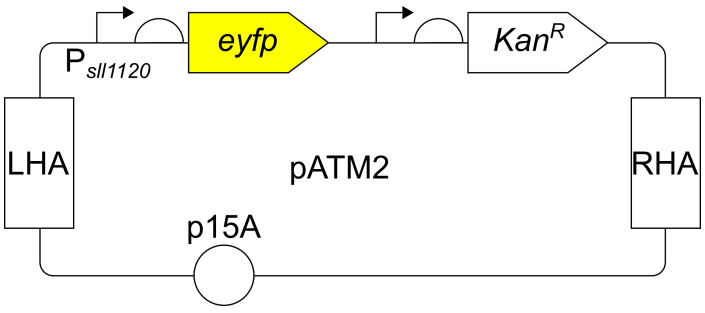


Figure 3.4. *E. coli-Synechocystis* shuttle vector pATM2 containing an *eyfp* reporter gene. The plasmid contains the low-copy number origin of replication p15A that allows replication of the shuttle vector in *E. coli*. Left (LHA) and right (RHA) homology arms that integrate at the *ssl0410* loci of *Synechocystis'* genome. Between the homology arms an *eyfp* coding sequence under promoter $P_{sll1120}$ that encoded the enhanced yellow fluorescent protein and KanR that encodes an aminoglycoside phosphotransferase that confers resistance to kanamycin in both *E. coli* and *Synechocystis*.

3.2.2 Promoter library

To date, promoters have been the most widely used expression control part to engineer protein expression and therefore were the obvious choice of portable part. At the start of this project, there were no portable promoters and very few characterised cyanobacterial promoters available, other than strong native promoters, such as P_{psbA2}, $P_{rbcL,}$ and P_{cpcB}, which had not been characterised in *E. coli* nor offered much fine-tuning ability. This project was started by attempting to generate a library of synthetic promoters, which were functional in both *E. coli* and *Synechocystis*. With this library, it would be possible to assess the strength of each synthetic promoter clone by using EYFP fluorescence intensities as an indirect measure of promoter strength in both *E. coli* and *Synechocystis*. From this it would be possible to assess the portability of the synthetic promoter library between *Synechocystis* and *E. coli*, and therefore its suitability for the rapid prototyping concept.

3.2.2.1 Approach for designing library of promoters

Before a promoter library was constructed, a suitable library design first needed to be chosen. The library design needed to generate a promoter library that was functional in both organisms and which offered a wide range of expression strengths that were covered in small incremental steps of activity in both *E. coli* and *Synechocystis*. Many different methodologies have been described generating promoter libraries (120, 136, 444-447), many of which have been successfully used to produce a wide range of expression levels in *E. coli* (120, 136, 444, 447). At the start of this project, no approach had been described for generating such a library in cyanobacteria, although one has since been published (352).

The Synthetic Promoter Library (SPL) approach previously described by Jensen and Hammer (120) was chosen as it is a well proven method for generating promoter libraries with a wide range of expression strengths that increase in a small step-wise manner, as demonstrated in *E. coli* and *Lactococcus lactis* (82). The library design generates a promoter library by randomising the spacer sequences that surround the -35 and -10 consensus sequences of a σ^{70} *E. coli* promoter. Conveniently, cyanobacterial type 1 promoters, which are recognised by group 1 sigma factors such as SigA that control expression of housekeeping genes, have been shown to possess both -35 and -10 elements with the consensus sequences TAGACA and TATAAT, respectively (398, 448), which are nearly identical to the consensus sequences for the -35 and -10 elements of an *E. coli* sigma 70 factor-dependent promoter, TTGACA and

TATAAT, respectively (449, 450). Therefore, if the SPL design conserved the -35 and -10 consensus sequence of a σ^{70} *E. coli* promoter it was likely that the promoter library would be functional in both organisms.

The SPL design that was subsequently used to generate the promoter library randomised 33 bases that surround the -35 (TTGACA) and the -10 (TATAAT) elements of a σ^{70} *E. coli* promoter (10 bases upstream of the -35 element, the 17-base spacer between -35 and -10 elements and 6 bp downstream of the -10 element; Figure 3.5). Downstream of the SPL, a 20 bp insulator was included (Figure 3.5), that was generated using a random DNA sequence generator (451), which was incorporated to provide an identical context downstream of the promoter library to minimise the contextual effect that different promoter sequences would have on the RBS strength (context dependence is introduced in section **4.1.2**). As a result, differences in fluorescence between different members of the promoter library would be attributable to the activities of the different promoters per se, not in combination with indirect effects on the RBS. The RBS was designed with a predicted translation initial rate (T.I.R.) of 12,800 using the RBS Calculator (85) in the context of the insulator and *eyfp*. This RBS was designed with a lower predicted T.I.R. so that stronger members of the SPL library would not have a deleterious effect on cells, which could affect the measured strength and could increase the rate of mutation within a promoter sequence, thus preventing proper assessment of strong promoters.

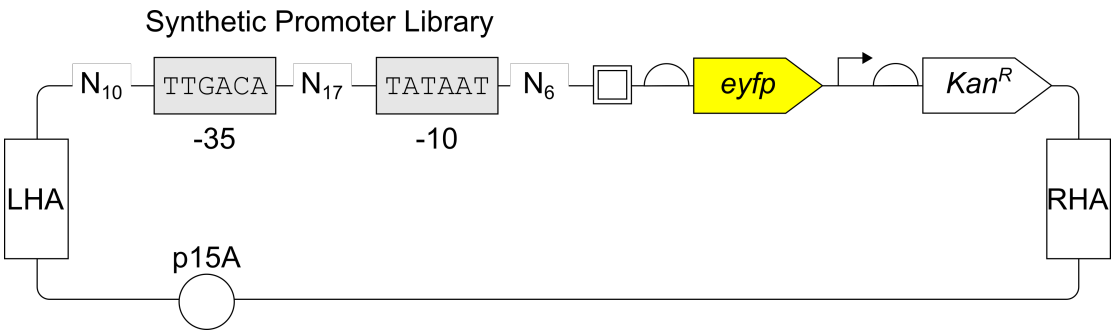


Figure 3.5 The Synthetic Promoter Library design in the context of pATM2. The Synthetic Promoter Library (SPL) design randomised 33 bases that surround the -35 (TTGACA) and the -10 (TATAAT) elements of a σ^{70} *E. coli* promoter (10 bases upstream of the -35 element, the 17-base spacer between -35 and -10 elements and 6 bp downstream of the -10 element). To provide context independence to the synthetic RBS and *eyfp* coding sequence, downstream of the SPL is a 20 bp insulator (white box). pATM2 backbone features as described in Figure 1.

The SPL promoter library was generated by inverse PCR using plasmid pATM2 (which contains *eyfp* in vector pSHUTTLE2) as a template with partially degenerate 5'-

phosphorylated primers. The primers introduced the SPL, insulator and synthetic RBS upstream of the *eyfp* in pATM2. The primer sequences corresponding to the SPL contains only the appropriate specific nucleotide at positions corresponding to the -35 and -10 elements and a mixture of nucleotides specified as "N" (for "any Nucleotide" in the standard IUPAC notation) at randomised positions. Oligonucleotide synthesis results in a mixed pool of oligonucleotides with the four nucleotides A, T, G and C, each incorporated at approximately the same 25% frequency at each N position to provide randomisation. The PCR products were then circularised by ligation and the ligation products used to transform *E. coli* to obtain a promoter library of clones with different synthetic promoter variants. Of the 10^4 *E. coli* transformant colonies obtained, 200 colonies were picked, 100 randomly and 100 by visual inspection, illuminating with blue light to visualise fluorescence, to ensure a wide range of expression strengths were isolated.

3.2.2.2 Identifying clones generated by the Synthetic Promoter Library

Of the 200 *E. coli* transformant colonies picked, 100 promoter clones were picked for analysis (Figure 3.6). Using flow cytometry, clones with florescence values weaker than or equivalent to the negative control pATM2 ($P_{sll1120}$ non-functional in *E. coli*) were deemed non-functional and were discarded from the library. Promoter clones that showed bimodality (Figure 3.6) were also discarded as they would be poor candidates for precise control of protein expression. This process reduced the library down to 50 SPL clones that were subsequently sequenced for further analysis. SPL clones with deletions within promoter sequence were discarded as that could affect the position of the transcriptional start site (TSS), which would in turn affect the mRNA transcript. Promoter clones were also discarded if they had any insertions, mutations or deletions within the insulator, RBS or *eyfp* CDS that would have a contributing effect on EYFP fluorescence. The resultant promoter library contained 37 SPL clones (Table A1). By using these quality constraints on the promoter library, it was possible to ensure that any difference in EYFP fluorescence intensity was most likely solely attributed to the strength of the promoter clone.

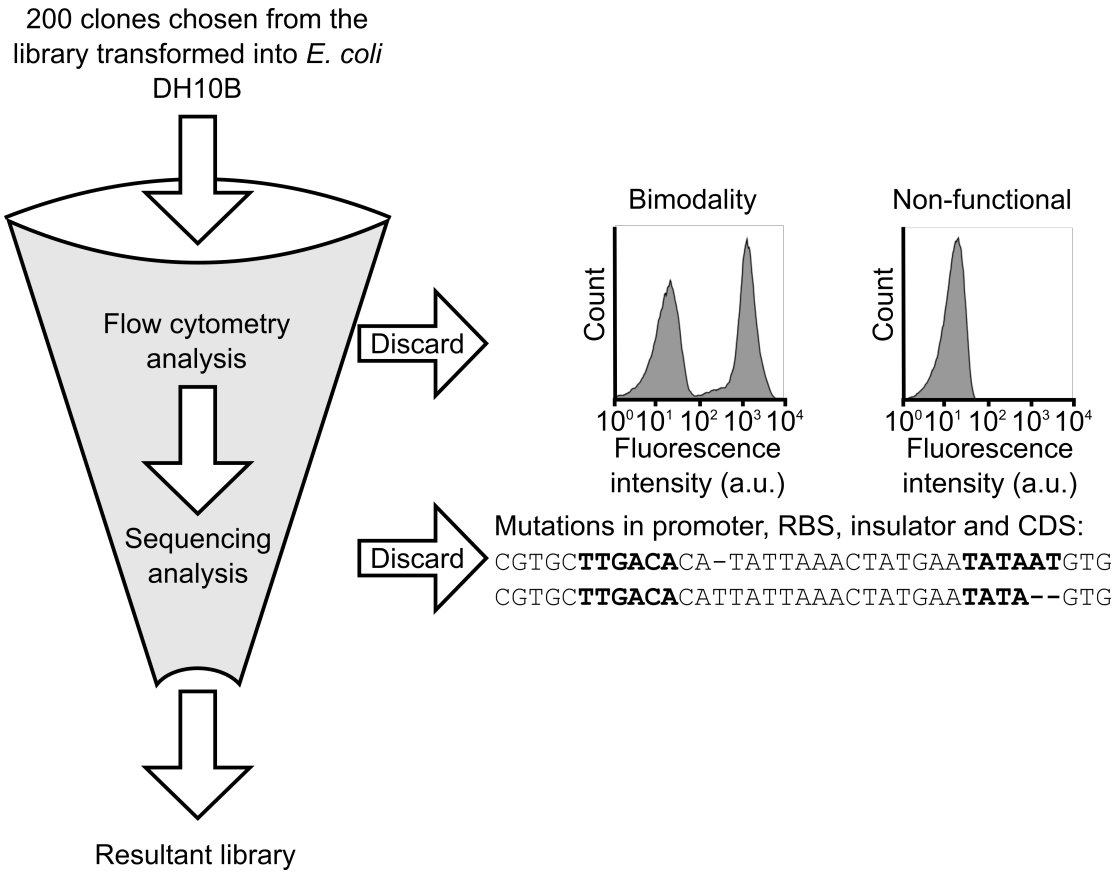


Figure 3.6. Screening cascade for promoters and RBS clones. Clones chosen from the promoter and RBS library were subjected to a quality control step to ensure that the strength for each of the characterised parts was solely accountable to the promoter or RBSs strength. Firstly, clones were analysed by flow cytometry and clones that displayed bimodality or were weaker than pATM2 (-ve control) were discarded. The resultant library was sequenced, clones with mutations in the promoter, RBS, insulator or *eyfp* that would affect fluorescence were discarded, yielding the final library.

3.2.2.3 Characterisation of the synthetic promoters in *E. coli* DH10B.

To establish whether the resultant 37 promoter clones would provide a wide range of strengths that increased in small increments, they were characterised in both *E. coli* and *Synechocystis*. Firstly, the library was characterised in *E. coli* DH10B, whereby *E. coli* was transformed with each of the SPL promoter clone plasmids, single colonies from each transformation plate were picked, grown overnight, sub-cultured and grown to mid-exponential phase (4 hours incubation at 37 °C). After incubation, the single-cell fluorescent distribution of each SPL clone was measured. A wide distribution of promoter strengths was observed in *E. coli* (Figure 3.7), ranging from barely detectable fluorescence (SPLc112) to the strongest promoter clone that was 14-fold stronger (SPLc31). Furthermore, the strength of the promoter clones increased in small increments as required for fine tuning protein expression.

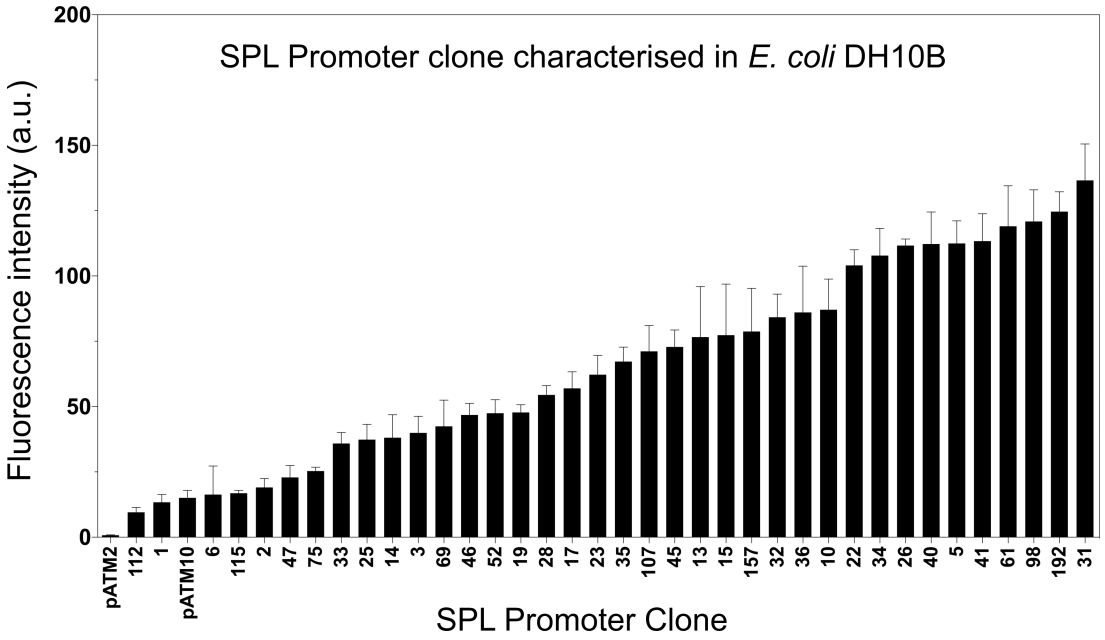


Figure 3.7. Synthetic Promoter Library characterisation in *E. coli* DH10B. *E. coli* DH10B cells were transformed with the plasmids containing the 37 promoter clones generated from the SPL. Transformant colonies were cultured overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the fluorescence of 10,000 cells were measured by flow cytometry. The fluorescence intensities presented represent the fluorescence of cells with wild-type background fluorescence subtracted. pATM10 uses the strong native promoter P_{psbA2} to drive expression of EYFP, using the same insulator and synthetic RBS as the SPL design. The error bars shown represent the standard deviation of five biological repeats.

3.2.2.4 Characterisation of the synthetic promoters in *Synechocystis* sp. PCC 6803.

Having observed a wide distribution of promoter strengths in *E. coli* the promoter clones were then characterised in *Synechocystis* to determine whether they offered similar properties in both organisms and could therefore be used as portable promoters for fine-tuning protein expression. To characterise the SPL clones in *Synechocystis*, wild-type cells (the glucose-tolerant derivative of the wild type) were transformed separately with each SPL clone plasmid and kanamycin-resistant transformant colonies were passaged four times until complete segregation was confirmed by PCR. These transformant colonies were then cultured under constant light in BG11 medium supplemented with kanamycin for 48h. Cultures were subcultured to a starting optical density (measured at 750 nm) of 0.1 and grown to exponential phase (OD_{750} 0.5) and then single-cell fluorescence intensities were assayed using flow cytometry. As in *E. coli*, a wide distribution of promoter strengths that increased over small increments of strength was observed (Figure 3.8), which ranged from very weak (SPLc115)

increasing 27-fold to the strongest promoter (SPLc25). Interestingly, over 80% of the SPL promoter clones were stronger than the positive control (pATM10), a plasmid that uses the strong native cyanobacterial promoter P_{psbA2} (452) to drive expression of *eyfp*, with the strongest promoter SPL clone 25 three times stronger. Having generated a library of 37 promoters that offered a wide range of strengths in both *E. coli* and *Synechocystis*, it was necessary to determine how these promoters performed between the two organisms and whether they could be used as portable promoters.

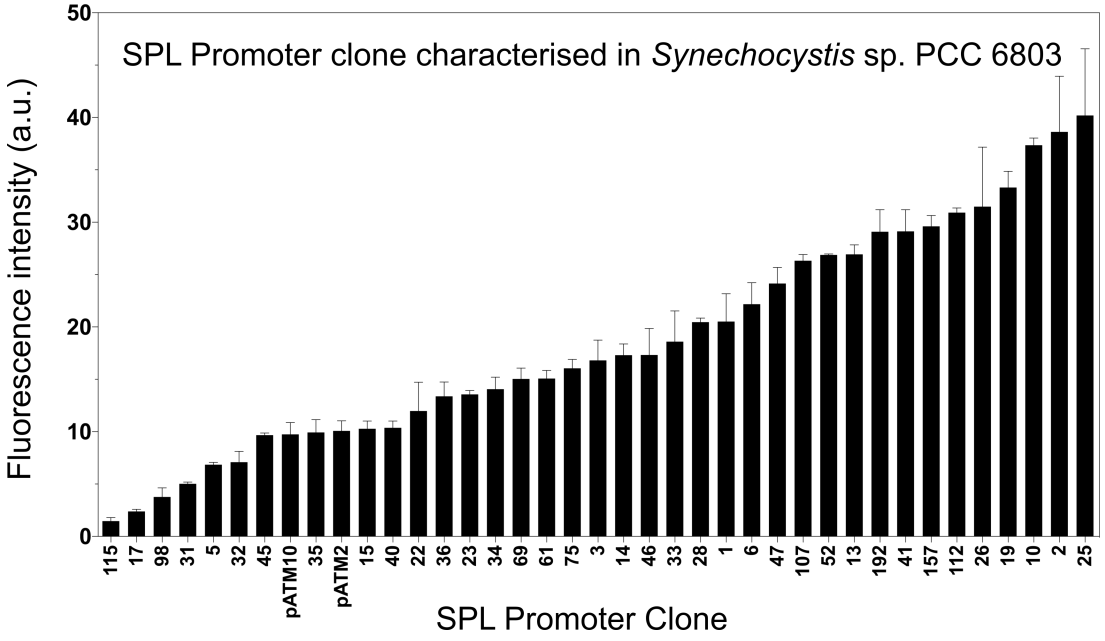


Figure 3.8. Synthetic Promoter Library characterisation in *Synechocystis* sp. PCC 6803. *Synechocystis* cells containing plasmids with different promoter clones isolated from the SPL were cultured in BG11 media in photoautotrophic conditions and constant light; and their fluorescence intensity of 10,000 cells measured in a mid-exponential phase of growth using flow cytometry. Fluorescence intensities presented represent the fluorescence of cells with background fluorescence subtracted. The positive control, pATM10, uses the strong native promoter P_{psbA2} to drive expression of EYFP, using the same insulator and synthetic RBS as the SPL design. The error bars shown represent the standard deviation of three biological repeats.

3.2.2.5 Portability of SPL promoter clones between *E. coli* and *Synechocystis*.

To assess the portability of the 37 SPL promoter clones, the fluorescence intensities between the two organisms were compared (Figure 3.9). No correlation, $R^2 = 0.05$, in the performance of the SPL clones in *E. coli* and *Synechocystis* was observed. The poor correlation suggests that the promoters in this library would be unsuitable as portable parts because their performance in *E. coli* could not be used as a prediction of their performance in *Synechocystis*. While no overall correlation of the promoter library between these two organisms was observed, there was a fair spread of the

promoter clones over a wide range of strengths in both organisms. It is possible that within the promoter library, subsets of promoters exist that have similar performances between *E. coli* and *Synechocystis* (Figure 3.10) and could potentially serve as portable promoters. Before further investigating these potential portable subsets, a novel promoter library design was created and used in an attempt to fill in the gaps of the missing portable promoters in possible portable subsets.

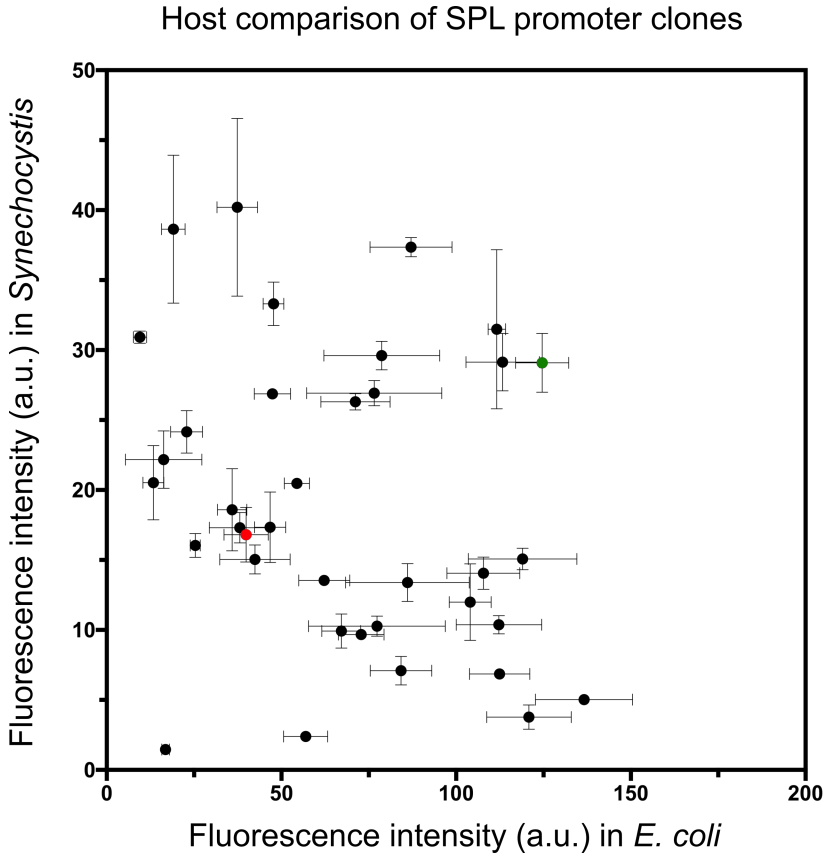


Figure 3.9. Portability of synthetic promoters from the SPL between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. Fluorescence intensities of the synthetic promoters in *E. coli* cells (as described in Figure 3.7) were plotted against activity of each synthetic promoter in *Synechocystis* (as described in Figure 3.8). SPL promoter clone 192 (green dot) and SPL promoter clone 3 (red dot) are used in Promoter Library 2 and RBS Library designs, respectively. Error bars represent standard deviations of five and three biological repeats for *E. coli* and *Synechocystis*, respectively.

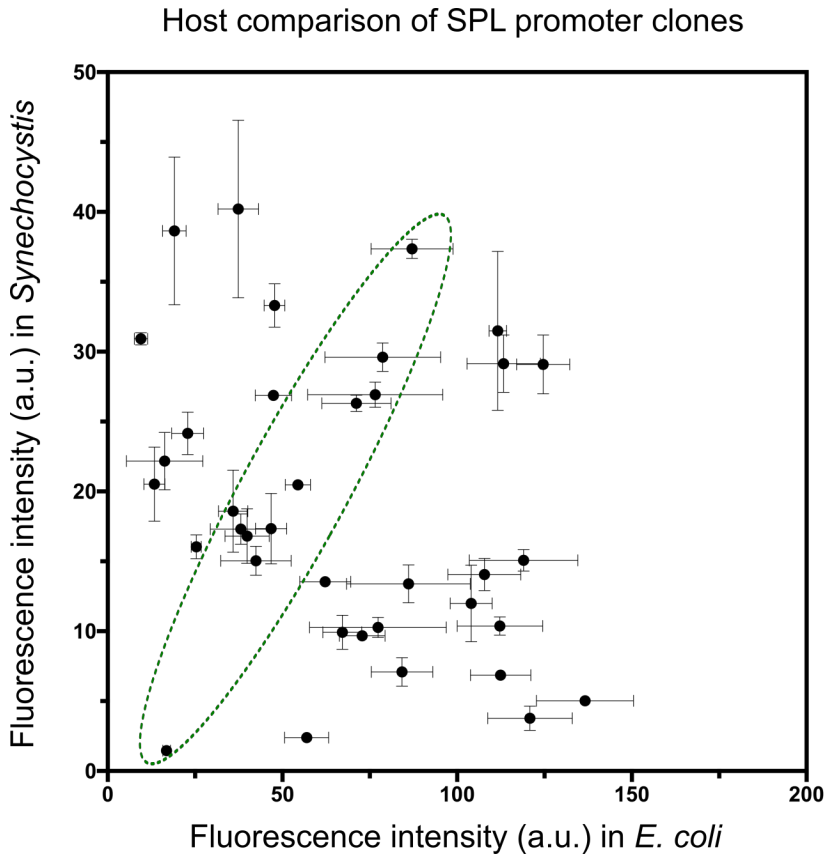


Figure 3.10. Potential subsets of portable promoters within the SPL. Subsets of potential portable promoters (green dotted oval) were highlighted on the SPL host comparison (described in Figure 3.9).

3.2.3 Promoter Library 2 – A targeted approach for generating a portable library of promoters

One approach to increasing the number of promoters in the promoter library would be to isolate and characterise more promoter clones from the SPL approach previously used (section **3.2.2**). However considering the lack of correlation that was previously observed, isolating 'useful' clones could be a somewhat intensive process. Instead, an alternative approach for generating a promoter library was designed, specifically for generate portable promoters.

3.2.3.1 Promoter Library 2 concept

While the original synthetic promoter library, generated using the SPL approach, was not entirely portable as a whole, some promoters within the set were portable and similarly strong in both *E. coli* and *Synechocystis*. It was hypothesized that the specific spacer sequence of those promoter clones might be accountable for the similar performance between the two organisms and if degeneracy was introduced to the -35 and -10 elements of those clones it might equally decrease the strength of those

promoter clones between the two organisms (Figure 3.11). It was also hypothesised that the strength of these promoters would decrease when degeneracy was introduced to the consensus sequences, based on the assumption that the -35 (TTGACA) and -10 (TATAAT) consensus sequences that were previously used in the SPL design give the strongest binding affinity with RpoD and SigA for *E. coli* and *Synechocystis*, respectively.

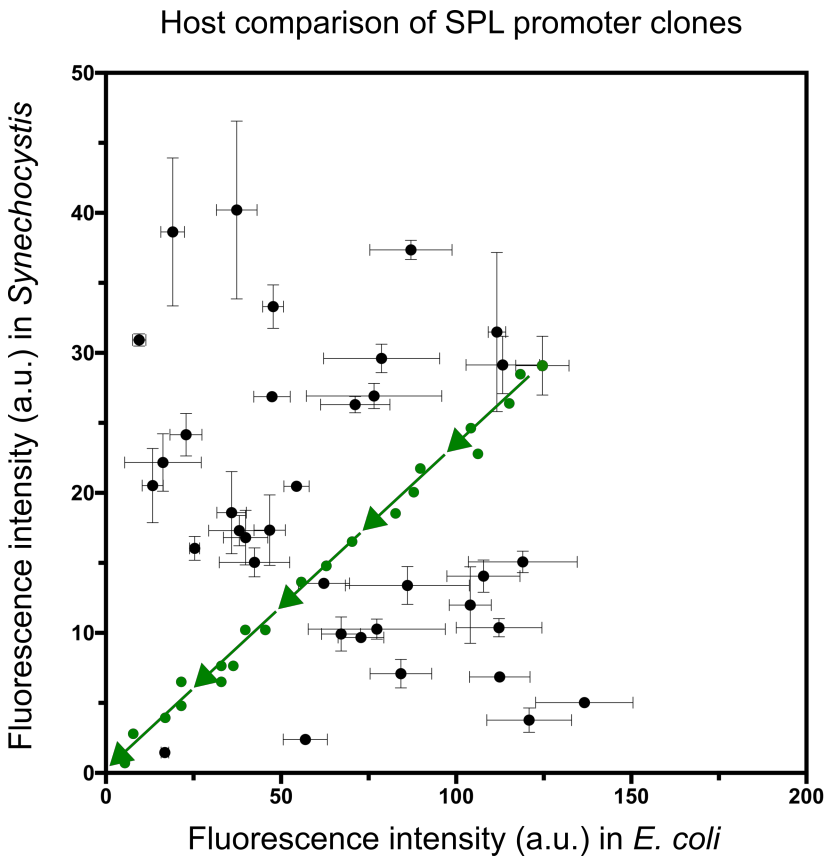


Figure 3.11. Illustration showing the hypothesized outcome of the targeted portable design for Promoter Library 2. The illustration shows the actual data from the original SPL promoter clones characterised in both *E. coli* and *Synechocystis* (black circles; as shown in Figure 3.9) and the hypothesized outcome of the Promoter Library 2 from degenerating the -35 and -10 consensus sequences (green circles).

3.2.3.2 Promoter Library 2 design

It has been shown that mutations within the -35 and -10 sequences change the promoter strength, with mutations that make the -35 and -10 sequences more similar to the consensus increasing the promoter strength (referred to as 'up mutations') (453), while mutations away from the -35 and -10 consensus sequences decrease the promoter strength (referred to as 'down mutations') (454). This is consistent with the hypothesis that randomising the -35 and -10 consensus sequence would create

weaker promoters. Previous 'up-' and 'down-mutations' have either been randomly introduced, for example by using UV mutagenesis (455), or they have been rationally introduced as point mutations (453, 454). With the lack of understanding on how to rationally introduce degeneracy to -35 and -10 elements without 'breaking' the promoter's activity and resulting in a library of non-functional clones, four different libraries designs were developed. Each library introduced varying degrees of degeneracy to the -35 and -10 sequences (Table 3.1), with the aim that at least one design would generate promoter clones that provide a wide range of strengths that increase in small increments. The first design (Design 1) contained a completely randomised -35 (NNNNNN) and -10 (NNNNNN) sequence. The second design (Design 2) was the most conserved of the four, with 3 bp randomised and 3 bp conserved in both the -35 (TTGNNN) and -10 (TANNNT) sequence. The bases that were fixed in the -35 and -10 elements were those that were most conserved within the *Synechocystis* -35 and -10 consensus sequences. The third design (Design 3) completely randomised the -10 (NNNNNN), but took forward the -35 sequence from Design 2 (TTGNNN). The fourth design (Design 4) used the -35 design (TTGNNN) and a semi-conserved -10 (WWWWWW), where "W" was an equal split of the adenine and thymine.

Table 3.1. Four designs used for the Promoter Library 2. Randomisation shown in bold, where "N" represents 25% split of the four bases A, T, C and G and "W" represents a 50% split of the two bases A and T.

Library 2 designs	-35 Sequence	-10 Sequence
Design 1	**NNNNNN**	**NNNNNN**
Design 2	TTG**NNN**	TA**NNN**T
Design 3	TTG**NNN**	**NNNNNN**
Design 4	TTG**NNN**	**WWWWWW**

Next, a suitable parental promoter that could be used for the four Promoter Library 2 designs needed to be identified. The parental promoter needed to be similarly strong in both *E. coli* and *Synechocystis*. SPL promoter clone 192 from the first promoter library (Figure 3.9, green dot) was chosen because it was a strong promoter in both *E. coli* and *Synechocystis*.

It is important to state that while the absolute values of SPL clone 192; 120 a.u. and 30 a.u. for *E. coli* and *Synechocystis*, respectively, don't match between the two organisms, they are both strong promoters relative to the strength of other promoter clones from the SPL promoter library. Using the relative strength of a promoter, rather

than the absolute value was deemed as an acceptable comparison of strength, as it is possible for the absolute fluorescence values in *Synechocystis* to have been effected by the abundance of photoactive pigments via energy transfer or by quenching. It is also possible that the absolute protein expression levels in *Synechocystis* might not be able to reach those of *E. coli*. As such, the direct comparison of absolute fluorescent values might not be entirely accurate and therefore relative comparison should serve well. It was hypothesized that as long as a subset of portable parts could maintain a relative ratio of the protein concentration profile of a pathway between the two organisms, this would be suitable for rapid prototyping. In this case, if the promoter clones generated from the Promoter Library 2 correlate well between the two organisms, it is probable that a relative ratio of protein levels between *E. coli* and *Synechocystis* would be close to 4:1, assuming this was the only factor contributing to protein expression levels.

So that the promoter clones from the first SPL promoter library and Promoter Library 2 could be compared, the four new library designs were constructed in pATM2 with the same 20 bp insulator, synthetic RBS and *eyfp* coding sequence that were originally used for the SPL. The Promoter Library 2 was generated by inverse PCR using plasmid pATM2 (which contains *eyfp* in vector pSHUTTLE2) as a template with partially degenerate 5'-phosphorylated primers. Different pairs of primers were used to introduce each of the four library designs, the insulator and RBS upstream of the *eyfp* of pATM2. To introduce degeneracy within the -35 and -10 elements the primer sequences contained a mixture of nucleotides specified as "N" (25% split of A, T, C and G) or "W" (for "Weak" in the standard IUPAC notation; 50% split of A and T). The PCR products were then circularised by ligation and the ligation products used to transform *E. coli*.

To identify which of the library designs would provide the greatest number of promoter clones over the widest range of expression strengths, the population of each library design was compared by flow cytometry in *E. coli*. This allowed the distribution of fluorescence in the population of each library design to be assessed, as a large number of individual cells could be independently measured. *E. coli* was transformed with the ligation product of each library design, the transformation was used to inoculate a culture that was grown to stationary phase (16 h) after which the cultures were analysed by flow cytometry (Figure 3.12). The promoter $P_{sll1220}$ that drives expression of *eyfp* in pATM2 is not functional in *E. coli* and so served as a negative control in *E. coli* and the parental promoter SPL clone 192 was used as a positive control. In

addition to comparing the range of expression strengths in the four designs for Promoter Library 2, the original SPL, which could be used as a reference for a wide range of strengths, was also analysed. The data showed that Design 1 and 3 had similar fluorescence distributions to the negative control (pATM2). Presumably the designs were too degenerate and produced a library of promoters that were predominately non-functional, and these designs were therefore discarded. The fluorescence distribution of Design 4 showed a wide range of strengths that was comparable to the SPL design, however the large majority of the population fell within the fluorescent range of the negative control and therefore the design was discarded. Design 2 offered a greater number of clones through the whole range of fluorescence intensities. Therefore, Design 2 was taken forward as the appropriate design for degenerating the -35 and -10 consensus sequences (Figure 3.13).

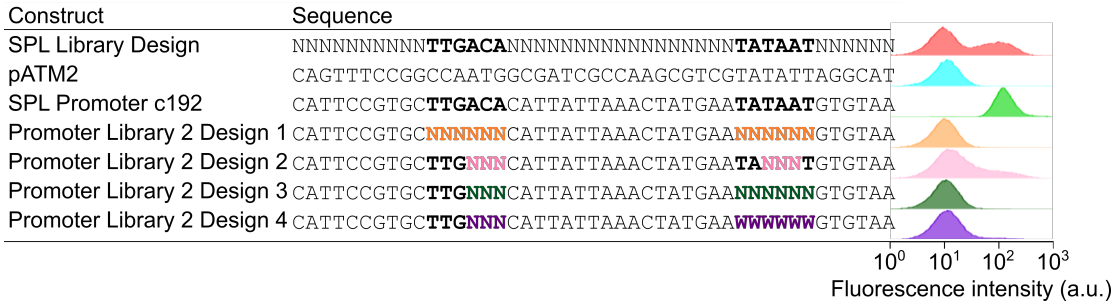


Figure 3.12. The range of promoter strengths generated in each of the four Promoter Library 2 designs. The sequence for each of the library designs shows the different degrees of randomisation introduced to the -35 and -10 elements (Bold; "N" represents 25% split of the four bases A, T, C and G; "W" represents a 50% split of the two bases A and T) of the four Promoter Library 2 designs. Flow cytometry histograms show 10,000 cell counts normalised to the maximum (in order to visualise distribution rather than absolute values) for pATM2 as a negative reference, SPL promoter clone 192 as a positive reference, a pool of several hundred transformants from the SPL Design as a positive reference for a wide dynamic range and a pool of several hundred transformants for each of the four library designs.

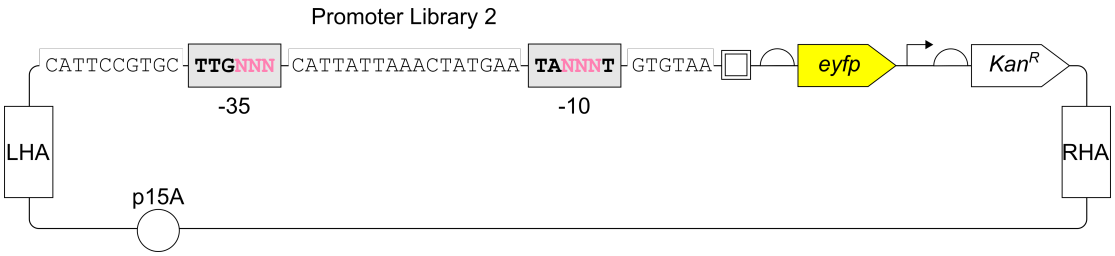


Figure 3.13. The Design of Promoter Library 2 in the context of pATM2. The Promoter Library 2 design conserved the spacers sequences of the parental promoter, SPLc192 that surround the -35 and -10 elements. Within the -35 and -10 elements, three bases were randomised, TTGNNN and TANNNT, respectively. To provide context independence to the synthetic RBS and *eyfp* coding sequence, downstream of the SPL is a 20 bp insulator (white box). pATM2 backbone features as described in Figure 3.1.

3.2.3.3 Identifying promoter clones from Promoter Library 2

To isolate promoter clones from the Promoter Library 2 *E. coli* was re-transformed with the ligation product. Out of the 10^4 *E. coli* transformant colonies obtained, 200 clones were picked from design 2 of the second library, 100 were isolated randomly and 100 were isolated by visual inspection, by illuminating with blue light to visualise fluorescence, and this ensured that a wide range of expression strengths were isolated. The same identification criteria used for clones generated with the first SPL approach described earlier was then applied to these new promoter clones (Figure 3.6); 100 clones were analysed by flow cytometry bimodal and non-functional clones were discarded, which excluded 23 clones. The remaining 77 clones were sequenced, and 13 further clones were excluded that had deletions within the promoter or deletions or mutations to the spacer, RBS and/or *eyfp*. The resultant library consisted of 64 promoter clones (Table A2).

3.2.3.4 Characterisation of Promoter Library 2 in *E. coli.*

To determine whether the 64 promoter clones would give a wide range of strengths decreasing in small increments from the parental promoter SPL clone 192 as hypothesized, the constructs were first characterised in *E. coli*. To characterise the promoters, *E. coli* was transformed with each of the 64 promoter plasmids, single colonies grown overnight, subcultured in fresh medium and grown to mid-exponential phase (4 h). At this point, the single-cell fluorescence distribution of each promoter clone was measured. A wide range of expression levels was observed in *E. coli* (Figure 3.14) with the strongest clone (clone 47) expressing 142-fold more fluorescence than the weakest clone (clone 80). Interestingly, approximately 50% of the promoter clones were stronger than the parental promoter, SPL clone 192, which suggests that

although the -35 (TTGACA) and -10 (TATAAT) consensus sequences present in SPL clone 192 might be the most common sequence in a σ^{70} *E. coli* promoter, they might not have the strongest affinities to the binding regions of RpoD. This library design produced clones that offered a wide range of strengths that increased in small increments of strength in *E. coli*. In order to determine the portability of these promoters they were next characterised in *Synechocystis*.

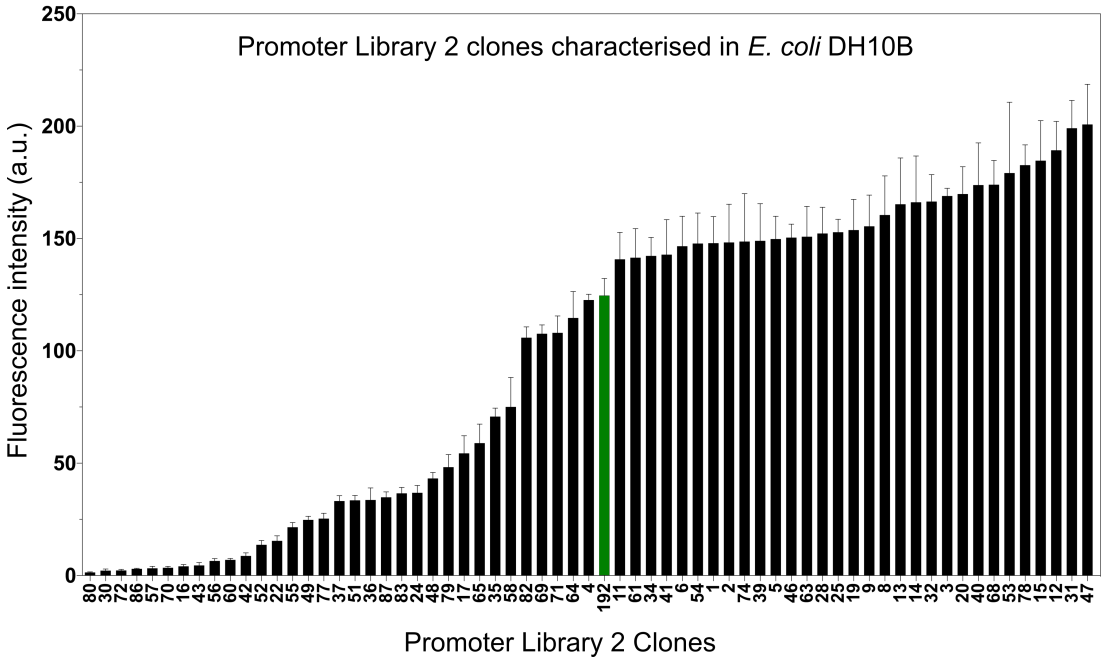


Figure 3.14. Promoter Library 2 characterisation in *E. coli* DH10B. *E. coli* DH10B cells were transformed with the plasmids containing the 64 promoter clones generated from the Promoter Library 2. Transformant colonies were cultured for overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the fluorescence of 10,000 cells were measured by flow cytometry. The fluorescence intensities presented represent the fluorescence of cells with wild-type background fluorescence subtracted. The parental promoter (SPL clone 192) used for Promoter Library 2 is shown in green. The error bars shown represent the standard deviation of five biological repeats.

3.2.3.5 Characterisation of Promoter Library 2 in *Synechocystis* sp. PCC 6803.

To characterise the 64 clones from Promoter Library 2 in *Synechocystis*, wild-type cells were transformed with the respective plasmids, kanamycin-resistant transformant colonies passaged and segregation was confirmed as before. These transformant colonies were cultured under constant light in BG11 media supplemented with kanamycin for 48h. Cultures were subcultured to a starting optical density (measured at 750 nm) of 0.1 and grown to exponential phase (OD_{750} 0.5) and then single-cell fluorescence intensities were assayed using flow cytometry. The data showed a 10-

fold increase in fluorescence from the weakest (clone 48) to the strongest promoter which was the parental promoter (SPL clone 192; Figure 3.15). While a 10-fold change in expression levels of Promoter Library 2 clones was observed, 50 of the 55 clones appeared non-functional, only measuring 6 a.u. of fluorescence greater than the wild-type background of *Synechocystis*, which has since been shown to be a result of chromosomal read-through from an upstream gene (124), confirming the lack of functionality. This library design did not produce promoters that offered a range of different strengths in *Synechocystis* as had been hypothesized. Interestingly, this data does indicate that *Synechocystis* had a considerably higher stringency for the consensus at the -35 and -10 regions compared to *E. coli*, which evidently tolerates diverse mutations.

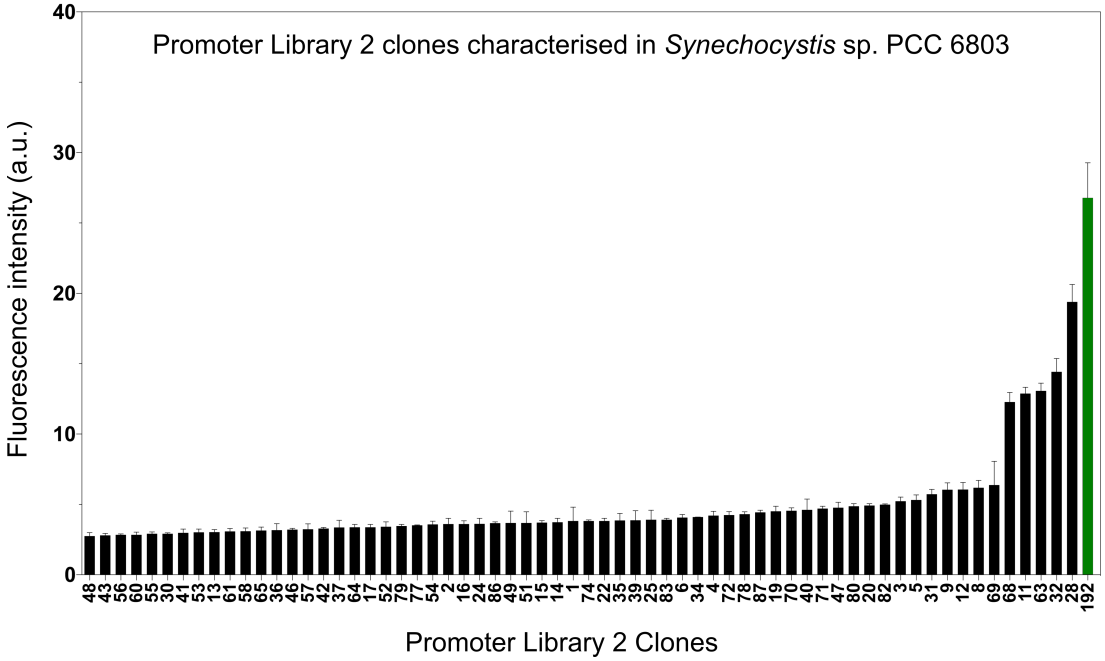


Figure 3.15. Promoter Library 2 characterisation in *Synechocystis* sp. PCC 6803. *Synechocystis* cells containing plasmids with different promoter clones isolated from the Promoter Library 2 were cultured in BG11 media in photoautotrophic conditions and constant light; and their fluorescence intensity of 10,000 cells measured in a mid-exponential phase of growth using flow cytometry. Fluorescence intensities presented represent the fluorescence of cells with background fluorescence subtracted. The parental promoter (SPL clone 192) used for Promoter Library 2 is shown in green. The error bars shown represent the standard deviation of three biological repeats.

3.2.3.6 Sequence level comparison of the consensus sequence for functional and non-functional clones in *Synechocystis* showed high stringency for the consensus

In an attempt to identify what sequence was responsible in the -35 and -10 elements for promoter functionality in cyanobacteria, the five strongest promoters were aligned (Table 3.2). Of the five strongest promoters, three clones (clone 32, 63 and 68) had the conserved -10 consensus sequence (TATAAT) and the two other clones (clone 11 and 28) contained the same sequence in the -10 element (TACAAT) that had only one mutation at the -10 position changing thymine for cytosine. This evidence suggests a strong preference towards the -10 consensus sequence. Less stringency was observed towards the -35 consensus sequence with each of the five strongest promoter clones having a different sequence in the -35 element. Interestingly, in the two clones with a mutation in the -10 element, the sequence for the -35 was either conserved (clone 28) or contained only a single mutation (clone 11). Furthermore, the sequences of the five strongest promoters were compared to the non-functional promoters. Ten clones were identified (clones 3, 5, 8, 9, 12, 15, 19, 32, 46 and 47; see Table A6) that had a single mutation away from one of the five strongest promoters and of these ten clones eight of them contained the single mutation in the -10 element. Although this is a small sample insufficient to fully determine specificity and stringency, these results suggest there was a high stringency towards the consensus sequence of the -10 region of a cyanobacterial type 1 promoter.

Table 3.2. Sequence comparison of the -35 and -10 elements of the five active promoter clones isolated from the Promoter Library 2. Mutations away from the -35 and -10 consensus sequence are shown in red.

Promoter Clone	-35	-10
SPL cone 192 (Parental promoter)	TTGACA	TATAAT
Library 2 clone 11	TTGAC**G**	TA**C**AAT
Library 2 clone 28	TTGACA	TA**C**AAT
Library 2 clone 32	TTG**CA**A	TATAAT
Library 2 clone 63	TTGAC**C**	TATAAT
Library 2 clone 68	TTG**CAT**	TATAAT

3.2.3.7 Portability of Promoter Library 2 between *E. coli* and *Synechocystis*

Despite the lack of promoters that were functional in *Synechocystis*, the fluorescence intensities of the 64 promoter clones were compared between the two organisms (Figure 3.16). The data showed no correlation between the strengths of these promoters in *E. coli* and *Synechocystis*. As such, a library of portable promoters had

not been generated, indicating that this new library design was not suitable for generating portable promoters as had originally been hypothesized.

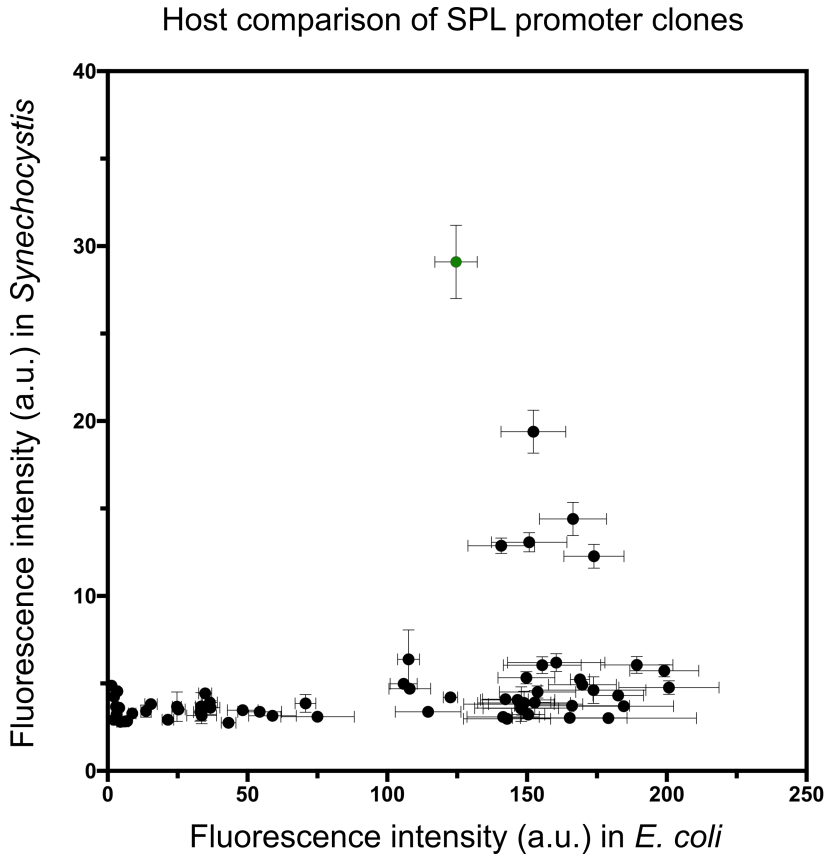


Figure 3.16. Portability of promoters from Promoter Library 2 between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. The fluorescence intensities of the synthetic promoters in *E. coli* cells (as described in Figure 3.14) were plotted against activity of each synthetic promoter in *Synechocystis* (as described in Figure 3.15). The parental promoter (SPL clone 192) used for Promoter Library 2 is shown in green. Error bars represent standard deviations of five and three biological repeats for *E. coli* and *Synechocystis*, respectively.

3.2.3.8 Promoters generated using the SPL approach were combined with those making up Promoter Library 2 in an attempt to increase the number of portable promoters.

Despite the fact that Promoter Library 2 predominately generated non-functional promoters in *Synechocystis,* the original SPL library of 37 promoter clones was combined with the 64 promoter clones from Promoter Library 2, to determine if the number of potential portable promoters would be increased. Combining the two libraries generated 101 promoter clones and gave a 287-fold dynamic range in EYFP expression in *E. coli* (Figure 3.17). However it made no improvement to the range of strengths in *Synechocystis* (Figure 3.18). By comparing the performance of each of

the 101 promoter clones between *Synechocystis* and *E. coli,* a large number of promoter clones were identified that expressed similarly low levels of expression between both organisms, which could serve as potential 'weak' portable promoters (Figure 3.19). The large majority of promoter clones from Promoter Library 2 were not suitable portable promoters in the combined promoter library, because while they changed strength in *E. coli* they did not change strength in *Synechocystis* (Figure 3.19). As such, these promoter clones were discarded from the combined promoter library, decluttering the library. The resultant decluttered library of 43 promoters contained 37 promoter clones generated in the SPL approach and six of the promoter clones from Promoter Library 2, five of which were the active clones and the sixth, which was a weak promoter in both *E. coli* and *Synechocystis* (Figure 3.20). This library is the largest promoter library reported for cyanobacteria to date. The library is also functional in *E. coli* with some of the promoters displaying similar performances between the two organisms providing potential subsets of portable promoters. This resultant promoter library was therefore taken forward as the final library of promoters for future use.

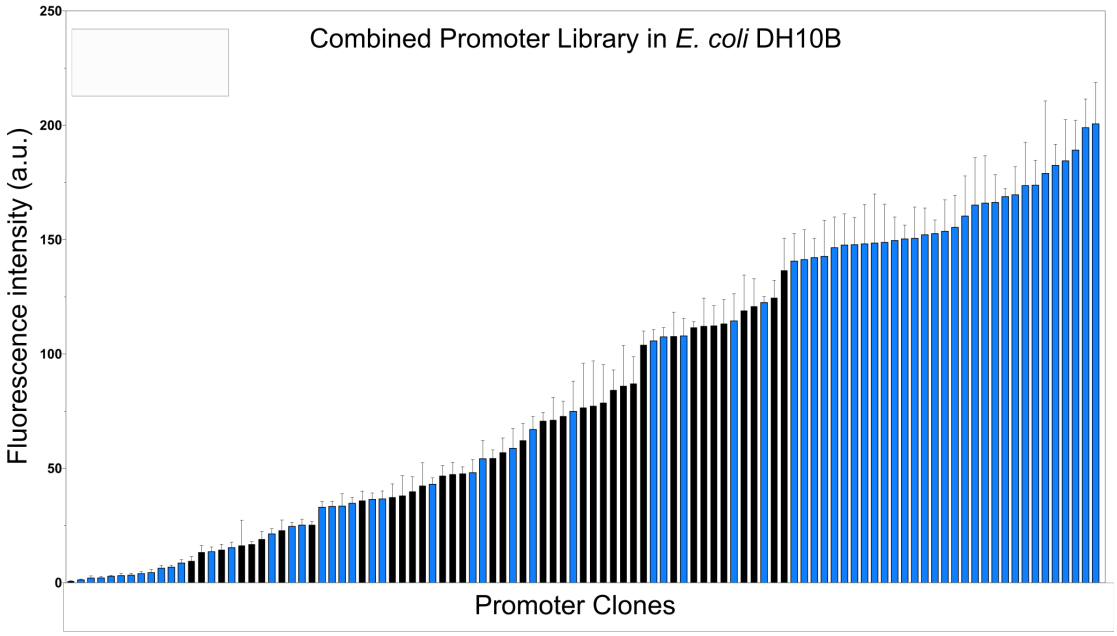


Figure 3.17. Promoters generated using the Synthetic Promoter Library and Promoter Library 2 characterised in *E. coli* DH10B. The fluorescence intensities for the 37 synthetic promoters generated using the SPL (black; described in Figure 3.7) and the 64 promoters derived from Promoter Library 2 (blue; described in Figure 3.14) characterised in *E. coli*. The promoter clone IDs can be found in Appendix Note A4.

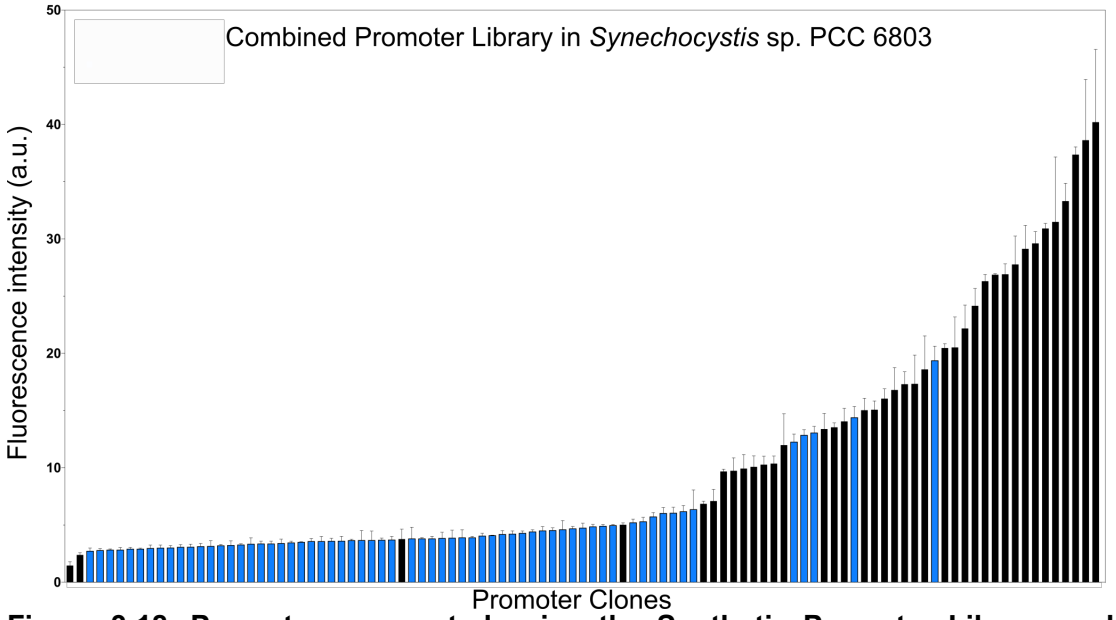


Figure 3.18. Promoters generated using the Synthetic Promoter Library and Promoter Library 2 characterised in *Synechocystis* sp. PCC 6803. The fluorescence intensities for the 37 synthetic promoters generated using the SPL (black; described in Figure 3.8) and the 64 promoters derived from Promoter Library 2 (blue; described in Figure 3.15) characterised in *Synechocystis*. The promoter clone IDs can be found in Appendix Note A5.

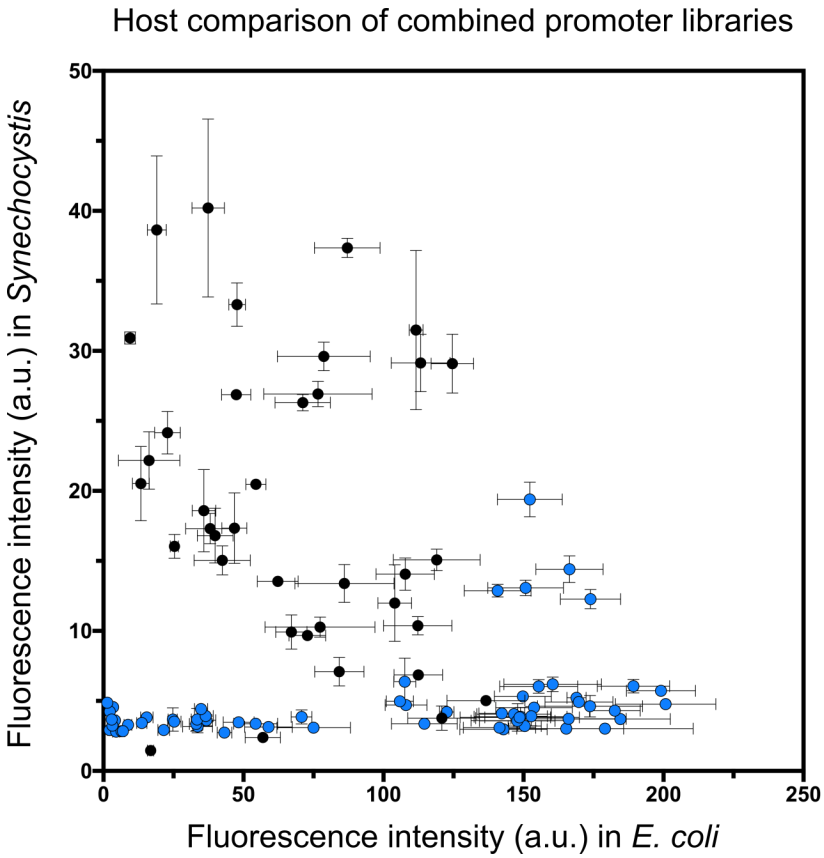


Figure 3.19. Portability of the 92 promoters generated using the Synthetic Promoter Library and Promoter Library 2 compared between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. Fluorescence intensities of the 92 promoters, 37 of which were generated using the SPL (black dots) and 64 that were generated in Promoter Library 2 (blue dots) in *E. coli* cells were plotted against the activity of each promoter in *Synechocystis*. Error bars represent standard deviations of five and three biological repeats for *E. coli* and *Synechocystis*, respectively.

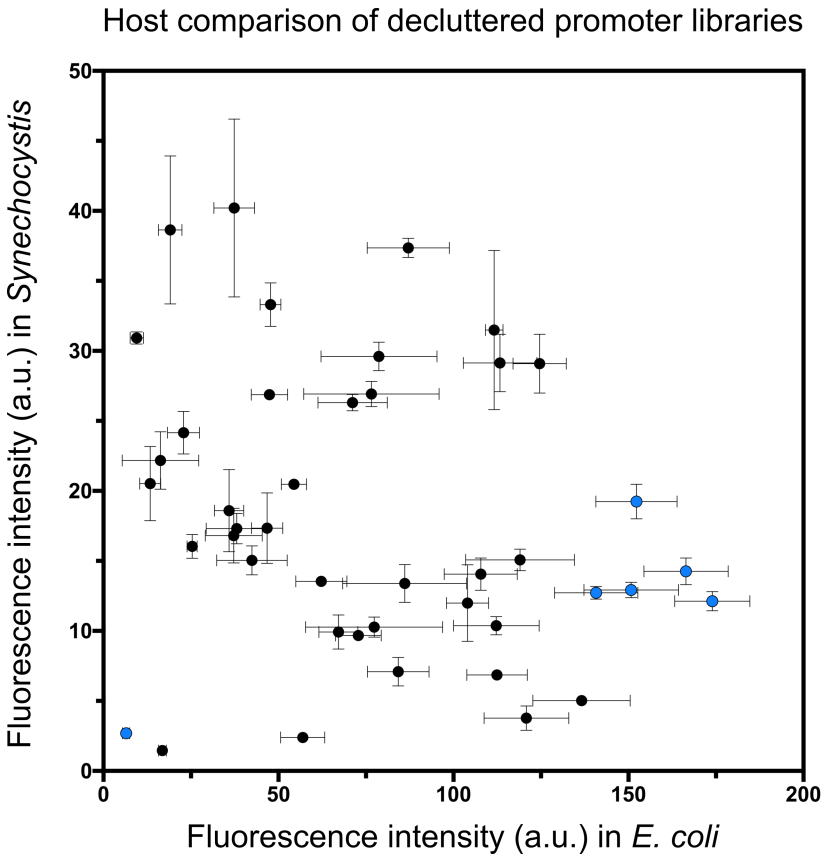


Figure 3.20. Portability of the final decluttered Promoter Library between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. Fluorescence intensities of the 43 promoters that make up the decluttered library, 37 of which were generated using the SPL (black dots) and 6 promoters taken forward from Promoter Library 2 (blue dots) in *E. coli* cells were plotted against the activity of each promoter in *Synechocystis*. Error bars represent standard deviations of five and three biological repeats for *E. coli* and *Synechocystis*, respectively.

3.2.4 Ribosome-binding site library

After generating the promoter libraries, the focus shifted to ribosome-binding sites (RBSs). There were no portable RBSs and very few characterised RBSs available in cyanobacteria at the start of the project and as such, the project next aimed to design and generate a library of RBSs, which were functional in both *E. coli* and *Synechocystis*. With this library it would be possible to assess how the RBSs perform between the two organisms and ascertain whether they hold the portable properties necessary for rapid prototyping.

RBSs were chosen because: i) as an expression control part they have been frequently used to control and fine-tune protein expression (85, 115, 456) over several orders of magnitude (85); and ii) to further fill the deficit of cyanobacterial parts. Two approaches have been previously described for generating a library of RBSs, both of which are based on changing the RBS DNA sequence. The first approach randomly mutates the

RBS sequence often generating an extremely large number of RBS clones, this approach can require screening a larger number of mutants before finding those with the desired properties (428, 457-459). The second approach uses computational tools, which are becomingly increasing popular as they overcome the potential need to screen a large number of mutants that would be generated by a random approach. To date three translation rate calculators have been developed: the RBS Calculator (85), the RBS Designer (129) and the UTR Designer (128), each of which are able to design synthetic RBSs with a user specified translation initiation rate (T.I.R.) for a specific 5'-UTR sequence and coding sequence. The three calculators use a similar approach to estimate an RBSs translation rate, whereby the free energies of key molecular interactions during translation initiation, such as mRNA secondary structures and interactions between mRNA and rRNA, are assessed using secondary software suites. The free energies are then considered by each calculators own respective thermodynamic model to give an estimation of translation rate. However, the RBS calculators remain limited in their predictive ability not only for model but also for non-model organisms, where the calculators predicted data tends to show a poor correlation with the experimental data. For example, the RBS Calculator has been used twice to generate RBSs for cyanobacteria. The first was for optimising the production of 2,3-butanediol in *Synechoccus elongatus* sp. PCC 7942 through synthetic RBSs, however the RBSs predicted strengths showed little correlation to their experimental strength (298). The second, which is also the only previously reported RBS library in cyanobacteria, used the RBS Calculator to generate an eight-member RBS library for *Synechococcus* sp. PCC 7002 (352), where again there was a weak correlation between the RBSs predicted and experimental strengths. Furthermore, unpublished work in our lab has found a similar weak correlation in *Clostridium* (John Heap, pers comm).

With future metabolic engineering in mind, as for the promoter library, the RBS library needed to offer a wide range of expression strengths in both *E. coli* and *Synechocystis*, which increase in a small step-wise manner to allow fine-tuning of protein expression. Considering that the random approach worked well for generating a promoter library with those properties and that the computational tools are still limited with their predictive power in non-model organisms, the RBS library was designed using a targeted degenerate mutagenesis.

3.2.4.1 Designing a degenerate RBS library by sequence randomisation.

The RBS library was designed with a similar rationale to that of the Synthetic Promoter Library; whereby the consensus sequence is fixed and the spacer sequences that surround the consensus are randomised. In the case of an RBS this meant conserving the consensus sequence of the SD sequence and randomising the sequences that surround it. It was possible to use such a design for the RBS library because *E. coli* and *Synechocystis* share a nearly identical SD and anti-SD sequence, differing by only one base pair, AUCACCUCCUUA and AUCACCUCCUUU (460), respectively. Therefore, in the RBS library design the SD sequence AAAGGAGGT was conserved and the surrounding sequence (15 bp upstream and five bp downstream of the SD sequence to align with optimal spacing) was randomised (Table 3.3). However, with no evidence that this design would encode a library of RBS clones with a wide range of expression strengths or in fact RBS portability; eight additional libraries that introduced increasing degrees of degeneracy within the RBS SD sequence were designed (Table 3.3). RBS library design 1 contained the fully conserved SD sequence AAAGGAGGT, which was surrounded by 15 randomised bp upstream and five randomised bp downstream. RBS library design 2 contained the SD sequence AAGGAGG, which was surrounded by 16 randomised bp upstream and six randomised bp downstream. RBS library design 3 contained the SD sequence AGGAGG, which was surrounded by 17 randomised bp upstream and six randomised bp downstream. RBS library design 4 contained the SD sequence GGAGG, which was surrounded by 18 randomised bp upstream and six randomised bp downstream. RBS library design 5 contained the SD sequence GAGG, which was surrounded by 19 randomised bp upstream and six randomised bp downstream. RBS library design 6 contained the SD sequence AGG, which was surrounded by 20 randomised bp upstream and six randomised bp downstream. RBS library design 7 contained the SD sequence GG, which was surrounded by 21 randomised bp upstream and six randomised bp downstream. RBS library design 8 contained the SD sequence G, which was surrounded by 22 randomised bp upstream and six randomised bp downstream. RBS library design 9 contained a string of 29 randomised bp.

Table 3.3. Nine RBS library designs. Fifteen bases upstream and five downstream of the Shine-Dalgarno sequence (bold) were randomised (N represents 25% split of the four bases A, T, C and G) in each of the nine designs. Through the nine designs the Shine-Dalgarno sequence was increasingly randomised from the fully conserved consensus in RBS library design 1 (red) through the a string of N's in RBS library design 9 (black).

RBS library design	RBS library sequence
RBS library design 1	NNNNNNNNNNNNNNN**AAAGGAGGT**NNNNN
RBS library design 2	NNNNNNNNNNNNNNN**N**AAGGAGG**N**NNNNN
RBS library design 3	NNNNNNNNNNNNNNN**NN**AGGAGG**N**NNNNN
RBS library design 4	NNNNNNNNNNNNNNN**NNN**GGAGG**N**NNNNN
RBS library design 5	NNNNNNNNNNNNNNN**NNNN**GAGG**N**NNNNN
RBS library design 6	NNNNNNNNNNNNNNN**NNNNN**AGG**N**NNNNN
RBS library design 7	NNNNNNNNNNNNNNN**NNNNNN**GG**N**NNNNN
RBS library design 8	NNNNNNNNNNNNNNN**NNNNNNN**G**N**NNNNN
RBS library design 9	NNNNNNNNNNNNNNN**NNNNNNNNN**NNNNN

To characterise these nine RBS library designs, a promoter was needed that: i) behaved similarly in both *E. coli* and *Synechocystis* so that transcription was similar between the organisms and therefore, the portability of the RBS library could be fairly assessed; and ii) was not so strong that the stronger members of the RBS library would have a deleterious effect on the cells, which could affect the measured strength and could increase the rate of mutation within an RBS sequence, preventing proper assessment of strong RBSs. Promoter SPL clone 3 from the Synthetic Promoter Library showed a similar level of mid-strength expression in both organisms (Figure 3.9, red dot) and was therefore chosen as the promoter for the RBS library (Figure 3.21).

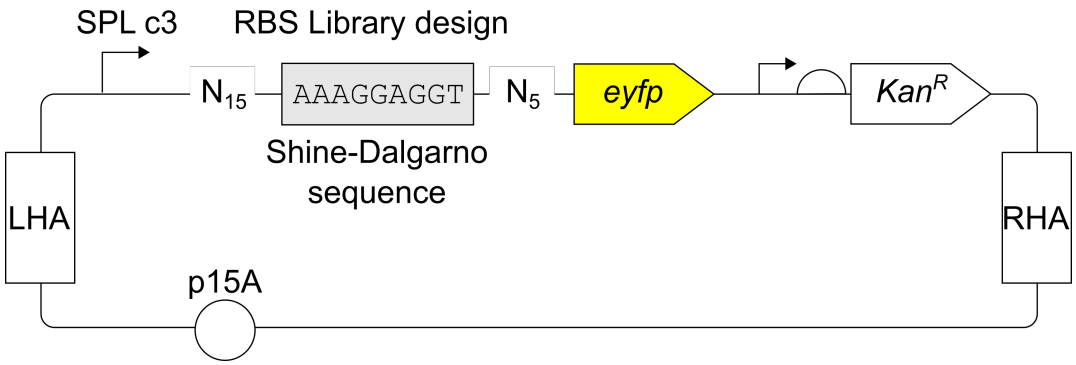


Figure 3.21. RBS library design 1 in the context of pATM2. RBS clones would be isolated and characterised in the pGT77 backbone (derived from pATM2). Expression of the *eyfp* would be driven by promoter SPL clone 3, previously characterised as a mid-strength promoter in *E. coli* and *Synechocystis*, and quantified using flow cytometry to characterise each RBS.

The nine libraries were generated by inverse PCR using plasmid pGT77 (pATM2 with SPL promoter clone 3 upstream of *eyfp*) as a template with partially degenerate 5'-phosphorylated primers. Different pairs of primers were used to introduce each of the nine RBS library designs upstream of the *eyfp* of pGT77. The primer sequences that correspond to the RBS libraries contain only the appropriate specific nucleotide at positions in the SD sequence for the corresponding design and also contain a mixture of nucleotides specified as "N" (25% mixture of A, T, C and G) at degenerate positions of the RBS library. The nine PCR products were then circularised by ligation and the ligation products used to transform *E. coli* to obtain nine RBS libraries.

3.2.4.2 Comparing the range of expression strengths from each RBS library.

To identify the RBS library design that would give the greatest number of clones over the widest range of expression strengths, the clonal populations of each library design in *E. coli* were compared. To do this, *E. coli* was transformed with the ligation products of each RBS library design, the transformation was used to inoculate a culture that was grown to stationary phase (16 h) at which point the cultures were analysed by flow cytometry (Figure 3.22). The plasmid pATM2 was used as the negative control because the promoter ($P_{sll1220}$) that drives expression of the EYFP in the plasmid is not functional in *E. coli* (unpublished data, John Heap, pers comm). The plasmid pGT77 was used as a positive control, which uses the promoter SPL clone 3 to drive expression of EYFP, the same promoter that is upstream of each RBS library design. The fluorescence intensity distributions of the clonal populations from the nine RBS library designs showed considerable differences, with the greatest number of RBS clones occurring over the widest range of fluorescence intensities in RBS library design 1. As the SD sequence was increasingly randomised through the library designs, the range of fluorescence intensities narrowed and the distribution moved towards that of the negative control, pATM2, with RBS library design 9 showing a comparable fluorescence distribution to pATM2, suggesting that the overall majority of clones in this library are non-functional in *E. coli*. This data showed a clear trend between the degree of randomisation in the SD sequence and the range of a library's expression strength.

The most suitable library designs to use for isolating RBS clones were RBS library designs 1 – 6, as their populations displayed the widest range of expression strengths. However, the library design also needs to generate RBSs that are portable between

the two organisms, and with no evidence on how to generate such RBSs for *E. coli* and *Synechocystis*, RBS clones were isolated from the RBS library designs 1, 4 and 9. This ensured that a wide range of expression strengths were isolated from library designs 1 and 4, but it also allowed investigation of the effect that any alternative SD sequence that was isolated from library 9 would have on portability.

Construct	Sequence
pATM2	TCGGGTAAGTTTATAATATAC**AAAGGAGGT**AGAA
RBS Library design 1	NNNNNNNNNNNNNNN**AAAGGAGGT**NNNNN
RBS Library design 2	NNNNNNNNNNNNNNN**N**AAGGAGG**N**NNNNN
RBS Library design 3	NNNNNNNNNNNNNNN**NN**AGGAGG**N**NNNNN
RBS Library design 4	NNNNNNNNNNNNNNN**NNN**GGAGG**N**NNNNN
RBS Library design 5	NNNNNNNNNNNNNNN**NNNN**GAGG**N**NNNNN
RBS Library design 6	NNNNNNNNNNNNNNN**NNNNN**AGG**N**NNNNN
RBS Library design 7	NNNNNNNNNNNNNNN**NNNNNN**GG**N**NNNNN
RBS Library design 8	NNNNNNNNNNNNNNN**NNNNNNN**G**N**NNNNN
RBS Library design 9	NNNNNNNNNNNNNNN**NNNNNNNNN**NNNNN
pGT77	AGAGCAGGATTAGAG**GGAGGT**CAGAGA

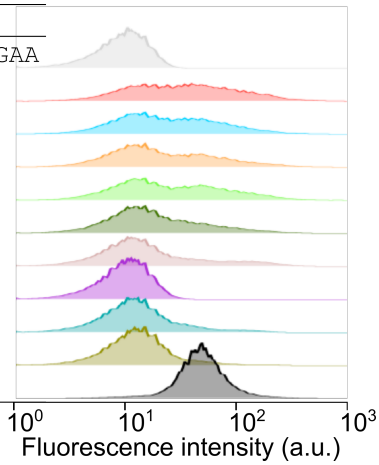


Figure 3.22. The range of RBS strengths generated in each of the nine RBS library designs. The sequence for each of the library designs shows the different degrees of randomisation ("N" represents 25% split of the four bases A, T, C and G) that was introduced to the nine RBS Library designs. Flow cytometry histograms show 10,000 cell counts normalised to the maximum (in order to visualise distribution rather than absolute values) for pATM2 as a negative reference, pGT77 as a positive reference and a pool of several hundred transformants for each of the nine library designs.

3.2.4.3 Identifying RBS clones from RBS library

To isolate RBS clones, *E. coli* was transformed with the ligation products for RBS library designs 1, 4 and 9. Out of the 10^{12} transformant colonies obtained (three independent transformations of the three library designs each obtaining 10^4 transformant colonies; $10^4 + 10^4 + 10^4$), 200 clones were picked, 100 chosen randomly and 100 by visual inspection, illuminating with blue light to visualise fluorescence, to ensure that a wide range of expression strengths were isolated. 100 of the chosen clones were subjected to the same screening cascade as the promoter libraries (Figure 3.6), whereby the RBS clones were analysed by flow cytometry and clones that were non-functional, weaker than the negative control (pATM2), and/or displayed a bimodal distribution were discarded. From this process, 30 RBS clones were excluded that were non-functional or showed a bimodal distribution. Each of the remaining 70 RBS were sequenced, RBS clones were again discarded if mutations had occurred within

the promoter or CDS, this excluded a further 12 clones. The resultant and final RBS library was made up of 58 RBS clones (Table A3).

3.2.4.4 Characterisation of the RBS library in *E. coli* DH10B showed a wide range of expression strengths

To determine whether the RBS library produced a wide range of strengths that increased in small increments, the 58 RBS clones were characterised in *E. coli*. Firstly, each RBS clone plasmid was transformed into *E. coli*, a single transformant colony was picked for each RBS clone and grown overnight, the culture was then sub-cultured and grown to mid-exponential phase (4 hours incubation at 37 °C). After 4 hours incubation, the single-cell fluorescent distribution of each RBS clone was measured (Figure 3.23). The data showed a wide range of fluorescence intensities that increased in small increments, with the strongest RBS (RBS clone 133) expressing a 455-fold increase in fluorescence compared to the weakest RBS (RBS clone 249). Interestingly, the library of 58 RBSs showed no relationship (R^2 = 0.149) between their measured fluorescence and their predicted ΔG_{tot} in *E. coli* (Figure 3.24), as calculated by the reverse engineering tool of the RBS Calculator v2.0 (85).

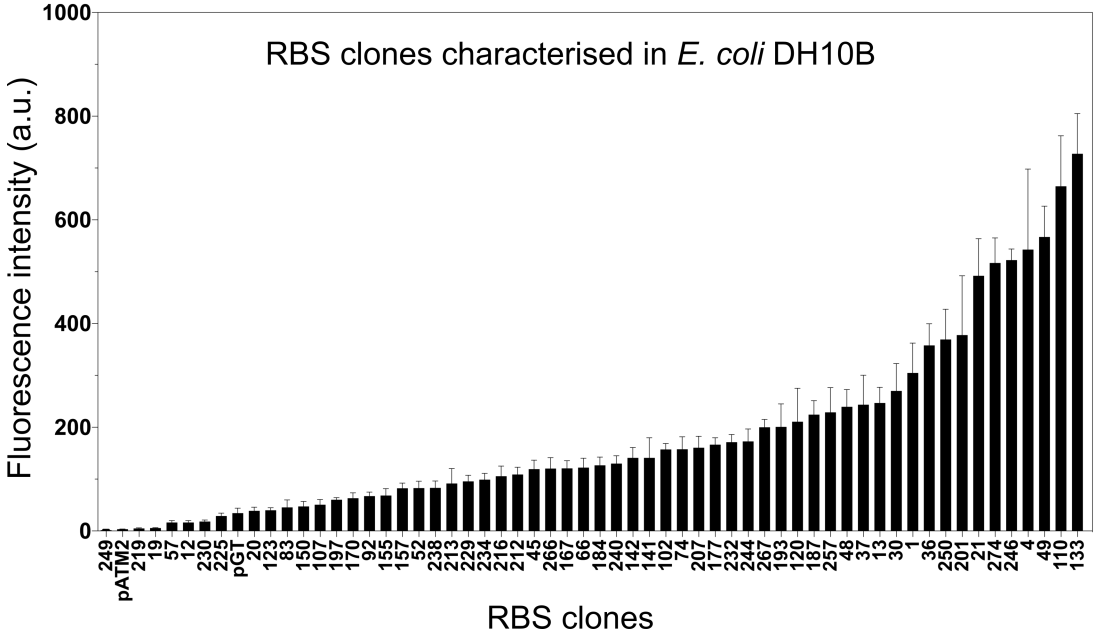


Figure 3.23. RBS library characterisation in *E. coli* DH10B. *E. coli* DH10B cells were transformed with the plasmids containing the 58 RBS clones (under promoter SPL clone 3) generated from the RBS library. Transformant colonies were cultured for overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the fluorescence of 10,000 cells were measured by flow cytometry. The fluorescence intensities presented represent the fluorescence of cells with wild-type background fluorescence subtracted. The error bars shown represent the standard deviation of five biological repeats.

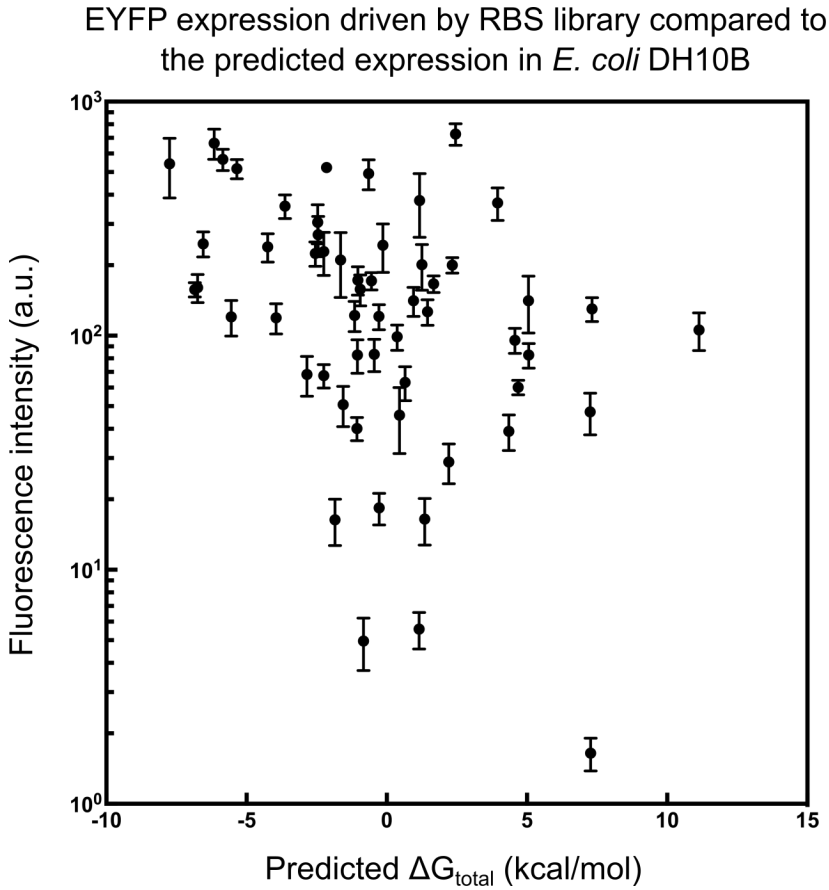


Figure 3.24. Comparison of the predicted and measured strength of the RBS library in *E. coli*. The log protein fluorescence of the EYFP reporter expression driven by the RBS library compared to their predicted ΔG_{tot} in *E. coli* DH10B calculated using the RBS Calculator v2.0. The error bars shown represent the standard deviation of five biological repeats. Linear regression R^2 = 0.149.

3.2.4.5 Characterisation of the RBS library of *Synechocystis* sp. PCC 6803 showed a wide range of expression strengths.

Next, the RBS library was characterised in *Synechocystis* to determine whether the 58 RBS clones had similar fluorescence intensities to what they did in *E. coli*. To characterise the RBS library clones in *Synechocystis*, wild-type cells were transformed with the plasmid for each RBS clone and kanamycin-resistant transformant colonies were passaged until complete segregation was confirmed by PCR. These transformant colonies were then cultured under constant light in BG11 media supplemented with kanamycin for 48h. Cultures were subcultured, diluting to a starting optical density (measured at 750 nm) of 0.1 and grown to exponential phase (OD750 0.5), at which point the single-cell fluorescence intensities were assayed using flow cytometry. Like in *E. coli*, a wide distribution of fluorescence intensities that increased in a small stepwise manner was observed (Figure 3.25) that ranged from very weak (RBS clone

249) to very strong (RBS clone 4) over a 152-fold change. Interestingly, the library of 58 RBSs showed no relationship (R^2 = 0.153) between their measured fluorescence and their predicted ΔG_{tot} in *Synechocystis* (Figure 3.26), as calculated by the reverse engineering tool of the RBS Calculator v2.0 (85).

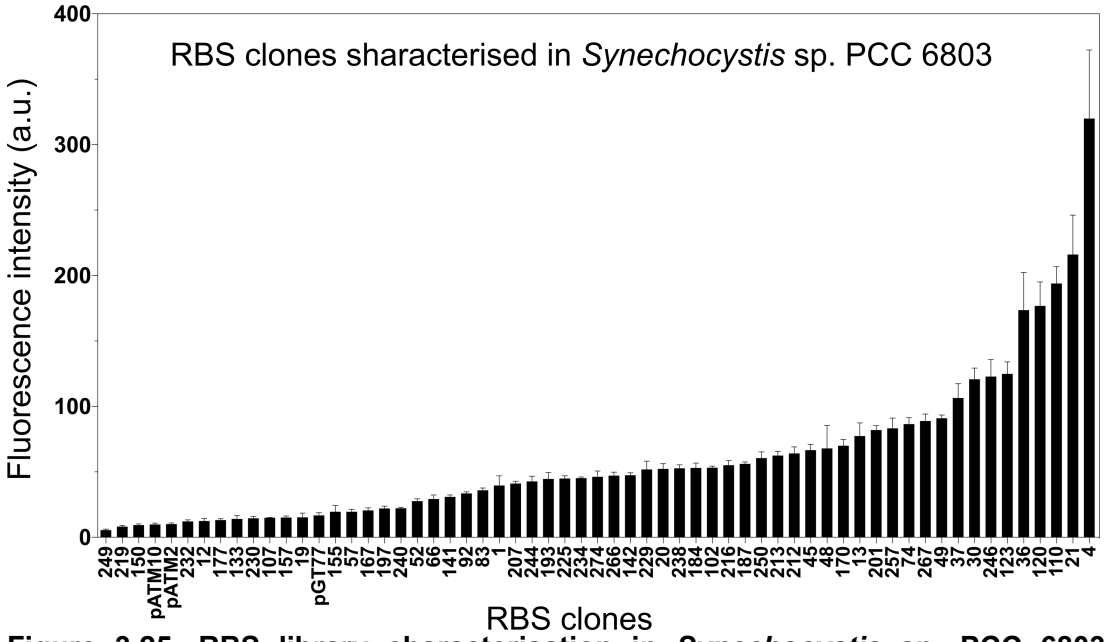


Figure 3.25. RBS library characterisation in *Synechocystis* sp. PCC 6803. *Synechocystis* cells containing plasmids with different RBS clones isolated from the RBS library were cultured in BG11 media in photoautotrophic conditions and constant light; and their fluorescence intensity of 10,000 cells measured in a mid-exponential phase of growth using flow cytometry. Fluorescence intensities presented represent the fluorescence of cells with background fluorescence subtracted. The error bars shown represent the standard deviation of three biological repeats.

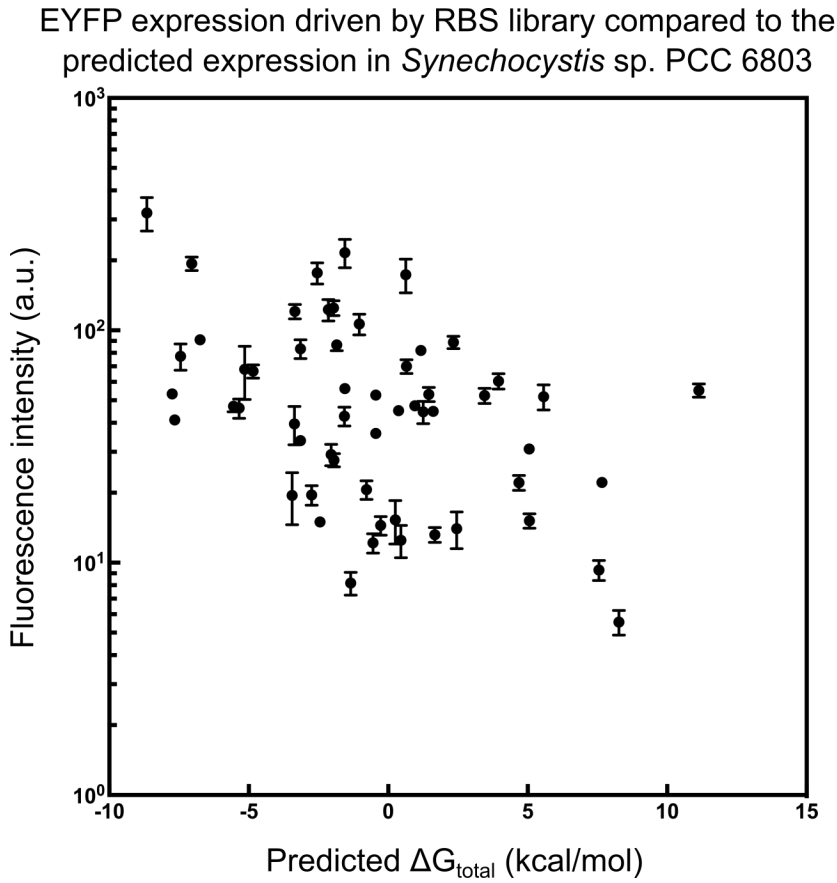


Figure 3.26. Comparison of the predicted and measured strength of the RBS library in *Synechocystis*. The log protein fluorescence of the EYFP reporter expression driven by the RBS library compared to their predicted ΔG_{tot} in *Synechocystis* sp. PCC 6803 calculated using the RBS Calculator v2.0. The error bars shown represent the standard deviation of three biological repeats. Linear regression R^2 = 0.153.

3.2.4.6 Portability of the RBS library between *E. coli* and *Synechocystis*.

To assess the portability of the RBS library and therefore its suitability for rapid prototyping the fluorescence intensities of the 58 RBS clones between *E. coli* and *Synechocystis* were compared (Figure 3.27). A weak correlation in the performance of the RBS library between *E. coli* and *Synechocystis*, R^2 = 0.34, was observed, which was better than the SPL promoter library correlation, R^2 = 0.05. Figure 3.28 shows the correlation of the three RBS library designs that RBS clones were isolated from, in which RBS library design 1 gave the strongest correlation of RBSs, R^2 = 0.57 and a comparably poor correlation was observed in RBS library design 4 and 9, R^2 = 0.10 and 0.12, respectively. While the overall library of 58 RBS clones did not have a strong correlation (R^2 = 0.34), it appeared that subsets of RBSs existed within the library that did offer the portable properties that were needed for rapid prototyping (Figure 3.29).

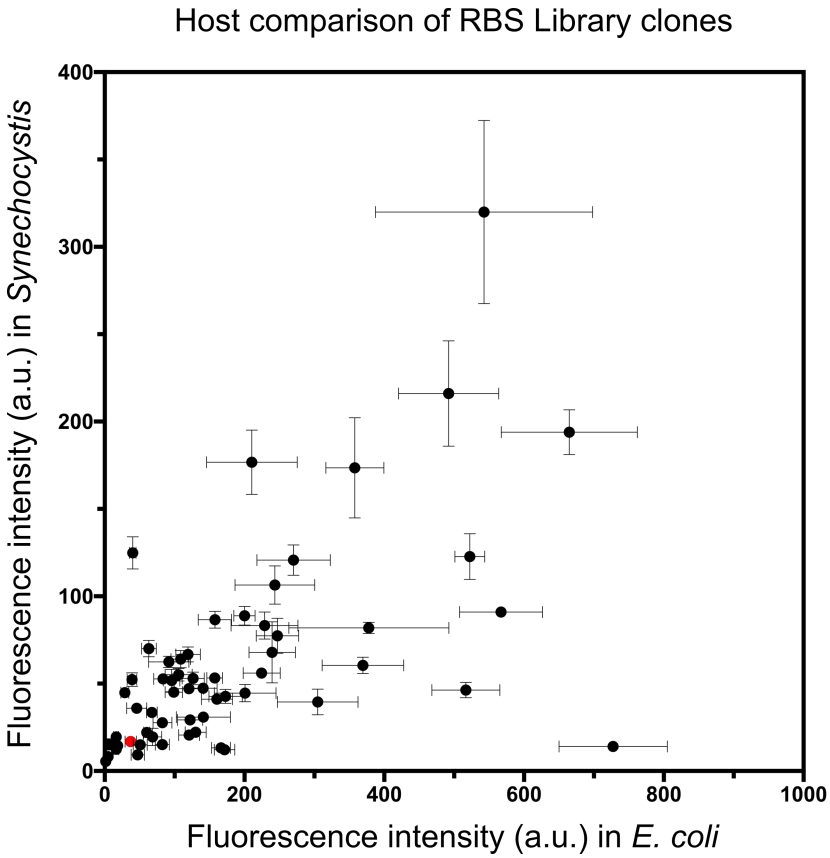


Figure 3.27. Portability of RBSs between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. Fluorescence intensities of the RBSs in *E. coli* cells (as described in Figure 3.23) were plotted against activity of each RBS in *Synechocystis* (as described in Figure 3.25). pGT77 (SPL promoter clone 3; red dot) was included as the positive reference containing the promoter used for the RBS library design. Error bars represent standard deviations of five and three biological repeats for *E. coli* and *Synechocystis*, respectively.

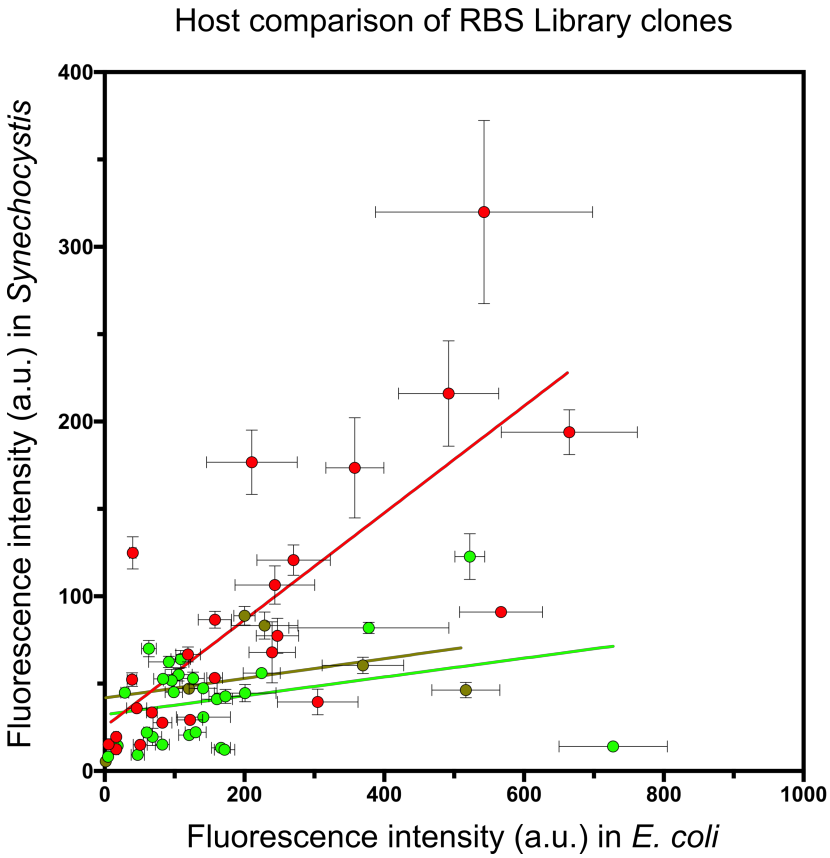


Figure 3.28. Portability of the three different designs used to isolate clones for the RBS library. The 58 clones that make up the RBS library were isolated from three different library designs, RBS library design 1 (red dots), RBS library design 4 (green dots) and RBS library design 9 (gold dots). The portability of the design 1 (red line, R^2 = 0.57), design 4 (green line, R^2 = 0.10) and design 9 (gold line, R^2 = 0.12) were then compared between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. Fluorescence intensities were compared between the two organisms as described in Figure 23.

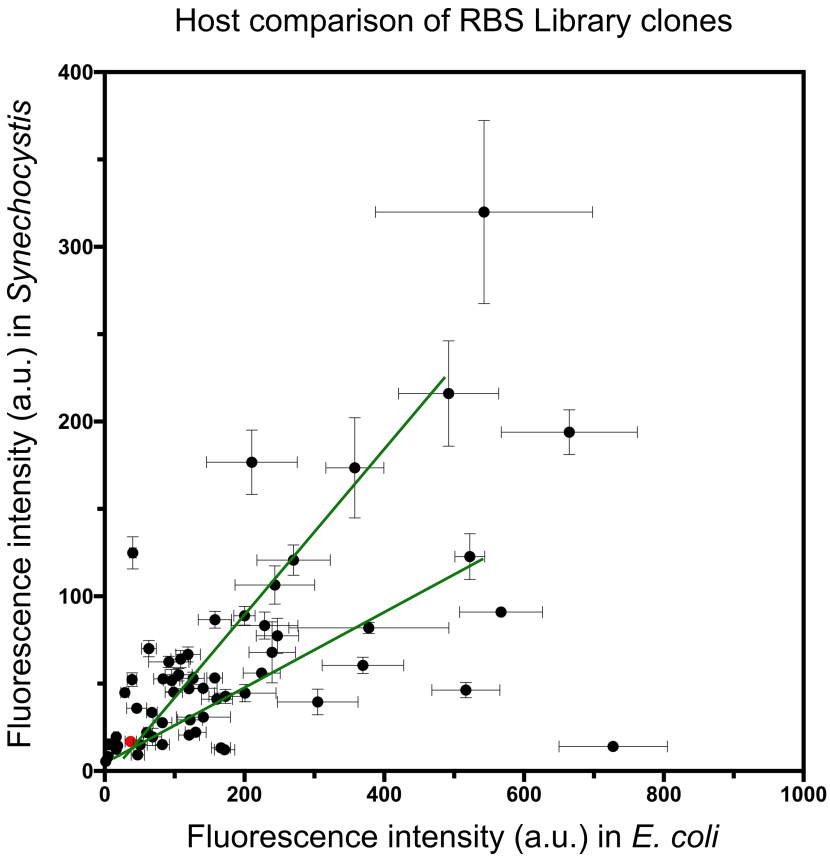


Figure 3.29. Potential subsets of portable RBSs within the RBS library. Subsets of potential portable RBSs (green lines) were highlighted on the RBS library host comparison (described in Figure 3.27). pGT77 (SPL promoter clone 3; red dot) was included as the positive reference containing the promoter used for the RBS library design.

3.3 Discussion

The overall goal of this chapter was to generate portable parts suitable for rapid prototyping, whereby they perform the same or similarly between the non-model and model organism, which in turn would allow protein expression levels to be preserved between the two organisms. To fine-tune protein expression levels, transcription and translation initiation rates were targeted and controlled using libraries of constitutive synthetic promoters and a library of synthetic RBSs, respectively. For these libraries to be suitable as portable parts for rapid prototyping, they would need to: i) offer a wide range of expression strengths in *E. coli* and *Synechocystis* that incrementally increase in a small step-wide manner, which would allow pathway optimisation through fine-tuning protein expression levels; and ii) perform portably as a set of parts between *E. coli* and *Synechocystis*, encoding similar protein expression levels in both organisms, which would allow its performance in *E. coli* to be used as a prediction of its performance in *Synechocystis*. This would allow rapid prototyping of metabolic pathway-encoding constructs in the fast-growing model organism, *E. coli*, and it would

remove the need to iterate the DBTL cycles in the slow-growing non-model organism, *Synechocystis*. The secondary goal was to fill the deficit of expression control parts for cyanobacteria by generating a library of promoters and a library of RBSs that offered a wide range of strengths that could be used to improve the pathway optimisation process in cyanobacteria, regardless of their portability and suitability for the rapid prototyping concept.

The overall goal of this chapter was to generate and/or identify a set of portable parts. Two promoter libraries and an RBS library were generated and in general the performance of these libraries did not correlate well, if at all between *E. coli* and *Synechocystis*. As such, none of the three libraries would serve well as portable parts suitable for rapid prototyping. However, within the promoter and RBS libraries, subsets of parts existed that appeared to offer portable properties suitable for rapid prototyping between *E. coli* and *Synechocystis* and therefore showed that they had the potential to serve as portable parts. In any case, to determine the suitability of subsets as potential portable parts, the Promoter library and RBS library were taken forward for further investigation (Chapter Four).

The secondary goal of this chapter, which was to fill the deficit of expression control parts, was successfully achieved. The promoter and RBS libraries generated are, to date, the largest set of cyanobacterial genetic parts available and span a wide range of expression strengths in *Synechocystis*, increasing in small incremental steps of strength. These libraries are not only perfectly suited for assembling metabolic pathway-encoding constructs, but also for fine-tuning the performance of that pathway. These tools should therefore become a useful part of the cyanobacterial genetic toolkit and should assist the fine-tuning of pathways to optimise the product titres and productivity. It is also worth noting that while the *E. coli* toolkit has an abundance of expression control parts, these two libraries are functional in *E. coli* and could, like in cyanobacteria, be used to fine-tune pathways in *E. coli* to optimise the product titres and productivity.

The first promoter library that was generated using the Synthetic Promoter Library (SPL) approach produced a library of promoters with a wide range of strengths in both *E. coli* and *Synechocystis*. The wide range of promoter strengths observed in both *E. coli* and *Synechocystis* indicates that the context that surrounds the -35 and -10 consensus sequences (the space sequences) is clearly important for the promoter's strength. This is most likely because regions other than the -35 and -10 elements of a

promoter's sequence are responsible for the recognition between the promoter and sigma factor subunit of RNAP holoenzyme. For example, the UP element of a promoter sequence (region between -40 and -60) interacts with the C-terminal domain of the α-subunit of an RNAP and the extended -10 element (-12/-11 sequence at the 5' end of -10 element) interacts with region 2.4 of the sigma-70 subunit (81). It is probable that much like for the -35 and -10 elements, the sequences of the UP element and the extended -10 element affect the binding affinities between the promoter and RNAP, especially considering a consensus sequence has previously been reported for the UP element (461). It is possible that varying the sequences of these elements affects the rate of transcription initiation and therefore promoter strength, which here allowed the SPL approach to generate a library of promoter with a wide range of strengths in both *E. coli* and *Synechocystis*.

Despite generating a library of promoters that offered a wide range of strengths in both *E. coli* and *Synechocystis*, the performance of the promoters did not correlate between the two organisms. Whilst there are similarities between the *E. coli* and *Synechocystis* transcriptional machinery, such as cyanobacterial type 1 promoters that possess both -35 and -10 elements with the consensus sequences TAGACA and TATAAT, respectively (398, 448), these are nearly identical to the consensus sequences for the -35 and -10 elements of an *E. coli* sigma 70 factor-dependent promoter, TTGACA and TATAAT, respectively (449, 450). The similarities in the promoters -35 and -10 consensus sequence is most likely explained by the similarities in the two organism's primary sigma factors, RpoD (accession number: NP_417539.1) the primary vegetative sigma 70 factor in *E. coli*, and SigA (accession number: ALJ69094.1) the primary sigma factor in *Synechocystis* (398), which share large similarities at the C-termini where the -10 element-binding domain differed by only one amino acid and the -35 element-binding domain was perfectly conserved (124). Despite the similarities, there are also reported differences in the transcriptional machinery between these two organisms, which are potentially the reason for the lack of portability in the promoter library. For example, the structure of their RNA polymerase (RNAP), which, while largely are composed of the same generic eubacterial subunits, do have one difference; the β' subunit in *E. coli* is spilt into two parts, the γ and β' subunits in cyanobacteria (398, 448). While the exact effect of these structural differences is unknown, it is easy to hypothesize that this could affect transcription and therefore explain differences in the promoter strength between the organisms.

In an attempt to further populate the promoter library and increase the number of available promoter clones that could be considered as a promoter for a portable subset, a novel library design for a second promoter library was designed. It was hypothesised that the novel library would be a more targeted approach for generating portable promoters. However, the approach did not work as anticipated, as it generated predominantly non-functional promoters in *Synechocystis*, apparently a result of a very high stringency towards the consensus sequence for the -35 and -10 elements in *Synechocystis*. Interestingly, *E. coli* showed little stringency towards the -35 and -10 consensus sequences, with approximately 50% of the promoter clones generated from the new library design proving stronger than the parental promoter.

Following the development of a Promoter library, an RBS library was generated. Rather than use the increasingly popular predictive design tools that are available, such as the RBS Calculator (85), a randomised library design was used. This was more suitable for two reasons: i) the predicative ability of computational design methods for generating RBSs is still very limited in non-model organisms, like cyanobacteria (352); and ii) with no prior knowledge on generating a portable RBS library between *E. coli* and *Synechocystis*, using a random approach made it more probable that different RBS sequences would be tested for portability (compared to a non-random approach, which could, although not necessarily, test a smaller sample of RBS sequences). The RBS library design followed a similar rationale to the SPL method, whereby consensus sequences were conserved and spacer sequences surrounding them were randomised. In the case of an RBS, that meant conserving the SD sequence and randomising the upstream and downstream sequences, which was only possible because *E. coli* and *Synechocystis* share a nearly identical SD and anti-SD sequences (460). However, there was a very weak correlation in the observed strength of the RBSs between *E. coli* and *Synechocystis*. While the translational machinery between the two organisms share similarities, such as SD consensus sequences and anti-SD sequence (460), it is possible that differences in the RBS performance could be a result of different mRNA folding. Differences in mRNA folding might have been caused by slight differences in the two organisms growth conditions, such as temperature, pH, salt concentration, or differences within the cell, which could result in different RNA structures, or stability of those structures, that could block either the 16S rRNA binding site or the standby site (87, 462, 463). Another factor which could have affected RBS strengths and therefore the portability of the RBS library, is the difference in optimal spacing between the 16S rRNA and the start codon of the two organisms (460). Differences in the mRNA concentration between the two organisms

could have been caused by different rates of transcription and/or mRNA degradation which could affect the portable performance of the RBS library between the two organisms.

3.4 Key Outcomes

- A library of 43 constitutive synthetic promoters was generated, which was functional in both *E. coli* and *Synechocystis* and would serve as a useful addition to the cyanobacterial toolkit for optimising the function of a genetic system.
- A library of 60 synthetic RBSs was generated, which was functional in both *E. coli* and *Synechocystis* and would serve as a useful addition to the cyanobacterial toolkit for optimising the function of a genetic system.
- The strengths of neither synthetic promoters nor synthetic RBSs correlated well between *E. coli* and *Synechocystis*, however subsets of portable parts appeared to exist within each library.

4. CHAPTER FOUR. Isolation and validation of a subset of portable promoters and RBS.

4.1 Introduction

4.1.1 Identifying subset combinations in datasets

In the previous chapter (Chapter Three), a library of promoters and RBSs was generated and characterised in both *E. coli* and *Synechocystis*. Little to no correlation was observed when the activities of these parts were compared between the two organisms. However, it did appear that subsets of promoters and RBSs that did correlate well between the two organisms existed within the whole library and that they may have the potential to serve as portable subsets. To identify the best subset of portable parts within each library, ideally every possible combination of parts within each library would be compared. However, such a task would result in a huge number of subset combinations being identified, for example, the number of possible six-part subsets in a set of 100 parts is 1,192,052,400 (Equation 4.1).

$$c = \frac{n!}{r!(n-r)!}$$

Equation 4.1. Equation for calculating the number of combinations in a set. *c* = number of combinations. *n* = the size of the whole set. *r* = the size of the subset. ! = factorial.

Furthermore, identifying every possible subset combination typically results in a combinatorial explosion, whereby there is an exponential increase in the number of combinations as the size of a set increases (Figure 4.1). Not only would identifying all of the subset combinations by eye be effectively impossible, there is no rational nor systematic approach to quantitatively comparing the subsets. A sensible solution for such a task is to use a computational search algorithm to identify all subset combinations within the set, furthermore with such an algorithm it would be possible to quantitatively compare all the subset combinations, making it feasible to identify the best subset from any dataset. Exhaustive (brute force) search algorithms are a perfect tool for such a function as they identify every possible combination of elements within a set, which in this case means identifying every possible subset within the promoter and RBS libraries.

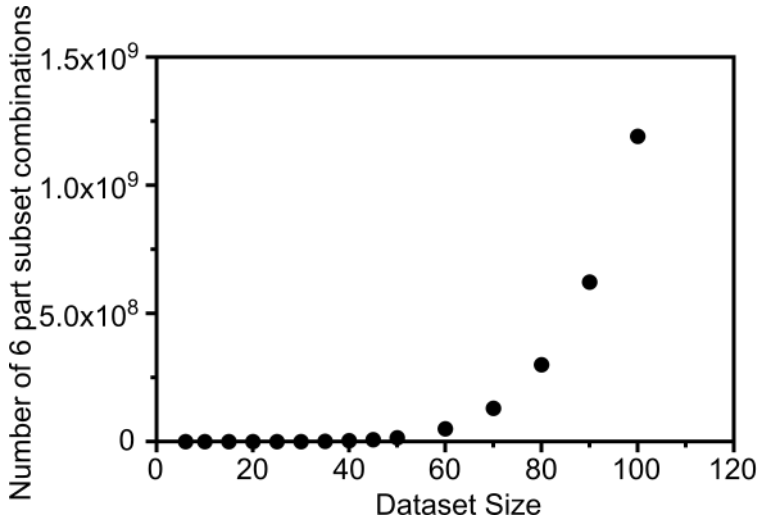


Figure 4.1. Relationship between dataset size and number of subset combinations. As the size of datasets increases the number of six-part subset combinations exponentially increases, resulting in a combinatorial explosion of computational complexity. Calculated using Equation 4.1.

While an exhaustive search algorithm would guarantee finding the best subset, they have an exponential complexity, which results in a combinatorial explosion of complexity for every new element added to a problem. When an algorithm has an exponential complexity, even small problems can have huge computational running times and be extremely computationally expensive. With larger datasets being generated, for example the library of >10,000 insulator variants (464) and the library of >500 transcriptional terminations (127), using the unbound search space of an exhaustive search algorithm is not appropriate as it would be too computationally expensive. One faster alternative to an exhaustive algorithm is a heuristic algorithm, which overcomes the speed and efficiency issues of an exhaustive approach by reducing the search space, that is, the number of candidate elements being considered for potential subsets. A heuristic algorithm, if designed correctly, should be: i) able to compute solutions that are nearly as close to the optimum that an exhaustive algorithm would have identified; while ii) keeping the time complexity low in all instances, especially compared to the exhaustive equivalent; and iii) robust so that it is always effective and efficient for any possible input. The time complexity of a heuristic algorithm depends on the heuristic function implemented, however, the exponential complexity will always be smaller than an exhaustive algorithm, which allows larger problems to be solved.

4.1.2 Context dependence

After identifying the subsets with the most optimal portable properties, context dependence, which has been one of synthetic biology's major challenges, will subsequently be addressed. One of synthetic biology's founding principles was modularity, which was inherited from other engineering disciplines, for example developing large-scale integrated circuits in electronics (465). The concept of modularity is that modular parts would be characterised separately and then composed into systems that perform predictably, based on the characterised properties of the individual parts. In reality, the local genetic context often has a significant effect on the performance of a part (466-470). As a result, context dependence has limited the ability of using modular parts to rationally design a genetic systems performance (394).

In a recent study, over 1,200 part combinations were assembled and characterised using frequently used prokaryotic transcription and translation expression control parts (471). The study quantitatively scored the changing activity of parts across different genetic contexts, revealing that the biggest part interaction effects occur at the 5' UTR:gene of interest (GOI) junction and the promoter:5' UTR junction. These part interaction effects result from changing DNA sequences, which in turn changes mRNA sequences and therefore secondary structures. Different mRNA secondary structures can affect translation initiation and translation elongation rates by interfering with the ability of the ribosome to bind to the mRNA at the SD sequence and the movement of the ribosome along the mRNA sequence, respectively (471). Furthermore, transcription can also be affected by its genetic context, typically the downstream RBS. The first 9 – 15 bases of the mRNA transcript that is synthesised during transcription, which is the initially transcribed sequence (ITS), causes the dissociation between the RNAP-promoter contacts, known as promoter escape (472). The sequence of ITS has been shown to affect the dissociation between the RNAP and promoter and therefore affects promoter escape efficiency, which in turn affects protein expression (473-475). As such, changing the genetic context that surrounds expression control parts, such as promoters and RBSs, typically changes the part's strength (476).

Significant work has been carried out to overcome context dependence of expression control parts. Most notably, in an attempt to minimise the contextual effects around the promoter:5' UTR junction, ribozymes and CRISPR-associated proteins were incorporated upstream of the SD sequence and were used to cleave upstream sequences generated from different promoters (477, 478). Recently, incorporating two SD motifs in the 5' untranslated region (UTR), referred to as the bicistronic design

(BCD), showed increased predictability of RBSs that express different CDSs when used in different combinations with promoters (476).

Rapid prototyping aims to optimise many different metabolic pathways and therefore it is important to understand how a portable subset will perform when used in different contexts, such as combinatorial assemblies and different genes. Here the strength of promoters and RBSs is expected to change as the surrounding genetic context is changed, however hopefully the change in strength in the portable parts is proportional between the two organisms (Figure 4.2). If this is the case, the portable subset would serve as an ideal set of parts to use to combinatorially optimise metabolic pathways in *Synechocystis* via rapid prototyping.

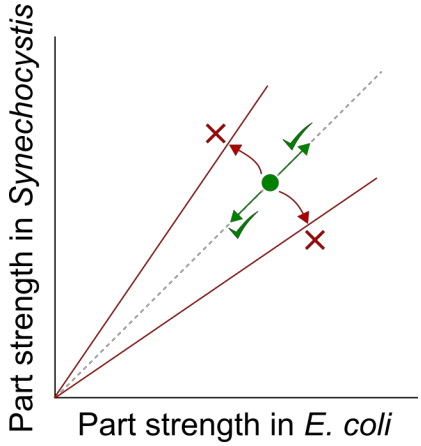


Figure 4.2. Illustrative example of the portability of a part in different genetic contexts. While the strength of each portable part is expected to change when used in different genetic contexts as long as the change in strength is proportional between the two organisms, the parts will still be portable (green arrows). However, if the change in strength is not proportional between the organisms (red arrows), the parts will no longer be portable.

4.1.3 Aims

The aim of this chapter is to develop and implement a heuristic algorithm that can search for subsets within a dataset, quantitatively compare them and identify the best subset with the best portable properties, in this case from the promoter and RBS libraries. The portability of the portable subset can then be validated by testing their performance when used in different contexts. Aims are as listed below:

- To design a scoring function that can be used to quantitatively score subsets based on the portable properties between the two organisms.

- To design an algorithm that incorporates the scoring function with a heuristic search function, which allows the best portable subset to be identified within datasets.
- To run the algorithm to identify the best portable subsets from the promoter and RBS libraries generated in Chapter Three.
- To validate the portable properties of the identified subsets when they are used in different contexts.

Declaration – In this chapter, work carried out on the design and writing of the computational algorithm was done together with Dr. John Heap.

4.2 Results

4.2.1 Designing a computational algorithm for identifying the best portable subset

To identify the subsets with the best portable properties, it was necessary to compare possible subset combinations within each library. To achieve such a task, an algorithm, SUBSETTER, was developed that heuristically searches the libraries for subsets of parts and then quantitatively scores and ranks the subsets based on their portability between the two organisms.

4.2.1.1 *Scoring the portability of a subset*

To allow the scoring and ranking of a subset, a way to measure their portability was required. To measure a subset's portability, the most desirable properties of a portable subset first needed to be determined by considering their exact purpose in rapid prototyping. The idea of rapid prototyping is to use portable parts (promoters and RBSs) to assemble and express metabolic pathways; the portable parts need to perform similarly between the two organisms by encoding similar relative levels and ratios of proteins within a pathway, which will subsequently allow the performance of a pathway in *E. coli* to be used as a prediction of its performance in *Synechocystis*. Considering this, three properties were identified as most desirable in a portable subset: 1) to allow protein expression levels to be accurately predicted, the subset of parts must have a strong correlation between the two organisms (hereafter referred to as the 'fit'; Figure 4.3); 2) the subset must offer a wide dynamic range of expression strengths (hereafter referred to as the 'range'; Figure 4.3) of 3) evenly spaced parts (hereafter referred to as the 'spacing'; Figure 4.3), which is essential for fine-tuning

expression levels. If the parts in a subset had a poor fit they would be far less accurate for predicting the expression levels of proteins and therefore the performance of the pathway between the two organisms. Additionally, if that parts in a subset offered a narrow range of strengths and/or the parts were clustered around one expression level, it would reduce the number of expression levels that can be used for fine-tuning and optimising expression of a metabolic pathway.

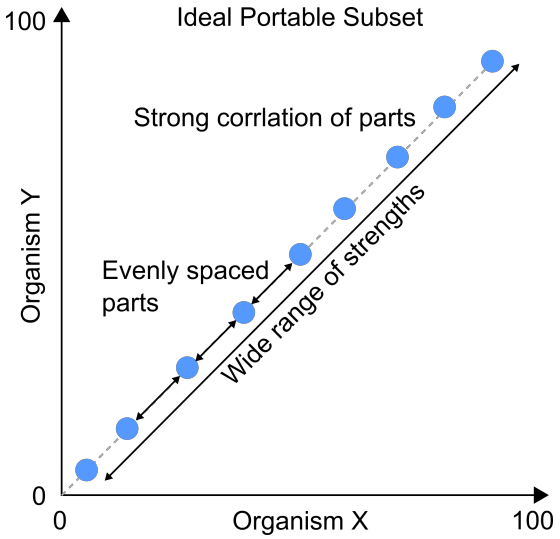


Figure 4.3. Illustration of an ideal portable subset. Strong correlation of parts that will provide accurate predictions of expression level in organism Y from organism X. The parts that make up an ideal portable subset but be evenly spaced over a wide range of strengths.

To implement a subset scoring function a way to measure the fit, range and spacing of a subset was required. Considering first the 'fit' of a subset, a portable parts performance in *E. coli* will be used as a prediction of its performance in *Synechocystis*. Therefore, for each portable part there will be a hypothetical value for what the strength of that part should ideally be in *Synechocystis* based on its strength in *E. coli*. By using the *y*-residual value (the vertical distance between a data point and a given gradient) of a part, the difference between the actual measured and hypothetical strength in *Synechocystis* can be measured. The sum of all the *y*-residual values in a subset can then be used as a measure of a subset's fit; where the larger the score the worse the fit (Figure 4.4a). Measuring the 'range' of a subset was next considered. It was important to consider the range on both the *x*-axis and *y*-axis to ensure that the range of strengths in both organisms was considered. If the range of only one of the two organisms was considered it would be possible for a subset with a large range in one of the organisms but a small range in the other to be scored highly, which would not be beneficial for rapid prototyping. As such, using the hypothenuse would allow the

range of a subset to be measured, where the range in both organisms has been considered (Figure 4.4b). The hypotenuse is the longest side of a right-angled triangle and is calculated using the other two sides. The two other sides of the triangle are calculated by taking the largest x and y values in a subset and defining them as one extreme point, and by taking the smallest x and y values in a subset and defining them as the other extreme point, and then measuring the distance between them.

The last of the three properties is the 'spacing' of the parts in a subset (Figure 4.4c). By measuring the differences between the ideal and actual sizes of the gaps between parts in a subset, a measure of a subsets spacing could be determined. To calculate a subset's spacing score, first the hypothetical 'ideal' spacing on the *x*-axis and the *y*-axis would be calculated for a given subset, based on the number of parts in a subset and its *x*-axis and *y*-axis range. It would then be possible to calculate the difference of the 'actual' *x*-axis and *y*-axis gaps of the given subset from the 'ideal' gaps. The sum of the *x*-axis spacing differences and the sum of the *y*-axis spacing differences would then be used in the subset scoring function.

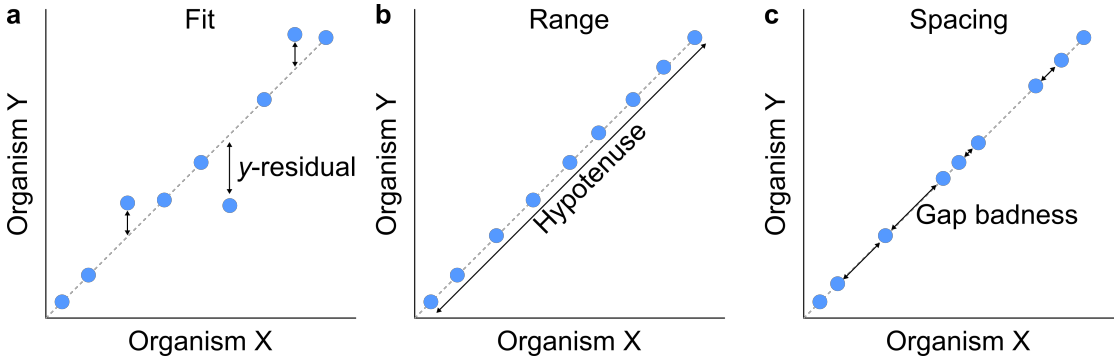


Figure 4.4. Desirable properties that compose the scoring function. The range of a subset was measured using its hypotenuse (a). The fit of a subset was measured as the sum of all the *y*-residual values (b). The gap penalty was taken as the sum of gap badness (c), where gap badness was the sum of all of the measured gaps.

Having established a way to score these three properties, a scoring equation to quantitatively rank the portability of a subset was designed, whereby subsets with better portability have larger positive scores. In this scoring equation, the range of a subset was used as a rewarding factor, as a bigger range would offer a greater range of expression levels. Meanwhile, the fit and spacing of parts in a subset were used as penalising factors, where a bad fit limits portable predictions and uneven spacing limits fine tuning of a pathway (Equation 4.2).

$$Subset\ score = \frac{Range}{\left(1+(Fit*Spacing)\right)}$$

Equation 4.2. Subset scoring equation. The hypothenuse is used as the range and is calculated using the range of the subset of both the x and y axis. The fit is calculated as the sum of the vertical distance between each data point and the given gradient (the sum of the *y* residuals). The spacing is calculated as the average of the sum of the gap badness from the x and y axis. Further details of these calculations are given in section **4.2.2.4**.

4.2.1.2 *Search for subsets within a library*

With the scoring function defined, the next step was to design a search function that would be suitable for identifying subset combinations in the promoter and RBS libraries. As previously discussed, running an exhaustive algorithm to identify every possible subset combination for a realistically sized dataset (< 50 parts = 15,890,700 six-part subsets) would be an unrealistic approach as it would have an exponential time complexity and it would therefore be computationally expensive. Furthermore, an exhaustive algorithm would presumably require the gradient to be calculated for each subset that is being scored by fitting a linear regression (forcing through the origin [0,0]) before scoring each subset, which again would be computationally expensive.

Using a heuristic search as a pragmatic solution to identifying good subsets, would enable bounds and constraints to be applied to the search function. This would focus the algorithm around the biological needs of a portable subset. To identify good subsets with respect to the biological needs of a portable subset, subsets would need to have a strong correlation of performance between the two organisms to allow the predictive ability. In the scoring equation, the 'fit' of a subset is scored based on how well it fits a given gradient. Considering this, the gradient could also serve as a good branch to identify subsets on, because it takes into consideration the performance of a subset in both organisms.

Here a heuristic search algorithm is developed that is bound by the biological objective, to identify good portable subsets, using gradients. Therefore, upper and lower bounds can be specified to constrain the search space and incremental gradients can be searched within the upper and lower bounds. For each gradient the dataset is scanned and a shortlist of parts is identified that would feasibly be good for a portable subset at that gradient. With the shortlisted set of parts at the given gradient each possible subset combination can then be identified and scored. By using this approach, the number of possible combinations is drastically reduced, for example for a shortlist of

six parts, there would only be 720 combinations at a given gradient, in contrast to the 1,192,052,400 combinations of six-part subsets in a whole set of 100 parts.

To implement such a function, the bounds of the search space would need be defined as the maximum and minimum gradient (Figure 4.5a) and the search space size can then be defined as the number of gradients (or increments; Figure 4.5b) between those bounds (Figure 4.5a). To identify subsets with the best portability between the two organisms, every subset was scored with respect to an ideal linear relationship passing through the origin (the origin means the 0 point of both axes). This would then provide a search space framework where the parts in a dataset could be shortlisted at each gradient (Figure 4.5c), based on how accurately the parts would be at predicting the expression level in *Synechocystis* from the expression measured in *E. coli* at the given gradient. Parts in a dataset were therefore shortlisted for a given gradient based on their *y*-residual values where a user-defined number of parts with the lowest *y*-residual values were taken forward. For the subsequent shortlist of parts, every possible subset combination was then identified and scored at the gradient (Figure 4.5d).

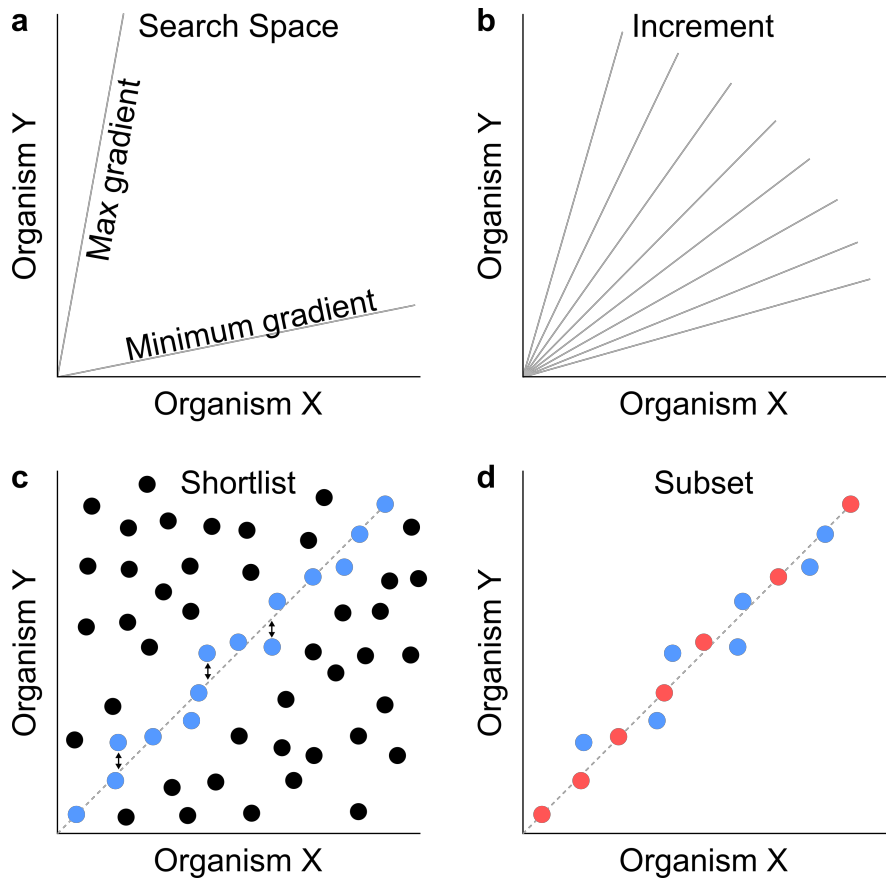


Figure 4.5. Search function parameters of the algorithm. The search function is carried out between a user-defined maximum and minimum gradient (a), where a specified number of increments are searched (b). For each increment that is searched the whole dataset is shortlisted (blue points) to a user-defined number of parts by using their *y*-residual values (c). For each shortlist every possible subset combination is scored (d) with respect to the current gradient and the best score is stored (red points) for that gradient. The search is carried out from the highest gradient through to the lowest gradient.

4.2.1.3 *Implementing a computational tool for search and scoring portable subsets*

Having designed a framework for identifying subsets using a heuristic search algorithm and an approach for scoring the portability of the subsets, the next step was to write a software implementation of the search algorithm using the scoring functions. A script was written using the programming language Python, compatible with both Python 2 and 3, named 'SUBSETTER'. The overall framework of the heuristic algorithm is illustrated in Figure 4.6. To run SUBSETTER it requires the user to input the dataset as a comma separated values file (.csv; Code snippet 1, line 3) and then define five parameters (Code snippet 1, line 5 – 10): the minimum gradient of the search space (gradMin), the maximum gradient of the search space (gradMax), the number of gradients between the gradMin and gradMax (numberofGradients), the number of parts to shortlist to (lengthofShortlists), the minimum length of a subset (minSubsetLength) and the maximum length of a subset (maxSubsetLength). Upon

running SUBSETTER it imports the module 'BIOPORT', which consists of the python code for the search and score functions, and calls the function fFindSubset (Code snippet 1, line 12) to return the best portable subset based on the user defined parameters.

```
1.from BIOPORT import *
2.#User inputs dataset:
3.inputCsv = "TEST.csv"
4.#User defines parameters:
5.gradMin = 0.1
6.gradMax = 10.0
7.numberofGradients = 200
8.lengthofShortlists = 16
9.minSubsetLength = 6
10.maxSubsetLength = 12
11.
12.fFindSubset(inputCsv,gradMin,gradMax,numberofGradients,lengthofShortlists,minSubsetLength
,maxSubsetLength)
```

Code snippet 1. Code for Subsetter.py. Subsetter.py is the master script that runs the SUBSETTER function. In this code, users specify the input parameters for the SUBSETTER function. Upon running SUBSETTER it runs the function fFindSubset and all of the users defined parameters are passed to the function. See comments in code marked with '#' for details.

Here an overview is given of the overall approach that the SUBSETTER uses to identify the best subset in a given dataset, the code is explained in greater depth in section **4.2.2**. The search function (fFindSubset) starts by creating a list of gradients between the user-defined gradient gradMax and gradMin (in descending order from gradMax to gradMin). For each gradient it then generates a shortlist of data points (each corresponding to a part) to a user-defined number (LengthofShortlist) using the *y*-residual values. The shortlisting is coded using the function fShortlister (further described in section **4.2.2.2**). All possible subsets of parts in the shortlist that are between the minSubsetLength and maxSubsetLength are identified using the function fAllSubsets (further described in section **4.2.2.3**) and then using the function fScore, all of the subsets are scored (further described in section **4.2.2.4**). The best subset (highest score) is then stored for that gradient. The overall search algorithm function, fFindSubset, then moves onto the next gradient and shortlists, identifies and scores all subsets. If the score of the best subset at the gradient is worse than the best subset score of the previous gradient, the subset is discarded, however if the best subset score is greater, it replaces the previous best subset and the previous gradients subset is discarded. This process is repeated until all of the gradients have been searched. Once SUBETTER has completed its search it will present a dot plot showing the best score for each gradient searched (Figure 4.7a) and also the best overall subset in the dataset (Figure 4.7b). The dot plot of best scores per gradient allows users to see the

landscape of subsets within the whole dataset, which is beneficial as it is perfectly conceivable that multiple high scoring subsets exist within a dataset, but without the dot plot it would be impossible to identify them. For example, the test dataset shown in Figure 4.7b shows three high scoring subsets. Using the dot plot allows users to ‘zoom in’ (by adjusting the search parameters and re-running the algorithm) on the areas of local optima where subsets have been initially identified and then if necessary the subsets can be compared.

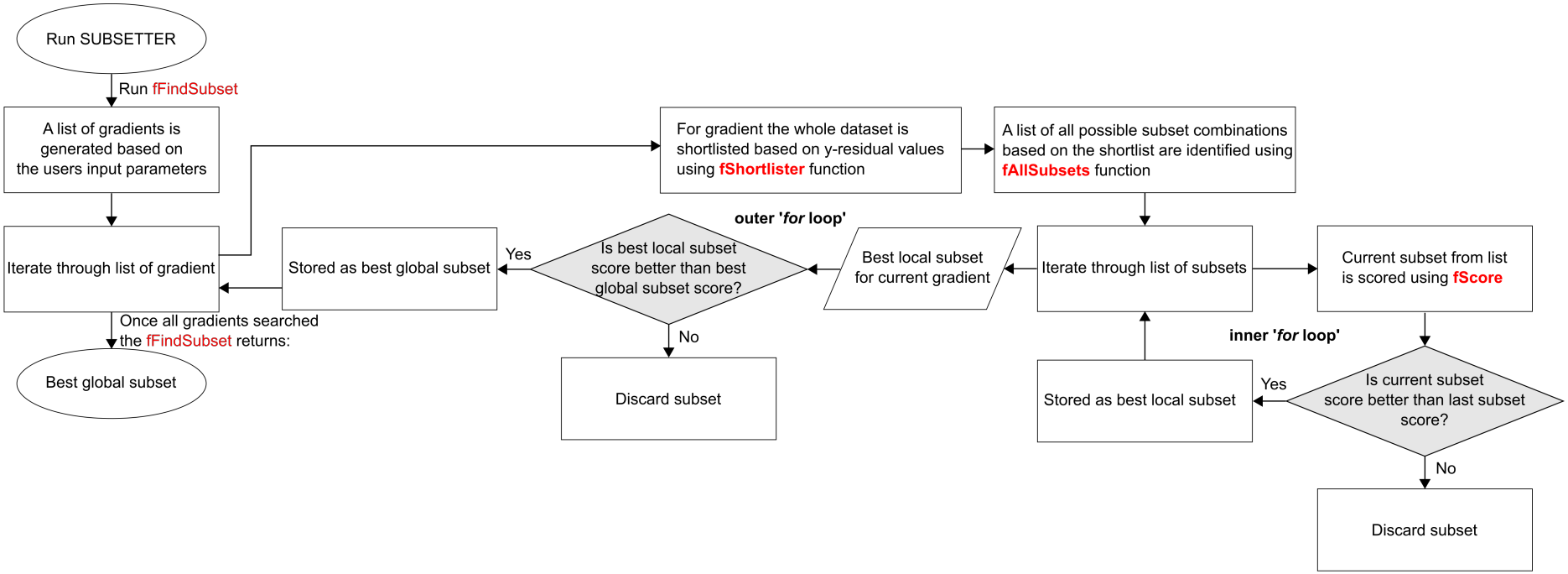


Figure 4.6. Flowchart of the SUBSETTER algorithm. SUBSETTER carries out a heuristic search on datasets, identifying the best portable subset. Defined gradients are used as a framework for the search. The algorithm shortlists datasets to a defined number of data points using *y*-residuals. From the shortlist every possible subset combination is scored, finding the best subset at that gradient. The algorithm search starts at the highest gradient and finds the best subset. The search moves to the next gradient, where the best subset is found. If the score is greater than the previous gradients score SUBSETTER stores that subset and discards the previous. If it is a lower score it is discarded. This process continues through all the gradients until the end of the program run. Functions are shown in red and decisions are highlighted in grey.

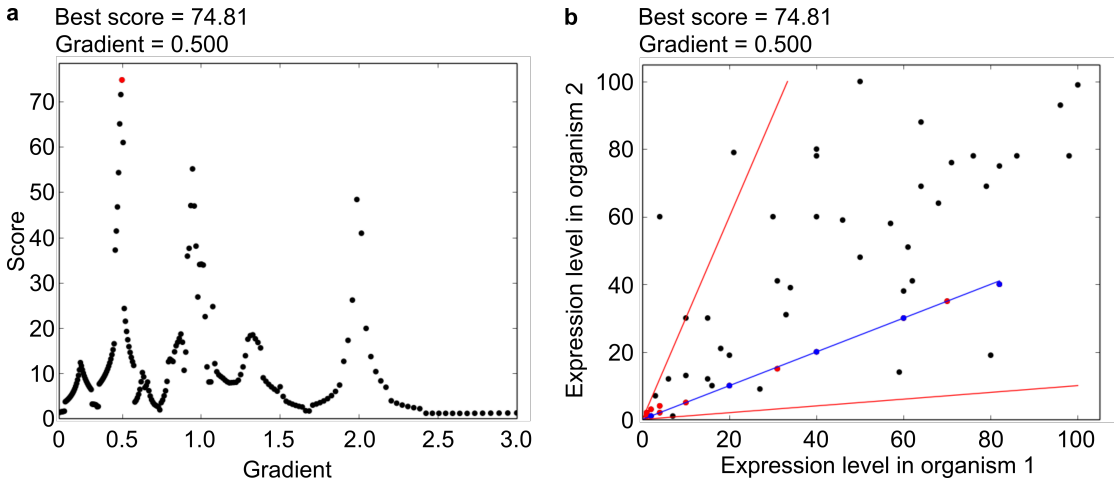


Figure 4.7. SUBSETTER search on test dataset. A SUBSETTER scan using the following parameters: gradMin = 0.1, gradMax = 3, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5 and maxSubsetLength = 8, was ran a test dataset composed solely for this purpose. (a) the dot plot shows the score of the best subset identified at each gradient that was searched and the highest-scoring subset (red dot) that had a score of 74.81 at a gradient of 0.5. (b) shows the *xy*-scatter plot with the test dataset, SUBSETTER's search space (red lines), the shortlisted parts (blue and red dots) for the gradient (blue line; 0.5) that the best subset (blue dots) was identified on.

Disclaimer - if the reader does not require understanding of the code of SUBSETTER and BIOPORT in greater depth, it is not necessary to read section 4.2.2. Please skip to the application of SUBSETTER in section 4.2.3.

4.2.2 Detailed explanation of the BIOPORT code

The BIOPORT module consists of all the code for the five functions, fFindSubset, fCsvToList, fShortlister, fAllSubsets and fScore, which were outlined in section **4.2.1.3**. Those functions are described here in detail.

4.2.2.1 fFindSubset

The code for the heuristic search was defined as a new function called fFindSubset (Code snippet 2). This function first initialises some variable, setting the best score of a subset (bestScore) to zero and by creating two empty lists, one for the gradients searched (termed 'allGradients') and the other for the best score of each gradient searched (termed 'bestScoresPerGradient'). fFindSubset then uses another defined function called fCsvToList, which converts the input dataset into a list of tuples (lists that cannot be changed, a basic type of variable in Python), termed 'theSet', and removing any text. The code can be found in Algorithm A1; line 95 – 104. The user-defined maximum and minimum gradients are then converted into angles; angles

(radians or degrees) are used to measure differences in the plane of two vectors in a linear space which would provide an even spacing of increments between the maximum and minimum gradients (Figure A12a), whereas if gradient values were used as the increment values the spacing between increments would have been unevenly distributed (Figure A12b). To convert the gradMin and gradMax values to angles (radians) the 'numpy.arctan' function was used, which gave gradMinRadians and gradMaxRadians, respectively (Code snippet 2; line 24 and 25). The function 'numpy.linspace' was used to generate a list of evenly spaced values (increments) between gradMinRadians and gradMaxRadians. The values in the list were converted back to gradient and an outer 'loop' was used to iterate through the list of gradients one by one. For each gradient the dataset (theSet) was shortlisted using the function fShortlister (described in section **4.2.2.2**), whereby a user-defined number of data points nearest the line of the current gradient were selected. For the given shortlist, all possible subsets were identified using the function fAllSubsets (described in section **4.2.2.3**) and scored using the function fScore (described in section **4.2.2.4**). Two '*if*' statements were used to compare the score of a subset to the best score at the gradient being searched (bestScoreForThisGradient) and the best score of a subset for the entire search (bestScore). In one or both cases, if the score of the current subset is greater than the current stored value of the bestScoreForThisGradient or bestScore then the score for the current subset replaces the previous best stored value. This process is repeated (using the 'for loop') as the rest of the subsets at the gradient are scored and compared to the bestScoreForThisGradient and bestScore. After all subsets have been scored at a given gradient, the next gradient is shortlisted, searched and then scored, the bestScore will increase each time a better subset score is found until no subset score is greater than the bestScore. The best score of each gradient searched (bestScoreForThisGradient) and the gradient of each gradient searched are both built up in the lists allGradients and bestScoreForThisGradient, which are subsequently plotted as is the subset responsible for the bestScore (See Algorithm A1; lines 49 – 93).

```
8.deffFindSubset(inputCsv,gradMin,gradMax,numberOfGradients,lengthOfShortlists,minSubsetLeng
th,maxSubsetLength):
9.      #Initialise
10.     bestScore = 0.0
11.     allGradients = [] #start empty list []
12.     bestScoresPerGradient = [] #start empty list []
13.     #generate set as list of lists
14.     theSet = fCsvToList(inputCsv)
15.     #OUTER FOR LOOP - for each value of gradient, generate a shortlist of points
16.     #first convert max and min gradients to angles (gives radians, not degrees)
17.     gradMinRadians = np.arctan(gradMin)
18.     gradMaxRadians = np.arctan(gradMax)
19.     #angles are in linear space, so use numpy linspace to generate range
20.     for value in np.linspace(gradMinRadians,gradMaxRadians,num=numberOfGradients):
21.             # convert evenly-space values back to gradients
22.             gradient = np.tan(value)
23.             #record the gradient for score vs gradient plot
24.             allGradients.append(gradient)
25.             #initialise
26.             bestScoreForThisGradient = 0.0
27.             #shortlist
28.             shortlist = fShortlister(theSet,gradient,lengthOfShortlists)
29.             #INNER FOR LOOP-for each shortlist, generate list of all subsets, and score
each subset
30.             for subset in fAllSubsets(shortlist,minSubsetLength,maxSubsetLength):
31.                     scoringResults = fScore(subset,gradient)
32.                     if scoringResults['score'] > bestScoreForThisGradient:
33.                             bestScoreForThisGradient = scoringResults['score']
34.                     #record best subset
35.                     if scoringResults['score'] > bestScore:
36.                             bestScore = scoringResults['score']
37.                             bestShortlist = shortlist
38.                             bestSubset = subset
39.                             bestGradient = gradient
```

Code snippet 2. fFindSubset function. The fFindSubset function creates a list of gradients, based on the user inputs defined in subsetter.py (Code snippet 1). The function uses an outer 'for loop' to iterate through the list of gradients to identify the best global subset for dataset. For each gradient, an inner 'for loop' is used to identify the best local subset for the current gradient. See comments in code marked with '#' for details.

4.2.2.2 fShortlister

To shortlist the dataset at each gradient a new function, fShortlister (Code snippet 3), was defined. It initialises, creating an empty list called 'setBeingShortlisted' (line 179), the list is filled using a 'for loop' that iterates through the dataset establishing the distance that each data point is from the line of the given gradient. This is calculated for each data point using the expected y-value (yExpected), which is calculated based on its measured x-value (x) given a linear model passing through the origin with the current gradient value. The y-residual (yResidual) is calculated as an absolute number (always positive number) by subtracting the yExpected from the measured y-value (yMeasured). The x, yMeasured, yExpected and yResidual values are then added to the setBeingShortlisted list using the 'append' function and this process is repeated for all dataPoints (line 185). The setBeingShortlisted list is then sorted (sortedSet) by yResidual values using the 'sorted' function (line 186), where point [3] denotes the

yResidual values in the setBeing Shortlisted list. The shortlist is then returned based on the sortedSet and the lengthofShortlists that the user defined (line 187).

```
176.#Define a function that shortlists the data points to specified number based on y-residuals
177.def fShortlister(setForShortlisting,gradient,lengthofShortlists):
178.    setBeingShortlised = [] #Make a new empty list
179.    for dataPoint in setForShortlisting: #for each data point calculate y-residual
180.            x = dataPoint[0]
181.            yMeasured = dataPoint[1]
182.            yExpected = gradient * x
183.            yResidual = abs(yExpected - yMeasured) #abs ensure always a positive number
184.            setBeingShortlisted.append([x,yMeasured,yExpected,yResidual])
185.    sortedSet = sorted(setBeingShortlisted,key=lambda point:[3]) #sorts by y-residuals
186.    shortlist = sortedSet[0:lengthofShortlists]
187.    return shortlist
```

Code snippet 3. fShortlister function. The fShortlister function initialises, creating a new empty list, a 'for loop' is used to establish which data points are nearest to the line of the current gradient being searched. The data points that are nearest the current gradient, based on the *y*-residual values, make up the shortlist. See comments in code marked with '#' for details.

4.2.2.3 fAllSubsets

To identify every possible subset from the shortlist of parts for the given gradient the function 'fAllSubsets' (Code snippet 4) was defined. The function initialises, making an empty list called 'subsetList', an outer 'for loop' uses the 'range' function to define a list of the possible lengths that each subset could be, based on user defined minSubsetLength and maxSubsetLength. Using an inner 'for loop' the fAllSubsets function generates a list of all possible subset combinations for each subset length. fAllSubsets returns a list of all possible subsets at the current gradient.

```
189.#Define a new function that returns every subset of the whole dataset
190.def fAllSubsets(theSet,minSubsetLength,maxSubsetLength):
191.    subsetList = [] #Makes a new empty list
192.    #Works through every subset length from minSubsetLength to maxSubsetLength
193.    for eachSubsetLength in range(len(theSet)+1[minSubsetLength:maxSubsetLength+1]:
194.            #For each subset length generate list of all possible subset combination
194.            for subset in list(itertools.combinations(theSet, eachSubsetLength)):
195.                    subsetList += [list(subset)]
196.    return subsetList
```

Code snippet 4. fAllSubsets function. The fAllSubsets function initialises, creating a new empty list, an outer 'for loop' uses a 'range' function to define a list of the possible lengths that each subset could be based on user defined parameters. Within the outer 'for loop' an inner 'for loop' identifies every possible subset at that given length. The function returns a list of all possible subsets at the current gradient. See comments in code marked with '#' for details.

4.2.2.4 fScore

To program the concept of the scoring function as described earlier (section **4.2.1.1**), a new function called fScore was defined (Code snippet 5). The function starts by initialising variable, setting the subset's overall score, sumXGapBadness score, sumYGapbadness score and fitPenalty score all to 0 (Code snippet 5; lines 202 – 205). fScore first calculates the length of a subset (subsetLength) using the 'len' function (line 207). It calculates the fitPenalty of a subset, by using a 'for loop' to add the yResidual value of each point in a subset to the fitPenalty score (line 209 - 212). Having calculated the fitPenalty, fScore then calculates the gapPenalty. To calculate the gapPenalty, sumXGapBadness is first calculated (line 214 – 229) by sorting the points in the subset by their x values using the 'sorted' function. It then takes the first (minimum) and last (maximum) of the sorted x values to calculate the x range. The x range is then used to calculate the optimum size of the gaps in the subset based on its x values (optimumXGap; line 221) by dividing the xRange by the number of gaps in the subset (calculated by: subsetLength – 1). A 'for loop' is then used to calculate the size of each x-axis gap (xGap) for each gap in the subset, the optimumXGap is then subtracted from each xGap to generate XGapBadness. The sumXGapBadness is the sum of all XGapBadness scores. The same process is then repeated for the sum of the *y*-axis gap badness (sumYGapBadness; line231 – 246). The mean of the sumXGapBadness and sumYGapBadness is then calculated to give the gapPenalty score (line 251 / 252). The range score (rangeBonus) is the last factor to be scored and this is achieved by treating the xRange and yRange as the shortest two sides of an equilateral triangle and then using the existing 'math.hypot' function to calculate the hypotenuse (line 249). The overall score is then calculated using the scoring equation (Equation 4.2; line 254).

```
199.#Returns the score and sub-scores of a subset given the specified gradient
200.def fScore(subsetToScore,gradient):
201.    #Initialise
202.    score = 0.0
203.    sumXGapBadness = 0.0
204.    sumYGapBadness = 0.0
205.    fitPenalty = 0.0
206.    #compute subset length
207.    subsetLength = len(subsetToScore)
208.    #compute fitPenalty
209.    for point in subsetToScore:
210.            yResidual = point[3]
211.            fitPenalty = fitPenalty + yResidual
212.    fitPenalty = fitPenalty + 0.01
213.    #sorts by x
214.    sortedSubset = sorted(subsetToScore,key=lambda point: point[0]) #sorts by x
215.    #take x value of first and last points to get xMin and xMax
216.    xMin = sortedSubset[0][0]
217.    xMax = sortedSubset[-1][0]
218.    #compute xRange
219.    xRange = xMax - xMin
220.    #compute optimumXGap
221.    optimumXGap = xRange / (subsetLength - 1)
222.    #compute sumXGapBadness
223.    for index in range(subsetLength)[1:]: #iterates through list [1, ... n]
224.            nextX = sortedSubset[index][0]
225.            thisX = sortedSubset[index-1][0]
226.            xGap = abs(nextX - thisX)
227.            #0.01 added to denominator to avoid division by zero
228.            xGapBadness = (abs(xGap - optimumXGap) / (optimumXGap + 0.01)
229.            sumXGapBadness = sumXGapBadness + xGapBadness
230.    #sort by y
231.    sortedSubset = sorted(subsetToScore,key=lambda point: point[1]) #sorts by y
232.    #take y value of first and last points to get yMin and yMax
233.    yMin = sortedSubset[0][1]
234.    yMax = sortedSubset[-1][1]
235.    #determine yRange
236.    yRange = yMax - yMin
237.    #compute optimumYGap
238.    optimumYGap = yRange / (subsetLength - 1)
239.    #compute sumYGapBadness
240.    for index in range(subsetLength)[1:]: #iterates through list [1, ... n]
242.            nextY = sortedSubset[index][1]
243.            thisY = sortedSubset[index-1][1]
244.            yGap = abs(nextY - thisY)
245.            #0.01 added to denominator to avoid division by zero
246.            yGapBadness = (abs(yGap - optimumYGap) / (optimumYGap + 0.01))
247.            sumYGapBadness = sumYGapBadness + yGapBadness
248.    #compute gap penalty as mean of sumXGapBadness and sumYGapBadness
249.    gapPenalty = (sumXGapBadness + sumYGapBadness) / 2
250.    gapPenalty = gapPenalty + 0.01
251.
252.    #RangeBonus calculated using hypotenuse
253.    rangeBonus = math.hypot(xRange, yRange)
254.
255.    #compute score
256.    score = (rangeBonus) / (1 + (gapPenalty * fitPenalty))
```

Code snippet 5. fScore function. The fScore function initialises and then computes the rangeBonus, gapPenalty and fitPenalty for the subset being scored. These scores are then used in the scoring equation (Equation 4.2) to provide a subset score. See comments in code marked with '#' for details.

4.2.3 Testing SUBSETTER and BIOPORT

To ensure that both the search and scoring functions of SUBSETTER perform correctly, a considerable amount of testing was carried out. Various test datasets were used, such as Anscombe's quartet (a set of four datasets which are nearly identical by typical descriptive statistics, yet appear intuitively when graphed/plotted) and isolation dummy datasets, which allowed the interactions between the scoring function properties to be assessed. Sufficient testing was carried out to ensure that not only the algorithm correctly searched for subsets (Figure A13) but also to ensure that each of the properties in the scoring function were appropriately weighted (Figure A13).

4.2.4 Application of SUBSETTER to search for portable subsets in the promoter and RBS libraries

4.2.4.1 SUBSETTER analysis of the Portable Subset from Promoter Library

Having previously demonstrated that SUBSETTER works as intended on various test datasets, it was next used to identify the best portable subset from the promoter and RBS library. For both the promoter and RBS library a minimum number of (minSubsetLength) five parts and a maximum number of (maxSubsetLength) of ten parts was specified, which would provide a minimum number of 25 expression levels (five promoters x five RBSs) or any number up to the maximum of 100 (ten promoters x ten RBSs), which based on previous combinatorial optimisation, should be more than enough to fine-tune metabolic pathways (115, 149).

The promoter library was initially analysed by running a wide search of the whole library to identify the overall distribution of subset scores across the possible search space. The following parameters were used: gradMin = 0.01, gradMax = 10.0, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5 and maxSubsetLength = 10, which showed that all the high scoring subsets were between the gradients of 0.01 and 1.0 (Figure 4.8a and b). The search was subsequently narrowed between these gradients (Figure 4.8c) and then re-run using SUBSETTER (parameters: gradMin = 0.01, gradMax = 1.0, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5 and maxSubsetLength = 10). From this run, SUBSETTER identified three high scoring subsets (Figure 4.8d); the first subset scored 44.67 (gradient 0.08), the second subset scored ~30 (gradient 0.13) and the third subset scored ~15 (gradient 0.38). Having identified three gradients for the three subsets, the search was then zoomed in on these local optima by running individual high-resolution searches that would find the best possible subset at gradients near to

these. To carry out the high-resolution searches, the gradMin and gradMax were narrowed around the gradients of interest, with the numberofGradients searched being maintained (numberofGradients = 200) in line with the previous runs, to provide a much finer-grained search. Furthermore, the lengthofShortlists was increased from 14 to 16, which allowed more parts to be considered for each possible subset, at the cost of increased computation time. The result of this increased the number of scores (for individual subset-gradient combinations) determined in a given SUBSETTER run from 2,888,600 to 11,226,800.

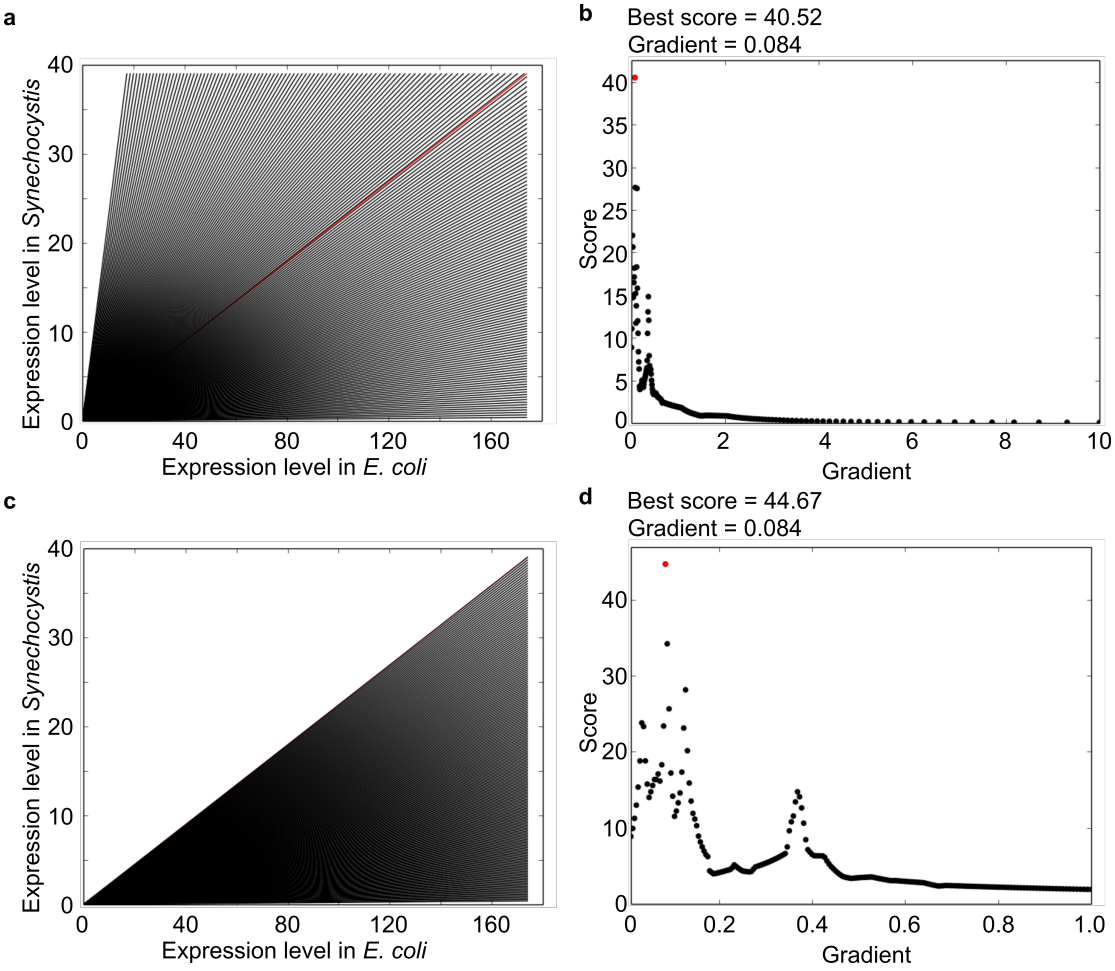


Figure 4.8. Using SUBETTER to identify promoter subsets within the promoter library. (a) A wide scan of the entire promoter library was carried out between the gradients of 0.01 and 10. (b) The search identified the highest-scoring subset (red dot) with a score of 16.60 at a gradient of 0.084, more notably however, it highlighted that all the high scoring subsets were located between the gradients 0.01 and 1.0. (c) A slightly narrowed SUBSETTER search was carried out between the gradients of 0.01 and 1.0. (d) This revealed three subsets of interest, the highest-scoring subset (red dot) with a score of 44.67 at a gradient of 0.084, and two other subsets of interest at the gradients 0.13 and 0.38.

SUBSETTER was first zoomed in around the subset at the gradient 0.08, which was the highest-scoring subset of the original wide scan, using the following parameters: gradMin = 0.07, gradMax = 0.1, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. From this run, the highest-scoring subset had a score of 45.46 at the gradient 0.085 (Figure 4.9a) that was composed of five promoters (hereafter referred to as SubsetP1; Figure 4.9b). SUBSETTER was then zoomed in around the subset at a gradient of 0.13, using the following parameters: gradMin = 0.11, gradMax = 0.14, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. The highest-scoring subset was found at a gradient of 0.126 with a score of 29.81 (Figure 4.9c) and was composed of five promoters (hereafter referred to as SubsetP2; Figure 4.9d). Lastly, SUBSETTER was zoomed in around the subset at a gradient of 0.38, using the parameters: gradMin = 0.35, gradMax = 0.40, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. The highest-scoring subset composed of five promoters (hereafter referred to as SubsetP3; Figure 4.9f) was identified with the gradient 0.370 and a score of 14.99 (Figure 4.9e).

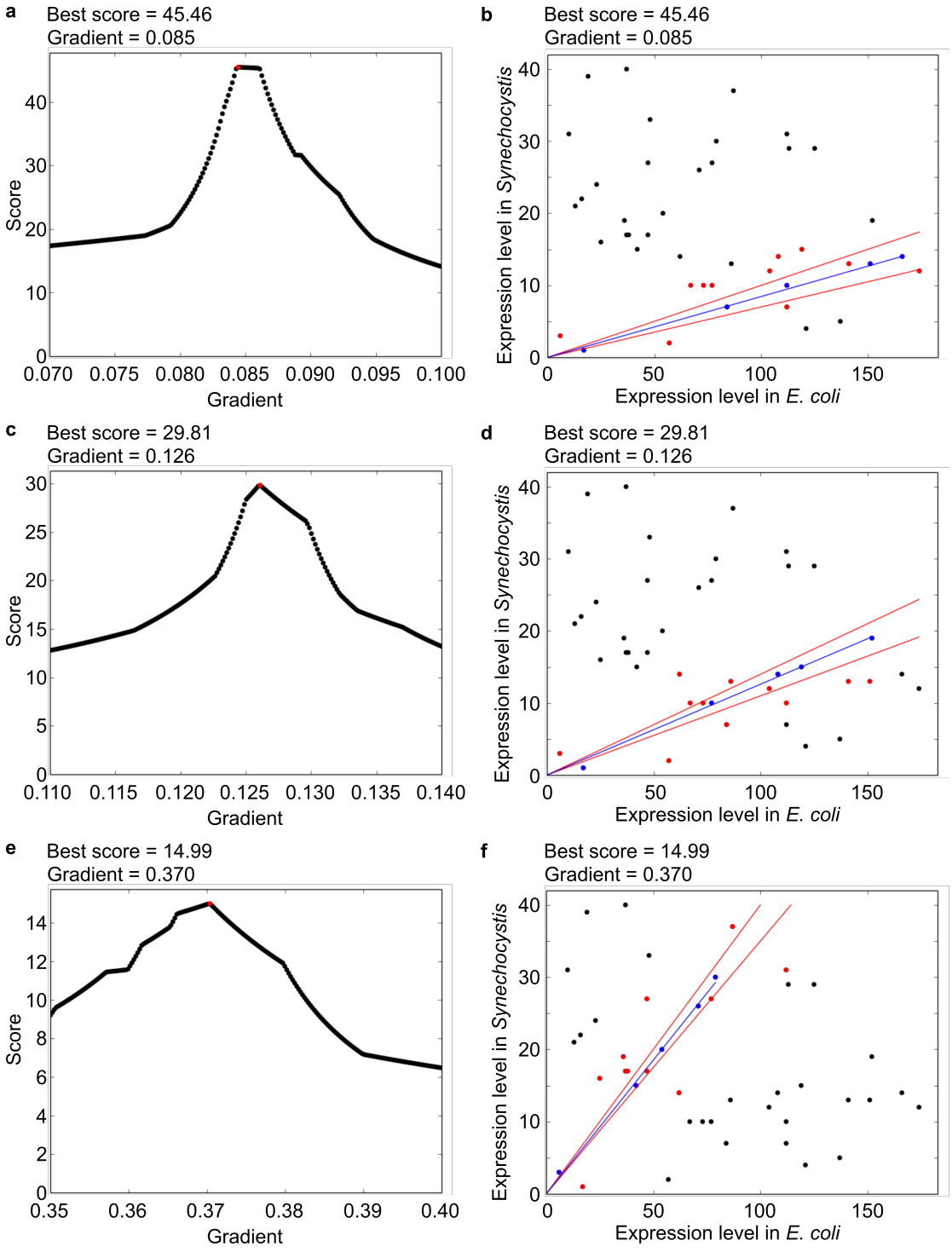


Figure 4.9. Zoomed in SUBSETTER searches around local optima in the promoter library. (a) A zoomed in search between the gradients 0.07 and 0.10 identified the best subset at the gradient 0.085 with a score of 45.46 (red dot). (b) The subset identified at gradient 0.085 (blue line) was composed of five promoters (blue dots) that made the best subset from the 16 shortlisted promoters (blue and red dots) between the searched space (gradients 0.07 and 0.10; red lines). (c) A zoomed in search between the gradients 0.11 and 0.14 identified the best subset at the gradient 0.126 with a score of 29.81 (red dot). (d) The subset identified at gradient 0.126 (blue line) was composed of five promoters (blue dots) that made the best subset from the 16 shortlisted promoters (blue and red dots) between the searched space (gradients 0.11 and 0.14; red lines). (e) A zoomed in search between the gradients 0.35 and 0.40

identified the best subset at the gradient 0.37 with a score of 14.99 (red dot). (f) The subset identified at gradient 0.37 (blue line) was composed of five promoters (blue dots) that made the best subset from the 16 shortlisted promoters (blue and red dots) between the searched space (gradients 0.35 and 0.40; red lines).

The three subsets were then more closely compared. While SubsetP1 had the highest score, this was largely attributed to its good fit (BestfitPenalty = 1.35), good spacing (BestgapPenalty = 1.70) and good range in *E. coli* (bestxRange = 149). However, it had a very low range in *Synechocystis* (bestyRange = 13). SubsetP2 also had a good fit (BestfitPenalty = 2.00) and good spacing (BestgapPenalty = 1.89) of promoters, but like in SubsetP1 it had a big range in *E. coli* (bestxRange = 135), although a small range in *Synechocystis* (bestyRange = 18). SubsetP3 also had a good fit (BestgapPenalty = 2.38) and good spacing (BestgapPenalty = 1.76) of the five promoters and whilst its range was smaller in *E. coli* (bestxRange = 73) compared to SubsetP1 and SubsetP2, it had the biggest of the three subsets in *Synechocystis* (bestyRange = 27). For this reason, SubsetP3 was chosen as the best portable subset of promoters, as it would provide the best range of expression strengths in *Synechocystis*. It was composed of the five promoters: SPLc28, SPLc107, SPLc69, SPLc157 and PLD2c56 (Table 4.1).

Table 4.1. Table of portable promoters that make up SubsetP3. Further described in Table A1 and A2.

Promoter ID	Sequence
SPLc28	CCCACATAACTTGACAGAAGTCCGCAGCGACTTTATAATACGGGG
SPLc107	CGCCCTACCATTGACATCCCGTTCCCGCCGCTGTATAATTGATTT
SPLc69	GTCGCCCCCATTGACAAATGTAAGTCGCGGAGGTATAATAGGTGC
SPLc157	CACCGGGCCTTTGACAATACGTATCAAGTACCTTATAATAAGGGG
PLD2c56	CATTCCGTGCTTGCGTCATTATTAAACTATGAATACCTTGTGTAA

4.2.4.2 SUBSETTER analysis of the RBS Library

Having identified a portable subset of promoters, the next step was to identify a portable subset of RBSs. The RBS library was initially analysed by running a wide search of the whole library to identify the overall distribution of subset scores across the possible search space (Figure 4.10a). The following parameters were used: gradMin = 0.01, gradMax = 10.0, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5 and maxSubsetLength = 10, which showed that all the high scoring subsets were between the gradients of 0.01 and 1.0 (Figure 4.10b). The search was subsequently narrowed between these gradients (Figure 4.10c) and then re-run

using SUBSETTER (parameters: gradMin = 0.01, gradMax = 1.0, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5 and maxSubsetLength = 10). From this run, SUBSETTER identified four high scoring subsets (Figure 4.10d); the first subset scored ~35 (gradient = ~0.15), the next (and best) subset scored 50.26 (gradient 0.249), the third subset scored ~35 (gradient = ~0.42) and the fourth subset scored ~20 (gradient = ~0.55). Having identified four gradients for the four subsets, the search was then zoomed in on these local optima by running individual high-resolution searches that would find the best possible subset at these gradients. The result of zooming in the search increased the number of scores determined in a given SUBSETTER run from 2,888,600 to 11,226,800.

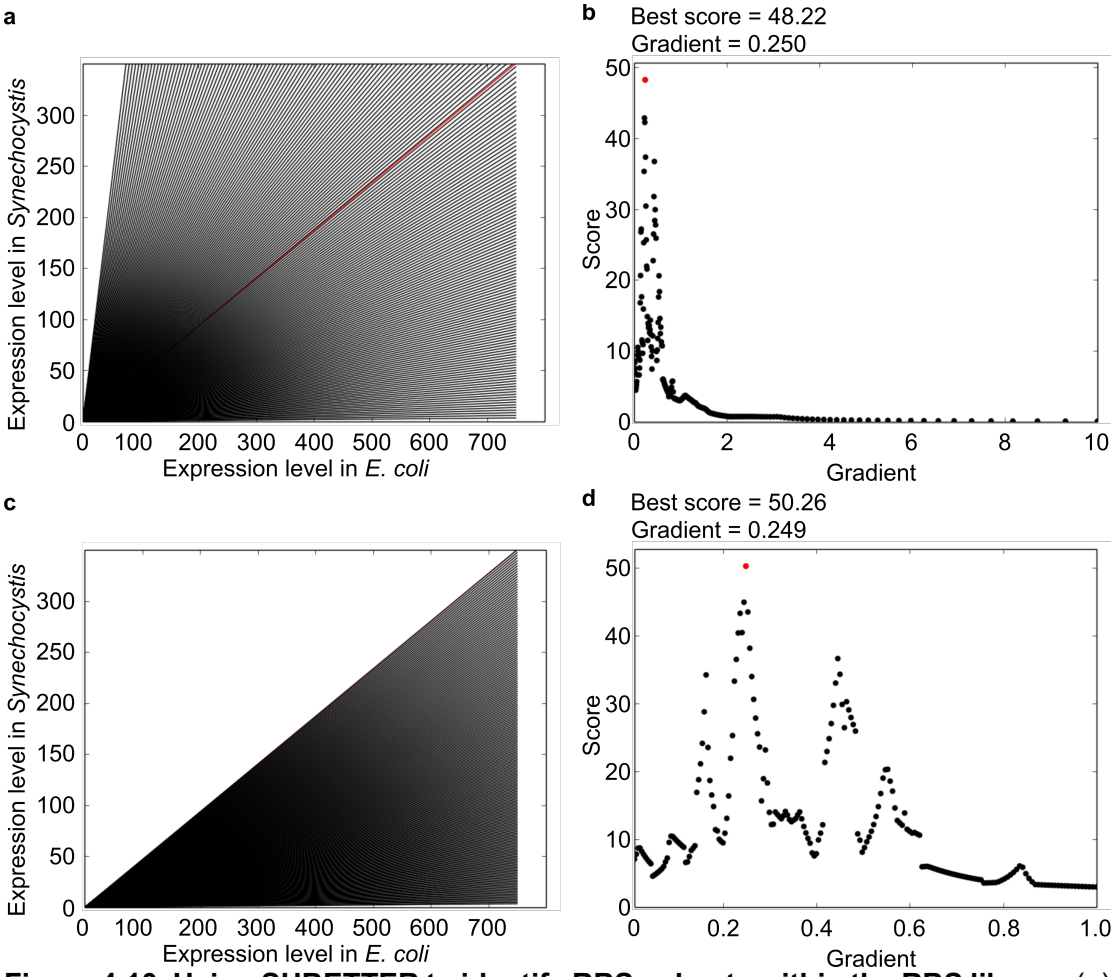


Figure 4.10. Using SUBETTER to identify RBS subsets within the RBS library. (a) A wide scan of the entire RBS library was carried out between the gradients of 0.01 and 10. (b) The wide scan identified the highest-scoring subset (red dot) with a score of 48.22 at a gradient of 0.250, more notably, however, it highlighted that all the high scoring subsets were located between the gradients 0.01 and 1.0. (c) A slightly narrowed SUBSETTER search was carried out between the gradients of 0.01 and 1.0. (d) This search revealed four subsets of interest, the highest-scoring subset (red dot) with a score of 50.26 at a gradient of 0.249, and three other subsets of interest at the gradients approximately around 0.15, 0.42 and 0.55.

The first subset that was identified around the gradient 0.15 (hereafter referred to as SubsetR1) was zoomed in on using the following parameters: gradMin = 0.1, gradMax = 0.2, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. From this run, SUBSETTER identified the highest-scoring subset with a score of 37.50 (Figure 4.11a; gradient = 0.162) that was composed of five RBSs (Figure 4.11b). The second subset that was identified from the wide scan at the gradient 0.249 (hereafter referred to as SubsetR2) was zoomed in on using the following SUBSETTER parameters: gradMin = 0.2, gradMax = 0.3, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. The highest-scoring subset was identified with a score of 50.26 (Figure 4.11c; gradient = 0.149) and composed of five RBSs (Figure 4.11d). The third subset initially identified around the gradient of 0.42 (hereafter referred to as SubsetR3) was zoomed in for a finer search using the following SUBSETTER parameters: gradMin = 0.4, gradMax = 0.5, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. The highest-scoring subset was identified with a score of 36.70 (Figure 4.11e; gradient = 0.448) and was composed of five RBSs (Figure 4.11f). The fourth subset that was identified at a gradient around 0.55 (hereafter referred to as SubsetR4) was zoomed in on using the following parameters: gradMin = 0.5, gradMax = 0.6, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength = 5 and maxSubsetLength = 10. From this run, the highest-scoring subset was identified with a score of 21.10 (Figure 4.11g; gradient = 0.551) that was composed of five RBSs (Figure 4.11h).

The four subsets were then more closely compared. SubsetR1 offered a good fit (bestfitPenalty = 4.76), spacing (bestgapPenalty = 2.74) and range in *E. coli* (bestxRange = 520), but a small range in *Synechocystis* (bestyRange = 82). Subset R2 showed similar properties to SubsetR1, with a good fit (bestfitPenalty = 9.01), spacing (bestgapPenalty = 0.51) and range of RBSs in *E. coli* (bestxRange = 223) but a narrow range in *Synechocystis* (bestyRange = 50). SubsetR3, had a good spacing (bestgapPenalty = 0.51) and big range in *E. coli* (bestxRange = 268) and *Synechocystis* (bestyRange = 115) but a poor fit (bestfitPenalty = 13.64). SubsetR4, was again similar to SubsetR1 and SubsetR2 showing a good fit (bestfitPenalty = 8.44) and spacing (bestgapPenalty = 0.78) of the five RBSs but a narrower range in *E. coli* (bestxRange =142) and *Synechocystis* (bestyRange = 75). From this, SubsetR3 was chosen as the RBS portable subset, principally because it offered the widest range of expression strengths in *Synechocystis* which will be most beneficial for metabolic

pathway optimisation. It was composed of the five promoters: RBSc30, RBSc267, RBSc184, RBSc92 and RBSc249 (Table 4.2).

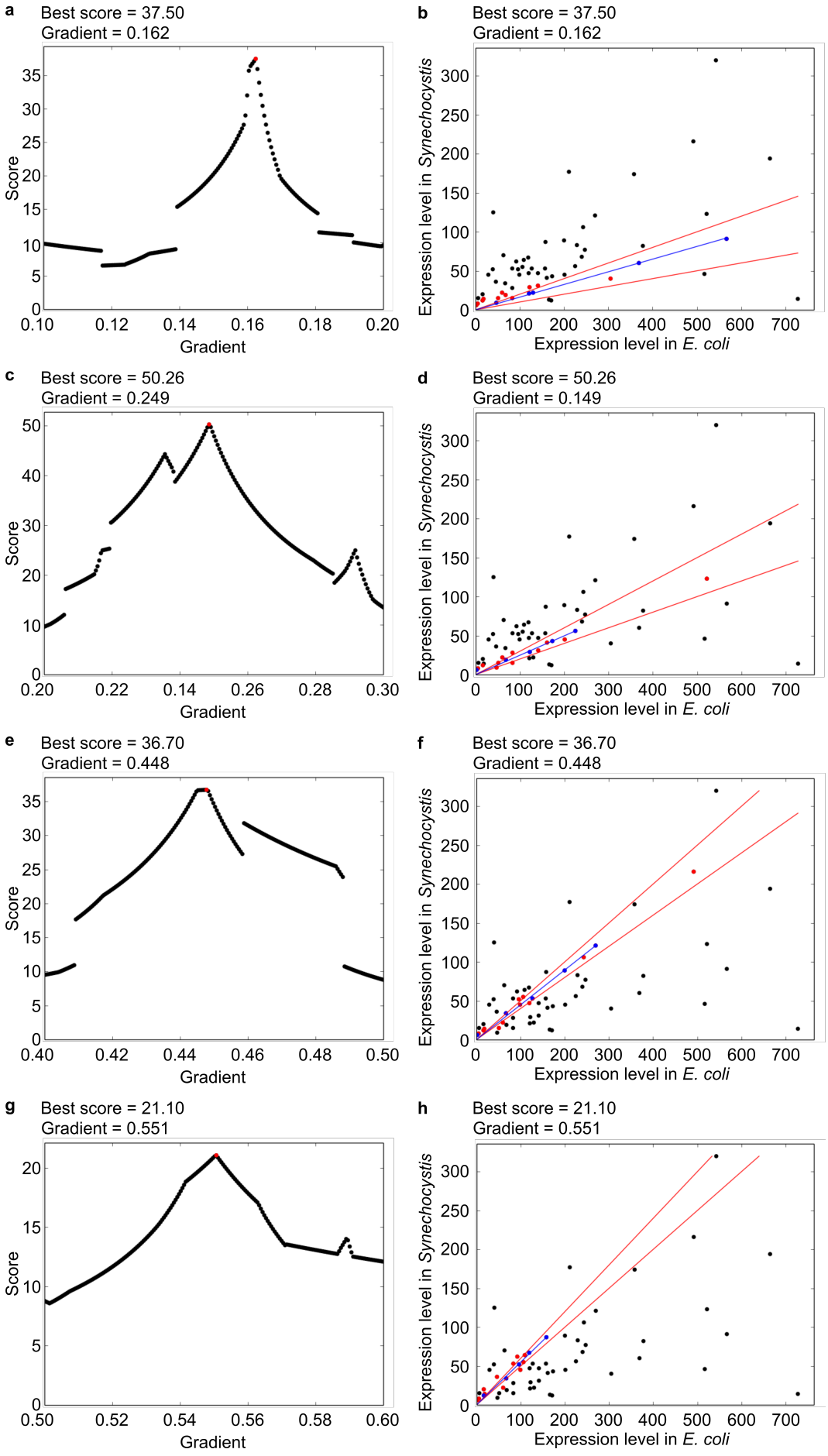

a
Best score = 37.50
Gradient = 0.162
Score
Gradient
b
Best score = 37.50
Gradient = 0.162
Expression level in Synechocystis
Expression level in E. coli
c
Best score = 50.26
Gradient = 0.249
Score
Gradient
d
Best score = 50.26
Gradient = 0.149
Expression level in Synechocystis
Expression level in E. coli
e
Best score = 36.70
Gradient = 0.448
Score
Gradient
f
Best score = 36.70
Gradient = 0.448
Expression level in Synechocystis
Expression level in E. coli
g
Best score = 21.10
Gradient = 0.551
Score
Gradient
h
Best score = 21.10
Gradient = 0.551
Expression level in Synechocystis
Expression level in E. coli

Figure 4.11. Zoomed in SUBSETTER searches around local optima in the RBS library. (a) A zoomed in search between the gradients 0.10 and 0.20 identified the best subset at the gradient 0.162 with a score of 37.50 (red dot). (b) The subset identified at gradient 0.162 (blue line) was composed of five RBSs (blue dots) that made the best subset from the 16 shortlisted RBSs (blue and red dots) between the searched space (gradients 0.10 and 0.20; red lines). (c) A zoomed in search between the gradients 0.20 and 0.30 identified the best subset at the gradient 0.249 with a score of 50.26 (red dot). (d) The subset identified at gradient 0.249 (blue line) was composed of five RBSs (blue dots) that made the best subset from the 16 shortlisted RBSs (blue and red dots) between the searched space (gradients 0.20 and 0.30; red lines). (e) A zoomed in search between the gradients 0.40 and 0.50 identified the best subset at the gradient 0.448 with a score of 36.70 (red dot). (f) The subset identified at gradient 0.448 (blue line) was composed of five RBSs (blue dots) that made the best subset from the 16 shortlisted RBSs (blue and red dots) between the searched space (gradients 0.40 and 0.50; red lines). (g) A zoomed in search between the gradients 0.50 and 0.60 identified the best subset at the gradient 0.551 with a score of 21.10 (red dot). (f) The subset identified at gradient 0.551 (blue line) was composed of five RBSs (blue dots) that made the best subset from the 16 shortlisted RBSs (blue and red dots) between the searched space (gradients 0.50 and 0.60; red lines).

Table 4.2. Table of portable RBSs in SubsetR3. Further described in Table A3.

RBS ID	Sequence
RBSc30	ATTTCGTATGAAGGTAAAGGAGGTTGGCA
RBSc267	GTTTCTTAGGTGTTGGAGGTTGGTCTA
RBSc184	CTGCGTGACAACAGATGTCGAGGTGAAAG
RBSc92	AAACCGAACAGGCCAAAGGAGGTCAGCT
RBSc249	GGCGCGAGTCAACGTTCCTGAGGAGTGGC

4.2.4.3 Gradient dependence: Do promoter and RBS subsets need to be isolated with the same gradient.

With the portable subset of promoters (Table 4.1) and RBSs (Table 4.2) identified, when combinatorially assembling any given gene, it would be possible to generate 25 expression levels (the result of multiplying the number of promoters by the number of RBSs). Looking forward, this would be ideal for predictively fine-tuning and combinatorially optimising pathways in *Synechocystis* via rapid prototyping in *E. coli*.

Here, the two best portable subsets had been identified (using SUBSETTER) independently to one another. As a result, the two subsets were chosen on different gradients (0.37 and 0.44 for promoter and RBS subsets, respectively). It was hypothesized that using subsets of promoters and RBSs with different gradients, such as these, together in a combinatorial assembly, would result in an “expression space” (Figure 4.12b), which would be disadvantageous for precise predictions of *Synechocystis* expression levels in *E. coli*. It was hypothesised that using subsets of

promoters and RBSs in a combinatorial assembly, which have the same gradients, would provide a stronger portable correlation between the two organisms (Figure 4.12a). This would in turn allow better prediction of expression levels in *Synechocystis* from *E. coli*.

In an attempt to improve the predictive ability of the portable subset between *E. coli* and *Synechocystis* as much as possible, alternative subsets of promoters and RBSs, which had the same gradient, would be chosen. If the promoter subsets and RBS subsets were obtained separately, then the best promoter subset and RBS subset would likely have different gradients. As it is preferential to have the same or very similar gradients, the same SUBSETTER algorithm would achieve this by evaluating both promoters and RBSs together, using a merged dataset. This approach was chosen to identify the new portable subset, however it would require the promoter and RBS library to be overlaid without introducing changes to the shape and/or distribution of either dataset.

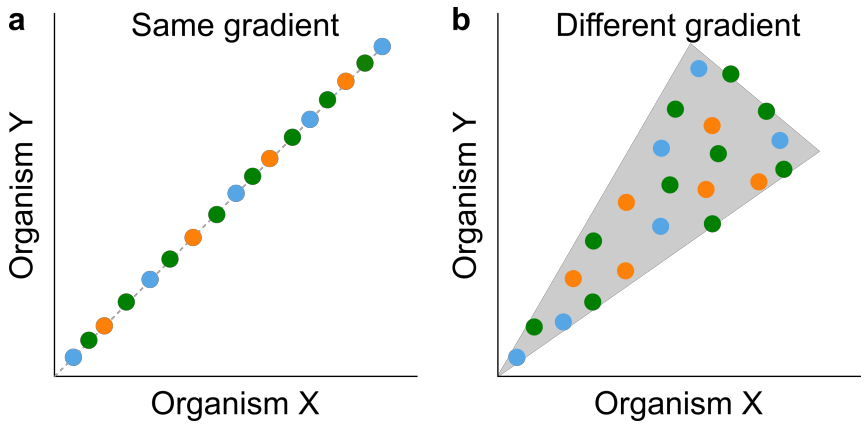


Figure 4.12. Schematic representation of using portable subsets in combinatorial assemblies. (a) Using portable subsets of promoters and RBSs together in combinatorial assembly's that have the same gradient should result in a strong correlation of expression levels between the two organisms, therefore providing highly accurate levels of prediction for rapid prototyping. (b) Using portable subsets of promoters and RBSs that have different gradients, would result in an "expression space", which would limit the predictive power of rapid prototyping.

To overlay the promoters and RBS libraries, both datasets needed to be transformed to the same scale so that the algorithm gives promoters and RBSs equal weighting. This is necessary and appropriate because all the values are relative, not absolute, as all the promoter strengths were determined in constructs with a single fixed RBS, and all the RBS strengths were determined in constructs with a single fixed promoter. To transform the two datasets, they were both scaled to a maximum 1000. In both the

promoter and RBS library the maximum value was a fluorescent intensity measured in *E. coli*; for the promoter library it was SPL clone 31 and for the RBS library it was RBSL clone 133. The two datasets were transformed and merged together and the resultant merged library (Fig. 4.13) could then be used for SUBSETTER analysis.

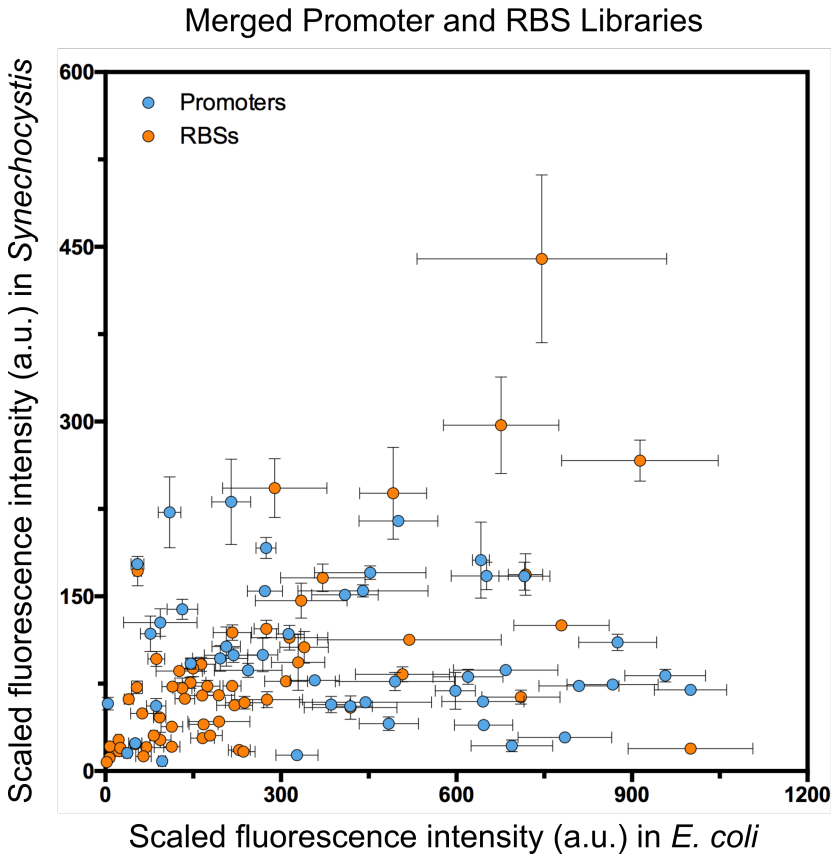


Figure 4.13. Host comparison of the merged promoter and RBS library between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. The fluorescence intensity values presented represent the fluorescence of cells in mid-exponential phase of growth measured using flow cytometry. The promoter (blue dots) and RBS (orange dots) library datasets were scaled to a maximum 1000. For the promoters it was SPLc31 and for the RBSs it was RBSc133. The *x*-axis error bars shown represent the standard deviation of five *E. coli* biological repeats. The *y*-axis error bars shown represent the standard deviation of three *Synechocystis* biological repeats.

4.2.4.4 SUBSETTER analysis of the merged dataset

Having overlaid the promoter and RBS libraries, the next step was to identify the best portable subset from the merged dataset. A minimum number of ten parts (minSubsetLength) and a maximum number of 15 parts (maxSubsetLength) was specified. The resultant number of expression levels that the subset would generate would depend on the composition of promoters and RBSs that make up the final subset.

The merged dataset was initially analysed by running a wide search of the whole library to identify the overall distribution of subset scores across the possible search space (Figure 4.14a). The following parameters were used: gradMin = 0.01, gradMax = 10.0, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength =10 and maxSubsetLength = 15, which showed that all the high scoring subsets were between the gradients of 0.01 and 1.0 (Figure 4.14b). The search was subsequently narrowed between these gradients (Figure 4.14c) and then re-run using SUBSETTER (gradMin = 0.1, gradMax = 1.0, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength =10 and maxSubsetLength = 15). From this run, SUBSETTER identified two high scoring subsets (Figure 4.14d), the highest-scoring subset had a score of 6.45 (gradient = 0.44; red dot Figure 4.14d) and the second subset a score of 6.44 (gradient = 0.37). Having identified two gradients for the two subsets, the search was then zoomed in on these local optima by running individual high-resolution searches that would find the best possible subset at these gradients.

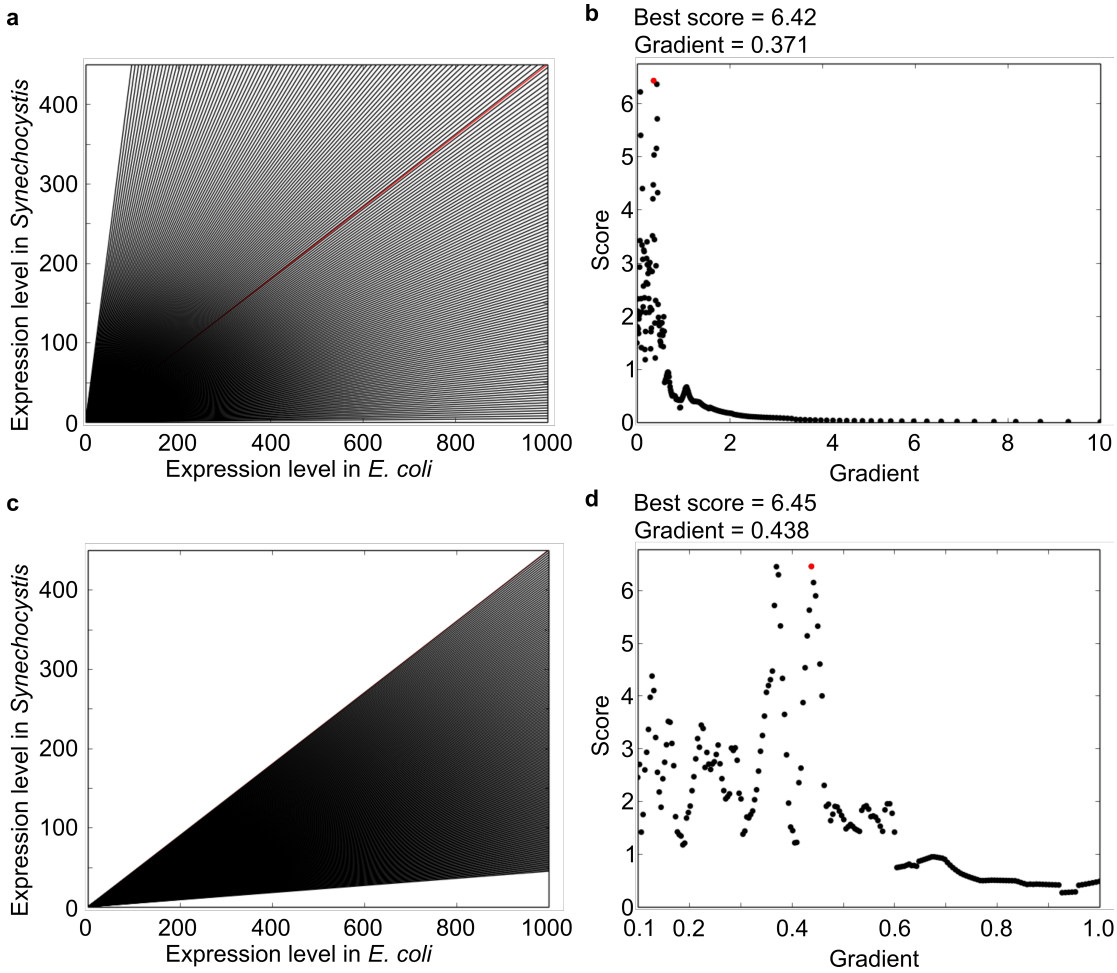


Figure 4.14. Using SUBETTER to identify a portable subset within the merged promoter and RBS library. (a) A wide scan of the entire merged library was carried out between the gradients of 0.01 and 10. (b) The wide scan identified the highest-scoring subset (red dot) with a score of 6.42 at a gradient of 0.371, more notably, however, it highlighted that all the high scoring subsets were located between the gradients 0.01 and 1.0. (c) A slightly narrowed SUBSETTER search was carried out between the gradients of 0.1 and 1.0. (d) This search revealed two subsets of interest, the highest-scoring subset (red dot) with a score of 6.45 at a gradient of 0.438, and the other subset scored approximately 6.4 at a gradient around 0.37.

The first subset that was identified around the gradient 0.37 (hereafter referred to as SubsetM1) was zoomed in on using the following parameters: gradMin = 0.32, gradMax = 0.4, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength =10 and maxSubsetLength = 15. From this run, SUBSETTER identified the highest-scoring subset, which had a score of 6.476 (Figure 4.15a; gradient = 0.369) and was composed of 12 parts (Figure 4.15b, blue dots). The second subset that was identified from the wide scan at the gradient 0.44 (hereafter referred to as SubsetM2) was zoomed in on using the following SUBSETTER parameters for: gradMin = 0.40, gradMax = 0.46, numberofGradients = 200, lengthofShortlists = 16, minSubsetLength =10 and maxSubsetLength = 15. From this run, SUBSETTER identified the highest-

scoring subset with a score of 6.476 (Figure 4.15c; gradient = 0.149), which was composed of 11 parts (blue dots, Figure 4.15d).

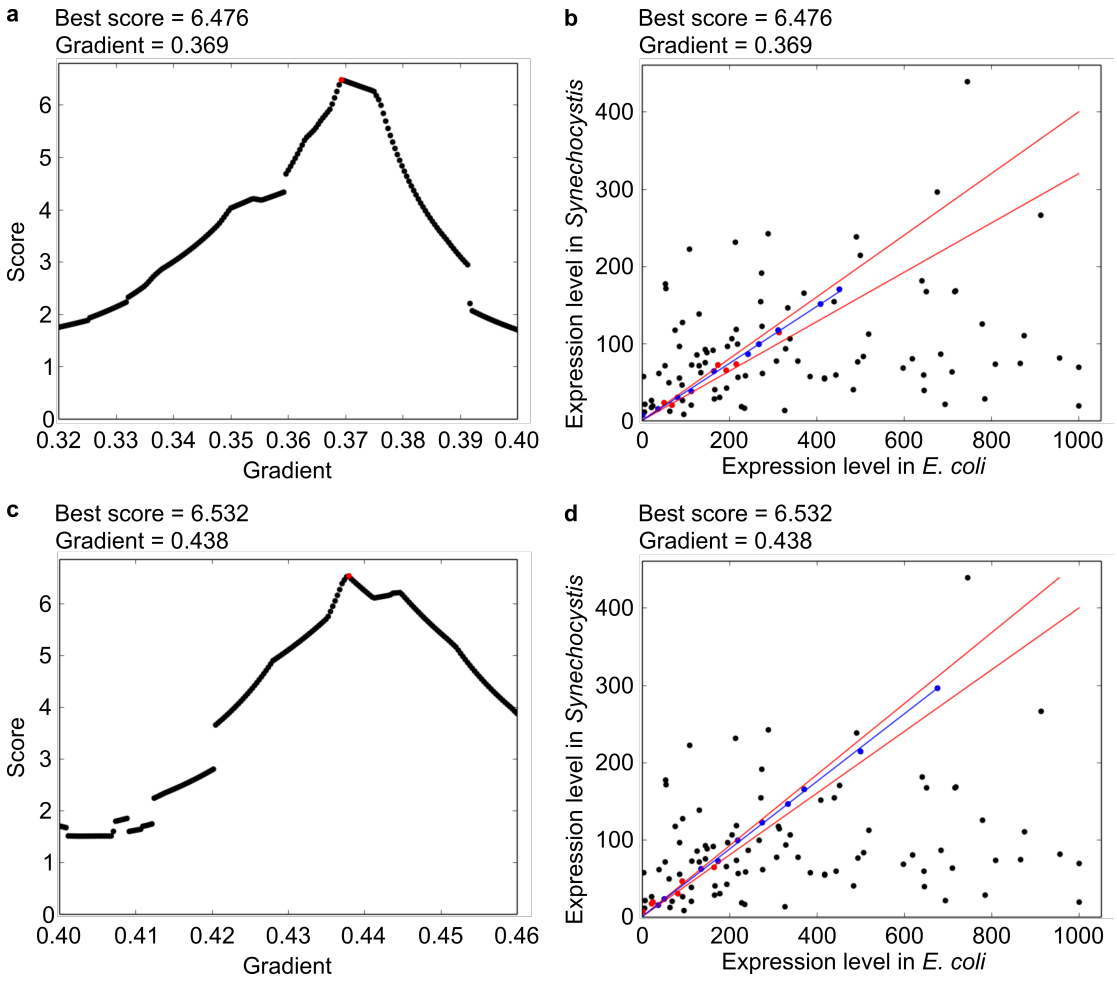


Figure 4.15. Zoomed in SUBSETTER searches around local optima in the merged promoter and RBS library. (a) A zoomed in search between the gradients 0.32 and 0.40 identified the best subset at the gradient 0.369 with a score of 6.476 (red dot). (b) The subset identified at gradient 0.369 (blue line) was composed of 12 parts (blue dots) that made the best subset from the 16 shortlisted parts (blue and red dots) between the searched space (gradients 0.32 and 0.40; red lines). (c) A zoomed in search between the gradients 0.40 and 0.46 identified the best subset at the gradient 0.438 with a score of 6.532 (red dot). (d) The subset identified at gradient 0.438 (blue line) was composed of 11 parts (blue dots) that made the best subset from the 16 shortlisted parts (blue and red dots) between the searched space (gradients 0.40 and 0.46; red lines).

With nearly identical scores, the two subsets were then more closely compared. SubsetM2 had a considerably wider range of expression strengths in both organisms (bestxRange = 639, best yRange = 281 for *E. coli* and *Synechocystis*, respectively) compared to the SubsetM1 (bestxRange = 450, best yRange = 163 for *E. coli* and *Synechocystis*, respectively), which would advantageously provide a bigger range of expression strengths for fine tuning and optimising metabolic pathways. The fit and

spacing scoring were very similar for the two subsets; bestfitPenalty = 23.3 and 21.46 and bestgapPenalty = 3.1 and 4.93, for SubsetM1 and SubsetM2, respectively. Comparing the composition of the two subsets revealed that SubsetM2 was composed of 11 parts, three of which were promoters and the other eight RBSs, while SubsetM1 was composed of 12 parts, six promoters and six RBS. Therefore, using SubsetM1 would provide 36 expression levels (the results of multiplying the number of promoters with the number of RBSs) rather than 24 that SubsetM2 would generate, which would be better suited for combinatorial assemblies and for fine tuning the protein expression levels when optimising metabolic pathway. The better balance between the numbers of promoters and RBSs also means that both transcription and translation are used for fine tuning, rather than primarily translation. From this, SubsetM1 was chosen as the portable subset, principally because it offered a greater number of expression levels (Figure 4.16). It was composed of the six promoters, SPLD2c56, SPLc69, SPLc46, SPLc28, SPLc107 and SPLc157 and the six RBSs, RBSc249, RBSc197, RBSc52, RBSc266, RBSc102 and RBSc257 (Table 4.3).

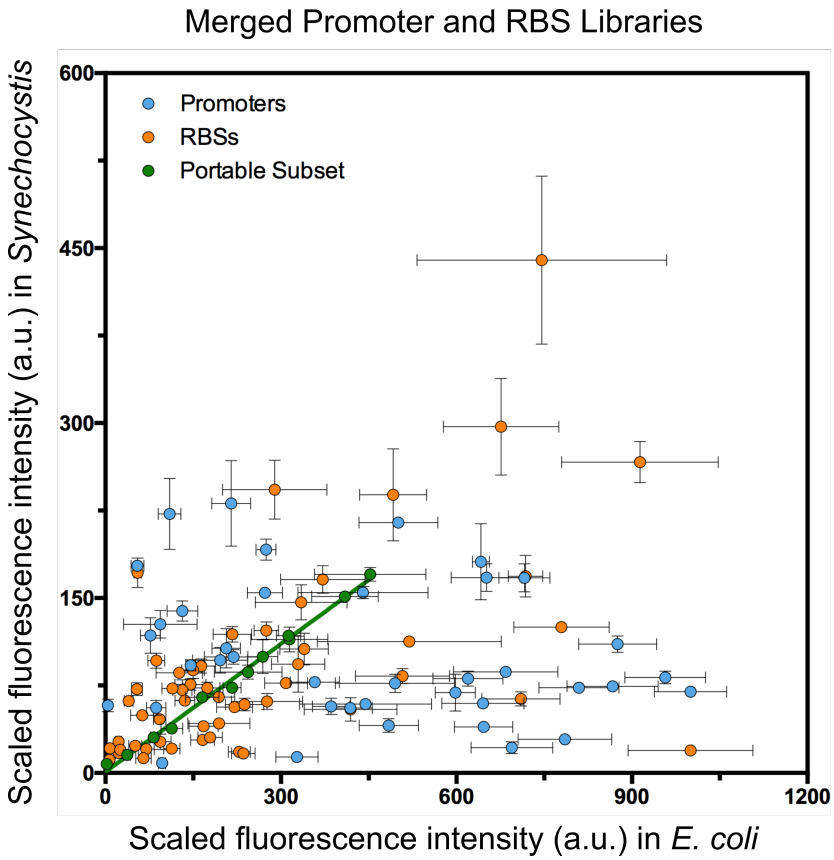


Figure 4.16. Portable subset highlighted on the merged promoter and RBS library. Figure same as Figure 4.13, but with the portable subset of 12 parts (six promoters and six RBSs) highlighted (green). The Pearson product-moment correlation for the portable subset showed a statistically significant positive correlation ($r = 0.94$, $n = 12$, $p = < 0.0001$).

Table 4.3. Promoters and RBSs that make up the portable subset. Further described in Table A1 and A3.

Portable Parts		Sequence (5' … 3')
Promoter	c56	CATTCCGTGCTTGCGTCATTATTAAACTATGAATACCTTGTGTAA
	c69	GTCGCCCCCATTGACAAATGTAAGTCGCGGAGGTATAATAGGTGC
	c46	AGCGCGGACCTTGACACTCCTAGTGACTTACACTATAATAGCTGG
	c28	CCCACATAACTTGACAGAAGTCCGCAGCGACTTTATAATACGGGG
	c107	CGCCCTACCATTGACATCCCGTTCCCGCCGCTGTATAATTGATTT
	c157	CACCGGGCCTTTGACAATACGTATCAAGTACCTTATAATAAGGGG
RBS	c249	GGCGCGAGTCAACGTTCCTGAGGAGTGGC
	c197	CTGGTATGGTTCAACGAGGATCGTT
	c52	ATTAGACGACGATCCAAAGGAGGTCCCAG
	c266	ATGATACCTGCAATTCAAGGAGGTCTTAT
	c102	TTTAGTATTGCTACTAAAGGAGGTTGCAC
	c257	CGGAGCAATCAACGAAAGGGGGAAGAGGG

4.2.5 Validation of the portable subset

The performance of the promoters and RBSs that make up the portable subset have been independently characterised in Chapter Three. However, it is important that these promoters and RBSs can be used together for combinatorial and fine-tuning approaches to optimise metabolic pathways via rapid prototyping. Such engineering involves changing the genetic context that surrounds the portable parts; it is well proven that expression control parts, including promoters and RBSs, are highly context dependent (394, 466-470) and therefore changing the genetic context will most likely cause differences in their performance. It is essential that these promoters and RBSs remain portable when they are not only used together but also when they are used to express different genes of interest. Therefore, the next step was to evaluate and validate the portability of the portable subset when used in different contexts. For this purpose, two different experiments were designed; the first combinatorially assembles the six portable promoters and the six portable RBSs to assess the effect that changing the context at the promoter-RBS junction has on portability of these parts; and second combinatorially assembles different coding sequences with the six portable RBSs to assess the effects that changing sequences at the RBS-CDS junction can have on the RBSs portability.

4.2.5.1 EYFP combinatorial build

In order to establish whether the six promoters and six RBSs of subset SubsetM1 are suitable as portable parts for the implementation of rapid prototyping and optimisation of metabolic pathways, it is essential that they can be used together and in any combination without affecting their portable performance. Furthermore, when using these six promoters and six RBSs in combinatorial assemblies, the 36 different expression levels (the product of multiplying the six promoters by the six RBSs) must be equally spaced over a wide range of strengths, which is necessary for optimising and fine-tuning expression of metabolic pathways. Therefore, the next step was to evaluate this by combinatorially assembling a construct encoding EYFP.

For this experiment, the six promoters and six RBSs from the portable subset (Table 4.3) were assembled in all possible combinations upstream of the *eyfp* coding sequence in pATM2 (Figure 4.17). Using pATM2 would provide an identical context to characterise the 36 constructs in, with any differences in the EYFP fluorescence intensity being a result of the combination of portable parts. Twelve primers were designed, six reverse primers each with a different primer tail representing one of six promoter sequences, and six forward primers each with a different primer tail that represented one of the six RBSs. These were used to amplify the six promoters and six RBSs in the 36 possible combinations upstream of the *eyfp* CDS in plasmid pATM2 using inverse PCR. The 36 PCR products were then circularised by ligation and the products were used to transform *E. coli*, from which sequence verified plasmids were obtained for each (pGT16-51; Table A8).

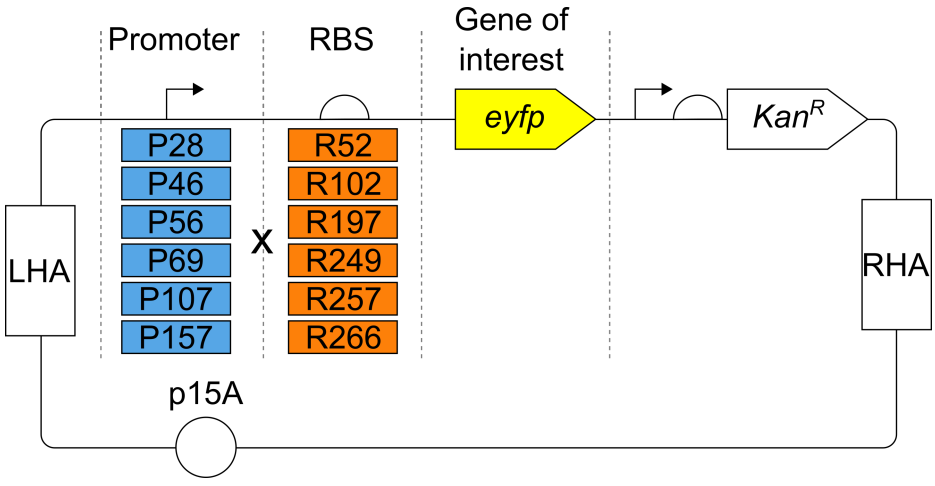


Figure 4.17. Combinatorial assembly of the EYFP reporter using the portable subset in the pATM2 shuttle vector. The portable subset of six promoters and six RBSs were assembled in all the possible 36 configurations (each promoter with each RBS) upstream of the *eyfp* CDS in the pATM2 backbone (features descried in Figure 3.1).

The 36 constructs were first characterised in *E. coli*. Each plasmid (pGT16-51) was individually transformed into *E. coli*, a single colony from each transformation was picked and grown overnight, the culture was then sub-cultured and grown to mid-exponential phase (4 hours incubation at 37 °C). After 4 hours incubation, the single-cell fluorescent distribution of each plasmid was measured. The 36 constructs gave a wide range of fluorescence intensities (Figure 4.18). To assess the portability of the EYFP combinatorial assembly, the 36 plasmids (pGT16-51) were then characterised in *Synechocystis*; wild-type cells were transformed with the 36 plasmids and kanamycin-resistant transformant colonies were passaged until complete segregation was confirmed by PCR. These transformant colonies were then cultured under constant light in BG11 media supplemented with kanamycin for 48h. Cultures were subcultured to a starting optical density (measured at 750 nm) of 0.1 and grown to exponential phase (OD750 0.5). At this point the single-cell fluorescence intensities were assayed using flow cytometry. Again, the 36 constructs gave a wide range of fluorescence intensities (Figure 4.18).

With the 36 constructs characterised in *E. coli* and *Synechocystis*, their performance was compared (Figure 4.18), which appeared to show a strong correlation (R^2 = 0.76) between the organisms. A Pearson product-moment correlation was run to determine the relationship of the 36 constructs between *E. coli* and *Synechocystis*, which showed there was a strong, positive correlation between the two organisms that was highly statistically significant (r = 0.87, n = 36, p = < 0.0001). Furthermore, the 36 expression levels generated using the six promoters and six RBSs offered a huge and well-distributed range of expression strengths between the two organisms, ranging from 0 – 18,560 a.u. and 0 - 6246 a.u. in *Synechocystis* and *E. coli*, respectively.

Disclaimer – The scales on the *x*-axis and *y*-axis on figures presenting flow cytometry data from Figure 4.18 onwards are different to those on the earlier figures as there was a change in the flow cytometer used in this project.

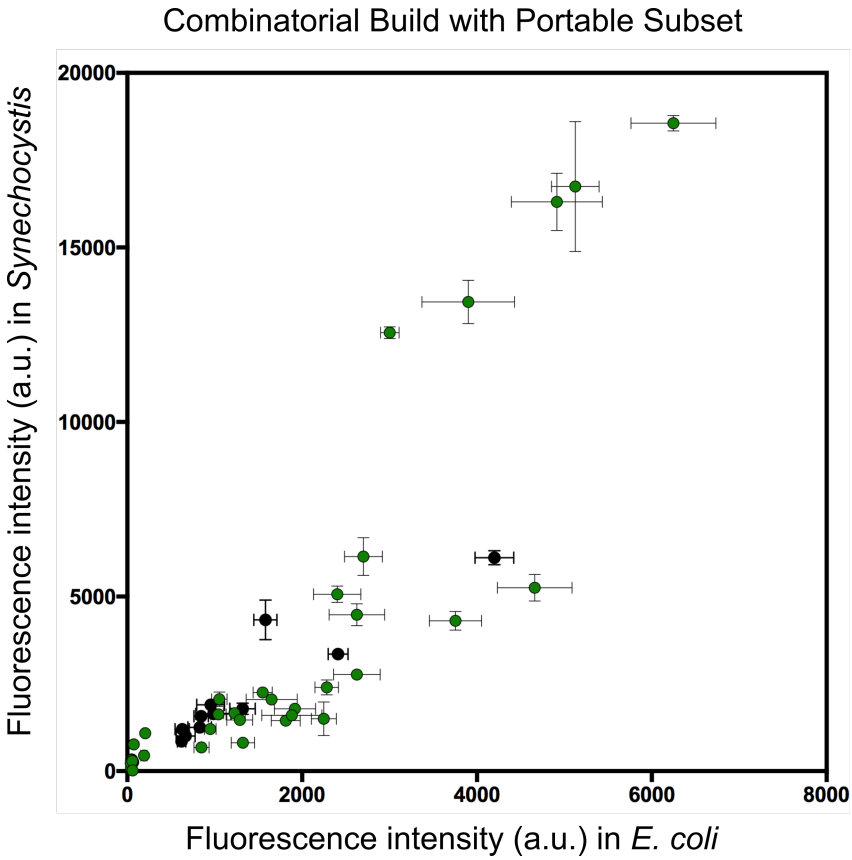


Figure 4.18. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the combinatorial assembly of EYFP. The fluorescence intensities of the 36 combinatorially assembled constructs encoding the EYFP expression in *E. coli* were plotted against the fluorescence intensities of the constructs in *Synechocystis* (green dots). *E. coli* DH10B cells were individually transformed with the 36 combinatorially assembled plasmids (pGT16-51). Transformant colonies were cultured overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the fluorescence of 10,000 cells were measured by flow cytometry. The fluorescence intensities presented represent the fluorescence of cells with wild-type background fluorescence subtracted. The error bars shown represent the standard deviation of five biological repeats. *Synechocystis* cells containing plasmids with the 36 plasmids (pGT16-51) were cultured in BG11 media in photoautotrophic conditions and constant light; and their fluorescence intensity of 10,000 cells measured in a mid-exponential phase of growth using flow cytometry. Fluorescence intensities presented represent the fluorescence of cells with background fluorescence subtracted. The error bars shown represent the standard deviation of three biological repeats. The portable subset of 12 parts was included as a control (black dots).

To further understand the distribution of the 36 ‘composite portable parts’ (composite refers to promoter-RBS combined part) the distribution was compared by highlighting each of the six promoters (Figure 4.19). This showed strong correlations of each of the six promoters between each organism (Pearson’s product-moment correlation: r = 0.70, 0.81, 0.84, 0.85, 0.95 and 0.91 for SPLc56, SPLc69, SPLc46, SPLc28, SPLc107 and SPLc157, respectively) and also showed that they were well distributed over the whole range of expression strengths. The same analysis was repeated to assess the

distribution of the six portable RBSs in the 36 'composite portable parts' (Figure 4.19). The data showed mostly strong correlations of the six RBSs between the two organisms (Pearson's product-moment correlation: r = 0.29, 0.85, 0.85, 0.98, 0.95 and 0.76 for RBSc249, RBSc197, RBSc52, RBSc266, RBSc102 and RBSc257, respectively), however it was apparent that the RBSs were not well distributed through the range of expression strengths and that they were actually clustered around similar fluorescence intensities (Figure 4.19). This suggested that the RBSs in the portable subset were more responsible for the big changes in expression strength, rather than the RBSs.

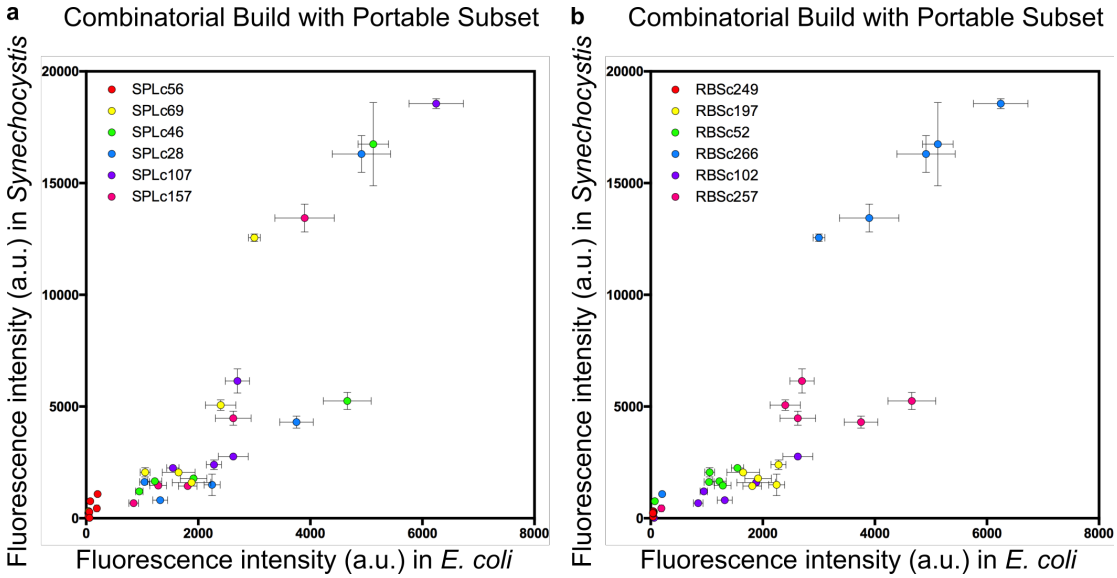


Figure 4.19. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the combinatorial assembly of EYFP. The data presented in this figure is the same as Figure 4.18, however a) presents the data with data points coloured by the promoter and b) presents the same data but with the data points coloured by the RBS.

4.2.5.2 Different reporters

The junction between a CDS and the upstream sequence, typically a 5'UTR including an RBS, is one of the most highly context-dependent areas of a gene. It is well documented that changing the coding sequence can affect an RBSs strength (394); typically caused by the mRNA sequence of the transcribed gene affecting ribosome binding to the mRNA and therefore affecting initiation of translation. Therefore, the strength of the six portable RBSs is expected to change when used to express different genes. However, considering that for a given RBS and CDS, the DNA sequence will be the same in *E. coli* and *Synechocystis*, the mRNA folding should be very similar between the two organisms. It was therefore hypothesised that the change in RBS strength should remain proportional between *E. coli* and *Synechocystis* (Figure 4.2).

In order to use the six RBSs as portable parts for rapid prototyping and optimising metabolic pathways, it was preferable that if changing genes affected the six RBSs strength, it should be proportional between the two organisms. To assess how the interaction at the RBS-CDS junction affected the strength and portability of the six RBSs, they were characterised upstream of five different reporters, GusA, YetiYFP, EYFP and two recoded EYFPs (Figure 4.20).

For this experiment, the mid-strength portable promoter SPLc46 and the set of six portable RBSs (Table 4.3; Figure 4.20) were used. The two recoded *eyfp* coding sequences were designed to provide different mRNA sequences and therefore different folding patterns and secondary structures, but the sequence would still encode the same amino acid sequence for the EYFP reporter. This was done to create different genetic contexts in order to assess the six RBSs from the portable subset, whilst using the same EYFP reporter, which therefore allowed the same assay to be used. This was achieved by recoding the CDS with synonymous codons so that the DNA sequence and therefore the mRNA sequence changed, but the encoded amino acids and therefore protein did not. In the first of the two alternative *eyfp* coding sequences (*eyfp* version 2), the first 35 bp of the CDS was recoded, which was previously identified as the area where secondary structures had the biggest impact on the translation initiation rate (85). For the second alternative *eyfp* coding sequence (*eyfp* version 3), the entire coding sequence was recoded (including the same first 35 bp as *eyfp* version 2). Furthermore, these two recoded *efyp* coding sequences would allow the differences of short-range and long-range mRNA interaction to be assessed. In order to minimise the effect that different translation elongation rates had on the protein expression level of the recoded *eyfp* CDSs between *E. coli* and *Synechocystis*, they needed to be designed with a similar translation elongation rate so that any differences in protein expression levels between the two organisms were a direct result of the RBSs activity. To recode the *eyfp* coding sequence, OPTIMIZER's guided random methodology (131) was used, where iterations of the optimisation were repeated until the CDS had a codon adaptation index (CAI) score similar to that of the original *eyfp* CAI in both organisms. CAI is the most prevalently used and well understood measure of synonymous codon usage bias. For a given gene the CAI is calculated by measuring the similarity between the codon usage of a given gene and the codon usage of a reference group of genes (organism specific), which is typically a group of highly-expressed genes in a given organism (479, 480). The CAI score (between 0 – 1) of a given gene is therefore a measure of how similar its codon usage is to the codon usage bias in the reference group of genes, with higher scores typically

resulting in better expression levels of a gene. The original *eyfp* coding sequence had a CAI score of 0.574 and 0.644 for *E. coli* and *Synechocystis*, respectively, *eyfp* version 2 had a CAI score of 0.685 and 0.650 for *E. coli* and *Synechocystis*, respectively and *eyfp* version 3 had a CAI score of 0.741 and 0.665 for *E. coli* and *Synechocystis*, respectively. Multiple iterations of the recoding were required to get CAI scores this similar. Along with the three variants of the *eyfp* coding sequence, two alternative reporters were chosen. These included *yetiYFP* (DNA 2.0), a fluorescent reporter with a similar phenotype to *eyfp*, which allowed the same assay to be used, and also *gusA* (408), a colourimetric reporter. These reporters would not only differ from the original *eyfp* by mRNA sequence, but also by the amino acid sequence (Figure 4.20). The alternative CDS were introduced to pATM2, and then inverse PCR was used to introduce SPLc46 and the six RBSs (See Plasmid Construct in Chapter Two for more details). Ligation mixtures were used to transform *E. coli* and transformant colonies were then sequenced to obtain correct clones for each plasmid.

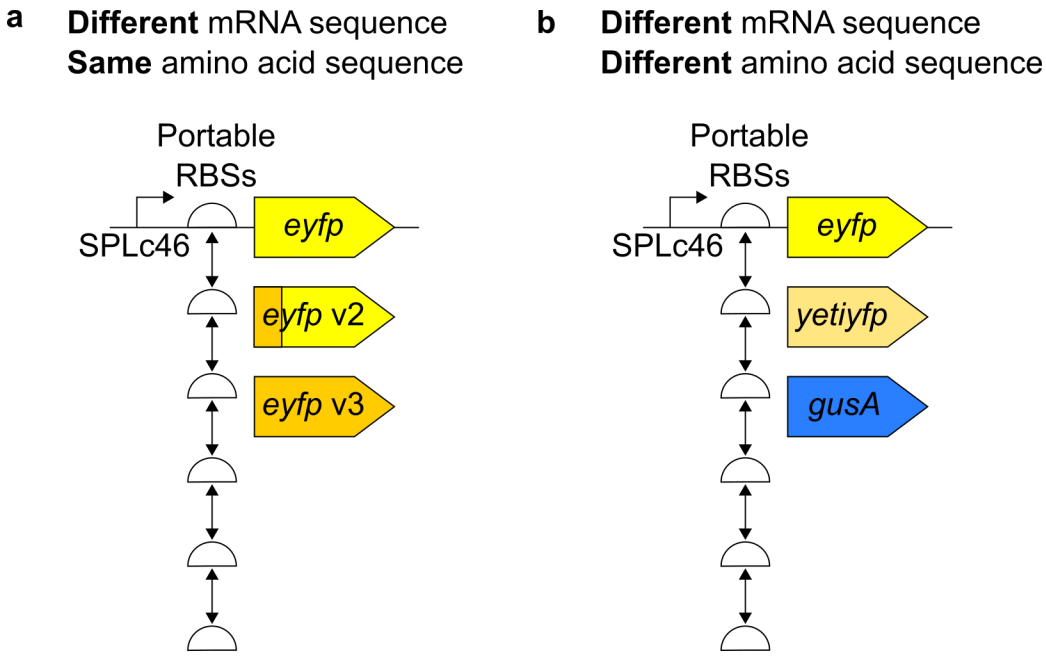


Figure 4.20. Reporters used for assessing the effects that context has on the portability of an RBS. The aim of this experiment is to assess the effect that different coding sequences have on the portability of the six RBSs. The six RBSs from the portable subset, RBSc249, RBSc197, RBSc52, RBSc266, RBSc102 and RBSc257 will be characterised in pATM2 using the five reporters *eyfp*, *eyfp* version 2, *eyfp* version 3, *yetiYFP* and *gusA*, along with the mid-strength portable promoter SPLc46. The three *eyfp* coding sequence variants will provide a different mRNA sequence but the same amino acid sequence to test the RBSs, while *yetiYFP* and *gusA* provide different mRNA and amino acids sequences. The *eyfp* coding sequence was recoded using the web tool OPTIMIZERS guided random methodology, *eyfp* version 2 had been recoded for the first 35 bp and *eyfp* version 3 recoded for the entire coding sequence (including the same first 35 bp as *eyfp* version 2). These two variants will allow us to assess the interactions that short- and long-range mRNA folding have on portability.

The 18 constructs with fluorescent reporters (*eyfp* version 2 and 3 and *yetiyfp*) were first characterised in *E. coli*. Each plasmid (pGT11, 12, 52 – 67) was individually transformed and characterised using flow cytometry as described in section **3.2.2.3**. The data showed a wide range of fluorescence intensities in each reporter, expressed with the six RBSs (Figure 4.21). To compare the six RBSs portable performance, the same 18 constructs (pGT11, 12, 52 – 67) were then characterised in *Synechocystis*. Wild-type cells were transformed, segregated and characterised using flow cytometry as previously described in section **3.2.2.4**. Like in *E. coli*, a wide range of fluorescence intensities for each reporter was expressed with the six RBSs (Figure 4.21). To assess the performance of the six RBSs, the fluorescence intensities of the 18 constructs was compared, along with the six constructs that use the same promoter (SPLc46) and the six RBSs to drive expression of the original *eyfp* CDS (pGT18, 24, 30, 36, 42, 28), between the two organisms (Figure 4.21). The data appeared to show a positive correlation of the fluorescence intensities from the four reporters between *E. coli* and *Synechocystis*. A Pearson product-moment correlation was then performed, which showed a strong, positive correlation between the two organisms that was statistically significant for *eyfp* ($r = 0.84$, $n = 6$, $p = 0.038$) and *eyfp* version 3 ($r = 0.86$, $n = 6$, $p = 0.027$). A strong positive correlation was seen for *eyfp* version 2, but it was not statistically significant ($r = 0.74$, $n = 6$, $p = 0.09$) and *yetiyfp* showed a weak positive correlation ($r = 0.23$, $n = 5$, $p = 0.71$). Interestingly, the *eyfp* version 2 and 3 encoded considerably higher levels of fluorescence (0 – ~80,000 a.u.), compared to the original *eyfp* sequence (0 - ~20,000 a.u.).

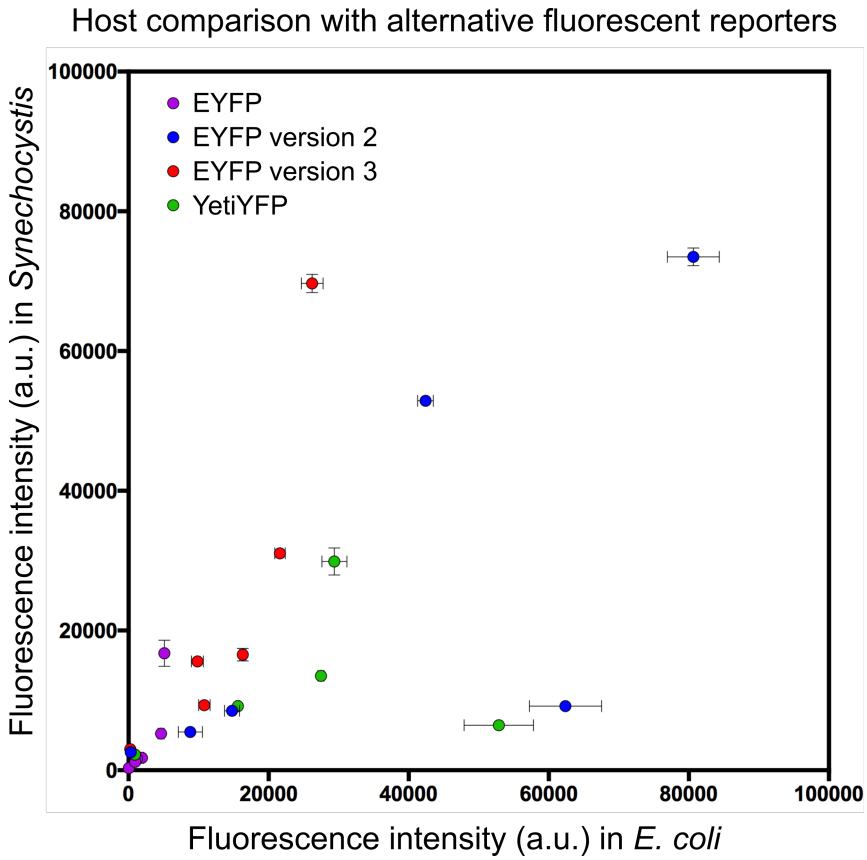


Figure 4.21. Host comparison between *E. coli* DH10B and *Synechocystis* sp. PCC 6803 of the fluorescence intensities from the four different fluorescent reporters. The fluorescence intensities of the four reporters (EYFP, EYFP version 2, EYFP version 3 and YetiYFP) was quantified in *E. coli* and *Synechocystis* using the six portable RBSs (RBSc249, RBSc197, RBSc52, RBSc266, RBSc102 and RBSc257) in pATM2 with the mid-strength portable promoter SPLc46. *E. coli* DH10B cells were individually transformed with the 24 plasmids (pGT11, 12, 18, 24, 30, 36, 42, 48, 52 - 67). Transformant colonies were cultured overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the fluorescence of 10,000 cells were measured by flow cytometry. The fluorescence intensities presented represent the fluorescence of cells with wild-type background fluorescence subtracted. The error bars shown represent the standard deviation of five biological repeats. *Synechocystis* cells containing plasmids with the 36 plasmids (pGT11, 12, 18, 24, 30, 36, 42, 48, 52 - 67) were cultured in BG11 media in photoautotrophic conditions and constant light; and their fluorescence intensity of 10,000 cells measured in a mid-exponential phase of growth using flow cytometry. Fluorescence intensities presented represent the fluorescence of cells with background fluorescence subtracted. The error bars shown represent the standard deviation of three biological repeats.

To characterise the glucuronidase activity, *E. coli* cells were transformed with the six *gusA* encoding constructs (pGT13, 68 – 72) and the glucuronidase activity was quantified in mid-exponential phase (4 hours incubation) using a colourimetric assay. To quantify glucuronidase activity, toluene-treated cells of *E. coli* expressing GusA hydrolysed the reaction substrate pnitrophenyl-β-D-glucuronide, forming a yellow product, which was measured using a spectrophotometer (Figure 4.22). To quantify glucuronidase activity in *Synechocystis*; wild-type cells were grown to mid-exponential phase and transformed with the *gusA* encoding plasmids (pGT13, 68 – 72). Segregated transformants were grown to mid-exponential phase and were subjected to bead beating. The released GusA then hydrolysed the reaction substrate pnitrophenyl-β-D-glucuronide, forming a yellow product, which was measured using a spectrophotometer (Figure 4.22). The glucuronidase activity from the two organisms was compared (Figure 4.22) and appeared to show a positive association between the *E. coli* and *Synechocystis* values, where *Synechocystis* values increased in line with *E. coli*. A Pearson product-moment correlation was run to determine the relationship between the two organisms, which showed that there was a strong, positive correlation between them that was statistically significant ($r = 0.81$, $n = 6$, $p = 0.05$).

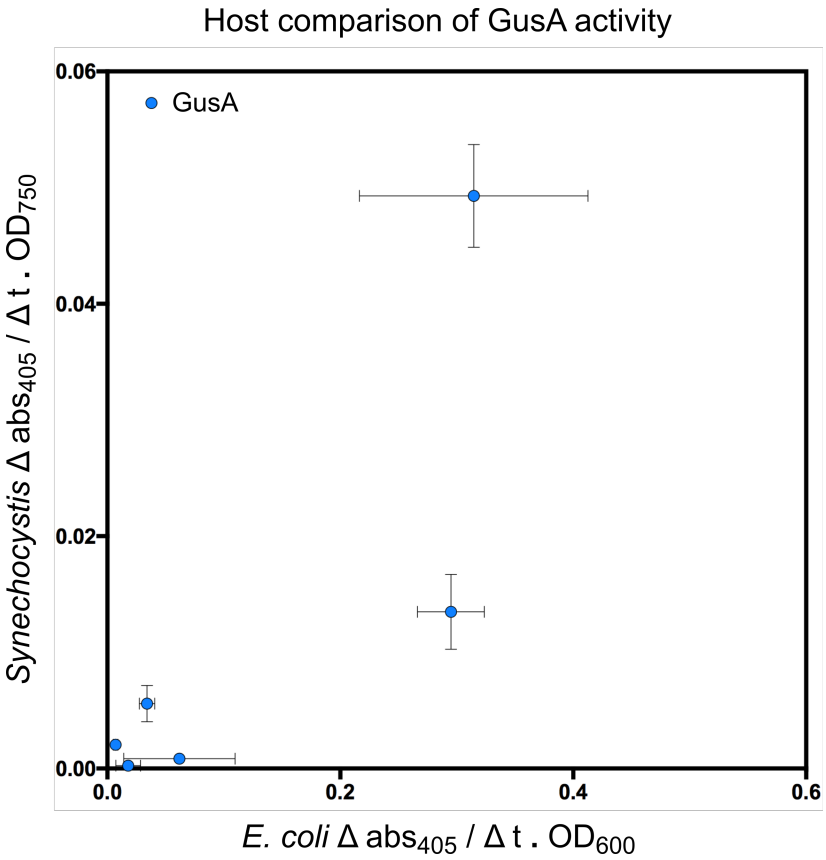


Figure 4.22. Host comparison of the glucuronidase activity between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. GusA activity was quantified in *E. coli* and *Synechocystis* using the six portable RBSs (RBSc249, RBSc197, RBSc52, RBSc266, RBSc102 and RBSc257) in pATM2 with the mid-strength portable promoter SPLc46. *E. coli* DH10B cells were transformed with the six plasmids (pGT13, 68 - 72). Transformant colonies were cultured overnight at 37 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 4 hours. After which, the cells were harvested and tested in the glucuronidase assay. Absorbance at 405 nm was normalised to OD600 nm and reaction time. Error bars represent standard deviations of five independent experiments. *Synechocystis* cells containing the fully segregated plasmids (pGT13, 68 - 72) were cultured in BG11 media in photoautotrophic conditions and constant light; during a mid-exponential phase of growth cells were harvested and tested in the glucuronidase assay. Absorbance at 405 nm was normalised to OD750 nm and reaction time. Error bars represent standard deviations of three independent experiments.

4.2.5.3 RBS rank orders and curating portable subset

Having found that the six RBSs showed, in general, a strong positive association for the protein expression levels between *E. coli* and *Synechocystis*, the change in strength of the six RBSs with the five different CDSs was assessed. The data showed that the RBSs strengths changed for each of the five CDSs (Figure 4.23), as had previously been predicted and whilst there was no particular pattern or order to the change in RBS strengths, there were a few general trends, such as RBSc249 being weak and RBSc266 being strong. When comparing the strengths of the RBSs it was

apparent that certain RBSs fell out of rank order; which meant that the RBSs did not increase in strength in *Synechocystis* when they did in *E. coli*. The SUBSETTER algorithm did not consider the effects that context dependence would have on the part's portable performance, hence why these validation experiments were carried out. With the data collected from these validation experiments it would be possible to further curate the portable subset, considering the effects that context dependence had on the parts in the subset. To curate the subset, a selection criterion was defined, which stated that if a promoter and/or RBS had fallen out of rank order on two or more occasions in the validation experiments it would be discarded from the subset.

To implement this curation, the RBSs that fell out of rank order when upstream of different CDSs were first identified. RBSc257 fell out of rank order for *eyfp* version 2, *yetiyfp* and *gusA*; RBSc102 fell out of rank order for *eyfp* version 3; RBSc52 fell out of rank order for *yetiyfp*; and RBSc197 fell out of rank order for *gusA*. The effects that part interactions had on the rank order of promoters and RBSs from the combinatorial assembly of *eyfp* were then considered. For the promoters, RBSc52 fell out of rank order for SPLc28, SPLc56, SPLc69 and SPLc157; and RBSc102 and RBSc197 fell out of rank order for SPLc56 and SPLc69, respectively. When considering the RBSs; SPLc69 fell out of rank order for RBSc52, RBSc197, RBSc249 and RBSc257; SPLc28 fell out of rank order for RBSc102, RBSc197 and RBSc249; SPLc157 fell out of rank order for RBSc52 and RBSc257; and SPLc107 fell out of rank order for RBSc257.

With the offending parts identified, the two promoters, SPLc28 and SPLc69, and two RBSs, RBSc52 and RBSc257, were discarded to generate the curated portable subset (Figure 4.24). Not surprisingly, with the curated portable subset, a stronger correlation of part strength between the two organisms was observed. A Pearson's product-moment correlation showed a strong, positive correlation between the two organisms, which was statistically significant for the *eyfp* combinatorial assembly ($r = 0.94$, $n = 16$, $p = < 0.0001$), *eyfp* version 2 ($r = 0.97$, $n = 4$, $p = 0.27$) and had very strong correlations for *eyfp* version 3 ($r = 0.90$, $n = 4$, $p = 0.1$), *yetiyfp* ($r=0.84$, $n = 4$, $p =0.15$) and *gusA* ($r = 0.77$, $n = 4$, $p = 0.23$).

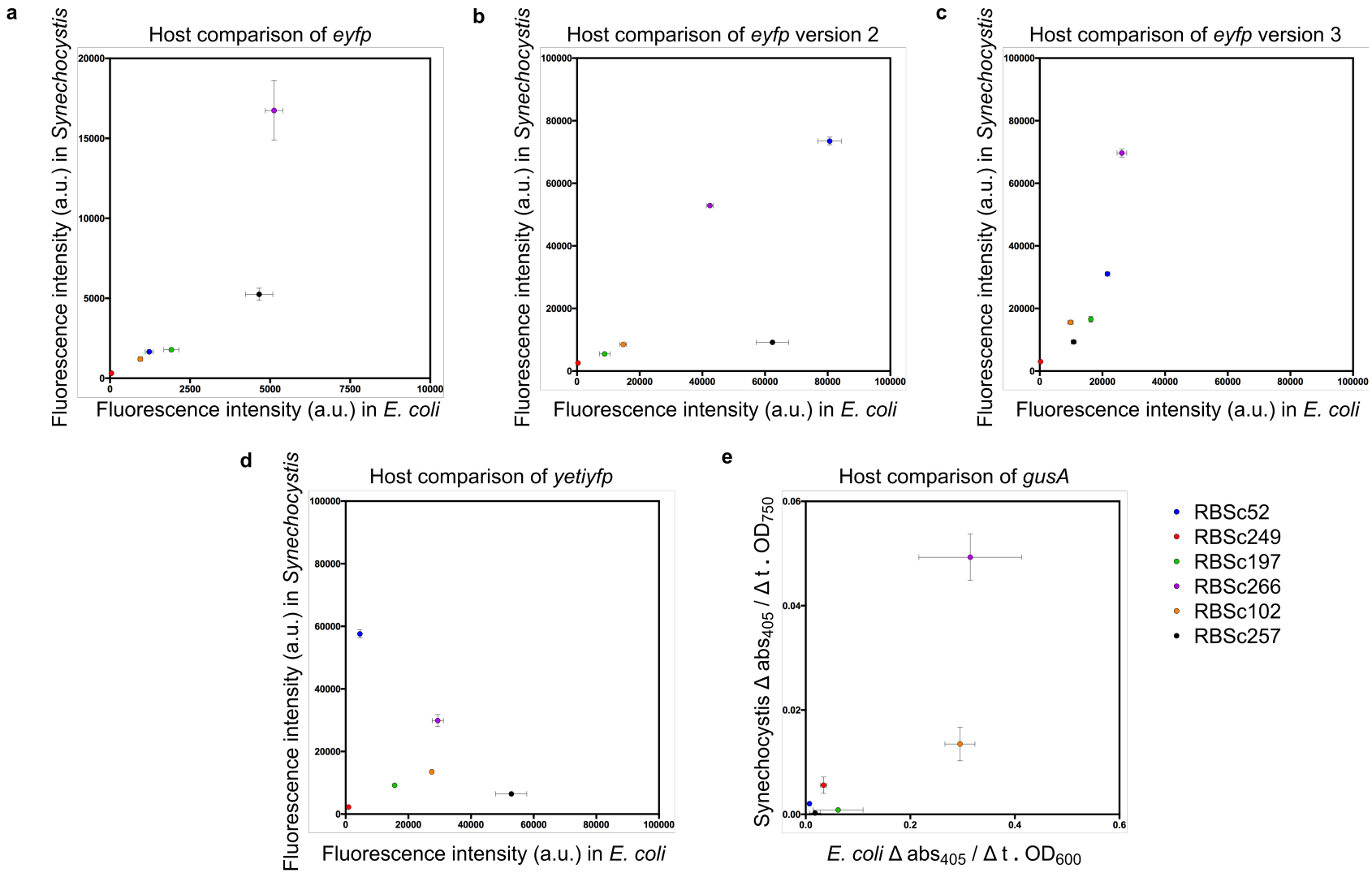


Figure 4.23. Host comparison of portable RBSs with different reporters. This figure presents the same data shown in Figure 4.21 and Figure 4.22, however in this Figure the six different portable RBSs are each coloured (RBSc52 = blue dot, RBSc249 = red dot, RBSc197 = green dot, RBSc266 = purple dot, RBSc102 = orange dot and RBSc257 = black dot). This allows the change in RBSs strength to be interrupted rather than used the portable performance of the overall six RBSs. The fluorescence intensities in graphs a – d were collected as described in Figure 4.21 and the glucuronidase activity in graph e was collected as described in Figure 4.22.

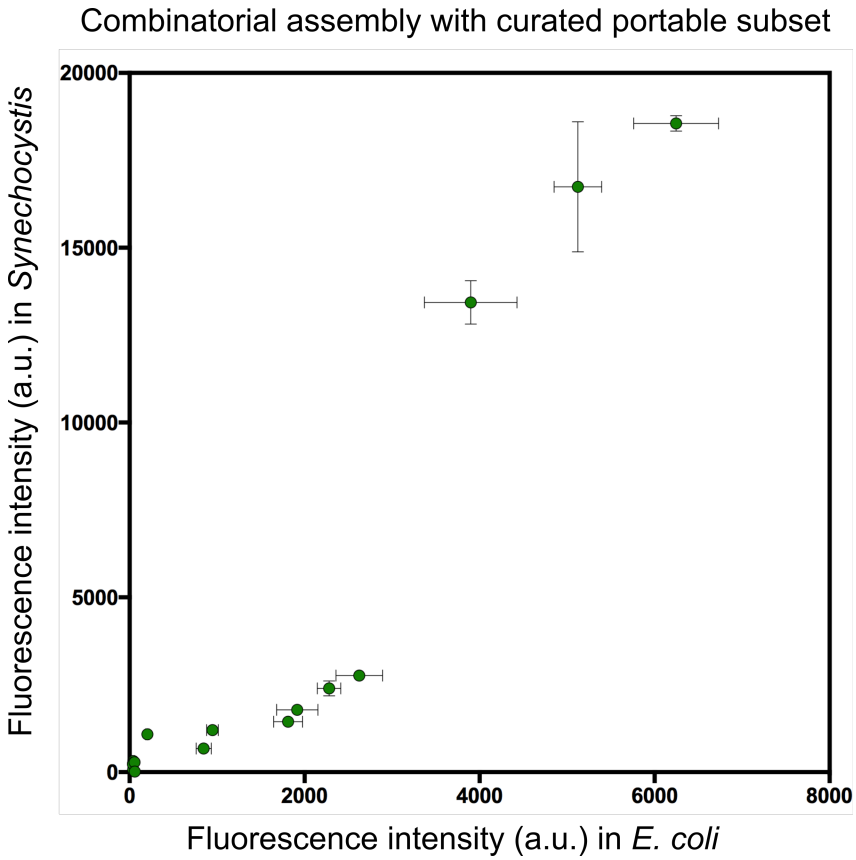


Figure 4.24. Host comparison of EYFP fluorescence intensities using the curated portable subset between *E. coli* DH10B and *Synechocystis* sp. PCC 6803. This figure is the same as Figure 4.18, but the data point that represent promoters and RBSs which were discarded during curation of the portable subset have been removed. Therefore, it now only presents the data points from the curated subset.

4.3 Discussion

The overall goal of this chapter was to quantitatively identify a subset of promoters and RBSs with the best portable properties between *E. coli* and *Synechocystis*, and then assess how the performance of the individual parts and the overall subset were affected when used in different contexts. To achieve this, SUBSETTER, a heuristic search algorithm that quantitatively ranks subsets based on their portable properties was first developed. Using SUBSETTER, the subset with the most optimal portable properties from the promoter and RBS libraries was identified. With the portable subset chosen, the portable performance of its parts was validated, which showed that while the strength of the portable promoters and RBSs changed when used in different contexts, their performance remained largely portable between the two organisms.

Developing the heuristic algorithm SUBSETTER was an ideal approach to identifying subset combinations with good portable properties and then quantitatively ranking them against one another. Using a heuristic search algorithm rather than an exhaustive

algorithm (to identify the best subset) is generally considered a trade-off between optimality and completeness for speed and feasibility and therefore sometimes does not guarantee finding the best solution. In this case, it was a necessary trade-off because an exhaustive search was not computationally possible for a realistically sized dataset. Furthermore, the dot plot output (best subset per gradient searched) that SUBSETTER generates allows users to zoom in around all the areas of local optima using much higher-resolution search parameters, therefore truly identifying the best possible subset and overcoming the trade-offs mentioned above. Not only does this tool identify portable subsets between *E. coli* and *Synechocystis*, it could be used for any pair of organisms, as the properties that make an ideal portable subset will most likely not change for different pairs of organisms, which therefore makes SUBSETTER a useful tool for future work in developing and identifying portable parts for different organisms.

It is not possible to directly compare an exhaustive search algorithm with the SUBSETTER's heuristic search. However, SUBSETTER's heuristic search function determined the score 11,226,800 times when identifying the best portable subset in the merged promoter and RBS library. If an exhaustive search had been used, 1,192,052,400 combinations would have had to be scored, 100 times more than the heuristic search scored. Furthermore, SUBSETTER implicitly determines a series of linear models (represented by their gradients) before scoring subsets. An exhaustive search function would presumably need to determine a linear model for every subset, against which to score that subset. This could be done by fitting a linear regression, which would be computationally more expensive. On the other hand, SUBSETTER will score many subsets more than once. Taking this into account, it would likely be difficult to predict and therefore compare the relative speeds of SUBSETTER's heuristic approach with an exhaustive approach. However, unlike an exhaustive search, SUBSETTER's heuristic search is completely independent to the size of a dataset (Figure 4.25), whereby the number of scores calculated do not change when the size of the dataset varies. As such, as long as SUBSETTER gives reasonably optimal solutions, which it appears to, SUBSETTER will be increasingly preferable for analysing increasingly large problems in comparison to using an exhaustive search, which scales exponentially (Figure 4.25). Whilst the exponential nature of an exhaustive search might not have impact on the computational run time when the dataset size is small, there are increasingly more instances where larger dataset sizes exist, for example the library of >10,000 insulator variants (464) and the library of >500

transcriptional terminations (127) and in these cases, it's crucial that the dataset size would be independent to the number of scores calculated.

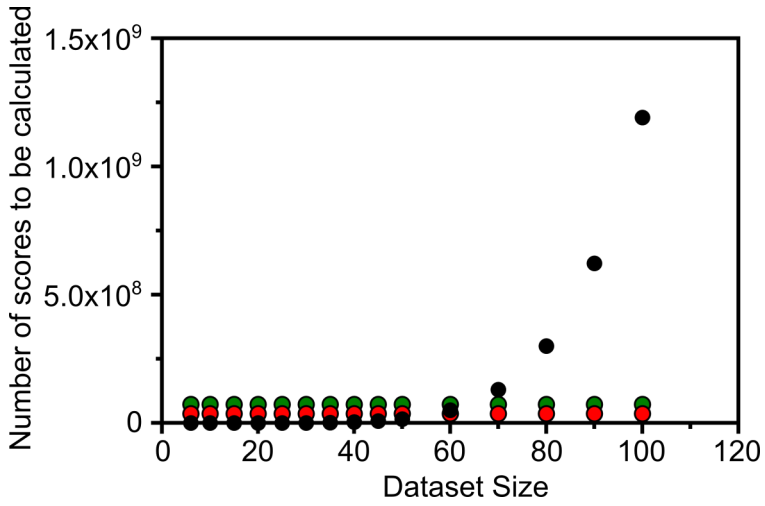


Figure 4.25. Relationship between dataset size and the number of scores to be calculated. The black dots show the exponential increase in the number of scores that need to be calculated as the size of the dataset increases (Calculated using Equation 4.1). The green and red dots show that the number of scores that need to be calculated when using SUBSETTER's heuristic search is independent of the dataset size (Calculated using Equation 4.3). Red and green dots used a shortlist of 10 points, red dots searched 100 gradients, green dots searched 200 gradients.

$$Number\ of\ scores\ to\ be\ calculated = S! * G$$

Equation 4.3. Equation to calculate the number of scores that need to be calculated using SUBSETTER's heuristic search. *S* = length of shortlist. *G* = number of gradients being searched. ! = factorial. In this equation, a minimum subset length of zero was assumed and the maximum subset length was assumed to be the same as the length of the shortlist (*S*).

SUBSETTER could also be used to identify subsets with different properties, for example, by narrowing the gradients around certain areas it would be possible to identify a subset of parts that was inactive in one organism and functional in another. Such a subset could be useful for cloning certain genes and/or pathways in one organism, such as the tractable, fast-growing *E. coli*, where low expression levels reduces cellular stress and metabolic burden, which could be advantageous for minimising bias in a combinatorial assembly, or for just assembling a construct that otherwise would be too toxic to clone, before transferring to a non-model organism for expression. Such a subset has been suggested to be beneficial for engineering non-model organisms such as *Geobacillus* (235).

Having identified a portable subset, its portable performance was validated by characterising the performance of the six promoters and six RBSs when assembled and expressed in different contexts. It was hypothesised that the effects of context dependence would cause a change in part strength, but as long as this change was proportional between the two organisms the parts would still be deemed portable. To establish this, the interactions between the portable promoters and portable RBSs, and the interactions between the portable RBSs and different coding sequences were assessed. The interactions between the promoters and the upstream DNA sequences were not assessed because while it has been reported that regions of more than 100 base pairs upstream of the promoter can affect the promoters strength (481), in this case, the sequence upstream of the promoters would remain conserved in the shuttle vector and therefore any effect on the promoters properties should already have been accounted for in the validation experiments. The first experiment revealed a strong correlation of the 36 constructs between the two organisms that stretched over a wide range of strengths, suggesting that the portable subset would provide not only the predictive performance that rapid prototyping requires but also the wide range of strengths needed for combinatorially optimising metabolic pathways. As expected, in the second experiment the six RBSs changed strength for different coding sequences, however in general the six RBSs still correlated well between *E. coli* and *Synechocystis* for the five coding sequences tested. With these validation experiments, it was established that the portable subset can provide a wide range of different and portable strengths between the two organisms and that the parts largely correlated well when used in different contexts.

The validation experiments highlighted a few parts that did not correlate as well as others when used in different contexts. While SUBSETTER had chosen the most optimal subset from the merged dataset, it did not consider the interactions that those parts would have with others. Using the data from the validation experiments, it was possible to curate the portable subset by considering the effects that different contexts had on the performance of the parts in the portable subset. Two promoters and two RBSs from the original portable subset were discarded, which resulted in a curated subset of four promoters and four RBSs.

During this chapter, a subset with the most optimal portable properties was identified, validated and then further curated after considering the effects of context dependences on portability. The end result was a set of portable parts that could be used to combinatorially assemble metabolic pathways and optimise them through rapid

prototyping. However, before testing the portable parts in the first implementation of rapid prototyping, a DNA assembly method that would be suitable for multi-part combinatorial assembly needed to be identified (Chapter Five).

4.4 Key Outcomes

- Development of a search algorithm that identifies the best portable subset in a dataset.
- It was established that, in general, when parts in the subset were used in different contexts, they changed strengths but remained portable.
- Having identified the effects that part interactions have on portability, it was possible to curate the portable subset, discarding parts that performed worse in different contexts.

5. CHAPTER FIVE. Combinatorially building multi gene pathways using the constraints of the genetic code – Start-Stop Assembly.

5.1 Introduction

In the previous chapter (Chapter Four), two subsets of portable parts were identified, both of which demonstrated a similar performance between *E. coli* and *Synechocystis* when characterised with different fluorescent and colourmetric reporters. To allow these subsets of portable parts to be used in combinatorial and rapid prototyping-based approaches to optimise a pathways performance, a suitable multi-part DNA assembly method was required.

New types of DNA cloning methods, which are typically referred to as DNA assembly, have been reported in recent years that are more suitable for combining multiple DNA sequence fragments in a single step, compared to conventional restriction-ligation cloning (160). For the most part, multi-part DNA assembly methods can be categorised as either: i) modular approaches, where a framework of steps, rules and design constraints, including pre-defined module formats, enables efficient multi-part assembly of individual constructs or of combinatorial mixtures of constructs. These advantages are however associated with fixed 'scar' sequences at the junctions between parts in the final assembled construct. Furthermore, a considerable amount of effort and resource is required in advance in order to obtain a suitable set of assembly vectors, as well as to prepare sequences of interest in the appropriate format. ii) Bespoke approaches, which are flexible, require minimal upfront effort and in general involve an *ad hoc* approach to designing the cloning strategies. They typically rely on sequence overlaps to direct assembly of DNA fragments and therefore have a greater need for custom oligonucleotide primers (generally one pair per junction between parts) and sequence verification (due to PCR steps). Furthermore, they are less suitable for combinatorial assembly due to their lower efficiencies (as fewer clones are obtained following transformation and therefore smaller libraries are generated) and there is a greater potential for bias, which is largely resultant of the repetitive sequences. Modular multi-part DNA assembly methods include Golden Gate assembly (163, 164) (and variants (137, 165-174)), BASIC (175), BioBrick assembly (177) (and variants such as BglBrick (178, 179)) and Gateway cloning (180-182). Bespoke multi-part DNA assembly methods include Gibson Assembly (183), AQUA cloning (184), Twin Primer Assembly (185), ligase cycling reaction (186), SLIC (187), SLiCE (188), overlap extension PCR (189), USER (190), CLIVA (191) and CPEC (192). Multi-part

DNA assembly is now frequently used in synthetic biology, which has enabled an increase in the scale, scope and speed of studies (160).

The development of effective metabolic pathway-encoding constructs is a key application for multi-part DNA assembly. The protein concentration produced by an expression construct is determined by the expression control parts used (including promoters, RBSs and transcriptional terminators) and also by the CDS, which affects expression levels through mRNA structure and codon usage. The design of a multi-protein expression construct, such as those encoding metabolic pathways, involves a multi-dimensional 'design space', where the concentration of each protein represents a dimension (55, 115, 142, 149). There will be regions of functional pathway-encoding designs, including local optima, in any such design space, but there will also be regions of poorly-functional or non-functional pathway-encoding designs. Natural metabolic pathways and the sequences that encode them are optimised by natural selection for the efficient formation of the intended product and also to avoid deleterious effects on the cell (112). As described in section **1.3.3.2**, heterologous or synthetic metabolic pathways can be non-optimal in varying ways, including poor flux through the pathway to the intended product, through the formation of by-products, the accumulation of intermediates (which may be toxic), excessive enzyme overexpression (113, 114) and also through deleterious effects on the cell that can result due to one or more of these issues (113, 115-118). These issues often rely on the profile of absolute and relative concentrations of the enzymes in a pathway. However, even if an optimal enzyme concentration profile is known, it is currently extremely difficult to rationally design an individual DNA sequence to express a protein of interest at a specific target concentration (142). This is a problem that increases in difficulty as the numbers of proteins in a multi-protein expression construct increase (142, 149).

Modular multi-part DNA assembly enables implementation of approaches to identify and optimise metabolic pathway-encoding constructs from the many possible designs for any given pathway. Firstly, efficient assembly is not limited to the construction of individual expression constructs, but it can be used to generate combinatorial libraries of multi-protein expression constructs, in which the expression of each protein is varied independently, or partially-independently, by using mixtures of expression control parts instead of individual expression control parts (115, 149). The design of such mixtures, in terms of the number of parts and the distribution of their expression levels, will dictate the distribution and granularity of the sampling of the design space with respect to the relevant protein. This enables effective tunable search approaches to be used

for large design spaces. These approaches are however limited by the throughput of screening after host cells have been transformed with the library. Secondly, once functional metabolic pathway-encoding construct designs are identified, modular DNA assembly allows rational optimisation through further and more fine-grained rounds of combinatorial searches, and/or by manually-designed substitutions of specific expression control parts (149). Relative to predictive modelling approaches, these approaches are practical and empirical and furthermore, they require little *a priori* knowledge of suitable enzyme concentrations, nor of the protein concentrations that will result from particular combinations of expression control parts and CDSs. The main requirements for these approaches are suitable expression control parts that span an appropriate range of strengths and also a suitable modular multi-part DNA assembly method.

Golden Gate assembly (163, 164) and its variants (137, 165-174) are the most widely used modular DNA assembly methods. They have frequently been used for assembling metabolic pathway-encoding constructs (137, 150, 171, 482, 483). Golden Gate uses type IIS restriction endonucleases, which are similar to classical type II restriction endonucleases, except that their restriction site is offset from their recognition site, rather than within it. IIS restriction endonuclease sites are asymmetrical, which confers directionality. Type IIS restriction endonucleases, in combination with appropriately-designed DNA sequences, can be used to generate DNA fragments with any desired cohesive end sequences (of the 4n possibilities, where 4 reflects the four bases A, T, C or G; and n is the length of the overhang) independently of the sequence of the recognition site. The cohesive end sequences can be non-palindromic, avoiding the self-ligation of palindromic cohesive ends like those generated by classical type II endonucleases. Golden Gate assembly uses a series of DNA fragments generated by a single type IIS endonuclease in each assembly step, so that each fragment has unique non-palindromic cohesive ends designed to anneal only to the correct end of the next DNA fragment in the series. Such a series of DNA fragments can be assembled efficiently and with directionality, in the intended order, by ligation. Furthermore, an efficient ‘one-pot’ assembly reaction containing both ligase and the appropriate type IIS endonuclease can be used (163, 164), because unique non-palindromic cohesive ends can only either i) ligate to the intended cognate cohesive DNA end, in which case the restriction site is not regenerated, so the ligation product is not subsequently restricted; or ii) re-ligate to the donor plasmid fragment from which they were originally excised, in which case the site is regenerated and can be subsequently restricted again. The overall effect is that the

concentration of the intended assembly product in the one-pot reaction increases over time, whereas the concentration of original or re-ligated donor plasmids decreases over time, resulting in very efficient assembly (163, 164). One-pot assembly reactions can take place under isothermal conditions, but are more efficient if the reaction temperature is cycled between the different optimal temperatures of the ligase and endonuclease (164). The Golden Gate approach that was originally described (163), defines a broad strategy, rather than specific details of how it might generally be applied. Subsequent publications from various groups have described particular implementations of Golden Gate, crucially including frameworks for the hierarchical assembly of numerous parts and also for the collections of vectors and parts in the relevant formats (165, 166). These implementations are mostly focused on specific organisms, such as plants (165-168), yeast (169-173) or *E. coli* (137, 174). There is a lack of modular, multi-part assembly systems that are widely applicable to many organisms with a minimal upfront cloning effort and if one was developed, it would be extremely useful.

All of the above Golden Gate approaches require specific 'fusion site' sequences to be defined for the unique cohesive ends, which are incorporated into the assembled construct as unavoidable scar sequences at the junctions between parts. Scars therefore impose design constraints on constructs assembled by Golden Gate and other modular multi-part DNA assembly methods. At the junction between some types of parts, scars can have a major impact on functional properties. A key example is the junction between a CDS and the upstream sequence. Scars at this junction are transcribed to mRNA, which potentially affects mRNA structure (which in turn affects functional properties (471)) and also RBS accessibility. Furthermore, DNA sequence scars can be problematic for regulatory approaches that are designed using RNA elements, such as riboswitches (484), riboregulators (426) and small transcription activating RNAs (STARs) (485). Scars at the junction between the RBS and the CDS are also within the region of the mRNA that binds to the ribosome and therefore scars can directly affect the ribosome binding affinities and can also alter translation initiation. A direct comparison was carried out to assess and highlight the impact that four frequently used DNA assembly scars had at a junction between a CDS and the upstream sequence (Figure 5.1).

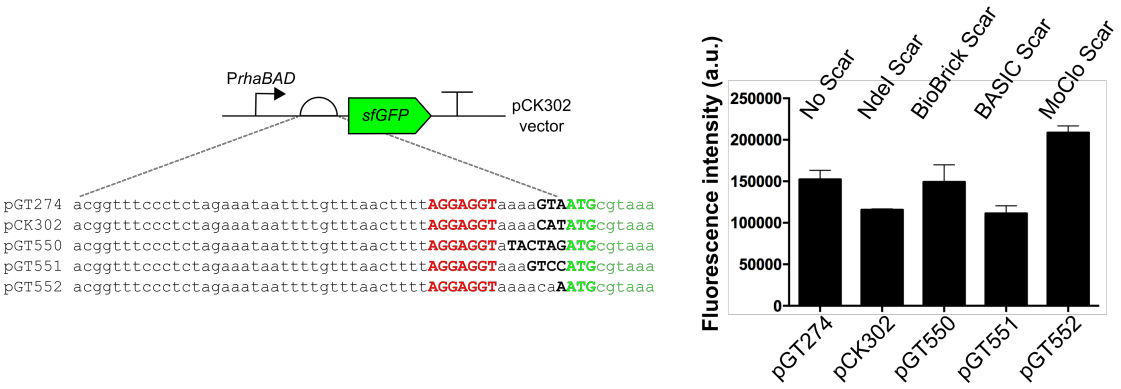


Figure 5.1. Effect of DNA assembly scars immediately upstream of the start codon on expression of *sfGFP*. The scar sequences (shown in bold text) that would result from from NdeI cloning, BioBrick assembly, MoClo assembly or BASIC assembly were each incorporated into derivatives of the reporter plasmid pCK302 (123) between the Shine-Dalgarno sequence (red text) of the RBS and the start codon (ATG) of *sfGFP* (green text) using inverse PCR. *E. coli* was independently transformed with each plasmid and transformants were characterised by flow cytometry. Fluorescence values presented represent the fluorescence of cells in mid-exponential phase of growth measured using flow cytometry with WT background subtracted. The error bars shown represent the standard deviation of three biological repeats.

In general, it is preferential to avoid imposing constraints on the design of a DNA construct. This is particularly important when the function of a genetic construct, such as a metabolic pathway-encoding construct, is sensitive to changes in protein expression levels, which in turn makes the constructs sensitive to sequence scars. A DNA assembly system that is capable of providing efficient modular multi-part assembly whilst introducing sequence scars at sensitive positions would be widely useful.

5.1.1 Aims

The aim of this chapter was to develop such a DNA assembly system, that had the efficiency of a modular DNA assembly methods for multi-part assemblies, while minimising the constraints on the design of DNA constructs. Furthermore, it was important that the DNA assembly method could easily be implemented in many different destination contexts, such as different organisms and/or expression vectors. Following the development of the DNA assembly system its suitability for combinatorial assembling metabolic pathway-encoding constructs would be demonstrated.

- Develop a multi-part modular DNA assembly method which is both functionally scarless and suitable for combinatorial assembly.

- Streamline the hierarchical framework of typical Golden-Gate based methods to make a DNA assembly method that is easily adaptable to new destination contexts and new organisms.
- Demonstrate the combinatorial ability to search expression spaces using carotenoid pathways.

Declaration – In this chapter, Dr Paweł Mordaka and Dr John Heap helped with the design of Start-Stop Assembly.

5.2 Results

5.2.1 A strategy for functionally-scarless assembly of expression units

To date, it has been unavoidable for modular multi-part DNA assembly methods not to generate scars between DNA ends. The DNA sequences of scars have been essential for specific and directional recognition and joining of DNA ends between parts, and as a result these sequences get incorporated as scars (typically and hereafter referred to as fusion sites) into the assembled construct. Here a strategy was devised to circumvent this issue, by exploiting already existing constraints on the design of expression constructs, such as conserved sequence motifs, which could be used as fusion sites for modular DNA assembly methods. By using such motifs as fusion sites, part junctions would be effectively scarless and the DNA assembly methods would not be introducing any additional constraints on the design of expression constructs. For such a motif to be suitable as a fusion site, it would need to be well-conserved, located at the boundaries between functional parts (for incorporating into a modular DNA assembly framework) that are sensitive to scars (to benefit from a scarless design), and to match the length of a cohesive DNA end that's cleaved by a known type IIS restriction endonuclease. The start and stop codons were identified as natural constraints of the genetic code that meet these requirements. The start and stop codons: i) occur at the beginning and end of every CDS, sites which are not only boundaries between functional elements but also particularly sensitive to scars; and ii) can be completely conserved. Modular DNA assembly methods typically use type IIS restriction endonucleases, such as BsaI, BbsI and/or BsmBI (163), which cleave 4 basepair (bp) overhangs, however the start and stop codons are 3 bp long. To use the start and stop codons as fusion sites in scarless and modular design, it is necessary for a type IIS restriction endonuclease to cleave 3 bp overhangs. Several such

endonucleases are known, including the commercially-available enzyme SapI, which was used here.

With the aim of designing a scarless, modular and hierarchical multi-part DNA assembly method, an architecture was designed for assembling a single 'expression unit' around using the start and stop codons as fusion sites, which would serve as the first assembly level (Level 1; Figure 5.2a). To allow independent control over transcription and translation, each expression unit was divided into four key functional parts; the promoter, RBS, CDS and transcriptional terminator. These four elements were delineated from one another and from a vector by five fusion sites (Figure 5.2a), which were named α (alpha), β (beta), γ (gamma), δ (delta) and ε (epsilon). When choosing the sequences for these fusion sites, it was important that they were all 3 bp in length so that they could all be assembled in the same one-pot restriction-ligation reaction using the same type IIS restriction enzyme.

The γ fusion site is the junction between the RBS and the CDS and was therefore first defined as the sequence ATG; the most common (486, 487) and most efficient (488) start codon. The δ fusion site represents the junction between the CDS and the terminator, which was defined as the sequence TAA; the most frequently-used stop codon in synthetic biology, and is efficient in a wide range of organisms (489). The β fusion site is the junction between the promoter and the RBS, which tends to mark the transcriptional start site (TSS). With no known constraints in the TSS sequence that could be used for the β fusion site, the next step was to try and identify a consensus or bias for the *E. coli* TSS sequence that could be used to minimise the contextual effects that this fusion site may have in the assembled construct. 3746 *E. coli* MG1655 transcriptional start sites were previously identified (490) and were aligned using WebLogo (491). While no consensus was found, a bias in the TSS sequence was identified (Figure 5.3). The alignment showed a preference for A or G at the TSS +1 position, a preference for C or T at the -1 position, but no preference at the -2 position. Using this information, all TSS with adenine at +1 were aligned to assess whether there was a bias at either the -1 or -2 positions; the data showed no bias in either position. Then all TSS with guanine at the +1 position were aligned to assess whether there was a bias at either the -1 or -2 positions; no bias at the -1 or -2 position was observed. Having identified a bias at the -1 and +1 positions of the *E. coli* TSS and no bias at the -2, the number of potential β fusion sites was narrowed to 16 different combinations. To minimise the risk of misassemblies, any of the 16 candidate fusion sites that contained 2 bp matching the previously defined γ (ATG) and δ (TAA) fusion

sites were removed, which left six potential β fusion site sequences. Next, the frequency of these six fusion sites at the TSS in the *E. coli* MG1655 was assessed. Whilst ACA appeared 178 times, it would have been a palindromic site and therefore a poor choice of fusion site, CCA appeared 175 times, was not palindromic and therefore was defined as the β fusion site (Figure 5.3). The sequences upstream of a promoter and downstream of a terminator (sequences flanking an expression unit) have no known natural constraints, which could have been used to define the sequences of the α and ε fusion sites. As such, these sequences were designed to be as different as possible from the three fusion sites already defined (β, γ and δ) in order to minimise mis-assemblies. CAG was assigned to the α fusion site and GGA to the ε fusion site (Figure 5.2a). By defining these five fusions sites with 3 bp sequences, it was possible to assemble the expression unit from the four individual parts in a single one-pot restriction-ligation reaction (Figure 5.2a).

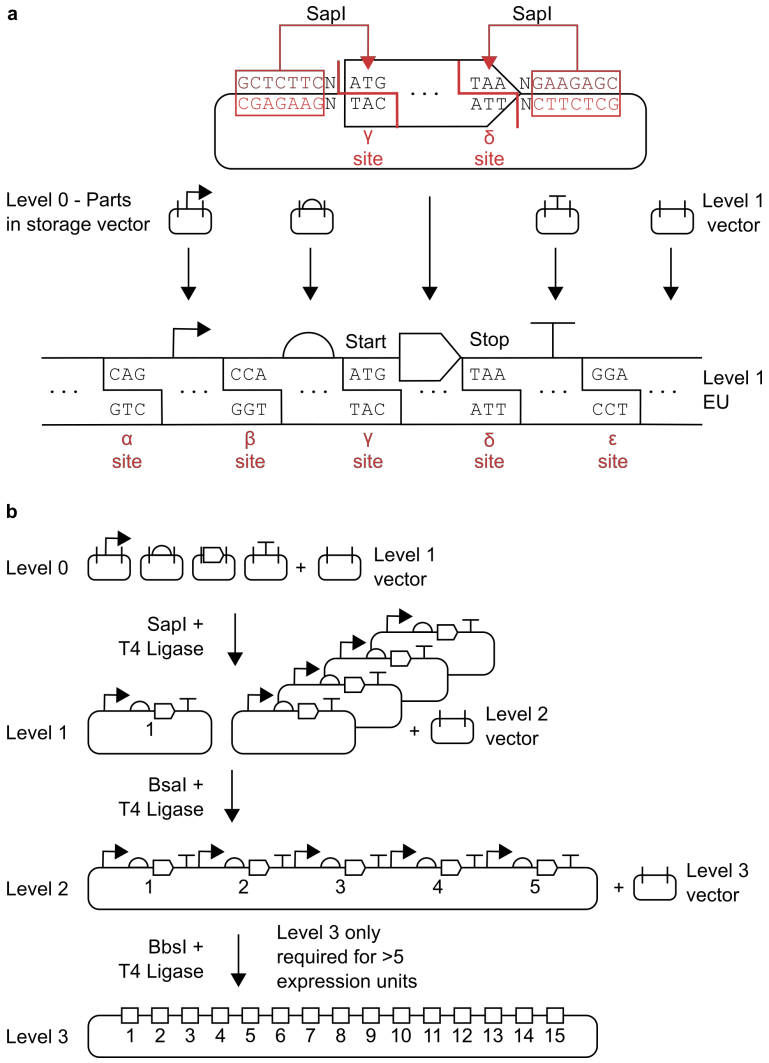


Figure 5.2. Schematic overview of Start-Stop Assembly. (a) Architecture of Level 1 assembly of expression units from genetic parts, showing detail for the CDS part in Level 0. Parts stored in Level 0 are flanked by inward-facing SapI restriction sites. Each type of part (promoter, RBS, CDS, terminator) uses different and unique fusion sites (α-ε sites) that allow the correct assembly of expression units. Coding sequences use the start codon ATG (γ site) and stop codon TAA (δ site) as fusion sites allowing functionally scarless assembly of expression units. (b) Schematic illustration showing the overall framework for Start-Stop Assembly. Up to 15 expression units can be hierarchically assembled from basic parts.

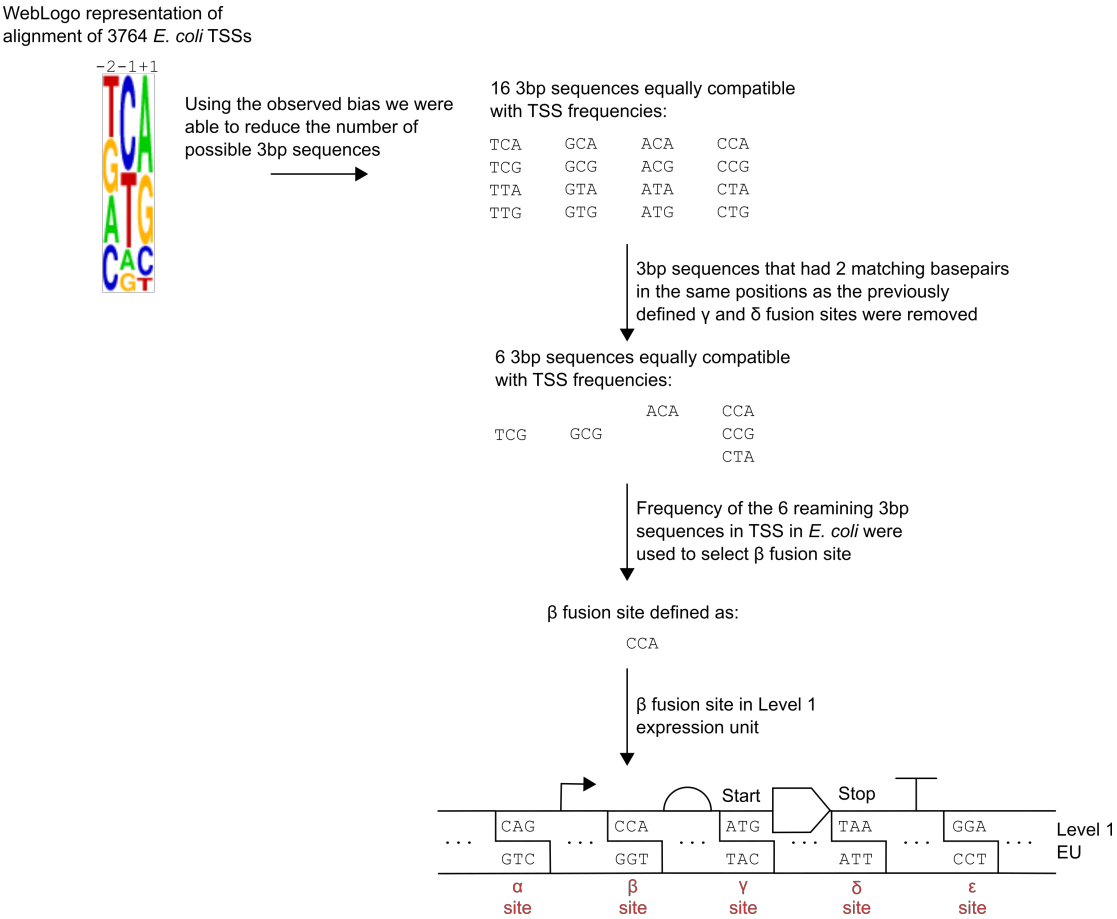


Figure 5.3. Choice of β fusion site. In an attempt to minimise the introduction of new sequences by the β fusion site at the junction between the promoter and the UTR/RBS, an attempt was made to try to identify a consensus sequence for *E. coli* transcriptional start sites (TSS). 3746, previously-described *E. coli* MG1655 TSSs (490), were aligned and is visualised in the figure using WebLogo (491). The alignment showed a preference for A or G at the TSS +1 position, a preference for C or T at the -1 position, and no preference at the -2 position. First, the candidate sequences for the β fusion site were limited to the 16 sequences equally compatible with the observed preferences in the TSS. Next, to avoid misassembly, any of the candidate sequences which matched the already defined γ (ATG) and δ (TAA) fusion sites at two positions were removed, which left six potential fusion sites. Finally, the frequencies of these six candidate sequences were compared among the 3746 previously-described *E. coli* MG1655 TSSs. The most frequently-occurring were ACA, which occurred 178 times, and CCA, which occurred 175 times. ACA is palindromic so would be a poor choice of fusion site and therefore the β fusion site was defined as CCA.

5.2.2 Requirements for Level 1 assembly from Level 0 parts

For an expression unit to be assembled in a one-pot restriction-ligation using the Level 1 architecture outlined above, the DNA parts (promoter, RBS, CDS and terminator) and vectors must meet certain specifications. Each DNA part must begin and end with the appropriate fusion sites for the type of part and in the case of the β, γ and δ sites these should be positioned at the TSS (if known), start codon or stop codon, respectively. Each part must also be flanked by an inward-facing SapI recognition site

(directionality exists because the recognition site GCTCTTC is asymmetrical) separated from the fusion site by a 1 bp spacer (Figure 5.2a, Figures A5 – A8). This configuration of recognition sequence and fusion site would ensure that when digested, SapI cleaves the DNA ends of each part with a 3 bp 5' overhang that corresponds to a 3 bp fusion site. It is possible to generate appropriately-formatted DNA parts using methods such as PCR and DNA synthesis; appropriately-formatted DNA parts can be used for assembling expression units directly as linear DNA or as circular plasmid DNA molecules. It is typical to store a DNA part in a vector, in which case the backbone of the vector should not contain any other SapI sites that could interfere with the assembly of an expression unit, affecting the assembly reactions efficiency, misassembly rate and potentially introduce bias. Furthermore, DNA parts should not contain any internal SapI sites, as they are used in Level 1 assembly; nor any BsaI sites or BbsI sites, as they are used in later assembly levels. These specifications define Level 0 (zero) of Start-Stop Assembly, which represents the initial 'entry' or 'storage' level of the system.

Most Golden Gate based systems use many entry or storage vectors, here a single Level 0 vector, pStA0, was designed and constructed that can be used for storage of parts of all types in *E. coli* (Table 5.1). To assist the use of pStA0, outward-facing BsaI restriction sites were included, which can be used for cloning individual parts into the storage vector using special storage fusion sites (F and R; Figure 5.4) and a *lacZα* gene was also included, which can be used for blue/white screening. To allow parts to be stored in Level 0 in the required Start-Stop Assembly format, a standard prefix and suffix sequences was defined (Table 5.2). The standard prefix and suffix should either be introduced to the ends of parts as: i) primer tails when they are PCR-amplified (Table 5.2); or ii) when designing parts sequences for DNA synthesis. The prefix and suffix sequence include inward-facing BsaI recognition sites and storage fusion sites (F and R) for cloning parts into pStA0 and inward-facing SapI recognition sites with α, β, γ, δ or ε fusion sites (depending on the type of part) for subsequent multi-part assembly of expression units in Level 1 (Figure A5 – A8, Table 5.2). Finally, assembly requires a Level 1 vector including two outward-facing SapI recognition sites with appropriately-positioned α and ε fusion sites, but without any other SapI sites, for the reasons described above (Figure A9).

Table 5.1. List of core Start-Stop Assembly vectors. The 15 vector names reflect their level in the hierarchy (number in blue) and their donor fusion sites (letters or numbers in red) or in the case of Level 3 vectors their acceptor fusion sites (numbers in green). Each vector contains an assembly cassette (including a *lacZα* gene for blue/white screening), a resistance marker (Amp^R, Tet^R, Kan^R or Cam^R) and a replicon (high-copy pMB1 or low-copy p15A). Each also has a unique ID number (pGT...). Genbank accession numbers are shown.

Plasmid Name	Accession Number	Level	Acceptor fusion sites	Donor fusion sites	Comments
pStA0	MG649420	0	F-R (BsaI)	α-ε (SapI)	Amp^R, pMB1, *lacZα*, ID = pGT400
pStA1AZ	MG649422	1	α-ε (SapI)	A-Z (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT401
pStA1AB	MG649421	1	α-ε (SapI)	A-B (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT402
pStA1BZ	MG649424	1	α-ε (SapI)	B-Z (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT403
pStA1BC	MG649423	1	α-ε (SapI)	B-C (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT404
pStA1CZ	MG649426	1	α-ε (SapI)	C-Z (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT405
pStA1CD	MG649425	1	α-ε (SapI)	C-D (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT406
pStA1DZ	MG649428	1	α-ε (SapI)	D-Z (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT407
pStA1DE	MG649427	1	α-ε (SapI)	D-E (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT408
pStA1EZ	MG649429	1	α-ε (SapI)	E-Z (BsaI)	Tet^R, p15A, *lacZα*, ID = pGT409
pStA212	MG649430	2	A-Z (BsaI)	1-2 (BbsI)	Kan^R, p15A, *lacZα*, ID = pGT417
pStA223	MG649431	2	A-Z (BsaI)	2-3 (BbsI)	Kan^R, p15A, *lacZα*, ID = pGT418
pStA234	MG649432	2	A-Z (BsaI)	3-4 (BbsI)	Kan^R, p15A, *lacZα*, ID = pGT419
pStA313	MG649433	3	1-3 (BbsI)	N/A	Cam^R, p15A, *lacZα*, ID = pGT415
pStA314	MG649434	3	1-4 (BbsI)	N/A	Cam^R, p15A, *lacZα*, ID = pGT416

Table 5.2. Prefix and suffix sequences. Prefix and suffix sequences should be added to genetic parts to allow cloning into Level 0 storage vector pStA0 and subsequent use in Start-Stop Assembly. They can be added by PCR using primer tails or included in the design of synthetic DNA sequences. The prefix and suffix sequences include inward-facing BsaI recognition sites (grey box) and corresponding storage fusion sites (F and R; bold) for cloning parts into Level 0 storage vector pStA0 using a Level 0 assembly reaction, as well as inward-facing SapI recognition sites (grey box) with corresponding donor fusion sites (α, β, γ, δ or ε depending on the type of part; bold) for subsequent multi-part assembly of expression units in Level 1 assembly.

Part	Prefix					Suffix			
Promoter	GGTCTCA	**TGTG**	GCTCTTCG	**CAG**	-Promoter-	**CCA**TGAAGAGC	**GACC**	AGAGACC	
	BsaI	F	SapI	α		β	SapI	R	BsaI
RBS	GGTCTCA	**TGTG**	GCTCTTCG	**CCA**	-RBS-	**ATG**TGAAGAGC	**GACC**	AGAGACC	
	BsaI	F	SapI	β		γ	SapI	R	BsaI
CDS	GGTCTCA	**TGTG**	GCTCTTCG	**ATG**	-CDS-	**TAA**TGAAGAGC	**GACC**	AGAGACC	
	BsaI	F	SapI	γ		δ	SapI	R	BsaI
Terminator	GGTCTCA	**TGTG**	GCTCTTCG	**TAA**	-Terminator-	**GGA**TGAAGAGC	**GACC**	AGAGACC	
	BsaI	F	SapI	δ		ε	SapI	R	BsaI

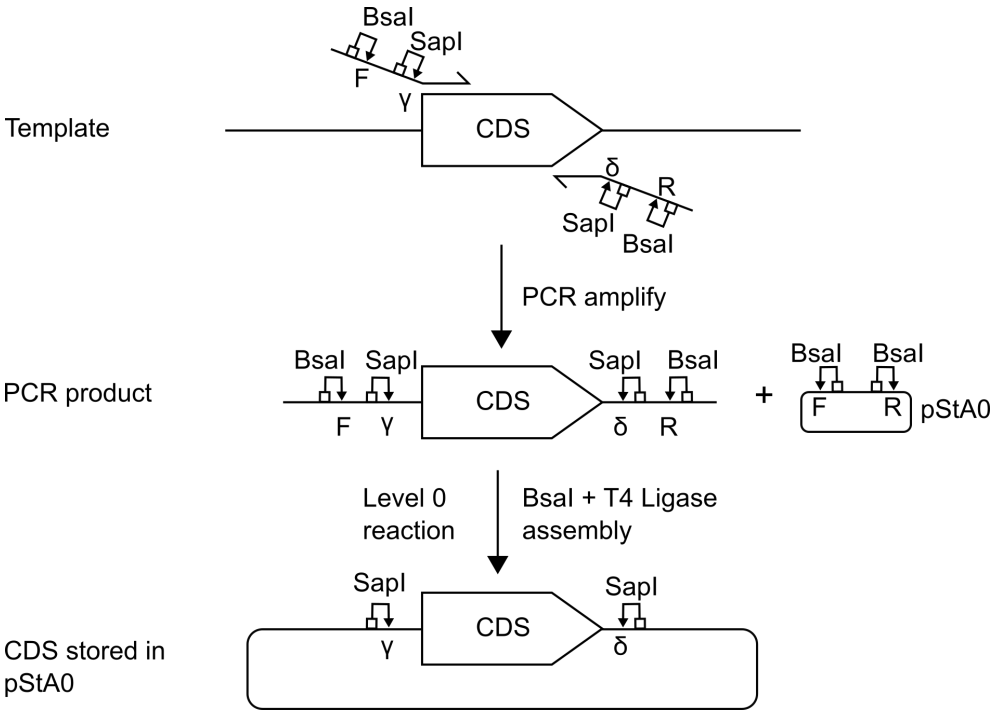


Figure 5.4. Strategy for PCR-amplification and cloning of genetic parts into Level 0 vector pStA0. To be used in Start-Stop Assembly, each genetic part must be flanked by the appropriate prefix and suffix sequence (Table 5.2), which can be added by PCR using primers with 5' tails. Different types of genetic part use different prefix and suffix sequences, because they include the differing fusion sites (Table 5.2). In each instance the prefix and suffix sequences contain the inward-facing BsaI recognition sites and corresponding storage donor fusion sites F (TGTG) and R (GACC), which are used to clone parts into Level 0 vector pStA0. Between the BsaI restriction sites are inward-facing SapI restriction sites and corresponding donor fusion sites that are not used during the storage cloning shown but are later used for excising cloned parts from pStA0 for use in Level 1 assemblies.

5.2.3 Validation of multi-part DNA assembly using 3 bp fusion sites

Type IIS restriction endonucleases that cleave 4 bp fusion sites have been typically used in modular multi-part DNA assembly frameworks and their highly efficient performances has been well validated in published studies (163, 165). However, type IIS restriction endonucleases that cleave 3 bp fusion sites have hardly been used. The proposed Level 1 architecture for Start-Stop Assembly uses five 3 bp fusion sites (α, β, γ, δ and ε), which have not been previously tested and could therefore cause differences in the assembly performance, such as the assembly fidelity and bias, compared to existing methods. As such, before incorporating the Level 1 architecture into an assembly hierarchy for constructing multi-gene pathways, the next step was to evaluate and validate assembly using just this Level 1 design. In order to do so, different types of DNA parts in the appropriate Level 0 context were required, which could be assembled as an expression unit in a recipient Level 1 vector. First, six promoters were chosen from the widely-used (447) library of constitutive *E. coli* promoters constructed by JC Anderson (122). The set of six promoters were chosen based on their previously-reported characterisation data (122) to offer a wide range of expression strengths that was evenly spaced within the range (Figure 5.5a). Secondly, a library of RBSs was constructed using a randomised library design (Figure 5.6). Like in Chapter Three, the RBS library design conserved the *E. coli* Shine-Dalgarno (SD) sequence (AGGAGGT) but randomised (N) 15 bp upstream and 7 bp downstream of the SD by incorporating the four nucleotides A, T, G and C at approximately the same 25% frequency at each N position. The RBS library was introduced upstream of *sfGFP* in the previously described reporter plasmid pCK302 (123) using degenerate primers and inverse PCR. The PCR product was then circularised by ligation and the ligation product was used to transform *E. coli* in order to obtain the RBS library. 96 RBS clones were randomly isolated from the library and characterised in a mid-exponential phase of growth (Figure 5.6). From this, six RBSs were chosen that offered evenly-spaced distribution of expression over a wide range of expression strengths (Shown in Figure 5.5b). Next, for the simple assessment of the Level 1 assembly fidelity and bias, the reporter gene that encodes the enhanced yellow florescent protein (EYFP) was chosen as the CDS to allow simple visualisation and measurements of expression. Lastly, four strong terminators that had dissimilar sequences were chosen from a previously-characterised library of terminators (127). Dissimilar sequences were preferable for minimising the risk of undesirable homologous recombination. All of the DNA parts stated above were cloned into the storage vector pStA0 and the sequence was verified (Table A8), ready for Level 1 assemblies. With the DNA parts stored in pStA0, next a Level 1 vector was needed to assemble the parts in. While the overall multi-level Start-

Stop Assembly hierarchy uses a series of nine Level 1 vectors as described later, for the purpose of validating the Level 1 assembly, at this point one Level 1 vector was used. The key features of the Level 1 vector with respect to these experiments are that the vector contains two outward-facing SapI recognition sites with appropriately-positioned α and ε fusion sites and a *lacZα* gene that can be used for blue/white screening (Figure A9).

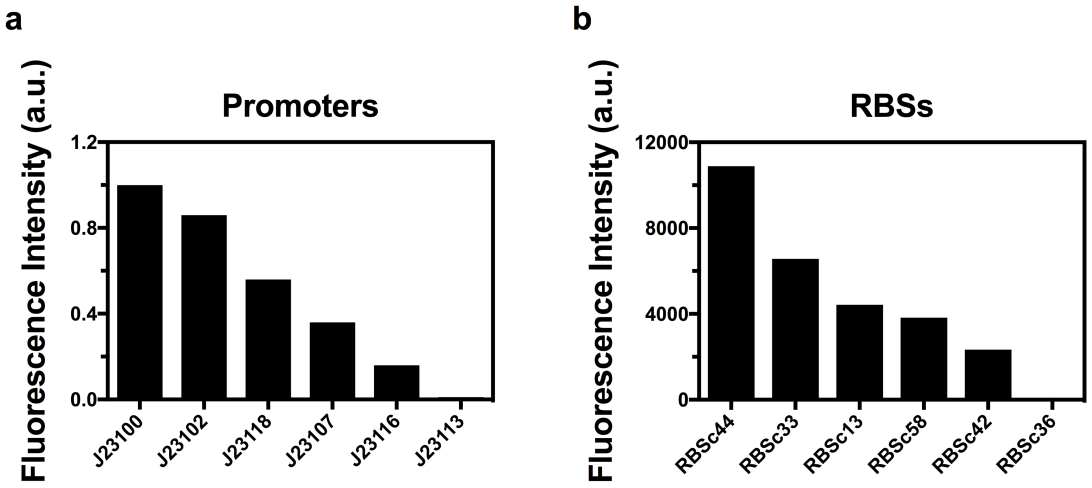


Figure 5.5. Promoters and RBSs stored in pStA0 in this study. (a) Fluorescence intensities reported by Anderson (122) for the six promoters were selected, relative to promoter J23100. (b) Fluorescence intensities of the six RBSs chosen from the characterised library described in Figure S4. These six RBSs give a wide and evenly-spaced distribution of expression strengths, and were stored in Level 0 vector pStA0. Promoters, in descending order of strength: P1 = J23100, P2 = J23102, P3 = J23118, P4 = J23107, P5 = J23116, P6 = J23113. RBSs, in descending order of strength: R1 = RBSc44, R2 = RBSc33, R3 = RBSc13, R4 = RBSc58, R5 = RBSc42, R6 = RBSc36.

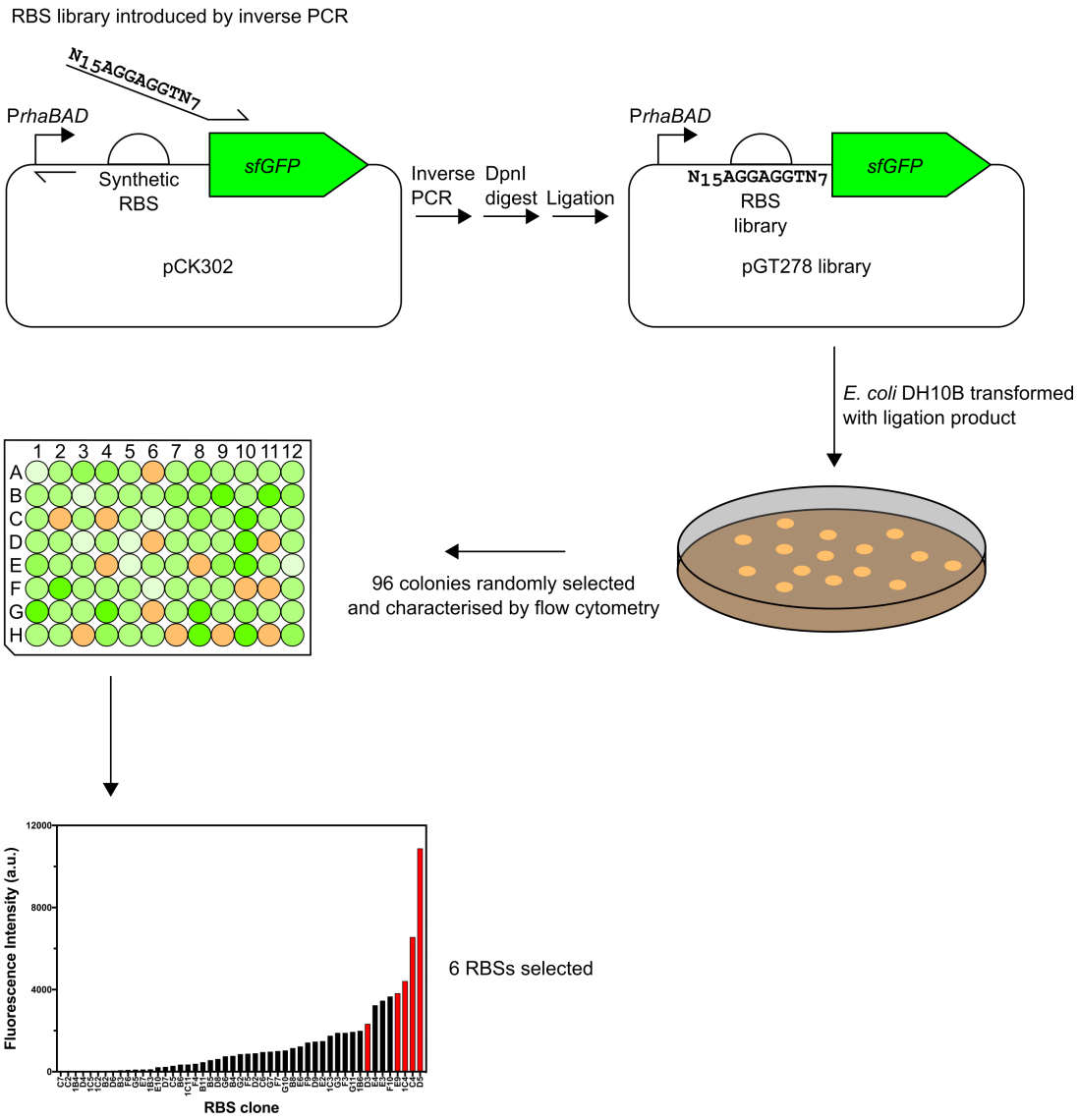


Figure 5.6. Design, generation and characterisation of a synthetic RBS library. The RBS library design conserved the *E. coli* Shine-Dalgarno (SD) consensus sequence (AGGAGGT) but randomised (N) 15 bp upstream and 7 bp downstream of the SD by incorporating the four nucleotides A, T, G and C at approximately the same 25% frequency at each N position. The RBS library was introduced to pCK302 upstream of *sfGFP* by inverse PCR using primers oligoGT448 and oligoGT463. The PCR reaction product was treated with *DpnI* to remove the template plasmid DNA. PCR fragments of the expected size (5.6 kbp) were excised and purified following gel electrophoresis, circularised by ligation and *E. coli* was transformed with the resultant ligation product by electroporation. 96 *E. coli* transformant colonies were randomly selected and characterised in mid-exponential phase of growth by flow cytometry (as described in Chapter Two – section **2.5.5**). The fluorescence intensity of each clone is plotted with the background fluorescence subtracted. From this RBS library, six RBSs were chosen to give a wide and evenly-spaced distribution of expression strengths (shown in red here, and separately in Figure 5.5b).

To validate the Level 1 assembly, two different assembly experiments were designed. The first assessed the assembly fidelity by assembling replicates of an individual expression construct (Figure 5.7) and the second assessed assembly bias by combinatorially assembling an expression construct (Figure 5.8). Before carrying out the assembly fidelity and bias experiments, reaction conditions for one-pot restriction-ligation reactions that contained SapI and T4 DNA Ligase were tested. Four conditions were compared, isothermal incubation at 23 °C (room temperature) for 5 hours, isothermal incubation at 37 °C for 1 hour, isothermal incubation at 37 °C for 5 hours and lastly, conditions whereby the reaction was subjected to 30 two-step cycles of 37 °C for 5 minutes, then 16 °C for 5 minutes (cycling). As previously reported (164, 492), the cycling conditions were considerably more efficient compared to the isothermal conditions and they were therefore discarded and only cycling conditions described as above were used hereafter.

To assess the assembly fidelity, the mid-strength promoter J23118, the mid-strength RBS RBSc13, the terminator L3S2P55 and the *eyfp* CDS were assembled in the Level 1 vector pStA1AZ (Table 5.1) in a one-pot restriction-ligation reaction that contained both SapI and T4 DNA Ligase. Five independent replicates of the assembly reaction were carried out, the five reaction mixtures were then used for five independent transformations of *E. coli* DH10B. Where blue/white screening using the *lacZα* gene of pStA1AZ, this showed approximately one blue colony (containing the unmodified pStA1AZ vector) per 1000 white colonies. By illuminating the transformant colonies on agar plates with blue light, a uniform level of fluorescence amongst colonies was observed (Figure 5.7). From each of the five experiment replicates two colonies were selected at random for further analysis. Plasmids from the ten clones were first purified and then sequenced, which showed all the ten constructs had been correctly assembled and that all five of the 3 bp fusion sites had performed as expected (Figure 5.7). In addition to the sequencing, the fluorescence of the ten clones and a pool of several hundred transformant clones from one transformation were analysed during mid-exponential growth phase using flow cytometry. The distribution of fluorescence in the pool of cells was very similar to the distribution of fluorescence of cells in the ten verified and isolated clones (Figure 5.7 shows an example and the others are shown in Figure 5.9a). These results indicate very high fidelity of assembly of the Level 1 architecture using five 3 bp fusion sites in a one-pot reaction containing SapI and ligase.

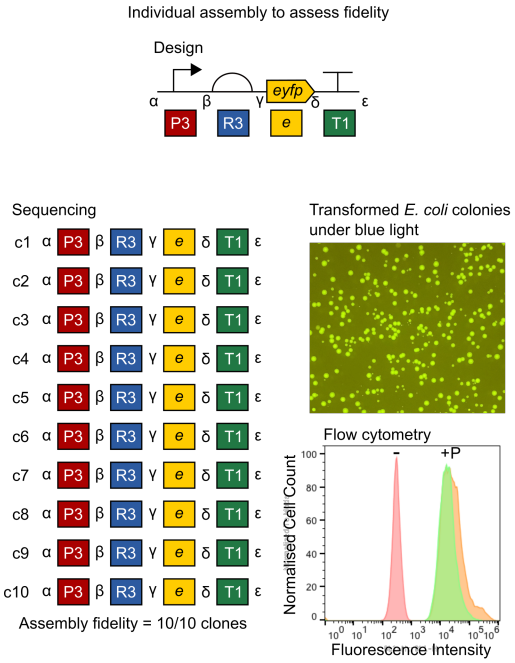


Figure 5.7. Assessment of fidelity of multi-part Level 1 assembly. Constructs were assembled from Level 0 parts (in storage plasmids) into Level 1 vector pStA1AZ. Promoters, in descending order of strength: P1 = J23100, P2 = J23102, P3 = J23118, P4 = J23107, P5 = J23116, P6 = J23113. RBSs, in descending order of strength: R1 = RBSc44, R2 = RBSc33, R3 = RBSc13, R4 = RBSc58, R5 = RBSc42, R6 = RBSc36. CDS: e = *eyfp*. Terminator: T1 = L3S2P55. Representative images of *E. coli* DH10B colonies transformed with assembly reactions are shown illuminated by blue light to visualise fluorescence. In each experiment ten randomly-selected colonies (c1-c10) were analysed by sequencing (with primers oligoGT234 and oligoGT235) and flow cytometry. Sequencing results show expected intact parts and fusion sites. Flow cytometry histograms show fluorescence intensity of 10,000 events (cells) normalised to the maximum (in order to visualise distribution rather than absolute values) for wild-type *E. coli* DH10B as a negative reference (-), the most fluorescent of the ten clones as a positive reference (+) and a pool of several hundred transformants (P). Assessment of assembly fidelity using assembly of an individual P3-R3-e-T1 expression unit. Two colonies were randomly selected for analysis from each of five replicate assemblies. The flow cytometry positive reference (+) strain is c3.

To assess the assembly bias using the five 3 bp fusion sites in the Level 1 assembly reaction, an equimolar mixture of all six promoters, an equimolar mixture of all six RBSs, terminator L3S2P55 and CDS *eyfp* were combinatorially assembled in the Level 1 vector pStA1AZ. The assembly reaction was performed using the same assembly procedures, *E. coli* transformation and analysis as described above for the fidelity experiment, except that the 30 colonies were randomly chosen for analysis from three individual assembly replicates and transformations (ten colonies from each assembly reaction). By illuminating the agar plates that contained the transformant colonies with a blue light, a large diversity in visual fluorescence was observed (Figure 5.8). 30 transformant colonies were randomly chosen from the three transformations and the sequencing data showed that 28 of the 30 clones showed assembly of expected parts

and correct assembly of all five fusion sites (Figure 5.8). Furthermore, a variety of different combinations of promoters and RBSs were assembled upstream of the *eyfp* CDS in these 28 correctly-assembled constructs. The sequences of the two misassembled constructs indicated a spurious ligation in each clone (clone 6 and clone 12). In clone 6 a fragment of *E. coli* genomic DNA and a fragment of vector DNA had taken the place of the RBS part between the β and γ fusion sites and in clone 12 a fragment of vector DNA had taken place of the entire expression unit between the α and ε fusion sites (Figure 5.8). The flow cytometry analysis of a pool of several hundred transformant clones showed the distribution of fluorescence of cells to span a wide range of values from the negative control (pStA1AZ) to the positive control (pStA1AZ::P3-R3-*eyfp*-L3S2P55; the construct from the previous fidelity assembly experiment, Figure 5.8). Furthermore, big differences between the fluorescence populations of cells from the individual ten clones was observed (Figure 5.9b). These results indicate that the combinatorial assembly using Level 1 architecture in a one-pot reaction is unbiased.

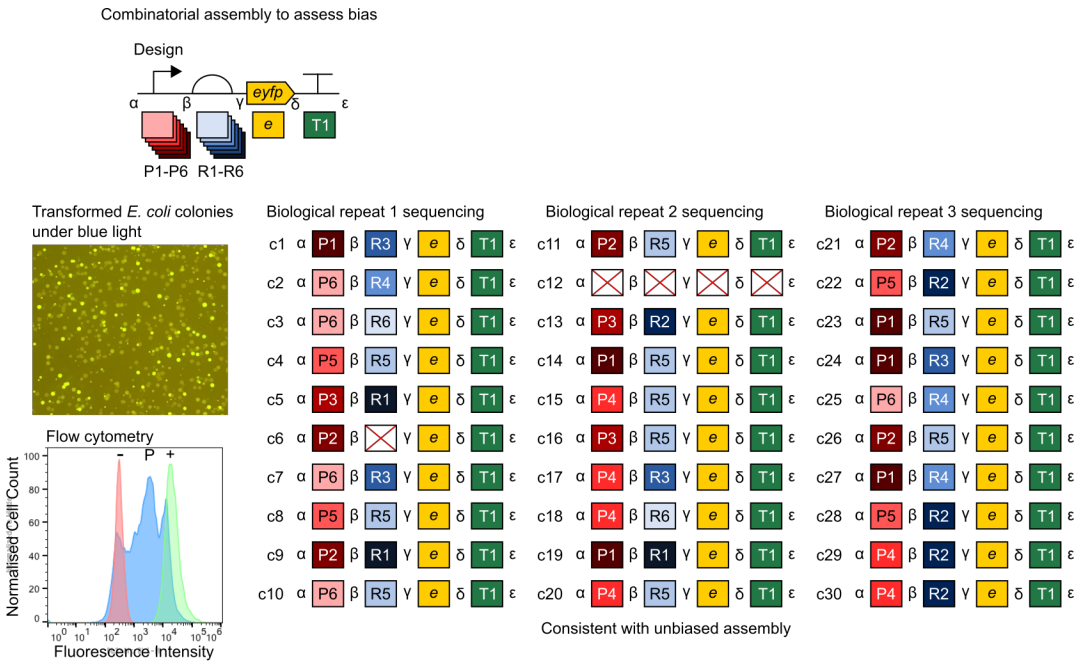


Figure 5.8. Assessment of bias of multi-part Level 1 assembly. Constructs were assembled from Level 0 parts (in storage plasmids) into Level 1 vector pStA1AZ. Promoters, in descending order of strength: P1 = J23100, P2 = J23102, P3 = J23118, P4 = J23107, P5 = J23116, P6 = J23113. RBSs, in descending order of strength: R1 = RBSc44, R2 = RBSc33, R3 = RBSc13, R4 = RBSc58, R5 = RBSc42, R6 = RBSc36. CDS: e = *eyfp*. Terminator: T1 = L3S2P55. Representative images of *E. coli* DH10B colonies transformed with assembly reactions are shown illuminated by blue light to visualise fluorescence. In each experiment ten randomly-selected colonies (c1-c10) were analysed by sequencing (with primers oligoGT234 and oligoGT235) and flow cytometry. Sequencing results show expected intact parts and fusion sites. Flow cytometry histograms show fluorescence intensity of 10,000 events (cells) normalised to the maximum (in order to visualise distribution rather than absolute values) for wild-type *E. coli* DH10B as a negative reference (-), the most fluorescent of the ten clones as a positive reference (+) and a pool of several hundred transformants (P). Assessment of assembly bias by combinatorial assembly of EYFP expression units using six promoters P1-P6, six RBSs R1-R6, eyfp and terminator T1. Ten colonies from one assembly were randomly selected for analysis. The flow cytometry positive reference (+) strain is c1. In the sequencing results the red cross indicates a misassembly between β and γ in place of the UTR/RBS.

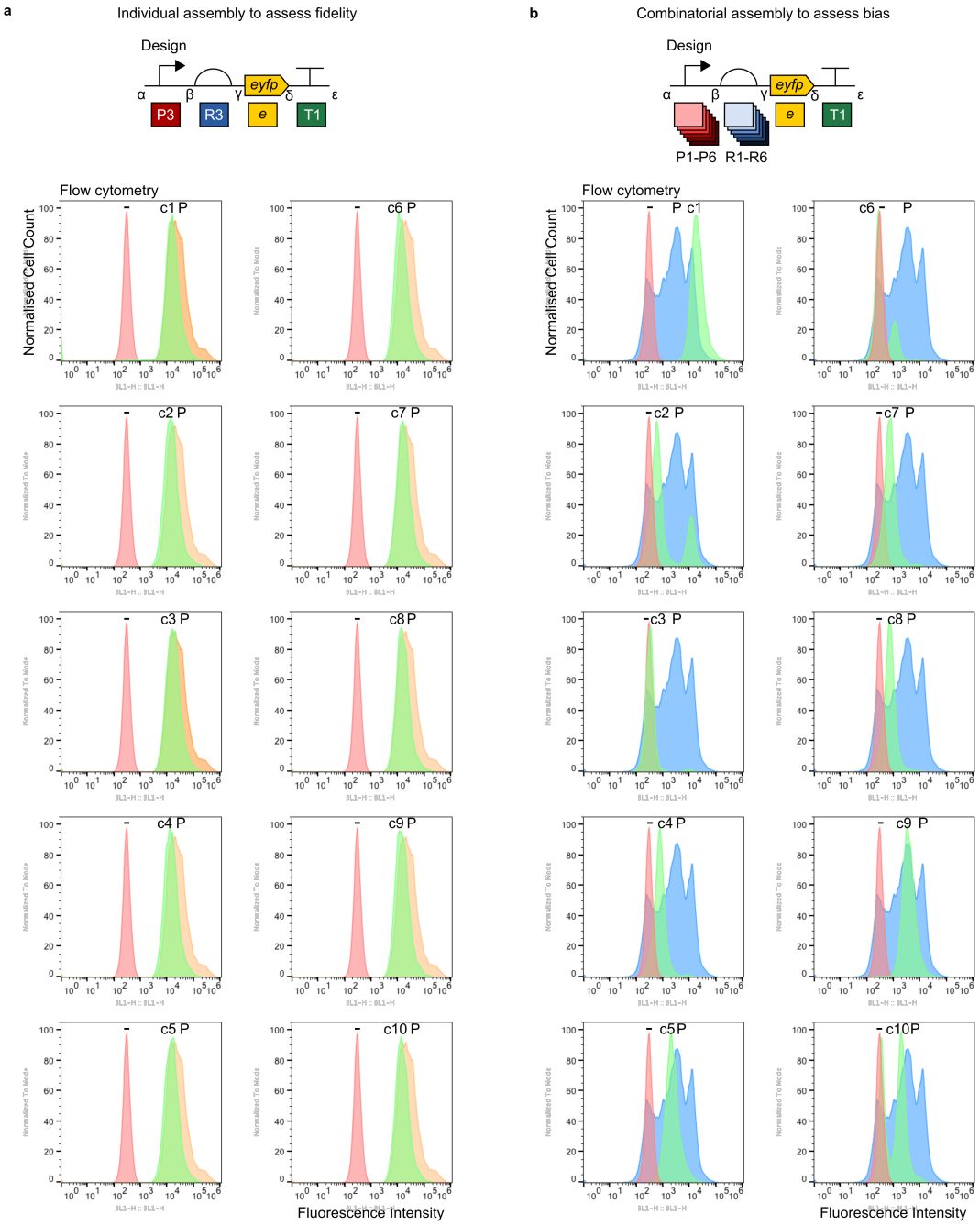


Figure 5.9. Flow cytometry data for individual clones used in assessment of fidelity and bias of multi-part Level 1 assembly. The experiments are described in the main text and Figure 5.7 and 5.8. Flow cytometry histograms show fluorescence intensity of 10,000 events (cells) normalised to the maximum (in order to visualise distribution rather than absolute values) for wild-type *E. coli* DH10B as a negative reference (-), each of the ten clones (c1-c10) and a pool of several hundred transformants (P). (a) Assessment of assembly fidelity using assembly of an individual P3-R3-e-T1 expression unit. The ten clones showed similar fluorescence intensity to one another. (b) Assessment of assembly bias by combinatorial assembly of EYFP expression units using six promoters P1-P6, six RBSs R1-R6, eyfp and terminator T1. The ten clones showed widely differing fluorescence intensity values.

5.2.4 A streamlined assembly hierarchy to minimise the number of destination vectors required for new contexts and organisms

Having validated the Level 1 architecture, the next step was to incorporate it into a multi-level hierarchical assembly framework that allows the assembly of multiple expression units in a single construct. It is possible to have different structures for hierarchical Golden Gate based methods; the structure of which is dictated by the identity and arrangement of fusion sites within and between levels. Golden Gate based methods use vectors at intermediate level (each level except the first or last) with acceptor and donor fusion sites for incorporating DNA parts from preceding levels and excising DNA parts for following levels, respectively. In the Level 1 architecture of Start-Stop Assembly (Figure 5.2a), the first (α) and last (ε) acceptor fusion sites must always be the same, which in general suits the assembly of individual expression units that are well defined units. However, using a defined first and last fusion site at each level of the assembly is typically not employed by Golden Gate based methods, rather the hierarchy allows different combinations of first and last fusion sites to be assembled, which provides greater flexibility when assembling a different number of parts. To allow different combinations of the first and last fusion sites to be assembled, sets of multiple vectors are required for later levels that can accommodate each different pair of acceptor fusion sites (for example, see Figure 5.10b). While this hierarchy performs well, it has one major shortcoming; it requires the complete set of multiple vectors (that can accept any fusion site configuration) to be constructed for each different destination context, such as plasmids with different copy numbers, or shuttle vectors for different organisms, or different integration sites in an organism. This is a significant drawback of using such multi-part DNA assembly in metabolic engineering, where metabolic pathway-encoding constructs regularly need to be expressed in many different organisms and contexts. With this issue yet to be addressed in previous modular multi-part assembly methods, a different hierarchy was designed for Start-Stop Assembly that would overcome this issue at the second multi-part assembly level, Level 2, where it would be most significant.

For many multi-part assemblies, Level 2 will be the destination (final) level. If five expression units were being assembled at Level 2, a typical hierarchy design (Figure 5.10b) would require five alternative combinations of acceptor fusion sites, and therefore five alternative destination vectors per destination context. In contrast, the Start-Stop Assembly hierarchy was designed to use one specific pair of fusion sites, named A and Z, that will always be used as the Level 2 acceptor fusion sites (Figure 5.10a). To implement this hierarchical design, a set of Level 1 donor fusion sites were

defined (A, B, C, D, E and Z), which are used for assembling Level 1 parts into a Level 2 vector. As mentioned A and Z make up the specific first and last fusion sites, which are used for part-vector junctions, while B, C, D and E are the fusion sites used for Level 1 part-part junctions. By defining a specific first and last fusion site at Level 2, only one Level 2 vector is required, regardless of the number of units being assembled from Level 1 (up to the five-unit maximum limit of Level 2, discussed later). As a result, only one new Level 2 vector needs to be constructed whenever an alternative vector for a new destination context is required (Figure 5.10a), rather than five alternative Level 2 vectors that the traditional design would require (Figure 5.10b). To assemble a number of units from Level 1 into one Level 2 vector, the first unit from Level 1 must begin with the A fusion site and the last unit from Level 1 must end with the Z fusion site. To manage this, while still allowing different numbers of expression units to be assembled at Level 2, each expression unit must be assembled in a certain Level 1 vector. The choice of Level 1 vector depends on the number and order of units being assembled. To allow the last expression unit in a construct to always end with the Z fusion site, a series of Level 1 ‘Z vectors’ were constructed, in which the second of the two fusion sites is replaced by a Z fusion site (Figure 5.10a).

This streamlined reorganisation of the hierarchical structure moves the prerequisite of additional vectors from Level 2, where as a destination level it is problematic as described earlier, to the intermediate level (Level 1) which is not a destination level. Therefore, the core set of Level 1 vectors constructed here can be used for assembling constructs of different sizes in any final Level 2 assembly destination vector. The user need only construct a single new (Level 2) destination vector per new context or organism.

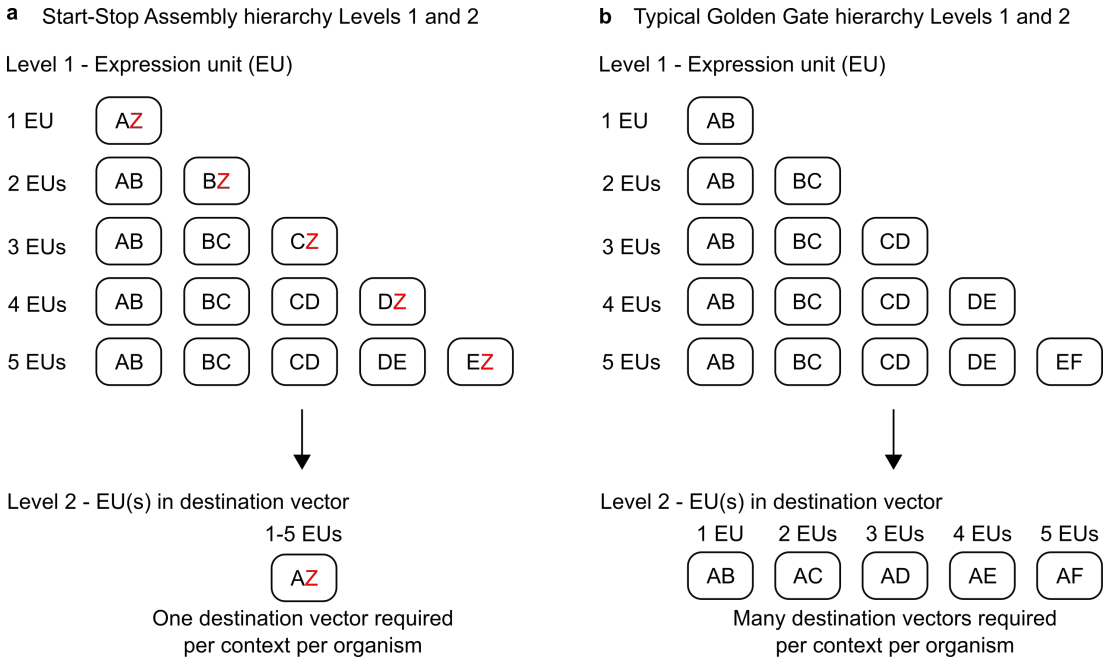


Figure 5.10. Alternative hierarchies achieved by alternative organisation of Level 1 donor and Level 2 acceptor fusion sites. Rounded rectangles represent vectors. Letters represent Level 1 donor fusion sites and Level 2 acceptor fusion sites. (a) Start-Stop Assembly uses alternative 'Z vectors' at Level 1 to minimise the number of destination vectors at Level 2. By assembling the last expression unit (of those being assembled at Level 1) in a Z vector at Level 1, in which the last donor fusion site is Z, then the recipient Level 2 vector can always use the same acceptor fusion sites, A and Z. While this requires more Level 1 vectors, it requires only one Level 2 vector. If alternative destination vectors are required for different organisms or contexts, only one new Level 2 vector needs to be constructed. (b) The typical Golden Gate framework requires multiple Level 2 vectors per context to accommodate all the possible Level 1 fusion site configurations. Therefore if an alternative destination vector is required several new vectors will need to be constructed per context or organism.

When combinatorially assembling pathway, the resultant pathway library is made up of a large number of pathway clones and therefore a high assembly efficiency is important to ensure as many clones as possible are obtained in the assembly reaction. In order to achieve this, the Level 2 assembly was limited to a maximum of five Level 1 parts (expression units), however efficient assemblies have been performed up to eight parts (165). In instances when more than five expression units are required, a third level (Level 3) in Start-Stop Assembly was designed, which allows up to 15 expression units to be assembled in a single construct (Figure 5.11). To assemble constructs in Level 3 vectors, the Level 2 vectors were designed with donor fusion sites (referred to by numbers 1, 2, 3 or 4), which allows constructs in Level 2 to be digested and used in a Level 3 assembly. To accept the Level 2 construct, the Level 3 vectors contain different combinations of the corresponding acceptor fusion sites. To implement this, multiple Level 2 vectors are required (Figure 5.11, Table 5.1). Despite

needing multiple Level 2 vectors to implement Level 3, the advantage of the streamlined hierarchy described above (and emphasised in Figure 5.10) is not lost, because when Level 3 assembly is used, the destination vector will be at Level 3, and Level 2 will serve as an intermediate level, not a destination level, so the Level 2 vectors already provided here can be used. Therefore, it is still only necessary for the user to construct one or possibly two new (Level 3) destination vector(s) per new context or organism.

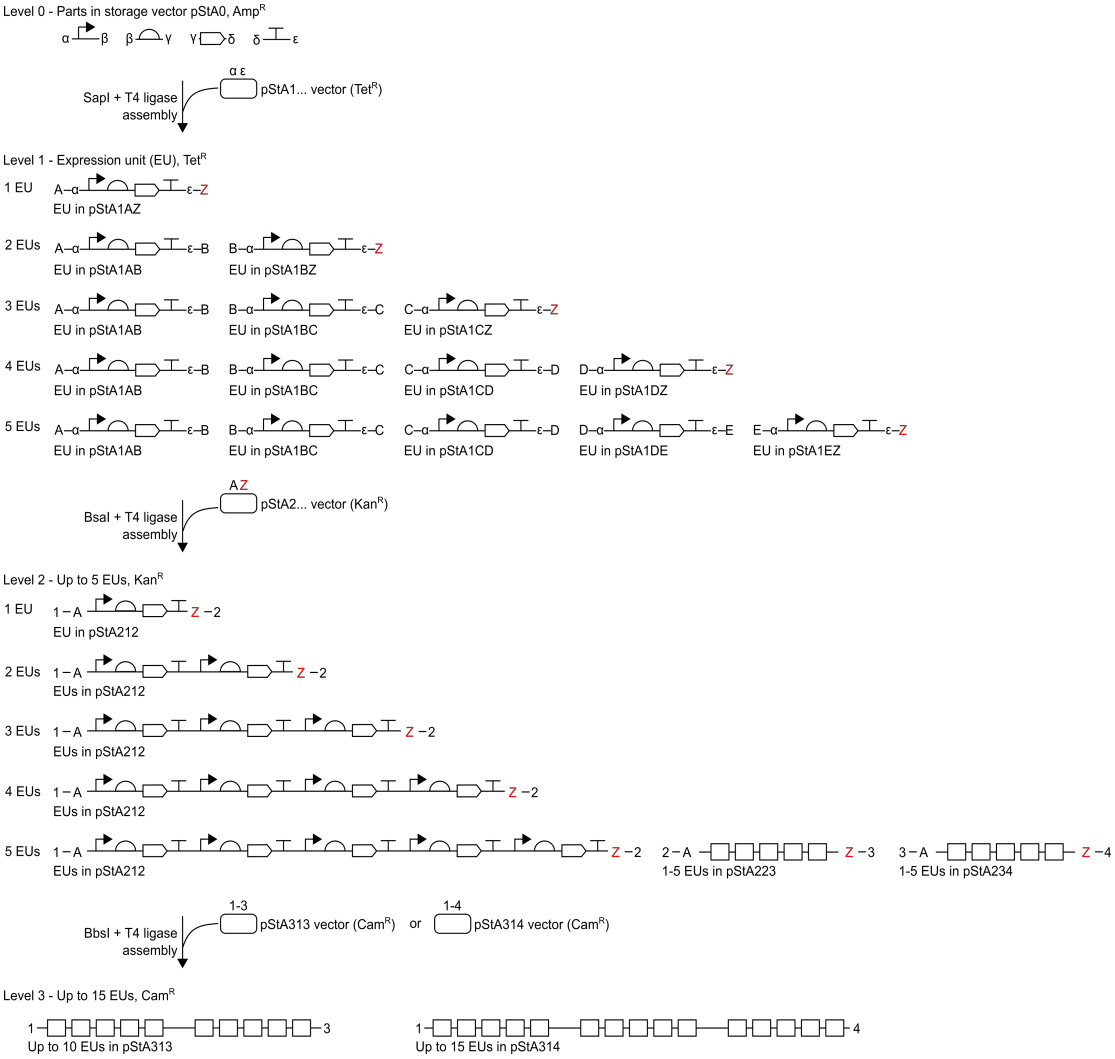


Figure 5.11. Detail of multi-part hierarchical assembly and fusion sites used in Start-Stop Assembly. Up to 15 expression units can be hierarchically assembled from basic parts (promoters, RBSs, CDSs and terminators) stored at Level 0. Level 1 expression units are assembled from four Level 0 parts, Level 2 constructs are assembled from up to five Level 1 expression units, and Level 3 constructs are assembled from up to three Level 2 constructs. Different type IIS restriction endonucleases (SapI, BsaI or BbsI) and different antibiotic-resistance markers (Amp^R, Tet^R, Kan^R or Cam^R) are used for each assembly level as shown. Acceptor fusion sites are shown inside donor fusion sites at Level 1 and Level 2. Only donor fusion sites are shown at Level 0, and only acceptor fusion sites are shown at Level 3. As described in the text, alternative sets of vectors are used at Level 1 depending upon the number of expression units being assembled, such that the first and last Level 1 donor fusion sites are always A and Z, respectively, which are always the Level 2 acceptor fusion sites. Multiple Level 2 vectors (pStA212, pStA223, pStA234) with different donor fusion sites are only needed if the assembly will be continued to Level 3, in order to assemble more than five expression units. Alternative Level 3 vectors (pStA313, pStA314) with different acceptor fusion sites allow for assembly of two or three Level 2 constructs.

5.2.5 Implementation of a complete Start-Stop Assembly system

With the functionally-scarless assembly of expression unit at Level 1 (Figure 5.2a) and the streamlined overall hierarchy (Figure 5.2b, Figure 5.11) established, the next step was to decide all of the remaining design details, so that Start-Stop Assembly could be implemented as a complete modular DNA assembly system with a widely-applicable set of 15 core vectors (Figure 5.2b, Figure 5.11).

Firstly, the specifics of the Level 2 assembly and Level 3 assembly were defined. Fusion sites used in the Level 2 and Level 3 assembly are located outside of the expression unit, in regions that are less sensitive to scar sequences. Therefore, unlike for the assembly of an expression unit in Level 1, it was not necessary to apply detailed design constraints like those described earlier for the Level 1 architecture and assembly. Rather, type IIS restriction endonucleases and fusion site sequences for Level 2 assembly and Level 3 assembly were chosen from previously reported and validated Golden Gate-based methods. For the Level 2 assembly, the type IIS enzyme BsaI was chosen and for the Level 3 assembly BbsI, both of which are frequently used in Golden Gate-based methods (165, 493). Both BsaI and BbsI cleave cohesive DNA ends with 4 bp overhangs that can be used as 4 bp fusion sites. The six fusion sites used in the Level 2 assembly were assigned 4 bp sequences A (GGAG), B (AATG), C (AGGT), D (GCTT), E (CGCT) and Z (TACT), as were the four fusion sites used in the Level 3 assembly 1 (TGCC), 2 (ACTA), 3 (TTAC) and 4 (CGAG). All of these sequences have previously been shown to provide high-fidelity assembly (165).

Start-Stop Assembly uses 15 core vectors, each of which contain a similar assembly cassette. Each assembly cassette contains a *lacZα* gene that can be used for blue/white screening in alpha-complementing *E. coli* strains when the *lacZα* gene is replaced by an assembled construct. Flanking the *lacZα* gene are the acceptor fusion sites, with outward-facing recognition sites for the corresponding type IIS restriction endonuclease, positioned appropriately to ensure cohesive DNA ends correspond to the fusion sites. Outside the acceptor fusion sites are the relevant donor fusion sites, with inward-facing recognition sites for the corresponding type IIS restriction endonuclease (except in Level 3 vectors, which lack donor fusion sites) positioned appropriately to ensure cohesive DNA ends correspond to the fusion sites. To transcriptionally insulate parts stored in pStA0 and assembled constructs from the rest of the vector, each of the assembly cassettes are flanked by the strong, *rho*-independent transcriptional terminators, T0 of phage lambda and T1 of the *rrnB* operon

of *E. coli* (494), in an attempt to prevent transcriptional 'read-in' and/or 'read-out' which might interfere with the function of either assembled sequences or the vector.

The core set of 15 vectors were constructed with the assembly cassettes and fusion sites in the specific arrangements corresponding to their level (Table 5.1 and Figure 5.11). The vectors at each level in the assembly hierarchy encode a different antibiotic resistance to prevent vector carry-over from one level to the next during assembly, in a similar vein to other multi-level assembly systems (165, 166, 493). The Level 0 vector encodes ampicillin resistance (Amp^R), the Level 1 vectors encode tetracycline resistance (Tet^R), the Level 2 vectors encode kanamycin resistance (Kan^R) and the Level 3 vectors encode chloramphenicol resistance (Cam^R). In an attempt to lessen the burden on cells caused by complete, functional expression units, the Level 1, Level 2 and Level 3 vectors were designed to use the low copy-number replicon p15A (10-12 copies per cell (161)). Using low copy-number replicons should reduce deleterious effects on the cell caused by the assembled intermediary and/or final constructs, which in turn should improve the assembly fidelity and reduce assembly bias in combinatorial assemblies. The Level 0 storage vector pStA0 was designed with the high copy-number replicon pMB1 (500 – 700 copies per cell); a low-copy number replicon was not necessary as pStA0 does not express functional expression units. The 15 core vectors were each derived from either pUC19, pACYC177 or pACYC184 (see plasmid construction section **2.8.4.1 – 2.8.4.4**), depending on the required combination of the antibiotic resistance marker and the replicon. Any SapI, BsaI and BbsI restriction sites were removed from any positions in the vectors other than those that were required for the assembly. To use Start-Stop Assembly in a different destination context (such as organisms or integration sites), alternative Level 2 or Level 3 destination vectors can easily be cloning by assembling suitable combinations of restriction sites and fusion sites (typically by transferring the entire assembly cassette from an existing Level 2 or Level 3 vector; Figure 5.12) into an alternative vector backbone and removing any other BsaI and/or BbsI sites.

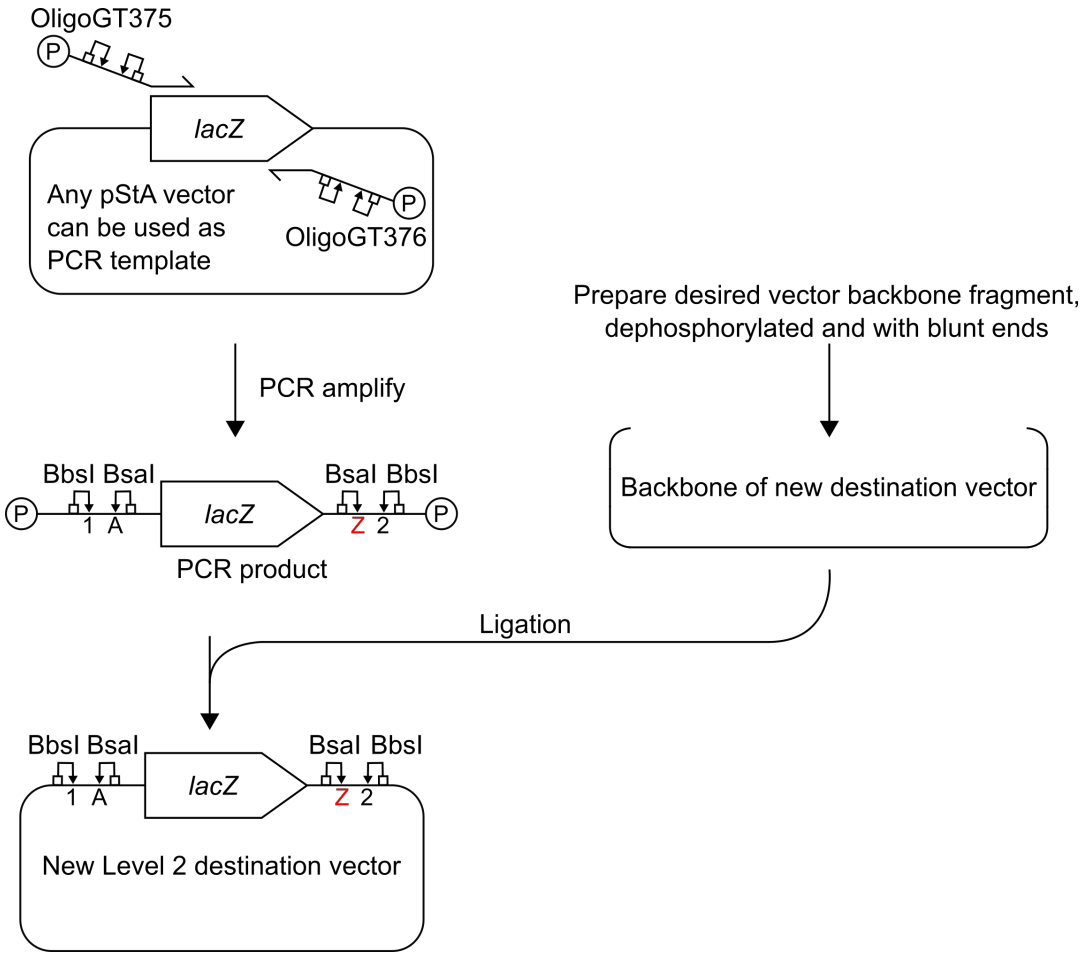


Figure 5.12. Strategy for construction of an alternative Level 2 destination vector. Prepare a linear, blunt-ended, dephosphorylated backbone fragment for the desired destination vector, for example by PCR-amplification. PCR-amplify the lacZα gene with primers oligoGT375 and oligoGT376 using any of the vectors in Table 1 as template. The primer tails contain outward-facing BsaI recognition sites, with corresponding A and Z acceptor fusion sites, and inward-facing BbsI recognition sites, with corresponding 1 and 2 donor fusion sites. This arrangement is the same as the assembly cassette of pStA212. Purify the PCR product and ligate it with the backbone vector fragment. Ligation of blunt-ended DNA fragments is not orientation-specific. Transform a suitable cloning strain of E. coli with the ligation product and use blue/white screening to identify blue colonies containing the desired plasmid with the assembly cassette. Verify the plasmid by DNA sequencing. BsaI sites and/or BbsI sites in the vector backbone should be removed.

Start-Stop Assembly can be used to assemble expression units in independent monocistronic, operons and/or hybrid configurations. Independent monocistronic expression units allow greater independence of expression units compared to operons and furthermore, both the promoter and RBS can be varied in monocistronic designs which provides more of a fine-grained control over expression compared to operons. Operon configurations can be advantageous in scenarios, such as: i) allowing the co-regulation of multiple expression units by a single promoter, typically found in nature (495, 496); ii) partially link the expression levels of multiple units in a combinatorial

assembly (115, 142, 149). To assemble operon configurations using the Start-Stop Assembly framework as described above, short spacer sequences were generated that could be used in place of promoters and/or terminators in the assembly of an expression unit at Level 1. These spacers can therefore be used to generate expression units at Level 1 that can be used at the start of an operon (with a spacer in place of a terminator), at the middle of an operon (with spacers in place of both promoter and terminator), and at the end of an operon (with a spacer in place of a promoter). Using the R2oDNA designer software (497), a set of 16 spacers (Table Table A7), each 12 bp in length and designed to be orthogonal and biologically-neutral, were generated. As the spacers were short in length (12 bp), it was possible to use these spacer parts as linkers by annealing partially complementary pairs of oligonucleotides to obtain double-stranded linkers with cohesive ends corresponding to fusion sites (Table A7). For each spacer sequence, two such linkers were designed, the first with α and β cohesive ends suitable for Level 1 assembly in place of a promoter; and the second with δ and ε cohesive ends suitable for Level 1 assembly in place of a terminator. It is possible to use these linkers directly in Level 1 assembly reactions.

5.2.6 Combinatorial Start-Stop Assembly of metabolic pathways provides exploration of design space, phenotypic diversity and high-performance pathway configurations

Start-Stop Assembly was designed to be an effective and generally-applicable platform for the implementation and optimisation of metabolic pathways. To validate this, metabolic pathways that were composed on numerous enzymes were combinatorially assembled in monocistronic, operon and hybrid configurations. The aim was that these pathway libraries would explore a wide design space that could be observed through phenotypic properties of the cell. To allow simple observational differences in the phenotypic properties caused by the combinatorial design, metabolic pathways that encode the production of carotenoids are well suited, because at several steps in the pathway different coloured carotenoids are produced. The different coloured products (red, yellow and orange) can be used as an indirect, semi-quantitative measure of a pathway's performance, allowing simple assessment of a pathways balance. Carotenoid pathways have been used for such a purpose before, with different carotenoid pathway configurations generating a huge phenotypic diversity and pathway performance (115, 148, 150). Additionally, carotenoids are valuable

compounds with useful properties, used in the pharmaceutical, food and cosmetic industries (498).

Five pathway libraries encoding carotenoid biosynthesis pathways were combinatorially assembled in *E. coli* using Start-Stop Assembly. Four of the pathway libraries (pGT531-534, Figures A1 – A4) encode the biosynthetic pathway that produces β-carotene (Figure 5.15a) by expressing the four heterologous CDSs (*crtE*, *crtB*, *crtI* and *icyB*) as well as the native *E. coli* CDS *dxs*, which has been previously reported as a rate-limiting step in the *E. coli* MEP pathway (115, 499-501). To check that different monocistronic, operon or hybrid configurations performed as expected in Start-Stop Assembly, each of the four β-carotene pathway libraries were assembled in a different configuration. The fifth pathway library (pGT535, Figure 5.13) encodes the biosynthetic pathway that produces astaxanthin (Figure 5.14a), which is an extension of the β-carotene and therefore expresses the same first five CDSs as the previous pathway libraries, as well as three additional ones (*idi*, *crtZ* and *crtW*). The eight CDSs were stored in pStA0 (Level 0 storage vector, Table A8) and were combinatorially assembled with six promoters and six RBSs and a defined terminator, which were used earlier. CDSs were combinatorially assembled at Level 1 by using equimolar mixtures of the six promoters and/or six RBSs where appropriate, which were then carried through to subsequent assembly levels through hierarchical assembly. Pathway libraries designed in an operon configuration used spacers in place of promoters and terminators where appropriate (Figure 5.13 and Figures A1 – A4). The first β-carotene pathway library (pGT531) was assembled in a monocistronic configuration in the destination vector pStA212 (Figure A1). As a result of the mixture of six promoter parts and mixture of six RBS parts used in this assembly, the total combinatorial design space (the maximum possible library size) was 6^{10} (which equals 6.05×10^7), which is the product of multiplying the number of parts (six) in the mixtures in each of ten combinatorial positions (Figure A1). In the second β-carotene pathway library (pGT532) the same five CDSs were used, but in an operon configuration, assembled in pStA223. The resultant pathway library was much smaller with a total combinatorial design space of 6^6 (which equals 4.67×10^4, Figure A2). The third β-carotene pathway library (pGT533) was constructed in pStA234 in a hybrid configuration that was composed of a three-unit operon and two monocistronic units. The hybrid pathway library had a total combinatorial design space of 6^8 (which equals 1.68×10^6, Figure A3). The fourth β-carotene pathway library (pGT534) was also assembled in a hybrid configuration, composed of a four-unit operon and one monocistronic unit, which was assembled in pStA313 with a total combinatorial design space of 6^7 (which equals

$2.80x10^5$, Figure A4). The astaxanthin pathway library (pGT535) was combinatorially assembled in pStA314 in a hybrid configuration, which was composed of a four-unit operon, a monocistronic unit, and a three-unit operon. The total combinatorial design space of the astaxanthin pathway library was 6^{11} (which equals $3.63x10^8$, Figure 5.13).

The five pathway libraries were constructed using one-pot assembly reactions at each level in the Start-Stop Assembly hierarchy. At each intermediate level the white transformant colonies (typically 99% of colonies were white and 1% were blue) were pooled together for plasmid DNA purification, which could be used as a combinatorial mixture in subsequent steps. Each transformation provided thousands and typically tens of thousands of transformant colonies. At the Level 1 assembly step this provided many-fold coverage of the design space, which was $6^2 = 36$. At the Level 2 and Level 3 assembly steps this typically provided partial coverage of the design space, which was between 6^6 (which equals $4.67x10^4$) and 6^{11} (which equals $3.63x10^8$). As such, the libraries of clones that were obtained as colonies on transformation plates at the end of Level 2 and Level 3 steps of the hierarchical assembly represent a sample of the possible design space.

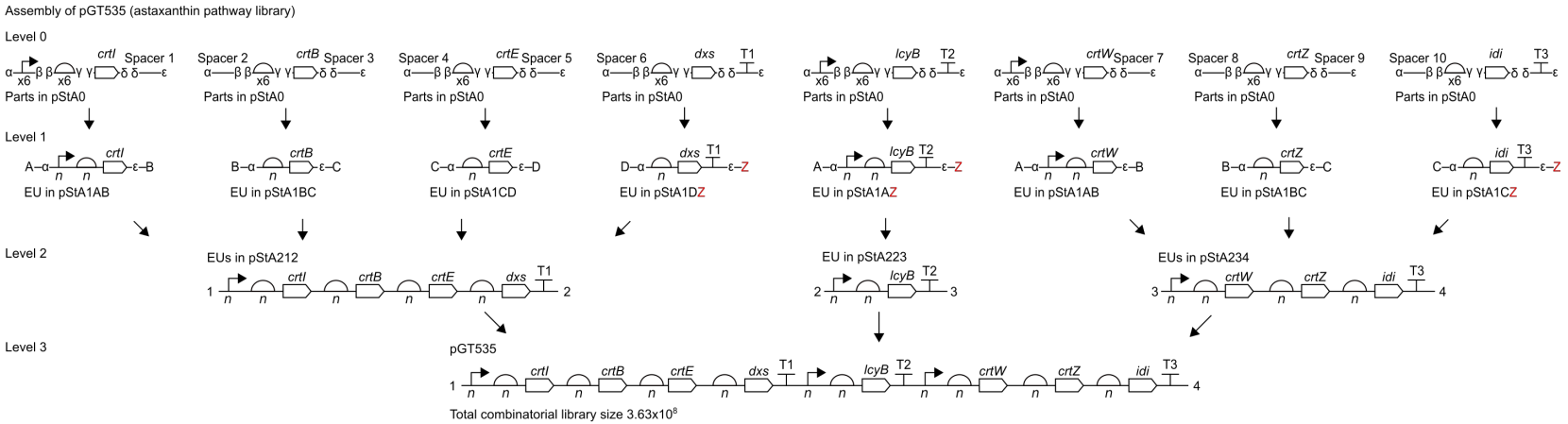


Figure 5.13. Combinatorial assembly of hybrid astaxanthin pathway library pGT535. A library of constructs each encoding an astaxanthin pathway was assembled hierarchically and combinatoirally as shown in the figure and descrined in section **5.2.6**. The destination vector was pStA314.Equimolar mixtures of six promoter parts and six RBS parts (Figure 5.5) labelled as 'x6' were used in Level 1 assemblies where shown. Spacer sequences were used in Level 1 assemblies (as linkers, Appendix Table A7) in place or promoters and/or terminators where shown. The uncertain representation of each promoter and RBS at each postition following assembly is represented by '*n*'. The total combinatorial design space was 6^{11} = $3.6x10^8$. T1 = terminator L3S2P55, T2 = terminator L3S2P21, T3 = terminator ECK120033737.

The five combinatorial pathway libraries were designed to vary the expression of each enzyme in the carotenoid biosynthetic pathways. This should produce a variety of different *E. coli* phenotypes with observably different colours and intensities that result from different compositions of the coloured carotenoids produced in the pathway (Figure 5.14 and 5.15). Different pathway clones will also contribute to differing levels of metabolic burden on the cell, which in turn affects cellular growth and will result in different colony sizes from the combinatorial assembly. The differences in both the colour and size of colonies expressing pathway clones was visually apparent. Images of plates of transformant colonies were analysed to measure these effects more objectively (Figure 5.14 and 5.15). In comparison to the controls (the empty vector pStA314 and an isolated clone pGT536 from the astaxanthin pathway library pGT535), the distribution of colony sizes in the astaxanthin pathway library pGT535 was shifted towards much smaller colonies and the colonies showed a wide distribution of colours which were not observed in the controls (Figure 5.14c). It is worth noting that the differences in colony colours were not associated with colony size (linear regression R^2 values <0.2, Figure 5.14c) and therefore colonies of similar colours but of very different sizes were observed. Similar results were obtained from the four different β-carotene pathway libraries, but with some differences in the colony size distribution and the colony colour distribution (Figure 5.15). This was presumably caused by the different designs of the pathway libraries. Finally, as a simple measure of the effectiveness of the pathway variants obtained and also to place the observed variation in the context of previously-reported carotenoid biosynthesis, ten clones were randomly chosen from the astaxanthin pathway library and they were analysed for their production of lycopene, the first coloured carotenoid in the pathway (Figure 5.16a). In simple, small-scale batch cultures that were grown over 24 hours, different lycopene concentrations were observed from 0 mg/g DCW to 108 mg/g DCW, which is a similar range to previous comparable studies (150, 203, 204). This indicates that the variants obtained represent effective pathway implementations.

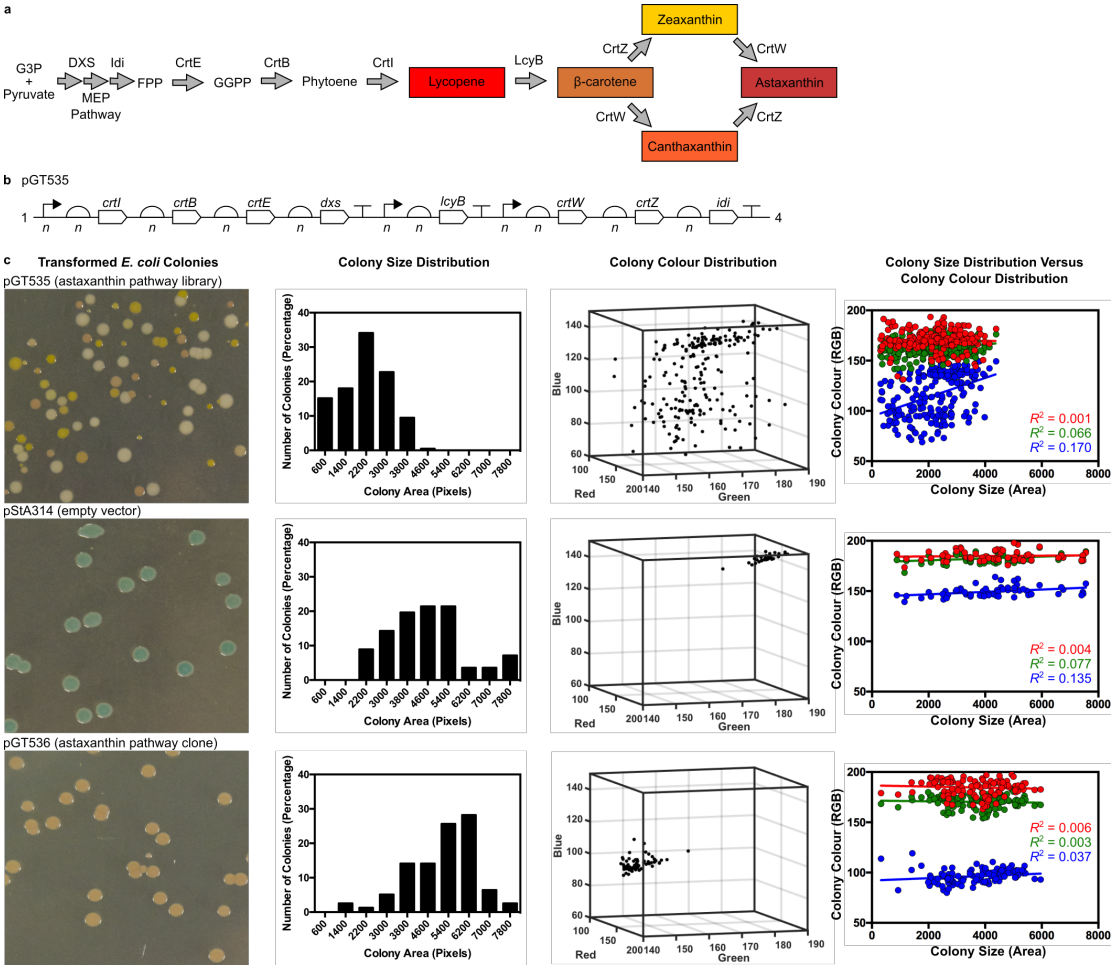


Figure 5.14. Combinatorial assembly of astaxanthin pathway using Start-Stop Assembly. (a) Astaxanthin pathway showing the endogenous enzymes of the *E. coli* MEP pathway and heterologous enzymes. Coloured products are shown in boxes, enzymes included in the combinatorial assembly are shown above arrows. Abbreviations: G3P (glyceraldehyde 3-phosphate), FPP (Farnesyl pyrophosphate), GGPP (Geranylgeranyl pyrophosphate). (b) Design of astaxanthin pathway library pGT535, composed of eight expression units combinatorially and hierarchically assembled into Level 3 vector pStA314, as outlined in the text and shown in detail in Figure 5.13. Mixtures of six parts of varying strengths were used at each promoter and RBS position shown, as described in the text. These are shown as '*n*' in the figure to reflect the sample of the combinatorial design space present after hierarchical assembly. (c) Phenotypic variation among *E. coli* clones of the astaxanthin pathway library (pGT535) was compared to controls pStA314 (empty vector) and an isolated clone pGT536 from the astaxanthin pathway library pGT535. Phenotypic variation is shown using representative images of colonies, histograms of colony size (measured as colony area in pixels, *x*-axis values represent the upper limit of each histogram bin) and the distributions of colony colours (represented using red, green and blue values extracted from colony images). For each colony, the three colour values are plotted separately in the corresponding colour. Linear regressions for each R/G/B colour are also shown in the corresponding colour (solid lines and R^2 values). There is little or no association between colony area and colony colour, reflected by low R^2 values.

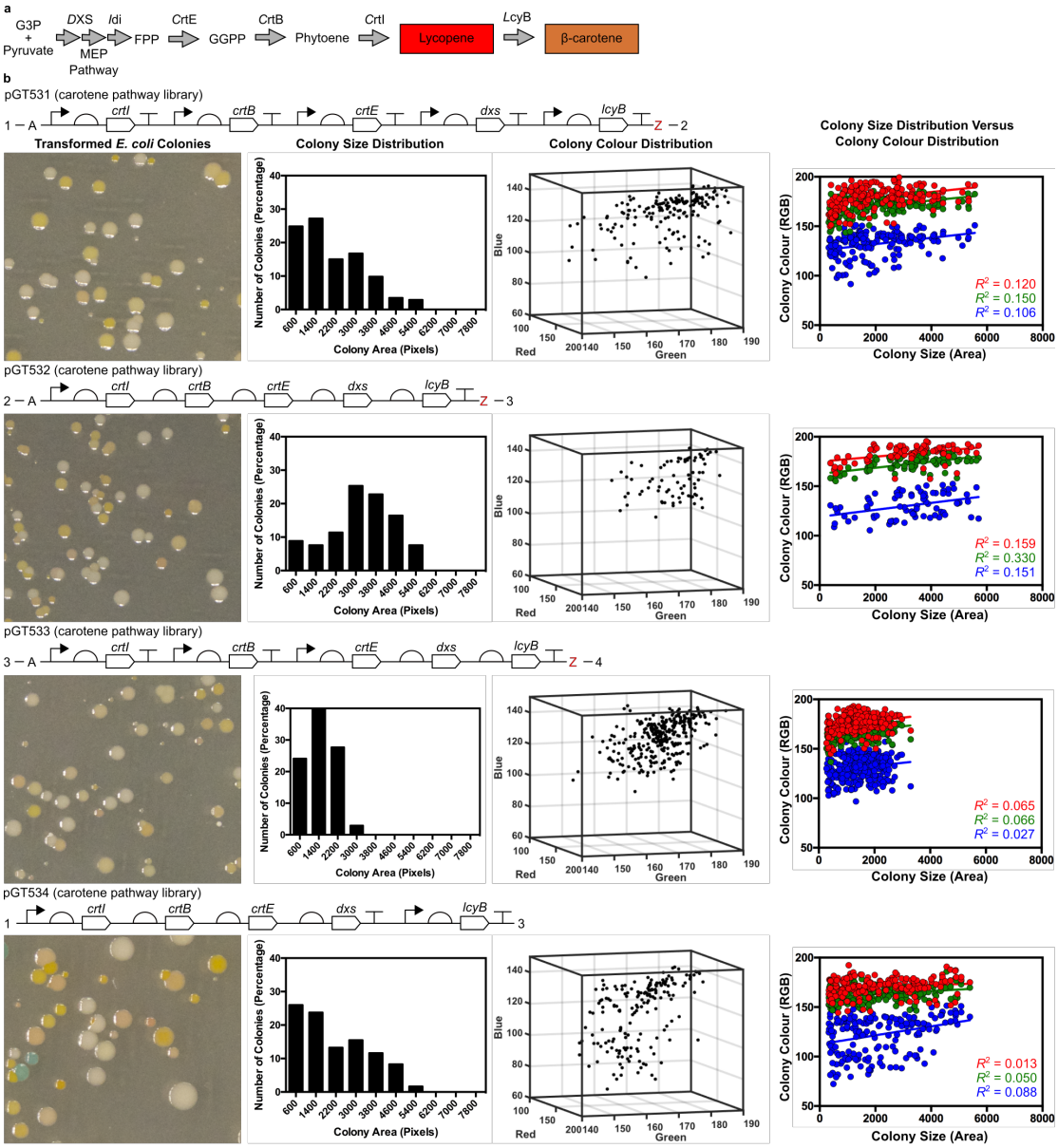


Figure 5.15. Analysis of the four β-carotene pathway libraries. (a) β-carotene pathway showing the endogenous enzymes of the *E. coli* MEP pathway and heterologous enzymes. Coloured products are shown in boxes, enzymes are shown above arrows. Abbreviations: G3P (glyceraldehyde 3-phosphate), FPP (Farnesyl pyrophosphate), GGPP (Geranylgeranyl pyrophosphate). (b) Designs of β-carotene pathway libraries (pGT531-534) and phenotypic variation among *E. coli* clones from those libraries. Variation was compared between the four β-carotene combinatorial pathway libraries (pGT531-534) and the control plasmids, pStA314 and pGT536 (data for the controls is shown in Figure 5.14c). Phenotypic variation is shown using representative pictures of the transformation plates, histograms of colony size (measured as cell area, *x*-axis represents the upper limit of each histogram bin) and the distributions of colony colours (measured using colony red, green and blue values extracted from colony images). For each colony, the three colour values are plotted separately in the corresponding colour. Linear regressions for each R/G/B colour are also shown in the corresponding colour (solid lines and R^2 values). There is little or no association between colony area and colony colour, reflected by low R^2 values.

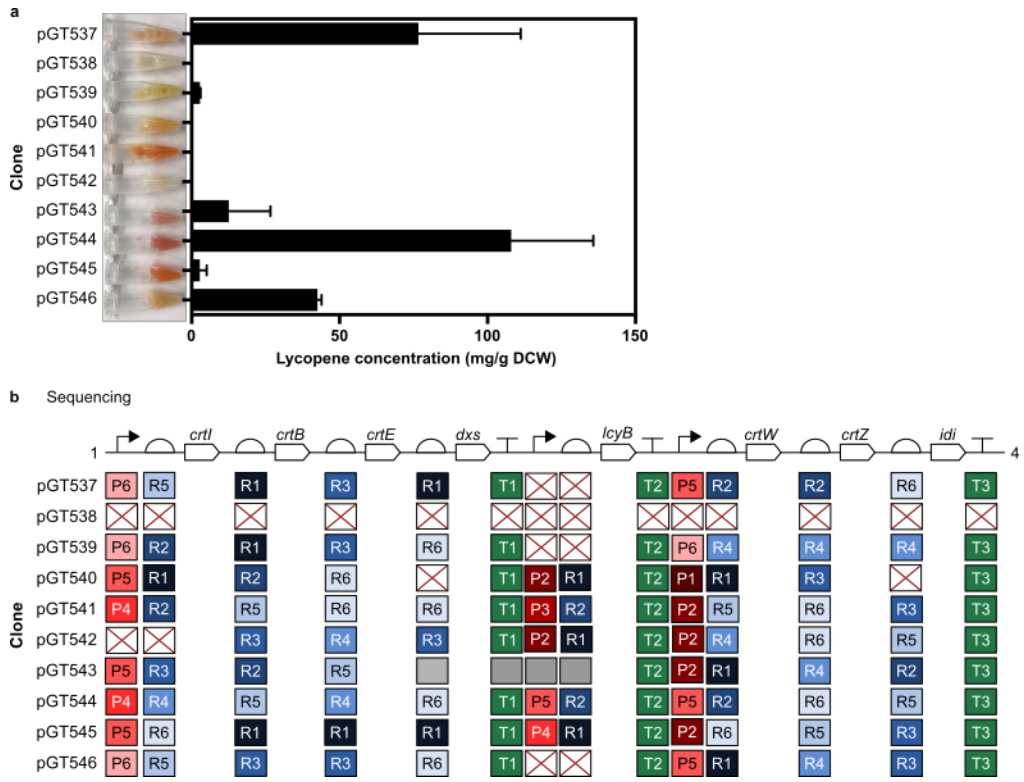


Figure 5.16. Lycopene content of *E. coli* clones from the astaxanthin library. (a) Ten *E. coli* colonies were randomly selected from the combinatorially-assembled astaxanthin pathway library pGT535 (Figure 5, Figure 5.13). Each clone was grown for 24 h in LB broth before lycopene content was determined as mg lycopene per dry cell weight (DCW). Error bars represent the standard deviation of three independent biological replicates. A photograph of a sample of the culture of each clone was taken (shown to left of graph), which shows the visible variation of colour in the ten cultures, attributed to their different profiles of the carotenoids produced in the astaxanthin pathway. (b) Sequencing data for the ten clones. Promoters, in descending order of strength: P4 = J23107, P5 = J23116. RBSs, in descending order of strength: R2 = RBSc33, R4 = RBSc58, R5 = RBSc42, R6 = RBSc36. Terminator: T1 = L3S2P55, T2 = L3S2P21, T3 = ECK120033737. Boxes with a red cross indicate pathway deletion. Grey boxes indicate unobtainable sequence data, which presumably indicate a mutation/ deletion.

The ten pathway clones that were analysed for their production of lycopene were then sequenced (Figure 5.16b). A surprisingly large number of deletions had occurred in six of the ten pathway clones. Analysing the sequences of the six clones with incomplete pathways indicated that none of the deletions were a result of a misassembly. As these deletions were not caused by misassemblies, they were presumably a consequence of biological selection, where incomplete pathways could have resulted from the burden that the pathway placed on the cell, which in turn could have led to an enrichment of incomplete pathway clones in the pathway library.

5.3 Discussion

The overall goal of this chapter was to develop a suitable multi-part DNA assembly method that would serve as the framework for implementing rapid prototyping as well as other combinatorial-based approaches. Both modular and bespoke DNA assembly methods impose constraints on the design and assembly of multi-part constructs. Modular assembly methods introduce scars at junctions, whereas bespoke assembly methods have a low efficiency and issues with bias and repetitive sequences. So far when using such methods, it has been necessary to operate within the constraints imposed by an assembly method. However, where possible it is favourable to avoid placing unnecessary constraints on the design of constructs that encode metabolic pathways and other systems. In this chapter, Start-Stop Assembly was developed as a multi-part DNA assembly method that offers the same key advantages of Golden Gate-based modular assembly frameworks; specifically, the efficient, unbiased, multi-part and hierarchical assembly. However, Start-Stop Assembly also allows CDSs to be assembled without introducing scars at the boundaries. The CDS boundaries are highly-sensitive sites, where scars affect mRNA structure and the activity of the RBS (as shown by the direct comparison of the impact of scars Figure 5.1) and potentially other functional RNA features and therefore scarless assembly at these boundaries is crucial. The scars imposed by other modular DNA assembly methods such as those shown in Figure 5.1 represent unavoidable design constraints. The second significant advantage of Start-Stop Assembly is the streamlined hierarchy, which requires only one new vector to be constructed when using a new destination context (Figure 5.10), whereas the traditional Golden Gate-based hierarchies required substantial work to create the necessary vectors for different destination contexts. As a result, the streamlined hierarchy of Start-Stop Assembly should facilitate more rapid and convenient development of engineered metabolic pathways for diverse non-model organisms in order to exploit their industrial potential.

The one notable disadvantage of using the start and stop codons as the fusion sites is the incompatibility of assembling fusion proteins from domains and/or tags, as this requires fusion sites which do not start or stop translation. However, metabolic engineering does not typically require construction of fusion proteins and if they are required, individual fusion proteins can be constructed in advance of the assembly and incorporated into Start-Stop Assembly as a Level 0 CDS part.

To date, SapI has been used once before in a Golden Gate-based method, which was developed to assemble a single expression composed of two parts (promoter and

CDS) in *Agrobacterium* based destination vectors (492). While the start codon ATG was used as a fusion site as one of the SapI fusion sites, the stop codon was not. This method was limited to assembling one expression unit from two parts, whereas Start-Stop Assembly offers the hierarchical assembly of up to 15 expression units from up to 60 individual parts, which is far better suited to the needs of metabolic engineering. There has been one description of a scarless and modular DNA assembly method similar to Start-Stop Assembly, termed ‘scarless stitching’ (149). This assembly method uses Golden Gate-based restriction-ligation and fusion sites, but it is limited to assembling only two parts in each step due to an additional blunt restriction-ligation reaction used to delete 'bridging' (scar) sequences. Hierarchical assembly of metabolic pathways using scarless stitching would require many more steps in series and in parallel than multi-part methods like Start-Stop Assembly.

In this chapter, Start-Stop Assembly has been focused towards constructing prokaryotic expression constructs. However by simply cloning suitable parts in the Start-Stop Assembly format and constructing a suitable destination vector it would be possible to use the same framework of fusion sites, hierarchy, storage vector and assembly vectors to construct eukaryotic expression constructs.

In this chapter, Start-Stop Assembly was evaluated as a platform for the implementation and optimisation of metabolic pathways. To test this, five carotenoid pathway libraries were assembled and analysed. The evaluation of Start-Stop Assembly demonstrated that: i) it is suitable for the wide exploration of design spaces, as demonstrated by the phenotypic diversity of colony colour and colony size; ii) the novel combinations and arrangements of fusion sites had no evident issues, as effective carotenoid pathway variants were readily generated; and iii) the core 15 vectors (Table 5.1) functioned as designed, having each been used during the assembly of carotenoid pathway libraries. The aim of these single-round experiments was not to maximise production, but to validate Start-Stop Assembly. However despite the small-scale experimental approach taken here, the results are promising when compared to other reports (150, 203, 204). The lack of association observed between the colony colour and the colony size (Figures 5.14c and 5.15c) indicates that while pathway variants can encode similarly well-functioning pathways, they can also impose differing degrees of burden on the cell, which could possibly be a result of the different effects that increased transcription and translation have on the cell (114, 502). These observations show that not only is the optimisation of metabolic pathways through the variation of their enzyme expression profiles important for obtaining

pathways that are high-performing, but also that pathways impose little burden on the cell. Furthermore, the data shows that combinatorial assembly is an ideal approach for identifying such pathways when the suitable expression control parts are available.

In this chapter, a novel multi-part DNA assembly method, Start-Stop Assembly, was developed to serve as an ideal framework for efficiently implementing rapid prototyping and combinatorial-based approaches, while imposing as few constraints as possible on the design of metabolic pathway-encoding constructs. With the necessary genetic tools (Chapter Three and Four) and framework (Chapter Five) now developed, it was possible to implement and evaluate rapid prototyping as a novel design principle for non-model organisms (Chapter Six).

5.4 Key Outcomes

- A functionally scarless, modular multi-part DNA assembly method was developed, called Start-Stop Assembly.
- Start-Stop Assembly is well suited for implementing combinatorial approaches to pathway design.
- Start-Stop Assembly is easily applicable to any new destination context.

6. CHAPTER SIX. Implementing rapid prototyping

6.1 Introduction

Cyanobacteria have the ability to directly utilise sunlight and CO_2 as their sole energy and carbon sources, respectively, making them promising candidates for sustainably producing products of value without the need for expensive feedstocks or creating competition with our food production. To date, cyanobacteria have been engineered to produce a range of products (263), but generally at low productivities and titres (as discussed in section **1.5**). To produce such products, many different strategies have been used, such as knocking out genes to enhance the production of native products (316), or introducing anywhere from one (503) to five genes (291) to produce non-native products. When a heterologous or synthetic metabolic pathway is introduced to an organism, there are different design principles that can be used to design the pathway. Such rationales that are used in model organisms were discussed in section **1.3**. In cyanobacteria, the design principles that are used to engineer heterologous or synthetic metabolic pathways lags far behind that of model organisms.

The current state-of-the-art design principle used to engineer metabolic pathways in non-model organisms remains very simple, using strong native or synthetic constitutive promoters, inducible promoters and occasionally defined RBSs, which are typically arranged as operons to overexpress an individual pathway, somewhat reminiscent of the design strategies used in the pre-synthetic biology era for model organisms. This design principle most likely results from the work that suggests that strong promoters are required to redirect carbon flux towards product formation, rather than towards biomass (302). This approach can still work well to express a pathway, for example, the single strong native *Synechocystis* promoter, P_{rbc}, was used to express pyruvate decarboxylase and endogenous alcohol dehydrogenase, which resulted in *Synechocystis* producing 5.5 gl^{-1} of ethanol (284). It is worth noting three iterations of the DBTL cycle was carried out to generate this well-performing pathway. Another successful example uses the inducible promoter, P_{llacO1}, to overexpress the two enzymes acetolactate synthase and 2-acetolactate decarboxylase to produce 2.38 gl^{-1} 2,3-butanediol (297). These two examples are very notable success stories directly converting CO_2 to a target product at a high titre, which used simple designs of two genes configured as an operon and overexpressed by a strong promoter, for the metabolic pathway-encoding constructs. However, using strong promoters does not always result in a well-functioning pathway. For example, 0.00017 gl^{-1} of isoprene was produced in *Synechocystis* when the single native P_{psbA2} was used to express the

pathway (504). Previously, it has been stated that the strong native and synthetic cyanobacterial promoters aren't strong enough to express the pathway enzymes at a level sufficient enough to shift a signification proportion of the carbon metabolism in *Synechocystis* to the product of interest. Therefore, stronger expression control parts are required to achieve better carbon partitioning (302).

When engineering the optimisation of productivities and titres of products in *Synechocystis*, there are factors other than the protein expression levels of pathway proteins that need to be considered. Metabolic control analysis has shown that limitations in the metabolic flux towards the product of interest may result from factors other than the enzyme expression levels in the pathway (379). Introducing modifications elsewhere in the metabolism of *Synechocystis* rather than further elevating the expression levels of the introduced exogenous pathway may prove more effective in increasing productivity (505). For example, by increasing the substrate availability by either i) eliminating competing pathways and/or ii) increasing the expression of enzymes that are anticipated to lead to higher concentrations of the substrate; and/or by increasing the thermodynamic force driving the reaction by adjusting cofactor specificity and/or availability. Furthermore, production rates of a product of interest have been shown to gradually change over the course of an experiment when factors such as nutrients and light limit growth and/or production (301). Furthermore, when *Synechocystis* is grown in limiting conditions, carbon flux can bypass the key metabolite pyruvate (a substrate/precursor of many pathways), immediately entering the TCA cycle (301).

Recently, more advanced design principles have been used to improve metabolic pathway performance. For example, a recent study identified optimal configurations of native and synthetic promoters with synthetic RBSs to express a thioesterase, phosphopantetheinyl transferase and carboxylic acid reductase, which resulted in the production of 0.905 gl^{-1} 1-octanol in *Synechocystis* when cultures were fed with octanoic acid (300). Furthermore, two synthetic promoter libraries and two synthetic RBS libraries have recently been generated (352, 383), one of which has been subsequently used to fine tune expression of ethylene-forming enzyme (383) and improve the ethylene production by 2.5-fold over the previously reported best cyanobacterial strain.

Despite these new approaches, engineering metabolic pathways in cyanobacteria remains limited by three predominant factors: i) their slow-growth rate, which limits the

ability to iteratively optimise the performance of metabolic pathways; ii) a lack of expression control parts for fine-tuning protein expression levels in a metabolic pathway (as discussed in section **3.1**); and iii) their genetic instability, whereby anywhere between a single nucleotide mutation to a complete deletion of the metabolic pathway can randomly occur, which results in the loss of pathway functionality (introduced in section **1.5.1.5**). As a result, implementing iterative and combinatorial approaches to metabolic pathway design has been a huge challenge in these organisms and as such, they have hardly been implemented. More often than not an individual metabolic pathway is constructed and performs poorly, but the optimisation process is too complicated considering the factors stated above. It is worth stating that unless multiple pathway variants are tested, it is not possible to ascertain exactly how optimal the performance of a pathway would be.

The current state-of-the-art design principles, outlined above, use individual pathway designs, which, when teamed with the limited ability to predictively design genetic pathways and also with the limited knowledge of what the protein concentrations should be in those pathways, lead to designing genetic pathways without a specification. This often leads to poor pathway performance and requires many iterations of the DBTL cycle to obtain an adequate level of performance. Combinatorial approaches have become an increasingly popular strategy for pathway design in model organisms (115, 149, 150), however I am not aware of any examples of its use in cyanobacteria. The advantages of using a combinatorial approach to design a pathway were previously introduced in section **1.3.3.2.3**. Briefly, combinatorial approaches provide an easy way to search large expression spaces without *a priori* knowledge of the required enzyme levels. They do however, require suitable expression control parts (such as promoters and RBSs) that span an appropriate range of strengths, a suitable modular multi-part DNA assembly method and an organism with a reasonably high transformation efficiency. It is most likely that each of these three factors have stalled the use of combinatorial approaches in non-model organisms, as many have few available genetic parts, low transformations efficiencies and modular DNA assembly methods have typically required large amounts of vector construction work to re-implement for any alternative organisms.

To date, these state-of-the-art design principles have used ‘conventional’ promoters and RBSs, which only consider the performance in the target non-model organism, unlike portable parts that have been developed in this project, which consider the performance in both the model and non-model organism. Rapid prototyping is a novel

design principle designed here to overcome some of the challenges associated with engineering non-model organisms, principally slow-growth rates and poor transformations efficiencies. Rapid prototyping aims to overcome these issues by optimising pathways in a fast-growing model organism with a high transformation efficiency, such as *E. coli*, which makes it easier and faster to implement combinatorial and iterative approaches. By using portable parts (promoters and RBSs) to the extent that the functionality of a pathway depends on the expression profile of the enzymes that result from activity of the portable parts, functionality should be similar in both non-model and model organisms, which therefore circumvents the need to further optimise the pathway in the non-model organism. In this chapter, the portable parts (Chapter Three and Four) and Start-Stop Assembly (Chapter Five) are combined to test whether rapid prototyping is a better pathway design principle for engineering metabolic pathways in non-model organisms, compared to the current state-of-the-art pathway design principles which are typically used.

It is of course possible that a combinatorially assembled pathway library could be transformed into a model organism with a high transformation efficiency and then that certain pathways are selected in *E. coli* and transferred to the non-model organism. This is of course the premise for rapid prototyping. However, the key difference in these approaches is that rapid prototyping uses portable parts rather than conventional parts. When conventional parts are used, the behaviour of the pathway variants in the model organism have little or no bearing on their performance in the non-model. However, by using portable parts, it should be possible to identify well performing pathways in the model organism, which perform somewhat similarly in the non-model organism and would therefore allow predictive pathway optimisation in the model organism.

6.1.1 Aims

The goal of this chapter was to determine whether implementing combinatorial and/or rapid prototyping-based approaches to engineering metabolic pathways in non-model organisms are better than the current state-of-the-art design principles that are currently implemented. To assess this, constructs encoding the lycopene pathway were engineered using different design principles that represent the current state-of-the-art design principles as well as combinatorial and rapid prototyping-based design principles. The expectation was that it would be easier to identify functional and high producing pathway variants in *Synechocystis* when portable and curated subsets were

used. This is because functional and high producing strains can be predictively identified in *E. coli* by using the portable parts or the curated subset of portable parts, rather than by using a subset of conventional parts. The aims of this chapter were:

- To pick a subset of 'conventional' promoters and RBSs from the libraries generated in Chapter Three, for comparison.
- To use the conventional, portable and curated subsets to assemble individual and combinatorial lycopene pathways.
- To carry out a prescreen of lycopene concentrations in *E. coli* from the conventional, portable, curated pathway libraries and chose clones based on their performance to take forward for characterisation in *Synechocystis*.
- To transform and segregate chosen lycopene pathways in *Synechocystis*.
- To characterise performance of lycopene pathways in both *E. coli* and *Synechocystis*.

6.2 Results

6.2.1 Choosing conventional parts

To implement the current state-of-the-art design strategies which are currently used for non-model organisms in *Synechocystis*, the first step was to choose a set of promoters and RBSs that would typically be used, which are referred to here as conventional parts. Conventional parts (promoters and RBSs) are different to the portable parts. While portable parts consider performance in both *E. coli* and *Synechocystis* and were chosen because of their similar performance between the two organisms, conventional parts are chosen with consideration only for the performance in *Synechocystis*. A set of conventional promoters and RBSs were chosen from the initial libraries that were previously generated in Chapter Three. Six promoters and six RBSs were chosen, which would provide 36 expression levels, the same number as the portable subset provides. To choose the six conventional promoters, only the *Synechocystis* characterisation data was considered. The strongest and weakest promoter clones were chosen and then four clones between them, which had an equal spacing of expression levels between them were chosen (Figure 6.1a). The chosen promoters were SPLc115, SPLc45, SPLc3, SPLc47, SPLc19 and SPLc25 (Table 6.1). This process was repeated to select six conventional RBSs from the RBS library using only the *Synechocystis* characterisation data (Figure 6.1b). The chosen RBSs were RBSc249, RBSc48, RBSc123, RBSc110, RBSc21 and RBSc4 (Table 6.1).

Table 6.1. Conventional promoters and RBSs.

Conventional Parts		Sequence (5' … 3')
Promoter	c3	CCAGATGACCTTGACAAGTAACTGCCCCAACAGTATAATATACCT
	c19	GCGGCTCGTATTGACAATTTAACTAAATATCAGTATAATTTAGTT
	c25	CCACGAAAAATTGACAAGCACTCTGCGCTCTGATATAATTAAGGT
	c45	CTCGCCACCTTTGACACAGCGGCCCGCTAACTATATAATAAACGT
	c47	GTGACGTCTATTGACAGCCAGGAGTTACCGAGATATAATGGAATA
	c115	CCGGACCCCCTTGACACTACGGCCCGCTATGGATATAATTCACCG
RBS	c4	TATGTCAGTAAAAAAAAAGGAGGTTCGTT
	c21	TACGAAAGATTTACGAAAGGAGGTTGGTT
	c48	AGTGTGTTCCAAGAGAAAGGAGGTATCTG
	c110	TAGGTATATGCAATCAAAGGAGGTATATG
	c123	TGTGTAGGTCCCACCAAAGGAGGTCTGTA
	c249	GGCGCGAGTCAACGTTCCTGAGGAGTGGC

Having chosen the subset of conventional promoters and RBSs by only considering the performance of the parts in *Synechocystis*, the performance of the subset was then compared between *E. coli* and *Synechocystis* (Figure 6.2). The subset of conventional promoters provided a wide range of strengths in *Synechocystis*, yet did not provide a wide range of strengths in *E. coli* relative to the library (Figure 6.2a). The subset of promoters also showed no correlation between the two organisms (R^2 = 0.003) and a Pearson product-moment correlation showed that there was no linear relationship in the performance of the conventional subset of promoters between the two organisms (r = 0.06, n = 6, p = 0.91). The conventional subset of RBSs provided wide range of strengths in both *E. coli* and *Synechocystis* (Figure 6.2b) and a moderate correlation between *E. coli* and *Synechocystis* (R^2 = 0.63). A Pearson product-moment correlation showed there was a strong linear relationship in the performance of the conventional subset of RBSs between the two organisms (r = 0.79, n = 6, p = 0.06). Comparing the performance of the conventional promoters and RBSs to the performance of the portable subset (r = 0.941, n = 12, p = <0.0001), the conventional subset (particularly the promoters) had a much weaker correlation in its performance between the two organisms. This is not surprising, as the portable subset was chosen based on their performance between the two organisms, whereas the conventional parts were not.

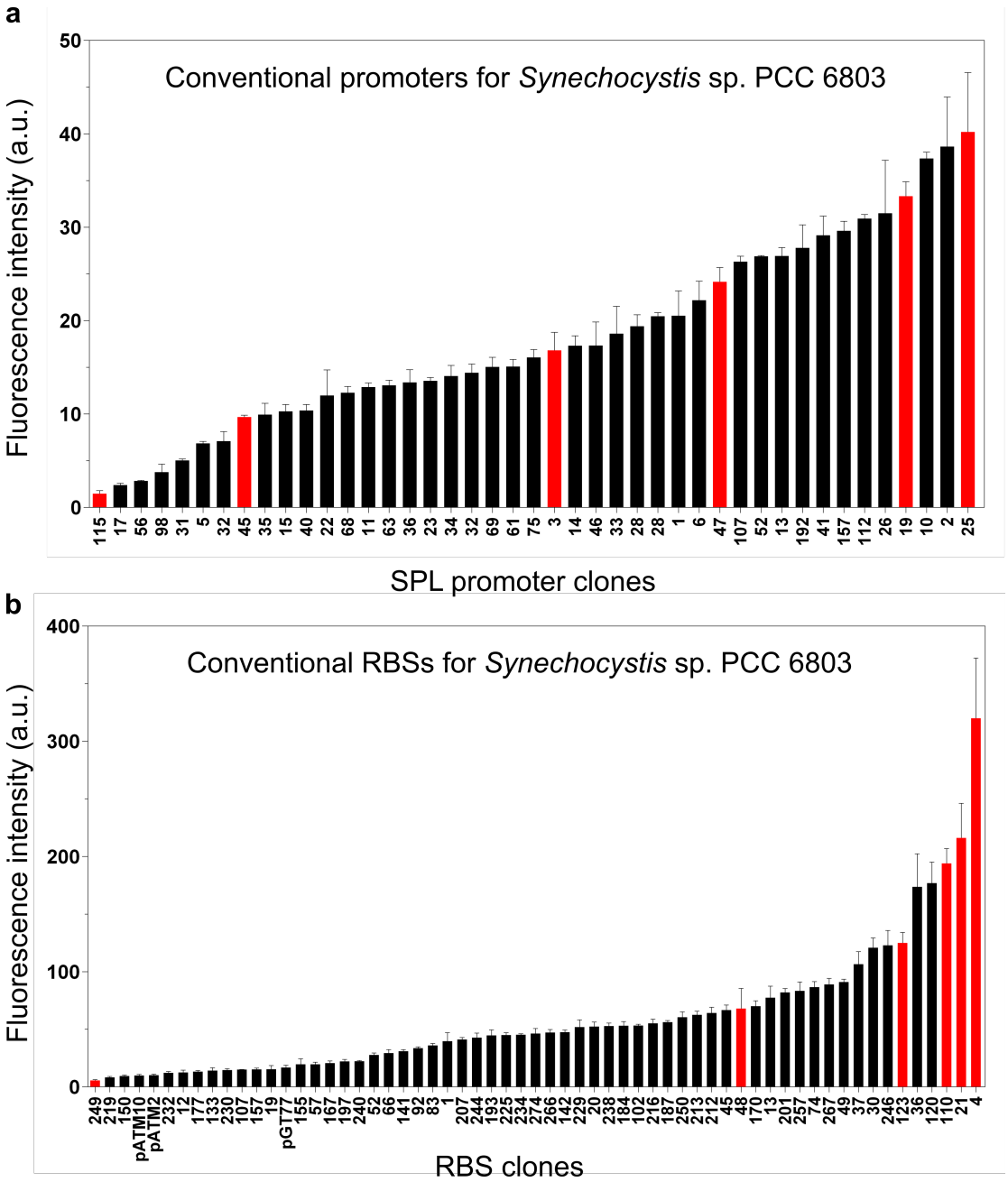


Figure 6.1. Conventional Parts selected in *Synechocystis* sp. PCC 6803. a) The experiment is described in the main text (section **3.2.2.4**) and Figure 5. Six conventional promoters were selected that gave a wide and evenly-spaced distribution of lycopene production (shown in red here). b) The experiment is described in main text (section **3.2.4.5**) and Figure 21. Six conventional RBSs that gave a wide and evenly-spaced distribution of lycopene production (shown in red here).

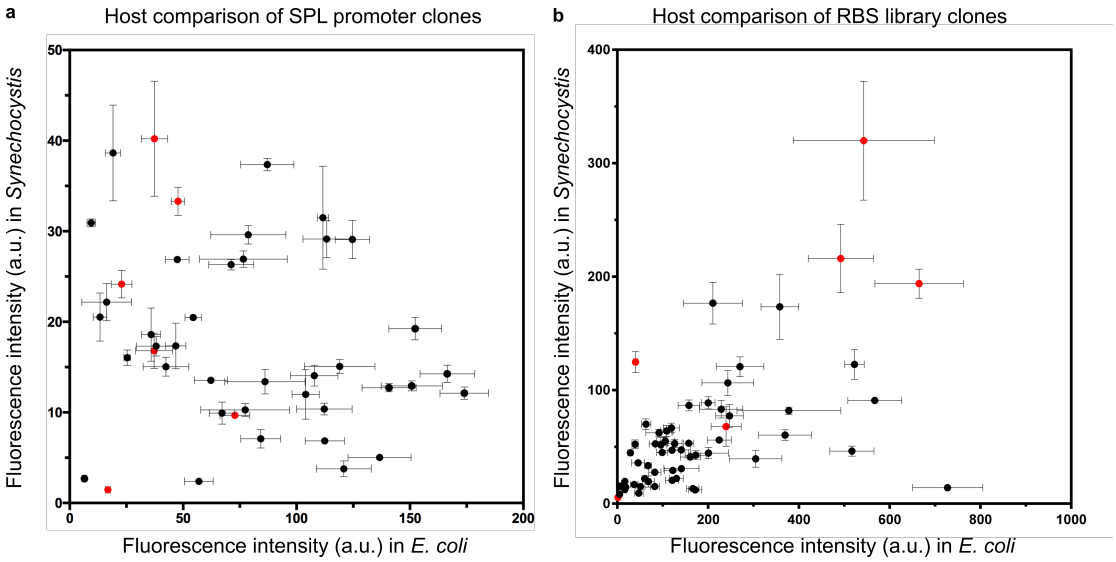


Figure 6.2. Subsets of conventional promoters and RBSs shown in the host comparisons of the SPL and RBS libraries. (a) The figure is the same as host comparison of final Promoter library between *E. coli* and *Synechocystis* as shown in Figure 3.20, but here the subset of conventional promoters are coloured in **red**. (b) The figure is the same as host comparison of RBS library between *E. coli* and *Synechocystis* as shown in Figure 3.27, but here the subset of conventional RBSs are coloured in **red**.

6.2.2 Design strategies for the lycopene pathway

To determine the benefit that rapid prototyping and combinatorial approaches would offer for engineering metabolic pathways in *Synechocystis* over the current state-of-the-art design principles, five different design principles were devised, which would be used for engineering the lycopene encoding pathway in *Synechocystis*. The first of the design principles was an individual design of the lycopene pathway, which represents the current state-of-the-art for pathway design in non-model organisms (Figure 6.4). The other four design principles were combinatorial approaches, which differed in their choice of parts and also in the choice of pathway clones that were transferred from *E. coli* to *Synechocystis* (Figure 6.5). These four design principles should allow evaluation of the expected advantages of combinatorial approaches and rapid prototyping.

To compare these five design principles a suitable metabolic pathway to engineer and express in *Synechocystis* was required. Like in Chapter Five, metabolic pathways that encode the biosynthesis of carotenoids are ideal for this purpose. They produce coloured products, which offer an indirect and semi-quantitative measure of the pathway's performance. Furthermore, carotenoids are valuable compounds with useful properties used in the pharmaceutical, food and cosmetic industries (498), which to date, have not been heterologously produced in cyanobacteria.

Each of the pathway designs encode the pathway which forms the product lycopene (Figure 6.3) using four heterologous CDSs, *crtE, crtB, crtI* and *dxs*, as previously used in section **5.2.6**. *dxs* encodes an enzyme in the upstream MEP pathway which is known to be rate-limiting in *E. coli* (115, 499-501) and cyanobacteria (overexpression of DXS resulted in a 1.5-times increase in the carotenoid content compared with the wild type organism (506)). Each of the four CDSs, in each of the five pathway designs were assembled as independent monocistronic expression units. Monocistronic configurations provide greater independence of each expression unit and potentially a wider and more fine-grained control of expression, as both promoters and RBSs can be varied for each unit. This was hypothesised as important for the independent precise and portable expression of each of these genes between the two organisms. To assemble the lycopene pathways in monocistronic configurations, terminators that functioned in both *Synechocystis* and *E. coli* were required. At this point in the project, no terminators had been characterised and demonstrated as functional in both *E. coli* and *Synechocystis*. Recent work in the Heap lab identified 20 *rho*-independent terminators that are functional in both organisms (data not shown; paper in preparation). The 20 terminators were a mix of native *E. coli* terminators and synthetic terminators which had been previously described (127). From this set of 20 terminators, four were chosen that had been previously characterised in *E. coli* (127). These included one previously reported synthetic terminator named L3S2P21 (hereafter referred to as T2) (127) and three native *E. coli* terminators (127), including the bacteriophage T7 terminator ECK120033737 (hereafter referred to as T3), the terminator for the *spy* gene ECK120029600 (hereafter referred to as T4) and the terminator for the ribosomal protein operon *rplM – rpsI* ECK120015170 (hereafter referred to as T6).

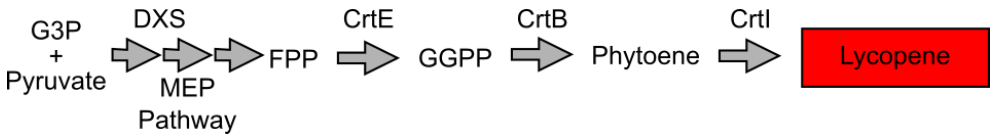


Figure 6.3. Lycopene pathway. Lycopene pathway showing the endogenous enzymes of the *E. coli* MEP pathway and heterologous enzymes. Coloured products are shown in boxes, enzymes included in the combinatorial assembly are shown above arrows. Abbreviations: G3P (glyceraldehyde 3-phosphate), FPP (Farnesyl pyrophosphate), GGPP (Geranylgeranyl pyrophosphate).

For Pathway design 1, four different constructs were designed, which were all based on the current state-of-the-art pathway design rationales for non-model organisms. The first pathway (pGT432) was designed with a classic strong over-expression rationale, with the strong conventional promoter SPLc19 and the strong conventional RBS RBSc21 driving expression of the four CDSs in the pathway (Figure 6.4). The second pathway (pGT433) was designed with a mid-strength expression rationale, with the medium strength conventional promoter SPLc3 and the medium strength conventional RBS RBSc123 upstream of the four CDSs (Figure 6.4).

The third pathway (pGT434) was designed with a rationale previously identified to reduce metabolic burden (114). This rationale used a strong promoter and a weak RBS on the basis that translation was more demanding on the cell than transcription, and that ribosomes are limiting, so for a given amount of protein expression, strong transcription and weak translation is thought to be less deleterious for the cell (as described in section **1.3.3.2.2**) (114). In this case, the strong conventional promoter SPLC19 and the weak conventional RBS RBSc48 were used (Figure 6.4). To observe the affects that 'overinitiation' of translation can have in *Synechocystis* a fourth rationale was developed, which was designed with the opposite configuration to pGT434. The fourth pathway (pGT435) used the weak conventional promoter SPLc45 and the strong conventional RBS RBSc21 (Figure 6.4).

Pathway design 2 and 3 used the six conventional promoters and the six conventional RBSs to combinatorially assemble each of the four CDSs in the lycopene pathway-encoding construct pGT547 (Figure 6.5). Conventional expression control parts are chosen with consideration only for the performance in the non-model organism. Due to the mixture of six promoter parts and the mixture of six RBS parts used, the total combinatorial design space (the maximum possible library size) was 6^8 (which equals $1.7x10^5$), which is the product of multiplying the number of parts (six) in the mixtures in each of eight combinatorial positions. The fourth pathway design (Pathway design 4) used the six portable promoters and six portable RBSs to combinatorially assemble each of the four CDSs in the lycopene pathway-encoding construct pGT548 (Figure 6.5). As for pGT547, with the mixture of six promoter parts and the mixture of six RBS parts, the total combinatorial design space was 6^8 (which equals $1.7x10^5$). The fifth pathway design (Pathway design 5) used the curated subset of portable parts (four portable promoters and four portable RBSs) to combinatorially assemble each CDS in the lycopene pathway-encoding construct pGT549. The CDS were assembled in a monocistronic configuration using the terminators outlined above (Figure 6.5). With

fewer parts, the mixture of four promoters and the mixture of four RBSs gave a total combinatorial design space of 4^8 (which equals $6.6x10^4$). Combinatorial variation was introduced by using equimolar mixtures of the promoter parts or RBS parts at each appropriate position in Level 1 assemblies, then propagated to subsequent assembly levels through hierarchical assembly.

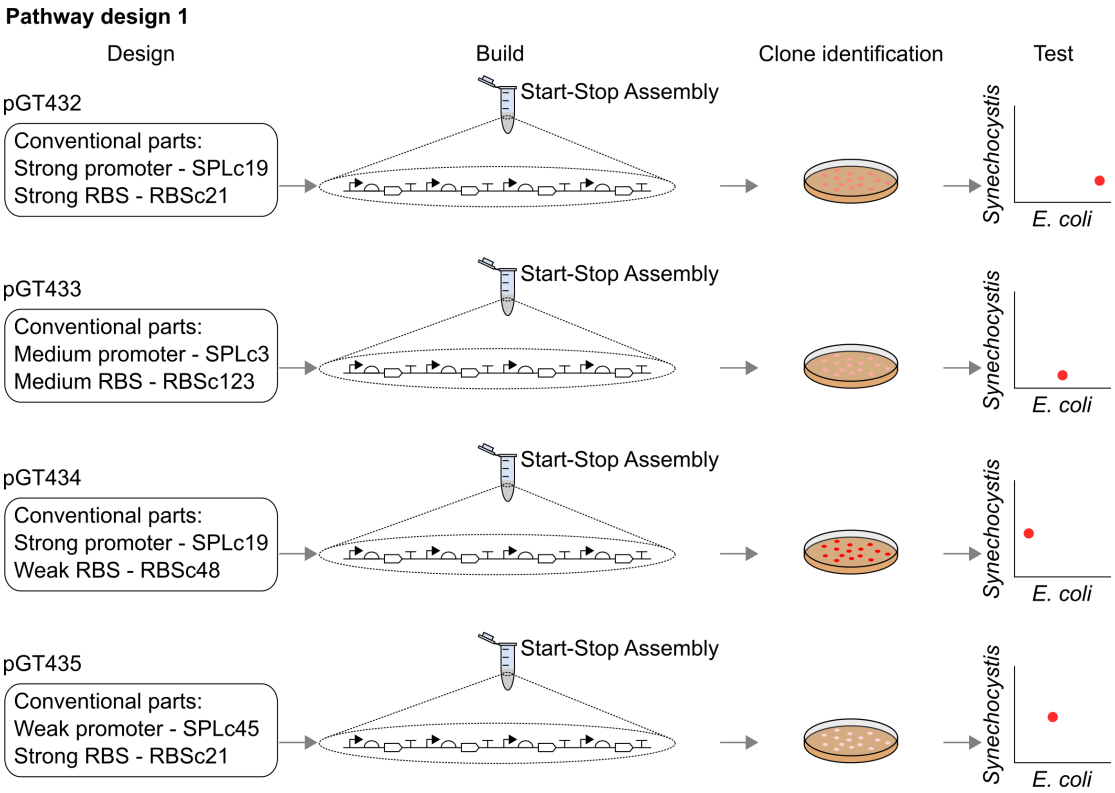


Figure 6.4. Design principle for Pathway design 1. Pathway design 1 represents the current state-of-the-art in pathway design for cyanobacteria, whereby individual constructs are assembled and tested. Four different individual constructs (pGT432 – 435) were designed, each construct was designed using a different rational. pGT432 followed the classic strong over-expression rationale. pGT433 followed a medium strength expression rationale. pGT434 followed the burden inspired design rationale, to use a strong promoter and weak RBS, pGT435 followed an opposite design using a weak promoter and strong RBS. Constructs would be assembled using Start-Stop Assembly, a pathway variant would be isolated, and the plasmid purified from an *E. coli* transformation plate. *E. coli* and *Synechocystis* would be transformed with the constructs and the resulting strains characterised.

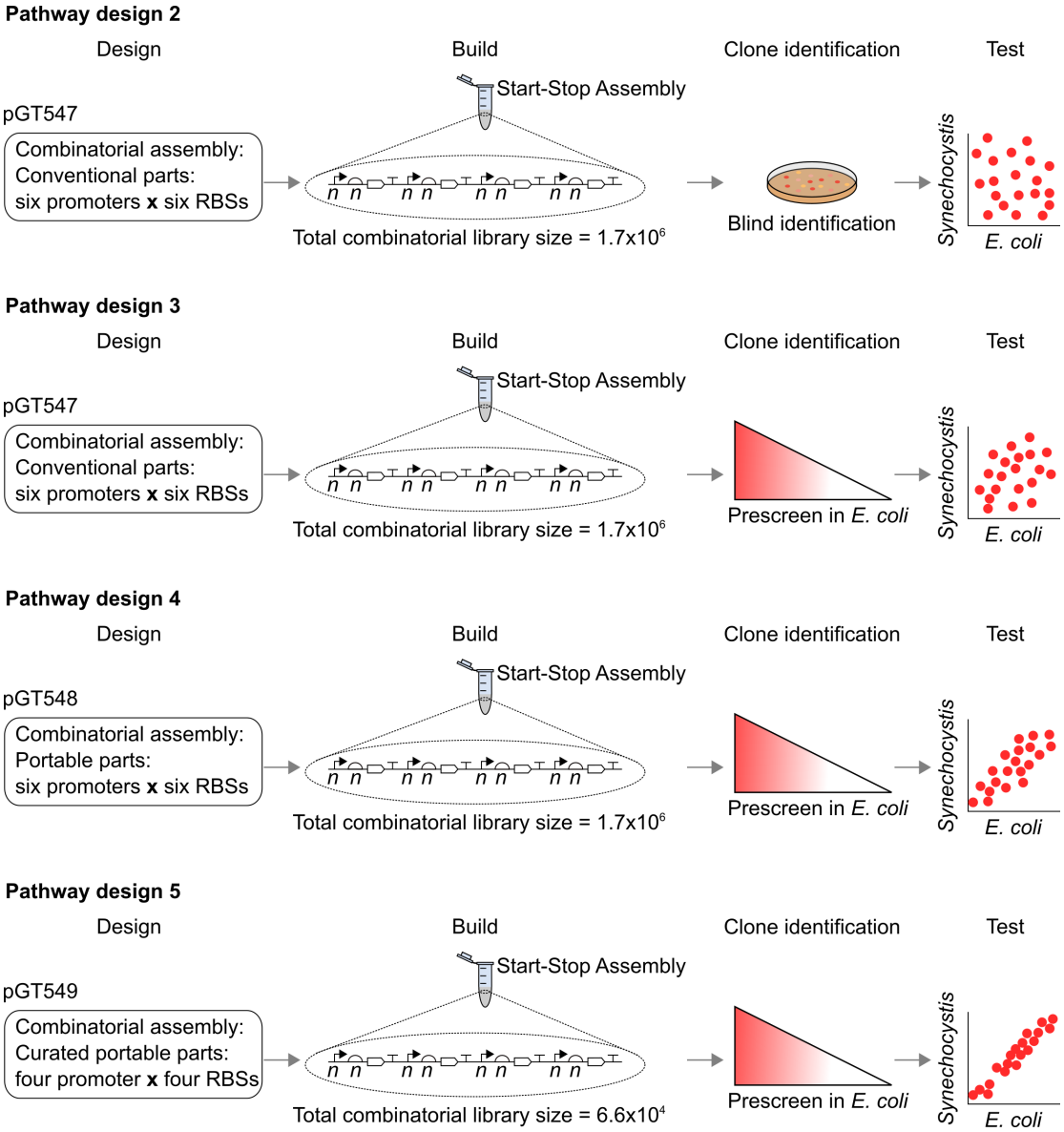


Figure 6.5. Design principles for Pathway design 2 – 5. Pathway design 2 – 5 all used a combinatorial approach to assemble the lycopene pathway-encoding construct. Pathway design 2 and 3 used the same combinatorially assembled lycopene pathway library (pGT547) that was assembled using the subset of six conventional promoters and six conventional RBSs. Lycopene pathway variants were blindly chosen in Pathway design 2. Lycopene pathway variants were chosen based on prescreen of pathway performance in *E. coli*. Pathway design 4 combinatorially assembled the lycopene pathway-encoding construct using the portable subset of six promoters and six portable RBSs (pGT548), lycopene pathway variants were then chosen based on prescreen of pathway performance in *E. coli*. Pathway design 5 combinatorially assembled the lycopene pathway-encoding construct using the curated subset subset of four promoters and four portable RBSs (pGT549), lycopene pathway variants were then chosen based on prescreen of pathway performance in *E. coli*. Lycopene pathway-encoding constructs would be assembled using Start-Stop Assembly, after which pathway variants would be isolated and the plasmids purified. *E. coli* and *Synechocystis* would be transformed with the constructs and the resulting strains characterised.

6.2.3 Assembling the lycopene pathways

To facilitate the testing of the lycopene pathways in *Synechocystis* and *E. coli*, a shuttle vector was constructed which could be used as a Start-Stop Assembly Level 2 destination vector, pGT270 (Figure 6.6) (see Materials and Methods section **2.8.4.7**). The destination vector was based on pATM2 but contained the appropriate Level 2 Start-Stop Assembly cassette (BsaI and BbsI restriction sites and the *lacZα* gene for blue/white screening) for assembling the lycopene pathways and a kanamycin-resistance gene for selection in both *E. coli* and *Synechocystis*, between the two homology arms that integrate into the genome of *Synechocystis* at the *ssl0410* locus.

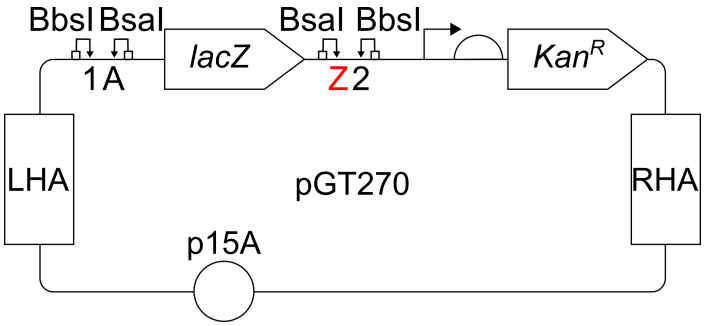


Figure 6.6. *E. coli-Synechocystis* shuttle vector pGT270 containing appropriate Level 2 Start-Stop Assembly cassette. The plasmid contains the low-copy number origin of replication p15A that allows replication of the shuttle vector in *E. coli*. Left (LHA) and right (RHA) homology arms that integrate at the *ssl0410* loci of *Synechocystis'* genome. Between the homology arms is a Level 2 Start-Stop Assembly cassette, which includes a *lacZα* cassette for blue/white screening, two outward-facing BsaI recognition sites with corresponding A and Z acceptor fusion sites for Level 2 assembly of multiple expression units and two inward-facing BbsI recognition sites with corresponding 1 and 2 donor fusion sites for Level 3 assemblies. In addition to the Level 2 Start-Stop Assembly cassette, there is a KanR cassette that encodes an aminoglycoside phosphotransferase that confers resistance to kanamycin in both *E. coli* and *Synechocystis*.

With the destination vector (pGT270) constructed, all the relevant expression control parts (promoters, RBSs, CDSs, terminators) were cloned into the Start-Stop Assembly Level 0 storage vector (pStA0) ready for use in the assembly. The four CDSs and three of the four terminators (T2, T3 and T4) were previously cloned into the Level 0 vector pStA0 (Table A8) in Chapter Five (detailed in the Materials and Methods section). The fourth terminator, T6, was cloned into pStA0 using inverse PCR and the PCR product was then circularised by ligation and product used to transform *E. coli*, from which the sequence verified plasmid (pGT424) was obtained (as described in Materials and Methods section **2.8.4.5**). The performance of the portable parts was characterised in a combinatorial assembly of the EYFP reporter in section **4.2.5.1.** To avoid introducing any additional scar sequences between the promoter and RBS, which would affect

their portable performance, the conventional and portable parts were cloned into pStA0 as single composite parts (promoter and RBS combined) using inverse PCR (as described in Materials and Methods section **2.8.4.5**). Having selected the set of conventional parts (as described in section **6.2.2**), they were cloned into pStA0 (as described in Materials and Methods section **2.8.4.5**), ready for use in assembly. It was only possible to clone 34 of the 36 conventional composite parts into pStA0. The portable composite parts that were previously chosen in Chapter Four (as described in section **4.2.5.1**) were then cloned into pStA0, however it was only possible to clone 33 of the 36 portable composite parts. It is possible that the five constructs couldn't be cloned due to reasons such as secondary structures. After multiple failed attempts at cloning each they were excluded and the end result was 34 conventional composite parts (rather than 36), 33 portable composite parts (rather than 36) and 25 curated portable composite parts (rather than 26).

With the parts cloned in the Level 0 storage vector pStA0, it was possible to assemble each of the lycopene pathways in pGT270 (Figure 6.7). In each case, the pathway was generated by hierarchical Start-Stop Assembly using one-pot assembly reactions at each step, then pooling all white colonies (typically 99% of colonies were white and 1% were blue) obtained at each intermediate step to prepare plasmid DNA for use as the combinatorial mixture of parts for the subsequent step. Each transformation yielded at least thousands of colonies and typically tens of thousands of colonies, providing many-fold coverage at the Level 1 assembly steps (at which the maximum design space was $6^2 = 36$) and typically partial coverage at the Level 2 assembly steps (at which the design space was between 4^8 (which equals 6.6×10^4) and 6^8 (which equals 1.7×10^5), so the libraries of clones obtained as colonies on plates at the end of each hierarchical assembly represent a sample of the possible design space.

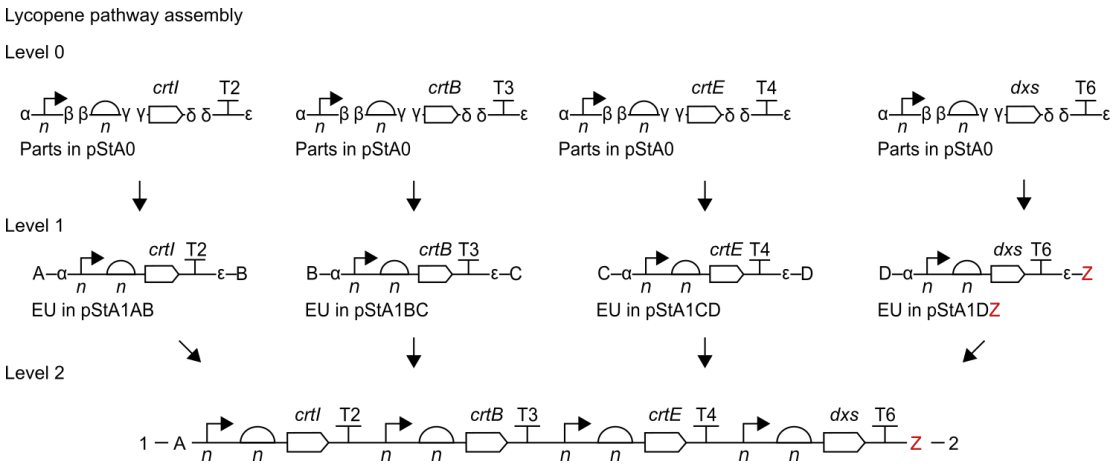


Figure 6.7. Assembly of lycopene pathways used in Pathway designs 1 – 5. Individual (Pathway design 1) and libraries (Pathway design 2 – 5) of constructs each encoding a lycopene pathway were assembled hierarchically and as needed combinatorially as shown in the figure and as described in the main text. The destination vector was pGT270. The uncertain representation of each promoter and RBS at each position following assembly is represented by '*n*'. In Pathway design 1, for pGT432 '*n*' represents the promoter SPLc19 and the RBS RBSc21. For pGT433 '*n*' represents the promoter SPLc3 and the RBS RBSc123. For pGT434 '*n*' represents the promoter SPLc19 and the RBS RBSc48. For pGT435 '*n*' represents the promoter SPLc45 and the RBS RBSc21. For Pathway design 2 and 3, the pathway library pGT547 was combinatorially assembled, where '*n*' represents the subset of six conventional promoters and six conventional RBSs. For Pathway design 4, the pathway library pGT548 was combinatorially assembled, where '*n*' represents the subset of six portable promoters and six portable RBSs. For Pathway design 5, the pathway library pGT549 was combinatorially assembled, where '*n*' represents the curated subset of four portable promoters and four portable RBSs. In instances of combinatorial assembly, equimolar mixtures of six promoter parts or six RBS parts were used in Level 1 assemblies. T2 = terminator L3S2P21, T3 = terminator ECK120033737, T4 = terminator ECK120019600 and T6 = ECK120015170.

After three attempts at assembling pGT434 (the third of the individual pathway designs), no colonies were obtained on the transformation plates and therefore the design was discarded. Pools of the non-blue colonies from the Level 2 transformation plates of the three other individual pathways, pGT432, 434 and 435, were obtained to prepare plasmid DNA for each of the three pathways, which could be used to transform *Synechocystis*.

The pathway clones from Pathway design 2 were blindly identified from the Level 2 *E. coli* transformation plate of the pGT547 pathway library. To achieve an unbiased blind identification, the twenty colonies nearest to a randomly chosen point on the edge of the plate were selected (pGT437 – 456). The single colonies were grown and their plasmids were purified, ready for transforming *Synechocystis*.

6.2.4 Pre-screen of lycopene pathways in *E. coli*

Having assembled the three combinatorial lycopene pathway libraries, pGT547 – 549, the next step was to prescreen a large number of clones from each pathway library in *E. coli* (for Pathway designs 3 – 5) before identifying a subset to transfer to *Synechocystis*. By doing this for the each of the three pathway libraries (Pathway design 3 – 5), it would be possible to compare the number of pathway clones that perform similarly between the two organisms and therefore it would be possible to determine whether using portable parts increases the number of well performing pathways in *Synechocystis*.

To prescreen the lycopene pathway clones from the three combinatorial pathway libraries in *E. coli*, 96 single colonies were selected from the *E. coli* transformation plates of each of the respective pathway libraries. For each pathway library, 96 clones were chosen that offered a wide range of lycopene concentrations, achieved by visual inspection of the reddish colour of the colonies. The lycopene concentration was then determined for the 288 (3 x 96) pathway clones using a high-throughput spectrophotometer-based screening method. The A503 nm absorbance then served as an indirect measure of lycopene concentration. In simple, small-scale batch cultures grown over 48 h a wide range of different lycopene concentrations from each of the three pathway libraries was observed (Figure 6.8 – 6.10). The ranges of lycopene concentrations produced by the three different pathway designs was very similar, with A503 nm values ranging between 0.0 and approximately 1.7. From this data, the ten top-producing clones from each of the pathway libraries and the ten clones that were evenly spaced from the tenth strongest to the weakest clone were chosen (Figure 6.8 – 6.10). In addition, five of the reddest colonies were also selected from the transformation plates of the three pathway libraries, pGT547-549. The plasmids from the chosen pathway clones were then purified for the *Synechocystis* transformation.

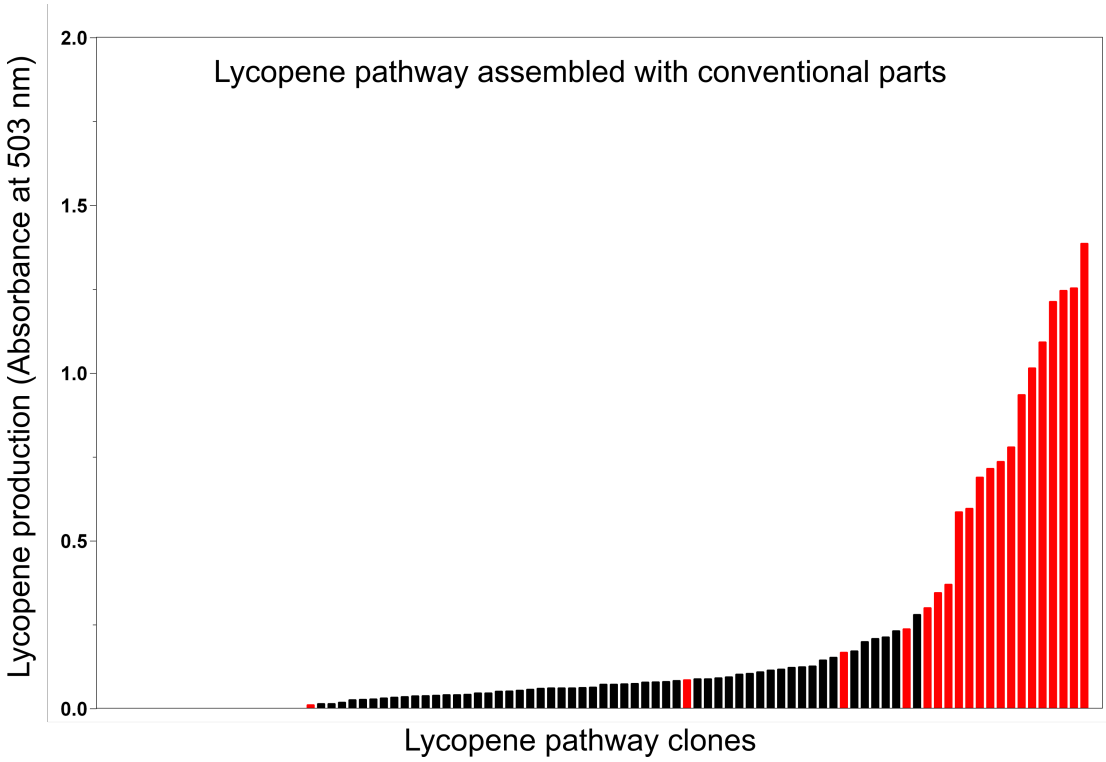


Figure 6.8. Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the conventional parts. 96 *E. coli* transformant colonies were selected and characterised after 48 h of growth by spectrophotometry (as described in Materials and Methods - Quantification of lycopene content of *E. coli* – Spectrophotometer (section **2.7.9**)). The absorbance at 503 nm of each clone is plotted with the background absorbance subtracted. From this library, 20 pathway clones were chosen, ten that produced the most lycopene and ten that gave a wide and evenly-spaced distribution of lycopene production (shown in red here). The lycopene pathway clone IDs can be found in Appendix Note A1.

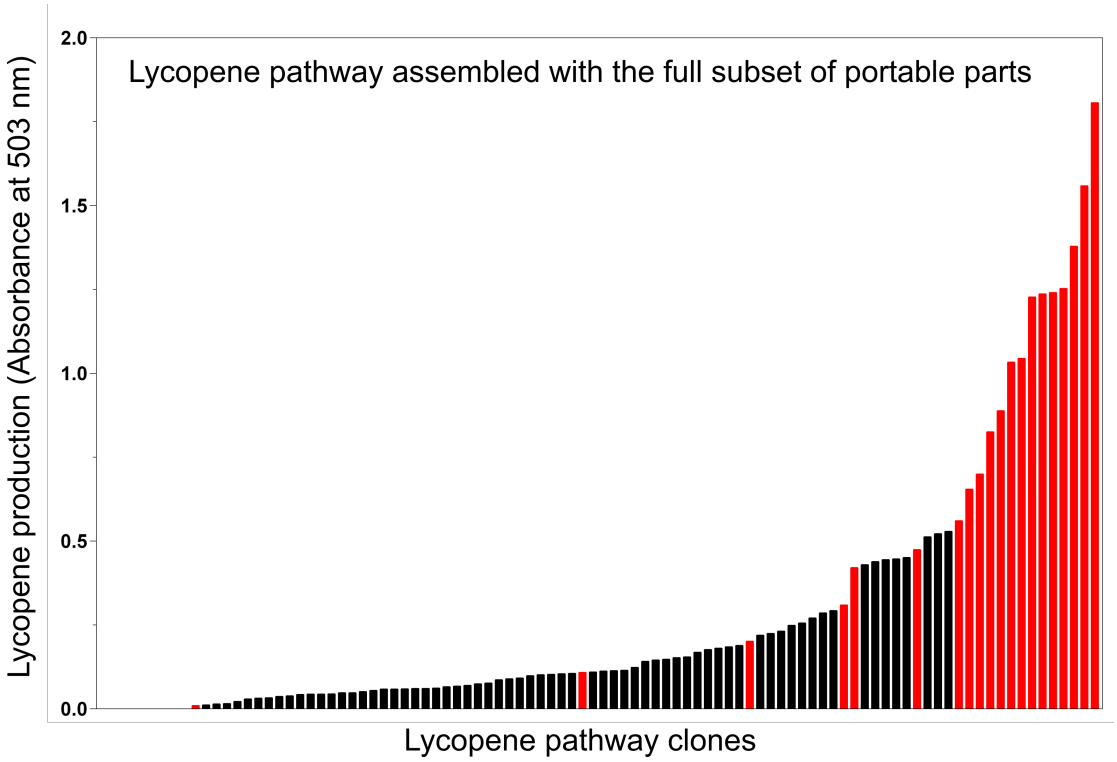


Figure 6.9. Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the full subset of portable parts. 96 *E. coli* transformant colonies were selected and characterised after 48 h of growth by spectrophotometry (as described in Materials and Methods - Quantification of lycopene content of *E. coli* – Spectrophotometer (section **2.7.9**)). The absorbance at 503 nm of each clone is plotted with the background absorbance subtracted. From this library, 20 pathway clones were chosen, ten that produced the most lycopene and ten that gave a wide and evenly-spaced distribution of lycopene production (shown in red here). The lycopene pathway clone IDs can be found in Appendix Note A2.

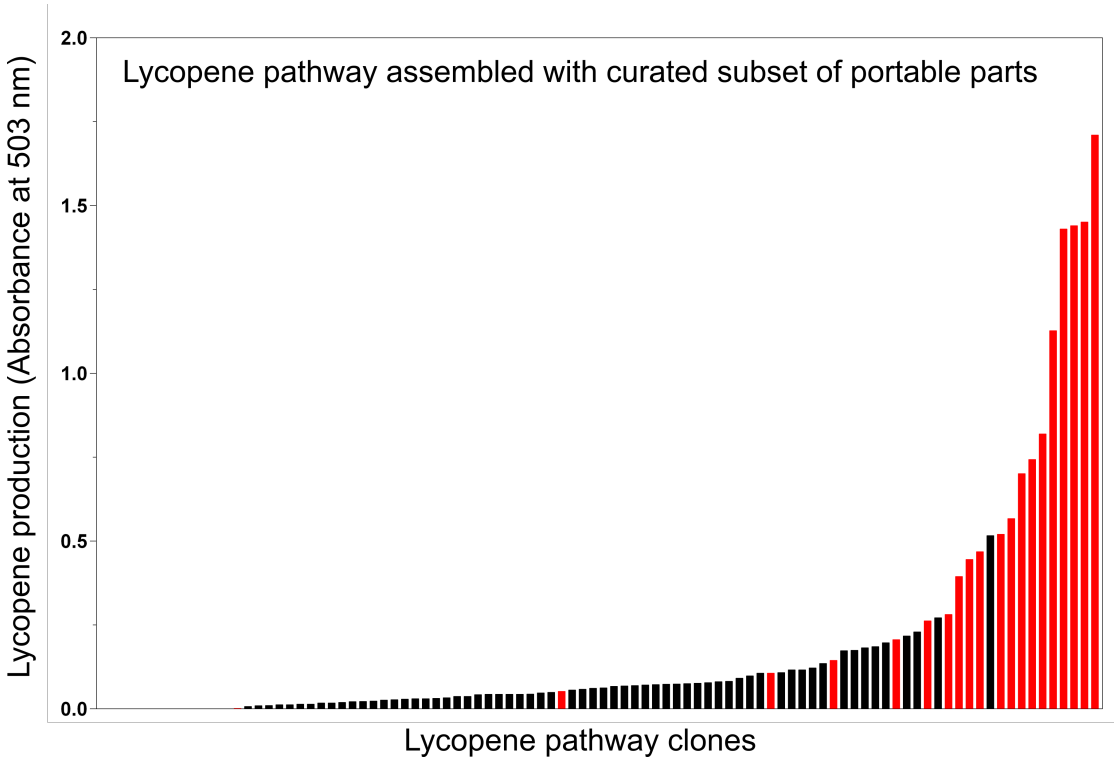


Figure 6.10. Prescreen of lycopene production in *E. coli* from the pathway clones combinatorially assembled with the curated subset of portable parts. 96 *E. coli* transformant colonies were selected and characterised after 48 h of growth by spectrophotometry (as described in Materials and Methods - Quantification of lycopene content of *E. coli* – Spectrophotometer (section **2.7.9**)). The absorbance at 503 nm of each clone is plotted with the background absorbance subtracted. From this library, 20 pathway clones were chosen, ten that produced the most lycopene and ten that gave a wide and evenly-spaced distribution of lycopene production (shown in red here). The lycopene pathway clone IDs can be found in Appendix Note A3.

6.2.5 Transforming and segregating pathways in *Synechocystis*

To test the lycopene pathway clones from the five pathway designs in *Synechocystis*, wild-type cells were transformed with each pathway clone plasmid (pGT432, 433, 435, 437 – 529) and kanamycin-resistant transformant colonies were passaged under continuous kanamycin selection in an attempt to obtain complete segregation of the pathways. The *Synechocystis* cells grew noticeably slower, with single colonies taking approximately two weeks to appear (rather than one week, which is more typical). After four passages, segregation was assessed by PCR.

To confirm presence of the full lycopene pathway, three different colony PCR reactions were required (Figure 6.11a; oligonucleotides shown in Table A4). These included: i) colony PCR reaction A that used a forward primer that annealed in the left homology arm and a reverse primer that annealed in the *crtB*; ii) colony PCR reaction B that used a forward primer in *crtB* and a reverse primer in *crtE*; and ii) colony PCR reaction C

that used a forward primer in *crtE* and a reverse primer in the right homology arm. A fourth PCR reaction (colony PCR reaction D) was also required to confirm whether each strain was partially or fully segregated into the *Synechocystis* genome, whereby flanking primers that anneal in the left and right homology arms were used. This reaction would give either a small 500 bp PCR product from WT template, no PCR product from a template with insertion of the complete pathway, as the fragment was too large to amplify, and/or potentially intermediate sized product from pathways with deletions (Figure 6.11b; oligonucleotides shown in Table A4). From these PCR reactions four possible outcomes of the *Synechocystis* transformations were identified: i) homozygote; ii) heterozygote; iii) mutated homozygote; or iv) mutated heterozygote. The homozygote contained the complete lycopene pathway and was completely integrated into the *Synechocystis* genome. Heterozygotes contained the full lycopene pathway but were not fully segregated into the *Synechocystis* genome. The mutated homozygotes contained a partial lycopene pathway that was fully segregated into the *Synechocystis* genome and the mutated heterozygotes contained a partial lycopene pathway that was not fully segregated. In all of the mutated homozygotes and mutated heterozygotes strains of *Synechocystis*, the kanamycin resistance gene had not been deleted, hence its growth on kanamycin selection (Figure 6.11).

The three individual constructs, pGT432, 433 and 435, from Pathway design 1 were all mutated homozygotes, having deleted all the pathway other than the kanamycin-resistance gene (Figure 6.11c). The twenty pathway clones that were blindly selected, pGT437 – 456, from Pathway design 2 contained one heterozygote (pGT450), three mutated homozygotes (pGT437, 443 and 453) and 16 homozygotes (pGT438 – 442, 444 – 449, 451, 452, 454 – 456; Figure 6.11c). The outcome of the twenty-four pathway clones selected after prescreening in *E. coli*, pGT457 – 476, 517 – 520, from Pathway design 3 gave six homozygotes (pGT462, 463, 475, 476, 518 and 520), two heterozygotes (pGT460 and 471), six mutated homozygotes (pGT459, 467, 468, 474, 517 and 519) and ten mutated heterozygotes (pGT457, 458, 461, 464 – 466, 469, 470, 472, 473; Figure 6.11c) in *Synechocystis*.

The twenty-four pathway clones selected after prescreening in *E. coli*, pGT477 – 496, 521 – 524, from Pathway design 4 gave ten homozygotes (pGT477, 479, 485, 487 – 489, 495, 521, 522 and 524), four heterozygotes (pGT481, 482, 486 and 492), four mutated homozygotes (pGT484, 493, 496 and 523) and six mutated heterozygotes (pGT478, 480, 483, 490, 491 and 494; Figure 6.11c). The twenty-five pathway clones selected after prescreening in *E. coli*, pGT497 – 516, 525 – 529, from Pathway design

5 gave nine homozygotes (pGT500, 505, 507, 508, 511, 516, 525, 527 and 529), six heterozygotes (pGT503, 504, 510, 513 – 515), four mutated homozygotes (pGT506, 512, 526 and 528) and six mutated heterozygotes (pGT497 – 499, 501, 502 and 509; Figure 6.11c).

Although not all clones were segregated, further passage was not deemed appropriate for the heterozygote strains as they had undergone subculturing for a considerably longer time than would have been normal in a usual transformation; two months rather than one month. It was felt that heterozygote strains were more likely to mutate rather than complete segregation. It was therefore decided to move forward with any *Synechocystis* strains that contained the full lycopene pathway, whether it was fully or partially segregated in the *Synechocystis* genome, and to characterise their lycopene production. As a result, 17 pathway clones from Pathway design 2, eight pathway clones from Pathway design 3, 14 pathway clones from Pathway design 4 and 15 pathway clones from Pathway design 5 were characterised in *E. coli* and *Synechocystis*.

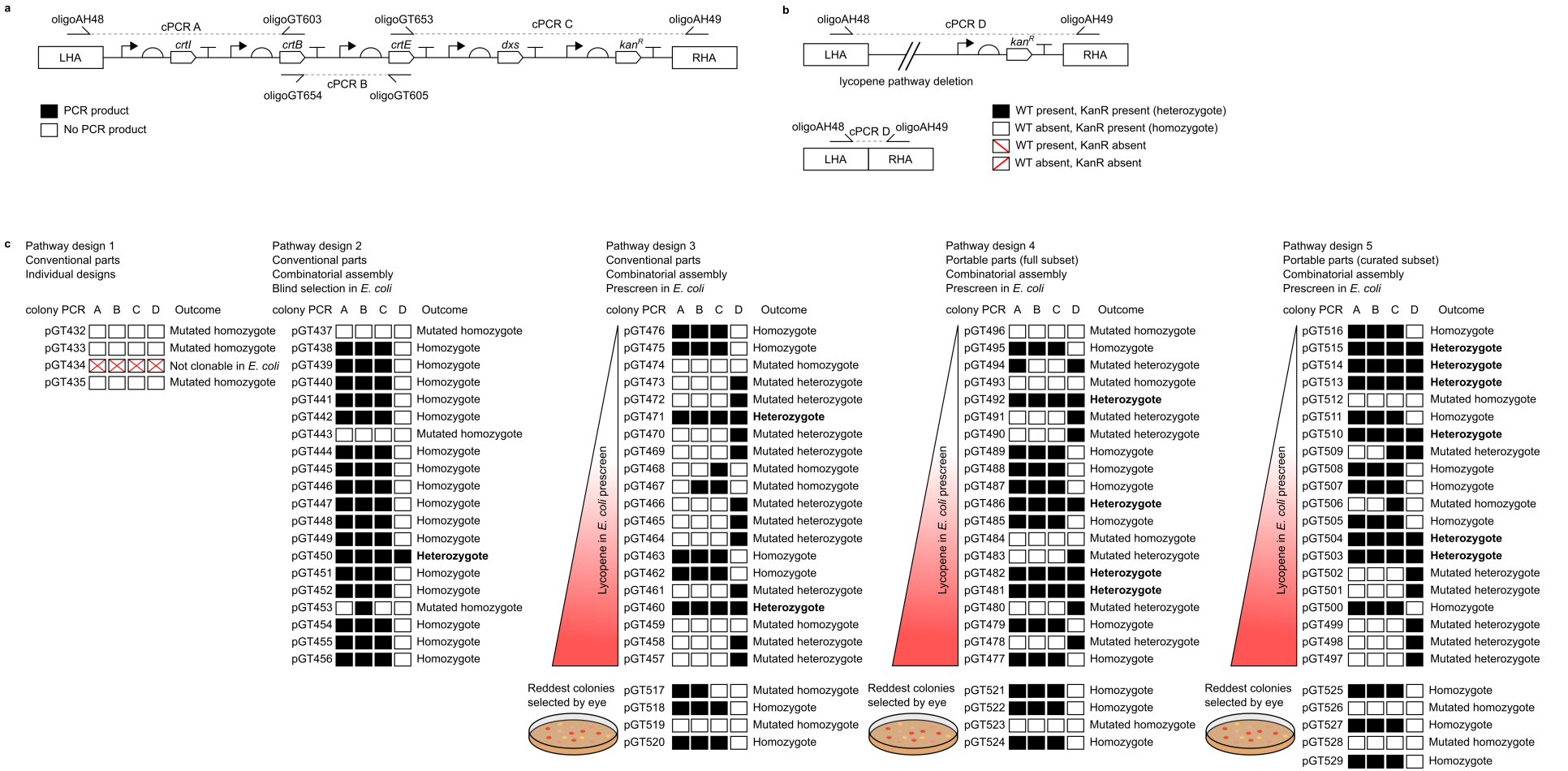


Figure 6.11. *Synechocystis* transformation outcome. Colony PCR was carried out to establish whether the each *Synechocystis* strain contained the complete lycopene pathway (a) and whether it was fully or partially segregated into the genome (b). The flanking primers (oligoAH48 and oligoAH49) used in cPCR D would give a small 500 bp PCR product from WT template, no product from template with insertion (as too large to amplify) and potentially intermediate sized product from pathways with deletions. The outcome of the 98 *Synechocystis* transformations (c) revealed four possible outcomes: i) homozygote; ii) heterozygote; iii) mutated homozygote; and iv) mutated heterozygote. Homozygotes and heterozygotes contained the full lycopene pathway, which were fully or partially segregated into the *Synechocystis* genomes, respectively. Mutated homozygotes and mutated heterozygotes contained none or partial pathways that were fully or partially segregated in the *Synechocystis* genome, respectively.

6.2.6 Characterising lycopene pathway clones in *E. coli*

The resultant 54 lycopene pathway clones from the four pathway library designs that had not mutated after the extended segregation period described above were then analysed for their production of lycopene using small-scale batch cultures in *E. coli*. Each lycopene pathway encoding plasmid (pGT438 – 442, 444 – 456, 460, 462, 463, 471, 475, 476, 477, 479, 481, 482, 485 – 489, 492, 495, 500, 503 – 505, 507, 508, 510, 511, 513 – 516, 518, 520 – 522, 525, 527 and 529) was introduced into *E. coli*, a single transformant colony was picked from each transformation and a pre-culture was grown for 24 hours. This was then sub-cultured and incubated for 48 hours, after which the lycopene concentration and dry cell weight (DCW) was measured. A wide range of lycopene concentrations was observed in *E. coli* from each of the four pathway designs (Figure 6.12), each producing between 0 – 40 mg/g DCW of lycopene.

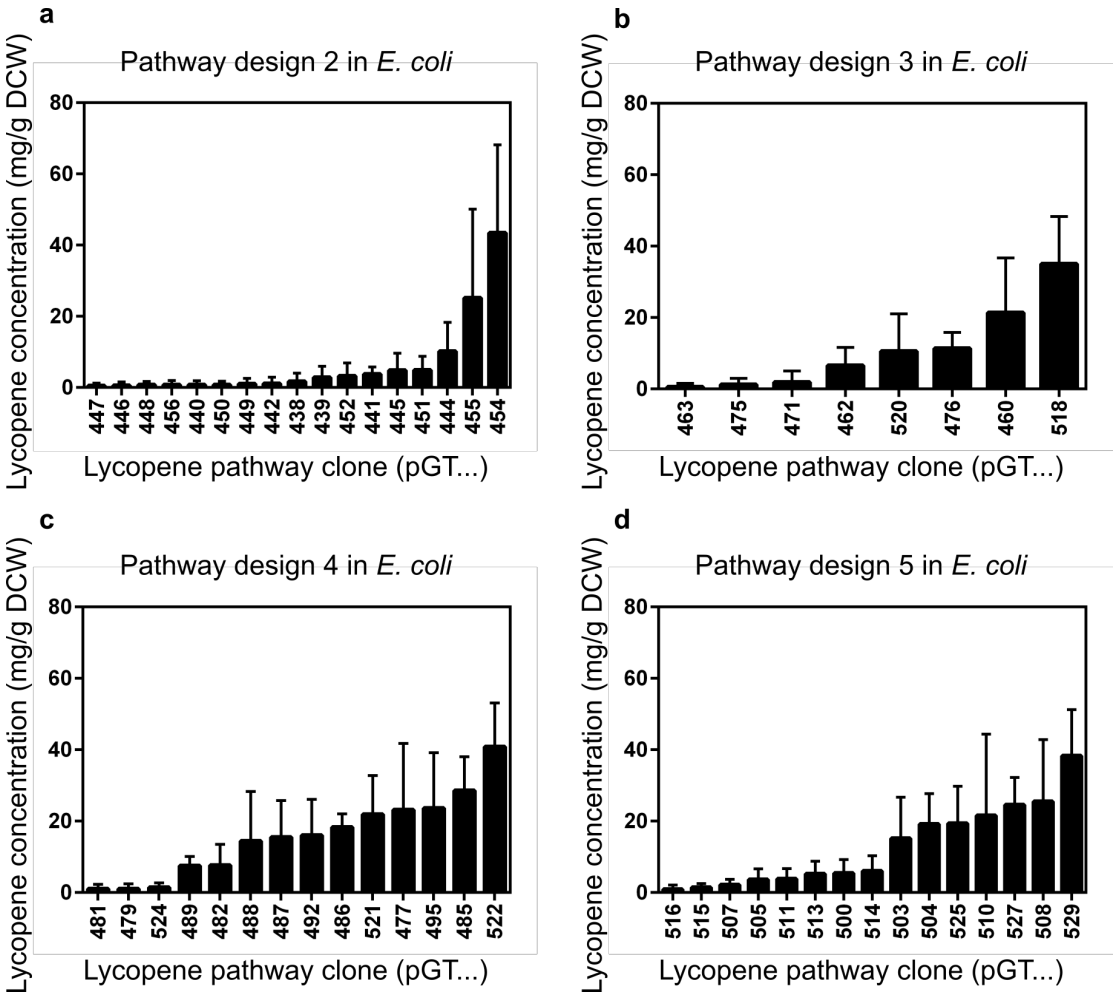


Figure 6.12. Lycopene concentrations in *E. coli*. *E. coli* DH10B cells were transformed with the plasmids containing the 54 lycopene pathway clones generated from the Pathway design 2 (panel a), Pathway design 3 (panel b), Pathway design 4 (panel c) and Pathway design 5 (panel d). Transformant colonies were cultured for 24 hours at 30 °C in LB media supplemented with the appropriate antibiotic and subcultured into fresh LB and grown for 48 hours. After which, lycopene concentration was determined as mg lycopene per dry cell weight (DCW). The error bars shown represent the standard deviation of three independent biological repeats.

6.2.7 Characterising lycopene pathway clones in *Synechocystis*

The resultant 54 lycopene pathway clones from the four pathway library designs that had not mutated after the extended segregation period were then analysed for their production of lycopene in using small-scale batch cultures in *Synechocystis*. Strains of *Synechocystis* cells that contained the complete lycopene pathway (pGT438 – 442, 444 – 456, 460, 462, 463, 471, 475, 476, 477, 479, 481, 482, 485 – 489, 492, 495, 500, 503 – 505, 507, 508, 510, 511, 513 – 516, 518, 520 – 522, 525, 527 and 529) were cultured under constant light in BG11 media pre-cultures, supplemented with kanamycin for 48 h. Cultures were subcultured to a starting optical density (measured at 750 nm) of 0.1 and grown for two weeks (336 hours), after which the lycopene

concentration and DCW was measured. Lycopene was detected in *Synechocystis* from the pathway clones from each pathway design, ranging between 0 – 1.5 mg/g DCW (Figure 6.13), which was approximately 30x less than it was in *E. coli*.

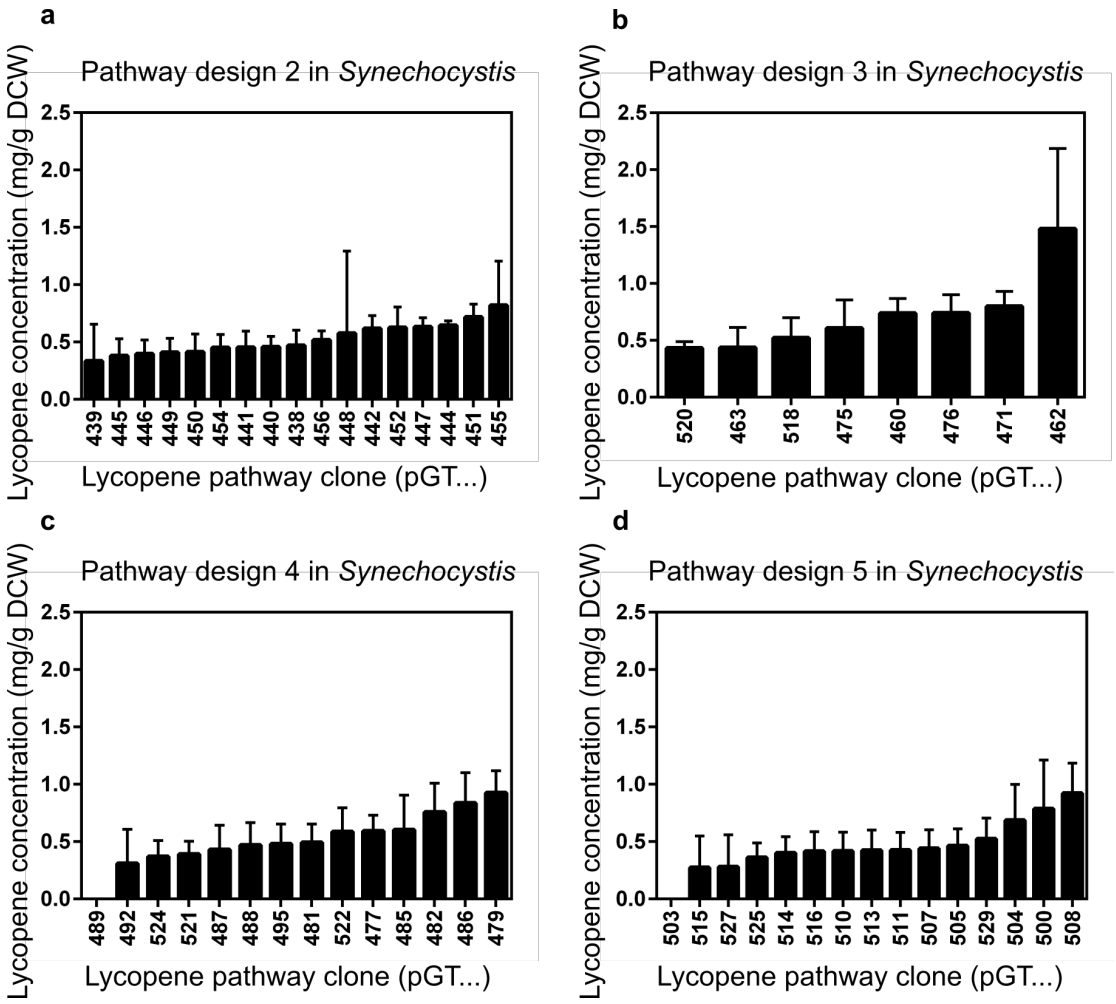


Figure 6.13. Lycopene pathway characterisation in *Synechocystis*. *Synechocystis* cells containing plasmids that encoded the 54 lycopene pathway clones generated from the Pathway design 2 (panel a), Pathway design 3 (panel b), Pathway design 4 (panel c) and Pathway design 5 (panel d) were cultured in BG11 media supplemented with appropriate antibiotic, in photoautotrophic conditions and constant light for 2 days; subcultured into fresh BG11 media and grown for two weeks. After which, lycopene concentration was determined as mg lycopene per dry cell weight (DCW). The error bars shown represent the standard deviation of three independent biological repeats.

6.2.8 Portability of the libraries of pathway-encoding constructs between *E. coli* and *Synechocystis*

To determine the differences between the four pathway designs and whether using portable parts offered any improvement over using conventional parts for engineering metabolic pathways, the performance of the pathway clones were compared between the two organisms.

It was not possible to assess the performance of four lycopene pathway clones from Pathway design 1 as they were either not cloneable in *E. coli* or mutated upon transformation into *Synechocystis*.

Assessing the performance between the 17 lycopene pathway clones that were generated and picked using Pathway design 2 principle (conventional parts and blindly picked), it was evident that, by eye, the lycopene concentrations in *E. coli* had a very different distribution compared to the distribution of the lycopene concentrations in *Synechocystis* and therefore appeared to show no correlation (R^2 = 0.042) of lycopene concentrations between the two organisms (Figure 6.14). A Pearson product-moment correlation was run to determine the relationship of the 17 constructs between *E. coli* and *Synechocystis*, which showed a weak positive correlation that was not statistically significant (r = 0.21, n = 17, p = 0.4277). Comparing the performance of the 17 clones by considering the pathways' homozygous or heterozygous nature in *Synechocystis* provided no further understanding of correlation between the two organisms (Figure A11a).

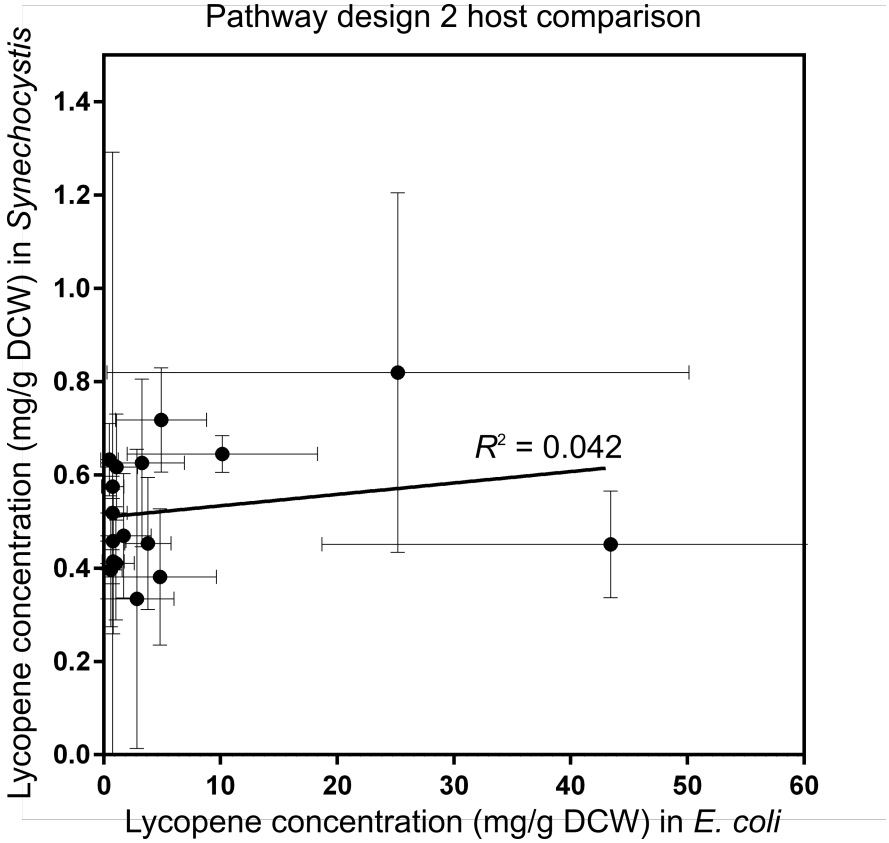


Figure 6.14. Host comparison of lycopene concentrations for pathways derived from Pathway design 2. Lycopene concentrations in *E. coli* (Figure 6.12a) were compared against those in *Synechocystis* (Figure 6.13a) for the pathway clones derived from Pathway design 2. Error bars represent standard deviations of three independent biological repeats for *E. coli* and *Synechocystis*.

Assessing the performance between the eight lycopene pathway clones that were generated and picked using Pathway design 3 principle (conventional parts and blindly picked), it was evident that, by eye, the lycopene concentrations in *E. coli* had a very different distribution compared to the distribution of the lycopene concentrations in *Synechocystis* and therefore appeared to show no correlation (R^2 = 0.025) of lycopene concentrations between the two organisms (Figure 6.15). A Pearson product-moment correlation was run to determine the relationship of the eight constructs between *E. coli* and *Synechocystis*, which showed there was a weak negative correlation that was not statistically significant (r = -0.16, n = 8, p = 0.71). Comparing the performance of the eight pathway clones by considering the pathways' homozygous or heterozygous nature in *Synechocystis* provided no further understanding of correlation between the two organisms (Figure A11b).

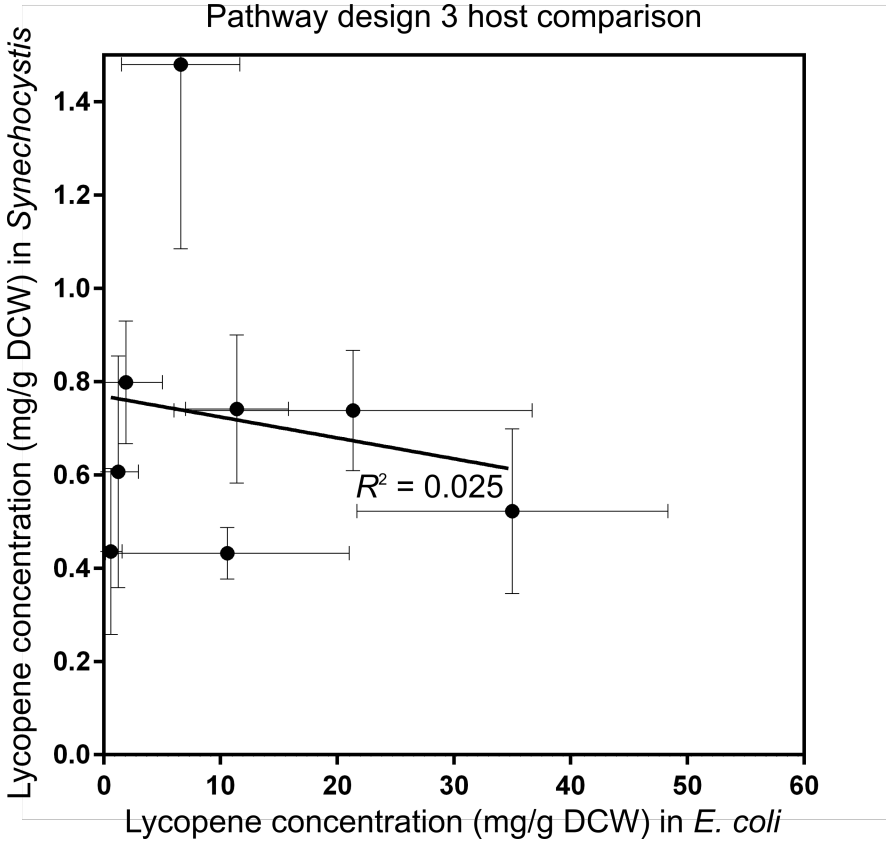


Figure 6.15. Host comparison of lycopene concentrations for pathways derived from Pathway design 3. Lycopene concentrations in *E. coli* (Figure 6.12b) were compared against those in *Synechocystis* (Figure 6.13b) for the pathway clones derived from Pathway design 3. Error bars represent standard deviations of three independent biological repeats for *E. coli* and *Synechocystis*.

Assessing the performance between the 14 lycopene pathway clones that were generated and picked using Pathway design 4 principle (portable parts and prescreened in *E. coli*), it was evident that, by eye, the lycopene concentrations in *E. coli* had a very different distribution compared to the distribution of the lycopene concentrations in *Synechocystis* and therefore appeared to show no correlation (R^2 = 0.003) of lycopene concentrations between the two organisms (Figure 6.16). A Pearson product-moment correlation was run to determine the relationship of the 14 constructs between *E. coli* and *Synechocystis*, which showed there was a no linear relationship (r = 0.06, n = 14, p = 0.85). Comparing the performance of the 14 pathway clones by considering the pathways' homozygous or heterozygous nature in *Synechocystis* provided no further understanding of correlation between the two organisms (Figure A11c).

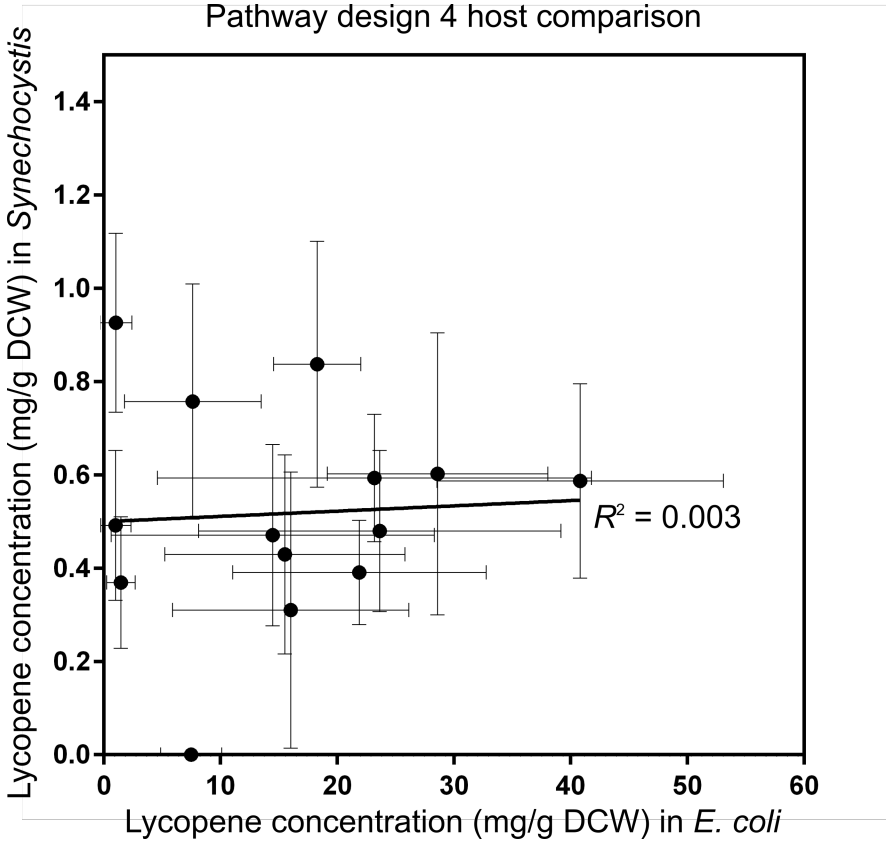


Figure 6.16. Host comparison of lycopene concentrations for pathways derived from Pathway design 4. Lycopene concentrations in *E. coli* (Figure 6.12c) were compared against those in *Synechocystis* (Figure 6.13c) for the pathway clones derived from Pathway design 4. Error bars represent standard deviations of three independent biological repeats for *E. coli* and *Synechocystis*.

Assessing the performance between the 15 lycopene pathway clones that were generated and picked using Pathway design 5 principle (curated subset of parts and prescreened in *E. coli*), it was evident that, by eye, the lycopene concentrations in *E. coli* had a very different distribution compared to the distribution of the lycopene concentrations in *Synechocystis* and therefore appeared show to no correlation (R^2 = 0.03) of lycopene concentrations between the two organisms (Figure 6.17). A Pearson product-moment correlation was run to determine the relationship of the 15 constructs between *E. coli* and *Synechocystis*, which showed there was a no linear relationship (r = 0.17, n = 15, p = 0.54). Comparing the performance of the 15 pathway clones by considering the pathways' homozygous or heterozygous nature in *Synechocystis* provided no further understanding of correlation between the two organisms (Figure A11d).

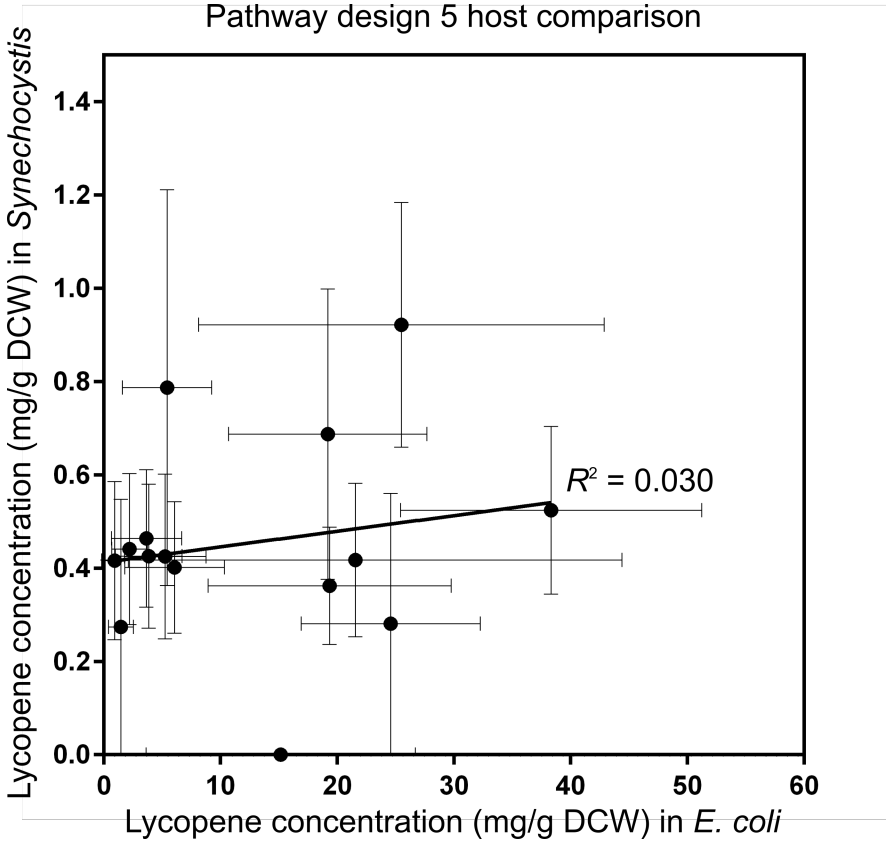


Figure 6.17. Host comparison of lycopene concentrations for pathways derived from Pathway design 5. Lycopene concentrations in *E. coli* (Figure 6.12d) were compared against those in *Synechocystis* (Figure 6.13d) for the pathway clones derived from Pathway design 5. Error bars represent standard deviations of three independent biological repeats for *E. coli* and *Synechocystis*.

By comparing the lycopene concentrations between *E. coli* and *Synechocystis* from the four pathway designs (Figure 6.14 – 6.17), it was apparent that there was a poor ability to predict the performance of a pathway in *Synechocystis* based on its performance in *E. coli*. None of the strong *E. coli* lycopene producers were strong lycopene producers in *Synechocystis* in any of the four pathway design principles, irrespective of whether conventional or portable expression control parts were used. Furthermore, the ranges of lycopene produced in *Synechocystis*, and *E. coli* for that matter, appeared to show little change between the three sets of expression control parts that were used to assemble the pathway (conventional parts, portable parts, curated subset of portable parts). The highest producing *Synechocystis* strain (pGT462) was assembled with the conventional parts.

6.3 Discussion

Engineering metabolic pathways in non-model organisms is an arduous and time-consuming task, particularly in those with low transformation efficiencies and/or slow-

growth rates, which have to date limited the use of iterative and combinatorial design approaches, respectively. To exploit the naturally diverse and industrially applicable properties of many non-model organisms, improved engineering of such organisms must be achieved. The goal of this chapter was to determine whether using combinatorial and rapid prototyping-based design principles to construct metabolic pathways in non-model organisms were better approaches compared to the state-of-the-art pathway design principles that are currently being used.

Here, the current state-of-the-art design principles used further highlighted the difficulties that surround engineering non-model organisms. For the four lycopene pathways designed using the Pathway design 1 principle, which represents current state-of-the-art design principles, one of the pathways was not cloneable in *E. coli* and the other three mutated during the *Synechocystis* transformation, which was most likely a result of cyanobacteria's genetic instability. Therefore, no functional lycopene pathways in *Synechocystis* were derived from this design principle. In general, several further iterations of the DBTL cycle would be carried out, attempting to rationally redesign and re-test the new pathway. However, as stated in Chapter One, this approach is particularly time-consuming and laborious, especially when working with a slow-growing polyploid that exhibits genetic instability, such as *Synechocystis*. This result in itself, highlights the issues that surround engineering non-model organisms using the current state-of-the-art design principles.

Implementing combinatorial approaches to design and assemble lycopene pathways, as used for Pathway designs 2 – 5, successfully generated lycopene pathways that were functional in both *E. coli* and *Synechocystis*, suggesting it is an improvement on the individual designs (Pathway design 1), which yielded no functional pathways. Comparing the four different design principles that used combinatorial approaches (Pathway design 2 – 5), they each generated functional lycopene pathways that produced similar ranges of lycopene concentrations, however the number of functional pathways differed between the four pathway designs. The pathways, at least for Pathway design 3 – 5, which were chosen based on their *E. coli* performance, were well-formed and fully functional. However, upon transforming *Synechocystis*, a large number of pathways were found to have mutated, where varying sizes of DNA fragments were lost from the pathway. This is most likely a result of the genetic instability of cyanobacteria, a phenomenon that up until recently has hardly been addressed. More specifically, Pathway design 2 gave the greatest number of complete pathways in *Synechocystis*, Pathway design 3 gave the lowest number, while Pathway

design 4 and 5 gave numbers more comparable to Pathway design 2. This demonstrates, for the lycopene pathway at least, that using the conventional parts along with a prescreen in *E. coli* (Pathway design 3) results in the greatest genetic instability. It appears that when using a prescreen of pathway performance in *E. coli*, the portable parts (Pathway design 4 and 5) improved the genetic stability of the lycopene pathway in *Synechocystis*. However, the portable parts still resulted in fewer complete pathways compared to Pathway design 2 and it is therefore hard to argue that it is a better approach, as it requires more effort not only to prescreen many pathway clones' performance in *E. coli*, but also to generate the portable parts.

While the portable parts generated functional pathways in both *E. coli* and *Synechocystis*, they offered no predictive power between the two organisms. As a result, none of the strong *E. coli* lycopene producing constructs generated with the portable parts were strong lycopene producing constructs in *Synechocystis* and therefore, this implementation of rapid prototyping was not possible. As a result, this design principle was not suitable for engineering metabolic pathways in *Synechocystis* with the current portable tools developed here. In this project, only portable promoters and RBSs were targeted and it might be conceivable that, with the development of more reliably predictive portable tools, rapid prototyping could be a worthwhile design principle for non-model organisms (further discussed in Chapter Seven). However, with the current portable tools that have been developed, that's not currently the case.

The high rate of genetic instability seen here is most likely a result of the pathway activities, where poor pathway balance has led to deleterious effects on the cell, through various factors. These factors could include: i) poor flux through the pathway to the intended product; ii) the formation of by-products; iii) the accumulation of intermediates (which may be toxic); and iv) excessive enzyme overexpression. It is also possible that genetic instability might be associated to the long periods in which *Synechocystis* strains were subcultured during attempts to fully segregate the pathways. While it is important to fully segregate a pathway into the *Synechocystis* genome, to prevent *Synechocystis* from reverting back to its wild-type phenotype, it has also been shown that repeated subculturing over long periods of time can lead to increased genetic instability. For example, over the course of three subcultures, a previous study found that the production of ethylene completely disappeared, as the pathway was deleted (327). It is also possible that *E. coli* might experience similar levels of genetic instability if the strains that contained the metabolic pathways were subcultured an equivalent number of generations to *Synechocystis*. While

cyanobacterial genetic instability has been predominately ignored as is often the nature of negative results, there are more instances in which it is being reported in cyanobacterial metabolic engineering (118, 237, 300, 301, 320, 327, 379, 507). In Chapter Five, the fidelity of assembling non-metabolic reporter constructs proved to be very efficient, however when assembling metabolic pathways, many broken pathways were identified and presumed to be a result of biological selection. It is probable that broken pathway variants in *Synechocystis* are also a result of biological selection. Furthermore, in this case the longer and multiple periods of culturing provided a greater chance for biological selection to act. If cyanobacteria are to be used as microbial cell factories, generating stable strains will be as necessary as generating well-performing pathways. It is possible that a fine line might exist between partial segregation of a pathway in the cyanobacterial genome and with the genetic instability of the construct for the best product titre. This is far from ideal for the commercialisation of cyanobacteria strains, where the risk of a pathway completely mutating or deleting is likely. Strains used in commercial contexts will have to be completely segregated. In an ideal situation, the metabolic coupling of product synthesis and microbial growth would be a favourable approach for maximising a pathways performance, not only with respect to product titres and productivities (508-510), but also to pathway stability.

The combinatorial approach used here successfully generated 54 strains of *Synechocystis* that contained full and functioning lycopene pathways. The top producing strain (*Synechocystis* strain pGT462) produced 0.006 gl^{-1} of lycopene and contained a fully segregated pathway in the genome. Not only is this the first report of engineered lycopene production in cyanobacteria, but when comparing the titre from *Synechocystis* strain pGT462 to products made via the MEP pathway, the combinatorial approach has resulted in a higher product titre. Here, 0.006 gl^{-1} of lycopene (C_{40}) was produced in *Synechocystis* strain pGT462, which is a much higher titre than other MEP pathway derived products, such as 0.00017gl^{-1} of isoprene (C_5) (504), 0.0045 gl^{-1} of limonene (C_{10}) (311), 0.00071 gl^{-1} of bisabolene (C_{15}) (311) and 0.0021gl^{-1} of squalene (C_{30}) (316), which have previously been produced in cyanobacteria.

The aim of these small-scale experiments was to compare the production of lycopene between *E. coli* and *Synechocystis* and not to maximise production, although the results are encouraging when compared to equivalent products. It is possible that the

production could be improved further not only by optimising pathway, but also by optimising the conditions.

From this project, combinatorial approaches seem to be a worthwhile implementation to improve pathway design in cyanobacteria, which not only generates well performing, but also stable metabolic pathways. Furthermore, the combinatorial approach was successfully used for first reporting of lycopene production in cyanobacteria. In this project, lycopene was produced in the *Synechocystis* wild-type background. It could be possible to use FBA identify potential knockout or knockdown modifications as a stoichiometric metabolic model is available for *Synechocystis* sp. PCC 6803 (511), to increase lycopene production. To date, there are very few examples of FBA predicted strategies being experimentally implemented in cyanobacteria (512, 513), but nonetheless, it offers a potential route for identifying issues in the distribution of carbon flux. Alternatively, two studies have identified target genes that can be knocked out to increase the production of lycopene in *E. coli* and this could be mirrored in *Synechocystis*. The first of which increased lycopene production by 130% relative to the control by knocking out the central carbon metabolic gene *zwf* (glucose-6-phosphate dehydrogenase) (514). Incidentally, a mutant of *Synechocystis* with *zwf* knocked out has been generated in the Heap lab (Dr John Heap, pers. comm). The second strategy knocked out three genes (*gdhA*, *aceE* and *fdhF*), which gave a 2-fold increase over the parental strain (515). It is worth noting that the metabolism of *Synechocystis* and *E. coli* are quite different and therefore the outcome of these strategies might be different. Furthermore, a recent report showed that *ipi*, *dxs* and *ispD* were all bottlenecks in the MEP pathway and increasing their expression increased the MEP-derived product isoprene 40-fold compared to the strain without these modifications in *Synechocystis* (308). Here *dxs* has been expressed in the lycopene pathway as a previously identified bottleneck, however *ipi* and *ispD* have not and could therefore be added to the pathway.

Another approach to increase the lycopene production would be to optimise the performance of the pathway. This could also be carried out in the knockout strains described above to further improve the lycopene production. One approach to optimising the performance of the pathway could be to perform a second iteration of combinatorial pathway optimisation. In this round of combinatorial optimisation it would be possible to rationally reduce the size of the combinatorial library. If the rational design was done appropriately, it would reduce the library size by several orders of magnitude and it would enrich the resultant library with high producing pathway clones.

6.4 Key Outcomes

- The first report of the direct and sustainable production of lycopene from CO_2.
- Genetic instability is as important as poor pathway performance for metabolic engineering in cyanobacteria.
- In this implementation of rapid prototyping the portable parts do not provide a predictive performance between *E. coli* and *Synechocystis* and therefore in this instance, rapid prototyping was not a viable design principle.
- Combinatorial approaches to a pathway design are a great improvement over conventional approaches, which allowed the generation of many functional and stable pathways in *Synechocystis*.

7. CHAPTER SEVEN. Discussion

The overall aim of this thesis was to establish whether combinatorial and rapid prototyping-based design principles were better at generating functional and stable metabolic pathways in cyanobacteria, compared to the state-of-the-art pathway design principles that are currently being used. The question asked in this project also fits into the broader challenge of trying to establish the best approach for engineering metabolic pathways in non-model organisms. While the fast-growth rate and tractability of model organisms have allowed the easy implementation of combinatorial and iterative optimisation of metabolic pathways, these approaches have been difficult to implement in non-model organisms, which often have slow-growth rates and low transformation efficiencies. In an attempt to generate effective (well-performing) metabolic pathways and to obtain them more quickly and easily in non-model organisms, the concept of rapid prototyping was applied as an *in vivo* optimisation process in the cyanobacterium *Synechocystis* sp. PCC 6803. To apply this concept, this project started by developing the necessary parts and framework, after which the concept was implemented and evaluated against the state-of-the-art design principles that are currently used in cyanobacteria. To determine whether the rapid prototyping of metabolic pathway-encoding constructs was an advantageous approach for optimising metabolic pathways in *Synechocystis*, it was tested and compared to current state-of-the-art and other prototyping based approaches.

7.1 Rapid prototyping of metabolic pathway-encoding constructs

In the first implementation of rapid prototyping it was not possible to use the performance of a pathway in *E. coli* as a prediction of its performance in *Synechocystis* and therefore in this instance, rapid prototyping offered no benefit as a novel pathway design principle. For this first implementation of rapid prototyping, portable parts were used that had been chosen on the basis of their portable performance between *E. coli* and *Synechocystis*, when characterised using non-metabolic fluorescent and colourimetric reporters (Chapter Four). It is possible that the lack of predictive pathway performance between the two organisms was caused by the portable parts not performing as expected between *E. coli* and *Synechocystis*, which would in turn led to different protein concentration profiles of the pathways between *E. coli* and *Synechocystis* (Figure 7.1). It is also possible that the lack of predictive pathway performance between the two organisms could have resulted from a lack of, or differences, in the compatibility of protein expression profiles between the metabolism of the two organisms (Figure 7.1). Currently it is not possible to establish whether the

differences in the performance of the pathways (assembled with portable parts) between the two organisms were caused by different protein concentration profiles, by differences in metabolic compatibility, or by a combination of the two.

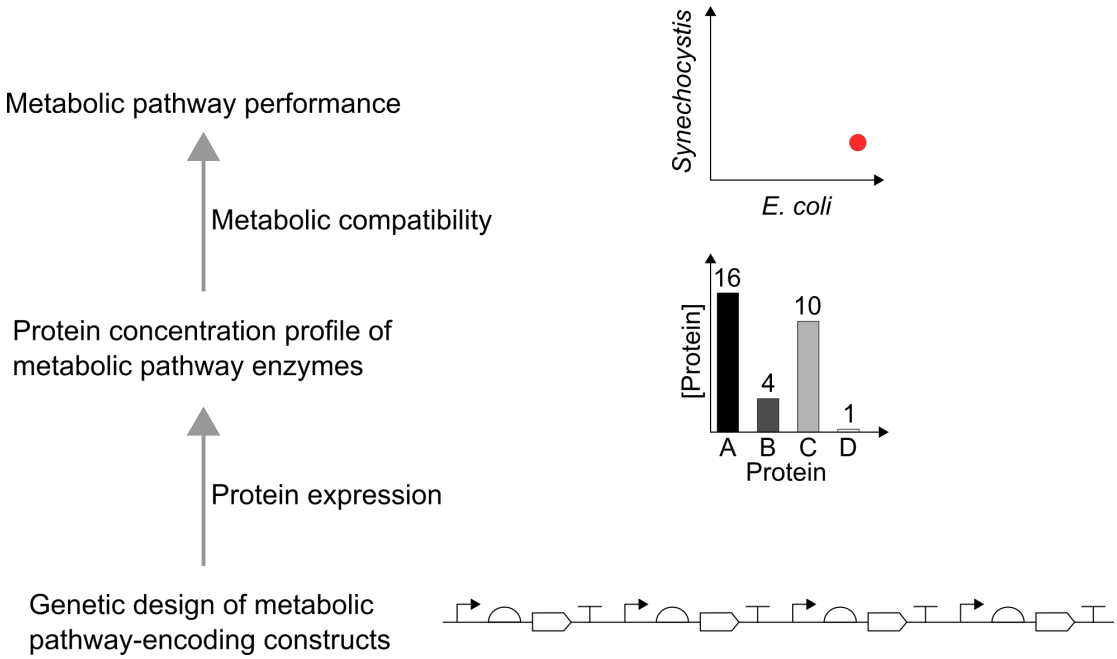


Figure 7.1. Relationship between genetic design and pathway performance. A given genetic construct could give different pathway performance outcomes in different organisms due to differences in the concentrations of proteins expressed from a given genetic design, or the compatibility of that protein expression profile with the metabolism of the organism, or both. Metabolic pathway performance refers to the productivity, titre and yields and, in the case for cyanobacteria, genetic stability.

7.2 Factors that affect protein concentration profiles being preserved

It is possible that the portable parts preserved protein concentration profiles between the two organisms, but that differences in the metabolism of the two organisms led to different optimal pathway protein concentration profiles being required to achieve the similarly functioning pathways. In order to establish whether this was the case, it would be necessary to measure the protein concentrations of the four enzymes in the pathway for each pathway variant in both *E. coli* and *Synechocystis*. It would also be necessary to sequence each pathway variant to determine which promoters and RBSs drive the expression of each gene within each pathway. With the promoter and RBSs identified and the protein concentrations determined, it would be possible to determine, using modelling-based approaches, whether certain promoters and/or RBSs offer better portable properties than others. It would also be possible to determine whether certain strengths of expression control parts, most likely stronger strengths, are unsuitable for portable metabolic engineering, and whether the weaker promoters and

RBSs are suitable on their own for portable metabolic engineering between *E. coli* and *Synechocystis* (Figure 7.2).

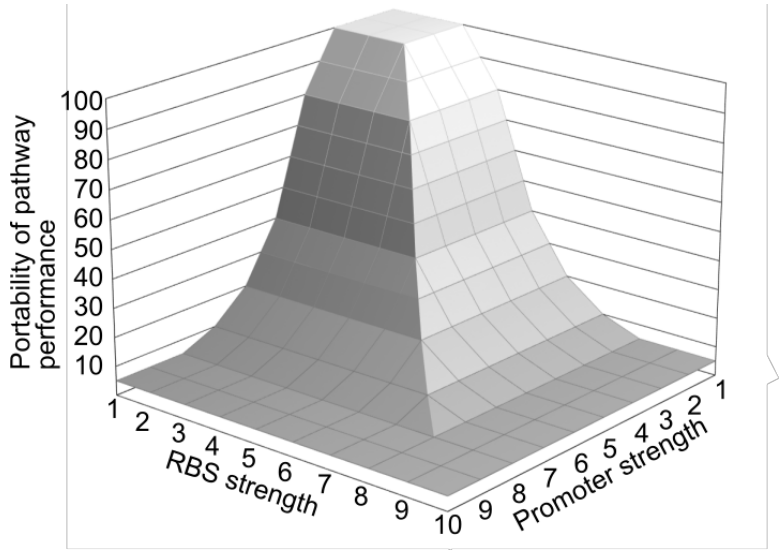


Figure 7.2. Relationship between promoter strength, RBS strength and the portability of a pathway's performance. Figure shows an illustrative example of the relationship between portable part strength and the portability of a pathway performance. The figure shows that the portability of a pathway's performance is sensitive to the strength of promoters and RBSs used. The pathway's performance is portable when weak promoters and weak RBSs are used, but as the strength of promoters and RBSs increase the portability of a pathway's performance decreases.

It is also possible however, that protein concentration profiles were not preserved by the portable parts between *E. coli* and *Synechocystis*. This could be because of differences in the transcription and translation machinery between the two organisms, which has previously been described in section **3.3**. Furthermore, if the central dogma of molecular biology is considered, there are many factors that affect protein concentration, and in this project, only transcription initiation and translation initiation were targeted. It is possible that, while controlling protein expression using portable promoters and portable RBSs had been sufficient for non-metabolic fluorescent and colourimetric reporters, it was not sufficient for the portable control when a metabolic pathway composed on multiple enzymes was expressed. In this project, the codon usage was assumed to be similar enough between the two organisms that it could be overlooked (Figure A10). However, differences in the two organism's codon usage may well have led to different rates of translation elongation, which in turn would have resulted in different protein concentrations between the *E. coli* and *Synechocystis*. Furthermore, it is possible that other factors that influence protein expression levels, including promoter escape (472) and post-transcriptional factors such as translational enhancers (516), antisense RNAs (517), mRNA degradation (where different RNases

have previously reported in *E. coli* and *Synechocystis* (518)) and protein degradation, could have contributed to the differences in the portability of the lycopene pathways when expressed using portable parts.

It is also possible that the portable performance of these parts and/or the pathways were affected by other factors that had not had to be considered before, when only non-metabolic fluorescent and colourmetric reporters had been expressed. Expressing metabolic pathways and other genetic systems can put an unnatural extra genetic load on a cell, typically referred to as burden. Burden is often caused by the additional demand that a heterologous and/or synthetic metabolic pathway has on the cellular resources, such as amino acids, nucleotides, energy and their transcriptional and translational machinery. It is possible that differences between the cellular resources of *E. coli* and *Synechocystis* could result in different burden-based responses in the two cells, which in turn would result in different pathway performances. Furthermore, burden-based studies have demonstrated that the stronger the RBS that is used to express the genetic circuit, the more sensitive the cellular burden and the circuit functionality is, to an increase in the level of transcription. Hence the burden derived design principle, to use a strong promoter and a weak RBS. It is possible that such a phenomenon is occurring in both *E. coli* and *Synechocystis* strains where the lycopene pathway has been assembled with medium to strong RBSs. However, the sensitivity of the pathway's performance and/or sensitivity to burden when the level of transcription increases (in this case by using stronger constitutive promoters) could be different between the two organisms. It is possible that the different sensitivities of these responses could break the portability of a pathway's performance between the two organisms, irrespective of whether the protein concentration profiles were preserved and irrespective of whether there was no difference in the metabolic constraints between the organisms. If this was the case, it could be possible to overcome the different sensitivities by constraining the design of the combinatorial design space by only using the weak RBSs from the portable subset along with the portable promoters.

To date, two studies have been reported that have used portable parts for cross-species expression of metabolic pathways. Both reports use expression systems that were orthogonal to the organisms host regulation (246, 409). The first of these examples, allows cross-species expression of metabolic pathways between *E. coli*, *B. subtilis* and *P. putida*, however while the orthogonal T7-based expression system allows expression in these three organisms, it does not encode similar pathway

performances between the three organisms (409). Therefore, as it stands this system would not serve well for rapid prototyping, it is possible that simple changes could be made that could make it more suitable, such as by using different viral RNAP and promoters. This was carried out by the second portable expression system, which was developed for *E. coli* and *Halomonas* and uses an orthogonal T7-like expression system, where the RNAP and the promoter were derived from Morganella phage MmP. This portable expression system was shown to perform portably between the two organisms when tested using the PHB pathway, whereby increasing the inducer concentration resulted in a predicable (R^2 = 0.92) increase in PHB production in both organisms (246). Therefore, if it is assumed that the lack of portability of a pathway in the first implementation of rapid prototyping was a result of protein concentration profiles not being preserved between *E. coli* and *Synechocystis*. It is possible that an orthogonal expression system would provide a better level of predictive performance between *E. coli* and *Synechocystis*, because it would express proteins without interference from the native functions of the host cell.

To achieve a truly predictive portability of protein concentration profiles, a central dogma that is completely orthogonal to both organisms may be required (519), whereby a given pathway can be expressed without being impeded by the native regulation of the host cell. An orthogonal central dogma would require combining orthogonal DNA replication systems, orthogonal transcription systems and orthogonal translation systems, somewhat like mitochondria or chloroplasts that have self-contained replication, transcription and translation machinery. Orthogonal regulation is not a new concept; with DNA replication systems (520), orthogonal transcription systems (246, 409) and components of orthogonal translation systems (521-524) having been previously developed; however they are yet to be combined into a truly orthogonal central dogma.

In this project, the terminators that were used in the assembly of the lycopene encoding metabolic pathways (Chapter Six) had previously demonstrated functionality in *Synechocystis*, however the efficiency of the termination of those terminators could be different between the two organisms. The four terminators had previously demonstrated strong transcriptional termination in *E. coli* (127), however the strong termination provided in *E. coli* might not be the case for *Synechocystis*. In this case, expression of any downstream genes in the lycopene pathway may get expressed by any of the upstream genes, which would result in greater concentrations of those proteins in *Synechocystis* compared to in *E. coli*. This is turn would affect the protein

concentration profiles between the organisms and therefore the performance of those pathway variants would be different in *Synechocystis* compared to *E. coli*. One approach to overcome this, would be to design the lycopene pathway as an operon, which would nullify the need to use terminators. Alternatively, the terminators used could be further characterised, to determine their efficiency of termination between the two organisms, providing confidence that downstream genes are not being expressed by upstream genes in *Synechocystis* and therefore affecting their portability.

7.3 Metabolic constraints on portability

The lack of portable performance (as shown in Figure 7.1) could have been caused by a lack of compatibility of protein expression profiles between the two organisms, whereby the protein expression profiles were preserved by the portable parts between *E. coli* and *Synechocystis*. However, the functionality of the pathway variants is different between the two organisms because the protein expression profiles have different compatibilities with the metabolism of the two organisms. For example, an optimal protein expression profile in *E. coli* might not be optimal in *Synechocystis*.

In this project, *E. coli* was grown on LB broth, a rich medium composed of a crude oligopeptide mix, which *E. coli* can grow on by freeing and catabolising amino acids from the oligopeptides (525). As a rich medium, it would presumably have provided a considerably greater concentration of carbon for *E. coli* to metabolise and convert to lycopene, compared to that of the atmospheric CO_2, which *Synechocystis* grows on. In addition to different concentrations of carbon provided in the growth mediums, the upstream metabolism of *E. coli* and *Synechocystis* is very different. *Synechocystis* fixes carbon as CO_2 using the CCM and the carbon then enters the CCB cycle as 3PG, which can either enter glycolysis or be redirected to regenerate RuBP or other sugar-based products (Figure 1.7). In this project, *E. coli* obtained its carbon predominately from amino acids. Analysis of LB broth has previously identified 17 different amino acids (525), which can be metabolised into one of seven metabolic intermediates: pyruvate, α-ketoglutarate, succinyl-CoA, fumarate, oxaloacetate, acetyl-CoA and acetoacetate (12). The amino acids, alanine, cysteine, glycine, serine, threonine and tryptophan that are metabolised into pyruvate can then be used in the MEP pathway (12). The differences in upstream metabolism between the two organisms will also include different side reactions where carbon flux can be redirected away from lycopene production to varying degrees. As a result, there would have been different fluxes of carbon towards the lycopene pathway in *E. coli* and *Synechocystis*. Based

on the concentration of lycopene that *E. coli* produced, more carbon was directed towards lycopene. It is possible that a different optimal protein expression profile was needed in *E. coli* compared to *Synechocystis*, to accommodate the greater flux of carbon being directed towards lycopene. One approach that might improve, but not overcome these differences in carbon uptake and flux could be to provide similar carbon concentrations in the growth medium. This could either be done by growing *E. coli* on a defined, minimal medium where the carbon concentration can be controlled and/or by enriching the CO_2 concentration for *Synechocystis* to grow on.

In addition to differences in the growth conditions and upstream metabolism, a lack of compatibility of the protein expression profiles between *E. coli* and *Synechocystis* could also have arisen due to different metabolic fluxes upstream of the lycopene pathway. Whilst the lycopene pathway is preceded by the MEP pathway in both *E. coli* and *Synechocystis*, it is possible that the thermodynamic and kinetic rates of these reactions differ between the two organisms. As such, whilst these reactions occur in both organisms, they could occur at very different rates. Different reaction rates would result in different carbon fluxes towards the lycopene pathway. To accommodate these different carbon fluxes, the optimal protein concentration profiles may differ between the organisms.

While understanding and comparing the kinetic properties between organisms is laborious, using modelling methods such as the Max-min Driving Force (MDF) (50), which combines thermodynamics and stoichiometric properties of metabolic pathways to provide some quantitative understanding of the differences between the metabolic fluxes of a pathway between the two organisms. This will identify reactions that have low thermodynamic driving forces and therefore limiting reactions. In this instance, there may be different limiting reactions in the lycopene and MEP pathways between the *E. coli* and *Synechocystis*, or different side reactions that draw carbon flux away from the lycopene pathway.

In addition to the metabolic differences outlined above, cofactor specificity could also have contributed to differences in the performance of a given pathway between the two organisms. Cofactors, such as ATP, coenzyme A, FAD, NAD(P)H/NAD(P) and metal elements, are essential for the activity of many enzymes, in both *E. coli* and *Synechocystis*. However, enzymes can have different specificities for such cofactors, particularly NADPH and NADH, which are the predominant reducing equivalents. In cyanobacteria, NADPH is produced by photosynthesis and is therefore more abundant

than NADH and is the major reducing equivalent (350, 526, 527). Whereas, in heterotrophic bacteria, *E. coli* in this case, NADH is more abundant than NADPH (528). While the three-enzyme pathway engineered here to produce lycopene does not contain any cofactor dependencies, DXS has been shown to use the cofactor thiamine diphosphate and the metal ions Mg^{2+} and Mn^{2+} in *E. coli* (529). Their availability may be different in *Synechocystis*, therefore affecting the portable performance and changing the carbon flux that is directed towards the carotenoid pathway between the two organisms. Furthermore, upstream reactions, such as the CBB cycle (526), have cofactor specifies and could therefore have contributed to the lack of portable performance between the two organisms. If *in vivo* rapid prototyping is to be implemented in the future metabolic pathways, the cofactor specificities will challenge the portability of the metabolic pathway.

The different metabolic constraints between *Synechocystis* and *E. coli*, or between any other pair of organisms, will affect the portability of the performance of a metabolic pathway (Figure 7.1). In some specific cases, the performance of a given metabolic pathway may be so strongly affected by the metabolic environment of the host organism that just prototyping the protein expression profiles of pathway enzymes will not allow portable performance of a pathway. To avoid such issues, one approach would be to make a modified strain of *E. coli*, or whatever the prototyping host is in any future cases, which better reflects the metabolic conditions of the target non-model organism. For example, one modification to *E. coli* which could make it better reflect the metabolic conditions of *Synechocystis* would be to increase the availability of NADPH, which has previously been achieved by replacing the NAD(+)-dependent glyceraldehyde-3-phosphate dehydrogenase GapA with the NADP(+)-dependent GapB from *B. subtilis* and addition of NAD kinase (530).

7.4 Alternative approaches to overcoming engineering non-model organisms.

In this project, an *in vivo* approach of rapid prototyping was tested and evaluated but did not perform as intended. This type of *in vivo* prototyping relies on portable expression of protein concentration profiles and overlooks differences in the physiology of the two organisms, as discussed in section **7.2** and **7.3**.

One alternative would be *in vitro* prototyping whereby a pathways performance would be tested and optimised in the cell-free lysate of a non-model organism prior to *in vivo* expression. Such cell-free prototyping strategies have been demonstrated for

optimising the production of *n*-butanol and polyhydroxyalkanoates bioplastics in *E. coli* (390, 391). It has also been demonstrated that cell-free prototyping can be used to measure the cost of a pathways gene expression to predict the metabolic burden that the given pathway would impose on the cell, as was demonstrated with the luciferase and β-carotene pathways (139). Furthermore, cell-free systems are now being developed for non-model organisms, such as *Bacillus megaterium* and *Streptomyces venezuelae* (531, 532). Such an approach may be beneficial instead of the *in vivo* rapid prototyping approach attempted here, because it would not rely on expression control parts performing portably between two different organisms. Furthermore, differences between the two organism's metabolism and cofactor specificity would not be an issue for *in vitro* prototyping. To date, no cell-free methods have been developed for cyanobacteria and therefore any *in vitro* prototyping would have to start with the development and characterisation of the necessary tools, which would be a time-consuming task.

Another alternative approach to rapid prototyping, for cyanobacteria at least, would be to use a fast-growing strain such as *Synechoccus* UTEX 2973, which has a considerably faster growth rate than *Synechocystis* (353). Furthermore, a genetic toolkit of inducible and constitutive promoters and riboswitches has recently been developed for this cyanobacterial strain, making it a considerably more attractive host for metabolic engineering compared to other cyanobacterial strains (533). It has also been shown that the growth rate of *Synechocystis* can be increased by overexpressing RuBisCO, however this only occurred under certain conditions (534) and the growth rate was still not as fast as in *Synechoccus* UTEX 2973. Nevertheless, using a faster growing strain such as *Synechoccus* UTEX 2973 would speed up iterations of the DBTL cycle, making the iterative approach to metabolic pathway optimisation more useable.

In this project the prototyping approach in which pathways were combinatorially optimised; whereby the lycopene pathway-encoding construct was combinatorially assembled and pathway variants were blindly chosen in *E. coli* and transferred and characterised in *Synechocystis*; proved to be an effective approach for generating functional, well-performing and stable pathways. An alternative approach to allow implementation of combinatorial pathway optimisation in a non-model organism, would be to invest time improving the transformation efficiencies of those organisms to enhance the effectiveness of the combinatorial approaches, such as in *Clostridium saccharoperbutylacetonicum* strain N1-4 (388).

Recently, incoherent feedforward loops (iFFLs) were engineered into *E. coli* promoters using transcription-activator like effectors (TALEs). These promoters showed near identical expression, without any re-tuning, of the three gene predeoxyviolacein when expressed in different genome locations and in plasmids with different copy numbers (535). It was recently suggested that this iFFL design might not just allow portability of a pathway's performance in different contexts, but it also might allow portability between different organisms (536), although this is yet to be tested.

7.5 Enhancing the cyanobacterial genetic toolkit

Despite the first implementation of rapid prototyping not offering the predictive performance of a pathway between *E. coli* and *Synechocystis*, this project has successfully developed the necessary genetic tools that the cyanobacterial toolkit require. In Chapter Three, the largest library of functional constitutive promoters and RBSs was developed, further enhancing the cyanobacterial genetic toolkit, which at the start of this project certainly lacked such parts. These libraries of promoters and RBSs offer a wide range of expression strengths that increase over small increments of strength. In Chapter Five, Start-Stop Assembly was developed, a multi-part DNA assembly method that offers the advantages of modular methods, such as reaction efficiency and yet it is scarless at the CDS boundaries. Furthermore, its streamlined hierarchy makes it easily applicable to any destination context such as different strains of cyanobacteria or different replicative or integrative shuttle vectors. Despite the fact that there is still a lot more work to be done in order to improve our understanding of portability, the promoter and RBS libraries and Start-Stop Assembly are ideal tools for assembling metabolic pathway-encoding constructs, as well as for other genetic systems and also for fine-tuning and combinatorially optimising protein expression levels.

7.6 Lycopene production

Synechocystis served as an interesting candidate to use as the non-model organism due to its potential to produce chemicals and biofuels in a renewable and sustainable manner. In this project *Synechocystis* was engineered to renewably produce lycopene directly from CO_2 for the first time. Lycopene is widely used in the healthcare industry owing to its antioxidant properties and its potential in the reduction of prostate cancer risk in humans (498). To date, lycopene production has been engineered using heterologous pathways in *E. coli* and *S. cerevisiae*, while higher titres have been

engineered; 448 mg/g DCW (150) and 55.56 mg/g DCW (537) respectively. However, in both cases lycopene was produced using the expensive substrate glucose rather than CO_2, which is abundant and free. In the case of *S. cerevisiae*, the culture conditions used were far more optimised for lycopene production, whereby batch-fed fermentation was used. Here however, small-scale experiments were used that were better suited for comparing the lycopene concentration of many strains in *E. coli* and *Synechocystis*, and titres could certainly be improved by optimising culture conditions.

In addition to optimisation the conditions in which *Synechocystis* is grown, the performance of the pathway could most likely be further improved by optimising the performance of the lycopene pathway itself. This could be achieved with further iterations of the DBTL cycle. Considering the success of the combinatorial prototyping seen in this project, further iterations of the DBTL could include a rational heuristic refinement of the original combinatorial assembly of the lycopene pathway-encoding construct, whereby the combinatorial assembly is focused around the local optimal of the well performing, stable pathway variants (Figure 7.3) identified in the first iteration of the combinatorial assembly. In theory this approach enriches the design space by decreasing the combinatorial search space by using fewer, but rationally chosen promoters and RBSs in the combinatorial assembly. The promoters and RBSs that are rationally chosen for the heuristic combinatorial assembly can be chosen by following two rationales: i) chose the promoter and RBS that most frequently occur at each CDS position in the pathway, plus the promoter and RBS that is one level stronger and one level weaker at each position; or ii) chose the promoters and RBSs that occurred at each CDS position in any of the well performing stable pathway variants and discard promoters and RBSs from each CDS position that did not occur in the well functioning stable pathway variants. Such heuristic refinement has proven to enrich the design space during the combinatorial optimisation of the lycopene pathway in *E. coli* (Dr John Heap, pers. comm). Moreover, it has also been used to increase the production of itaconic acid by 4-fold in *S. cerevisiae* (152).

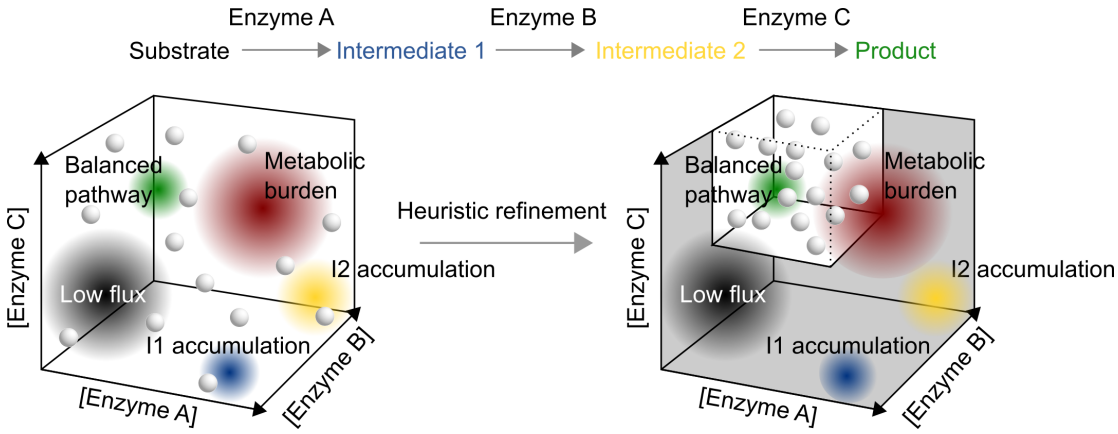


Figure 7.3. Heuristic refinement of a combinatorial search space. Combinatorial assembly typically provides a sparse sampling of the design space. Heuristic refinement can enrich the combinatorial search space of a combinatorial assembly for subsequent iterations of optimisation. To refine the search space the number and distribution of parts used can be focused around local optima identified at each CDS position in well performing stable strains generated in an initial sparse sampling.

7.7 Conclusion

While there are techniques and tools available to construct metabolic pathway-encoding constructs, it is currently very challenging to design a metabolic pathway with an optimal performance. This is because designing a pathway to express its proteins at an intended level is currently limited and therefore this typically results in an imbalance of the pathways protein concentration profile. Optimising the protein concentration profile through iterative and combinatorial approaches has proven effective in model organisms, but these approaches have been difficult to implement in diverse and industrially applicable organisms, such as *Synechocystis*. This PhD project tested whether *in vivo* rapid prototyping of metabolic pathway-encoding constructs was an effective and faster approach to optimising a pathways performance in *Synechocystis*. In this approach, metabolic pathway-encoding constructs were combinatorially assembled in the tractable model organism *E. coli*, where high producing strains were identified and subsequently transferred to *Synechocystis* for expression. The rapid prototyping concept relied on using portable expression control parts that would provide similar protein expression levels and therefore a similar protein concentration profile of a pathway between *E. coli* and *Synechocystis*, allowing the performance of a pathway in *E. coli* to be used as a prediction of its performance in *Synechocystis*. This concept was then tested using the lycopene pathway and evaluated against current design principles used in cyanobacteria. The first implementation of rapid prototyping did not offer the predictive performance of a pathway between *E. coli* and *Synechocystis*. Despite this, the project highlighted the limitations of using the current state-of-the-art approaches to design a metabolic

pathway-encoding construct and the advantages that using combinatorial approaches offer to overcome these limitations. Therefore, on the basis of this initial study, it was concluded that using a combinatorial approach to design and assemble a pathway, and then blindly identify pathway variants to transfer to *Synechocystis*, would be the best approach to engineering functional and stable metabolic pathways in cyanobacteria. If this approach was implemented in other non-model organisms, Start-Stop Assembly would serve as an ideal DNA assembly method, however suitable expression control parts would be required that cover a wide range of strengths in the given non-model organism.

7.8 Future work with respect to portability

- Sequence all 54 lycopene pathway variants to establish the promoters and RBSs that drive the expression of each gene in each construct.
- Using the promoter and RBS strengths established in Chapter Three, try to establish a relationship between the RBS and promoter strengths at each position in the pathway and the pathways overall performance between the two organisms, which would provide further insight into the performance of these pathways between *E. coli* and *Synechocystis*.
- In addition to the promoter and RBS strengths, the protein concentration profiles could be compared between the two organisms by quantitating the concentration of each enzyme in the pathway in each organism.
- Use the sequence data and protein concentrations to establish whether the lack of portable pathway performance was a result of the portable parts not preserving protein concentration profiles between *E. coli* and *Synechocystis* or a result of differences in the metabolic constraints between the two organisms.
- Test both rapid prototyping and combinatorial-based approaches using different metabolic pathways.

7.9 Future work with respect to increasing lycopene production

- Sequence the lycopene pathway of the top producing strain of *Synechocystis* to identify the promoters and RBSs responsible.
- Identify gene knockout candidates using OptKnock or an equivalent method for increasing lycopene production in *Synechocystis*.
- Generate knockout strains with identified genes and then transform them with the previously identified high producing pathway variant.

- Scale-up the culture size and optimise conditions, such as by increasing the CO_2 concentration, to increase the production of lycopene (538).
- Produce lycopene by growing the high-producing strain of *Synechocystis* in enriched CO_2 streams, which are off-gas streams from industrial and/or microbial processes, such as the CO_2 enriched off-gas from steel manufacturers or yeast fermentation.

7.10 Future directions

The aim of this project was to develop the tools to make cyanobacteria more of a viable and sustainable cell factory. Looking to the future, and as the world is becoming increasingly more focused on climate change and sustainability, the work presented in this thesis will hopefully assist in the development of cyanobacteria as a sustainable cell factory, producing products of interest directly from CO_2 and sunlight.

In the immediate future, the promoter and RBS libraries developed in this thesis will be a valuable addition to the genetic toolbox of cyanobacteria, by enabling protein expression levels to be optimised and fine-tuned in order to improve the performance of metabolic pathways. Furthermore, when these promoters and RBSs are used in conjunction with Start-Stop Assembly and the combinatorial approach to pathway design (as presented in Chapter Six), they will serve as a good platform for generating stable and effective metabolic pathways. Cyanobacteria have already proved capable of producing a wide range of products directly from CO_2, but these products have generally been at low titres; and with this improved platform, hopefully it will be possible to optimise the productivity of some of these metabolic pathways, which in turn may allow the genetically modified cyanobacteria to be commercially used for the first time.

Looking at how this project can progress further in the long term; with the difficulties that surround the engineering of cyanobacteria; a prototyping approach, such as rapid prototyping, still remains a viable and attractive approach to pathway optimisation and is therefore something that is worth exploring further. To implement rapid prototyping, genetic tools that provide a predictable and portable expression of metabolic pathways between two organisms are still required. Achieving this level of predicable and portable protein expression may require certain expression control parts to be used, which are truly orthogonal to each host (unlike those used in this project). Furthermore, it may even require an entirely orthogonal central dogma, which is becoming

increasingly possible with recent developments, for example an orthogonal ribosome has been developed that recognises orthogonal mRNA transcripts (539). If predictive and portable protein expression could be achieved between *E. coli* and *Synechocystis*, it would be possible to implement rapid prototyping. This would likely enhance the optimisation of metabolic pathways in *Synechocystis*, not only improving the productivity of these pathways, but also providing potential for cyanobacteria to be used commercially as a sustainable cell factory for producing products of interest.

Furthermore, rapid prototyping was designed as an *in vivo* prototyping approach for overcoming some of the limitations (such as slow-growth rates and low transformation efficiencies) that are encountered when working with organisms that are difficult to engineer. Whilst *E. coli* was used as a prototyping host for *Synechocystis* in this first implementation of rapid prototyping, rapid prototyping could be used to optimise the performance of metabolic pathways in various host organisms with industrially applicable properties, using either the same or different prototyping hosts. As introduced in Chapter One, a plethora of organisms exist with a wide range of industrially relevant properties. Successfully incorporating rapid prototyping to optimise the performance of pathways in other industrially relevant organisms could increase the number of sustainable cell factories that are being used commercially, which could assist in the global transition towards a more sustainable economy.

REFERENCES

1. Change,I.P.O.C. (2014) Climate Change 2013: The Physical Science Basis Cambridge University Press.

2. Rogelj,J., Elzen,den,M., Höhne,N., Fransen,T., Fekete,H., Winkler,H., Schaeffer,R., Sha,F., Riahi,K. and Meinshausen,M. (2016) Paris Agreement climate proposals need a boost to keep warming well below 2 °C. *Nature*, **534**, 631–639.

3. Grand View Research. (2017) Biotechnology Market Size, Share | Industry Analysis Report 2014-2025. Retrieved from https://www.grandviewresearch.com/industry-analysis/biotechnology-market.

4. Chemicals,C.O.I.O.B.A.R.T.A.T.A.M.O., Technology,B.O.C.S.A., Sciences,B.O.L., Studies,D.O.E.A.L.National Research Council (2015) Industrialization of Biology National Academies Press, Washington, D.C.

5. Lindee,S. (2012) Robert Bud, The uses of life: a history of biotechnology, Cambridge University Press, 1993, pp. xiii, 299, illus., £30.00, $49.95 (0-521-38240-8). *Medical History*, **38**, 342–342.

6. Demain,A.L., Vandamme,E.J., Collins,J. and Buchholz,K. (2016) History of Industrial Biotechnology. In *Industrial Biotechnology*, Microorganisms. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, Vol. 101, pp. 1–84.

7. Kinoshita,S., Udaka,S. and Shimono,M. (2004) Studies on the amino acid fermentation. Part 1. Production of L-glutamic acid by various microorganisms. *J. Gen. Appl. Microbiol.*, **50**, 331–343.

8. Perlman,D. (2012) Microbial Technology Elsevier.

9. Dashen,M.M., Ado,S.A., Ameh,J., Amapu,T. and Zakari,H. (2013) Screening and Improvement of Local Isolates of Aspergillus Niger for Citric Acid Production. *Bayero Journal of Pure and Applied Sciences*, **6**, 105–111.

10. Leroy,F. and De Vuyst,L. (2004) Lactic acid bacteria as functional starter cultures for the food fermentation industry. *Trends in Food Science & Technology*, **15**, 67–78.

11. de chimie,J.G.-L.A.1810 Extrait d'un mémoire sur la Fermentation.

12. Voet,D., Voet,J.G. and Pratt,C.W. (2008) Principles of Biochemistry John Wiley & Sons.

13. Flamholz,A., Noor,E., Bar-Even,A., Liebermeister,W. and Milo,R. (2013) Glycolytic strategy as a tradeoff between energy yield and protein cost. *Proceedings of the National Academy of Sciences*, **110**, 10039–10044.

14. Beard,D.A. and Qian,H. (2007) Relationship between thermodynamic driving force and one-way fluxes in reversible processes. *PLoS ONE*, **2**, e144.

15. Rottenberg,H. (1973) The thermodynamic description of enzyme-catalyzed reactions. The linear relation between the reaction rate and the affinity. *Biophys. J.*, **13**, 503–511.

16. Stephanopoulos,G. and Vallino,J.J. (1991) Network rigidity and metabolic engineering in metabolite overproduction. *Science*, **252**, 1675–1681.

17. Bailey,J.E. (1991) Toward a science of metabolic engineering. *Science*, **252**, 1668–1675.

18. Stephanopoulos,G., Aristidou,A. and Nielsen,J. (1998) Metabolic Engineering.

19. Stephanopoulos,G. and Sinskey,A.J. (1993) Metabolic engineering--methodologies and future prospects. *Trends Biotechnol.*, **11**, 392–396.

20. Mavrovouniotis,M.L. and Stephanopoulos,G. (1990) Computer-aided synthesis of biochemical pathways. *Biotechnol. Bioeng.*, **36**, 1119–1132.

21. Hatzimanikatis,V., Li,C., Ionita,J.A., Henry,C.S., Jankowski,M.D. and Broadbelt,L.J. (2005) Exploring the diversity of complex metabolic networks. *Bioinformatics*, **21**, 1603–1609.

22. Moriya,Y., Shigemizu,D., Hattori,M., Tokimatsu,T., Kotera,M., Goto,S. and Kanehisa,M. (2010) PathPred: an enzyme-catalyzed metabolic pathway prediction server. *Nucleic Acids Res.*, **38**, W138–43.

23. Carbonell,P., Parutto,P., Baudier,C., Junot,C. and Faulon,J.-L. (2014) Retropath: automated pipeline for embedded metabolic circuits. *ACS Synth. Biol.*, **3**, 565–577.

24. Rodrigo,G., Carrera,J., Prather,K.J. and Jaramillo,A. (2008) DESHARKY: automatic design of metabolic pathways for optimal cell growth. *Bioinformatics*, **24**, 2554–2556.

25. Henry,C.S., Broadbelt,L.J. and Hatzimanikatis,V. (2010) Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial chemicals: 3-hydroxypropanoate. *Biotechnol. Bioeng.*, **106**, n/a–n/a.

26. Yim,H., Haselbeck,R., Niu,W., Pujol-Baxley,C., Burgard,A., Boldt,J., Khandurina,J., Trawick,J.D., Osterhout,R.E., Stephen,R., *et al.* (2011) Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol. *Nat. Chem. Biol.*, **7**, 445–452.

27. Vallino,J.J. and Stephanopoulos,G. (1994) Carbon Flux Distributions at the Pyruvate Branch Point in Corynebacterium glutamicum during Lysine Overproduction. *Biotechnology Progress*, **10**, 320–326.

28. Stephanopoulos,G. (1994) Metabolic engineering. *Current Opinion in Biotechnology*, **5**, 196–200.

29. Nissen,T.L., Schulze,U., Nielsen,J. and Villadsen,J. (1997) Flux distributions in anaerobic, glucose-limited continuous cultures of Saccharomyces cerevisiae. *Microbiology*, **143 (Pt 1)**, 203–218.

30. Vallino,J.J. and Stephanopoulos,G. (2000) Metabolic flux distributions in Corynebacterium glutamicum during growth and lysine overproduction. Reprinted from Biotechnology and Bioengineering, Vol. 41, Pp 633-646 (1993).

31. Antoniewicz,M., Kraynie,D., Laffend,L., Gonzalezlergier,J., Kelleher,J. and Stephanopoulos,G. (2007) Metabolic flux analysis in a nonstationary system: Fed-batch fermentation of a high yielding strain of E. coli producing 1,3-propanediol. *Metab. Eng.*, **9**, 277–292.

32. Stephanopoulos,G. (1999) Metabolic fluxes and metabolic engineering. *Metab. Eng.*, **1**, 1–11.

33. Kim,H.U., Kim,T.Y. and Lee,S.Y. (2008) Metabolic flux analysis and metabolic engineering of microorganisms. *Mol Biosyst*, **4**, 113–120.

34. Vlassis,N., Pacheco,M.P. and Sauter,T. (2014) Fast reconstruction of compact context-specific metabolic network models. *PLoS Comput Biol*, **10**, e1003424.

35. Systems Biology Research Group. (2018) Available predictive genome-scale metabolic network reconstructions. Retrieved from http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms.

36. Oh,Y.-K., Palsson,B.Ø., Park,S.M., Schilling,C.H. and Mahadevan,R. (2007) Genome-scale reconstruction of metabolic network in Bacillus subtilis based on high-throughput phenotyping and gene essentiality data. *J. Biol. Chem.*, **282**, 28791–28799.

37. Senger,R.S. and Papoutsakis,E.T. (2008) Genome-scale model for Clostridium acetobutylicum: Part I. Metabolic network resolution and analysis. *Biotechnol. Bioeng.*, **101**, 1036–1052.

38. Förster,J., Famili,I., Fu,P., Palsson,B.Ø. and Nielsen,J. (2003) Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. *Genome Res.*, **13**, 244–253.

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

40. Alper,H., Jin,Y.-S., Moxley,J.F. and Stephanopoulos,G. (2005) Identifying gene targets for the metabolic engineering of lycopene biosynthesis in Escherichia coli. *Metab. Eng.*, **7**, 155–164.

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

42. Gu,D., Zhang,C., Zhou,S., Wei,L. and Hua,Q. (2016) IdealKnock: A framework for efficiently identifying knockout strategies leading to targeted overproduction. *Comput Biol Chem*, **61**, 229–237.

43. Maertens,J. and Vanrolleghem,P.A. (2010) Modeling with a view to target identification in metabolic engineering: a critical evaluation of the available tools. *Biotechnology Progress*, **26**, 313–331.

44. Kacser,H. and Burns,J.A. (1995) The control of flux. *Biochemical Society Transactions*, **23**, 341–366.

45. Westerhoff,H.V. and van Dam,K. (1987) Thermodynamics and control of biological free-energy transduction Elsevier Science Ltd.

46. Acerenza,L. (1993) Metabolic control design. *Journal of Theoretical Biology*, **165**, 63–85.

47. Nielsen,J. (1997) Metabolic control analysis of biochemical pathways based on a thermokinetic description of reaction rates. *Biochemical Journal*, **321 (Pt 1)**, 133–138.

48. Costa,R.S., Hartmann,A. and Vinga,S. (2016) Kinetic modeling of cell metabolism for microbial production. *Journal of Biotechnology*, **219**, 126–141.

49. Cintolesi,A., Clomburg,J.M., Rigou,V., Zygourakis,K. and Gonzalez,R. (2012) Quantitative analysis of the fermentative metabolism of glycerol in Escherichia coli. *Biotechnol. Bioeng.*, **109**, 187–198.

50. Noor,E., Bar-Even,A., Flamholz,A., Reznik,E., Liebermeister,W. and Milo,R. (2014) Pathway thermodynamics highlights kinetic obstacles in central metabolism. *PLoS Comput Biol*, **10**, e1003483.

51. Lee,S.J., Lee,D.-Y., Kim,T.Y., Kim,B.H., Lee,J. and Lee,S.Y. (2005) Metabolic engineering of Escherichia coli for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation. *Appl. Environ. Microbiol.*, **71**, 7880–7887.

52. Park,S.M., Shaw-Reid,C., Sinskey,A.J. and Stephanopoulos,G. (1997) Elucidation of anaplerotic pathways in Corynebacterium glutamicum via 13 C-NMR spectroscopy and GC-MS. *Appl. Microbiol. Biotechnol.*, **47**, 430–440.

53. Koffas,M.A.G., Jung,G.Y. and Stephanopoulos,G. (2003) Engineering metabolism and product formation in Corynebacterium glutamicum by coordinated gene overexpression. *Metab. Eng.*, **5**, 32–41.

54. Bro,C., Regenberg,B., Förster,J. and Nielsen,J. (2006) In silico aided metabolic engineering of Saccharomyces cerevisiae for improved bioethanol production. *Metab. Eng.*, **8**, 102–111.

55. Ajikumar,P.K., Xiao,W.-H., Tyo,K.E.J., Wang,Y., Simeon,F., Leonard,E., Mucha,O., Phon,T.H., Pfeifer,B. and Stephanopoulos,G. (2010) Isoprenoid pathway optimization for Taxol precursor overproduction in Escherichia coli. *Science*, **330**, 70–74.

56. Cann,A.F. and Liao,J.C. (2008) Production of 2-methyl-1-butanol in engineered Escherichia coli. *Appl. Microbiol. Biotechnol.*, **81**, 89–98.

57. Gonzalez,J.B., Fernandez,F.J. and Tomasini,A. (2003) Microbial Secondary Metabolites Production and Strain Improvement NISCAIR-CSIR, India.

58. Bailey,J.E., Sburlati,A., Hatzimanikatis,V., Lee,K., Renner,W.A. and Tsai,P.S. (1996) Inverse metabolic engineering: A strategy for directed genetic engineering of useful phenotypes. *Biotechnol. Bioeng.*, **52**, 109–121.

59. Jin,Y.-S., Alper,H., Yang,Y.-T. and Stephanopoulos,G. (2005) Improvement of xylose uptake and ethanol production in recombinant Saccharomyces cerevisiae

through an inverse metabolic engineering approach. *Appl. Environ. Microbiol.*, **71**, 8249–8256.

60. Ghosh,C., Gupta,R. and Mukherjee,K.J. (2012) An inverse metabolic engineering approach for the design of an improved host platform for over-expression of recombinant proteins in Escherichia coli. *Microb. Cell Fact.*, **11**, 93.

61. Brooks,A.N., Turkarslan,S., Beer,K.D., Lo,F.Y. and Baliga,N.S. (2011) Adaptation of cells to new environments. *Wiley Interdiscip Rev Syst Biol Med*, **3**, 544–561.

62. Wang,Y., Manow,R., Finan,C., Wang,J., Garza,E. and Zhou,S. (2011) Adaptive evolution of nontransgenic Escherichia coli KC01 for improved ethanol tolerance and homoethanol fermentation from xylose. *J. Ind. Microbiol. Biotechnol.*, **38**, 1371–1377.

63. Balderas-Hernández,V.E., Hernández-Montalvo,V., Bolívar,F., Gosset,G. and Martinez,A. (2010) Adaptive Evolution of *Escherichia coli* Inactivated in the Phosphotransferase System Operon Improves Co-utilization of Xylose and Glucose Under Anaerobic Conditions. *Appl Biochem Biotechnol*, **163**, 485–496.

64. Alper,H., Moxley,J., Nevoigt,E., Fink,G.R. and Stephanopoulos,G. (2006) Engineering yeast transcription machinery for improved ethanol tolerance and production. *Science*, **314**, 1565–1568.

65. Alper,H. and Stephanopoulos,G. (2007) Global transcription machinery engineering: a new approach for improving cellular phenotype. *Metab. Eng.*, **9**, 258–267.

66. Du,J., Yuan,Y., Si,T., Lian,J. and Zhao,H. (2012) Customized optimization of metabolic pathways by combinatorial transcriptional engineering. *Nucleic Acids Res.*, **40**, e142–e142.

67. Jovicevic,D., Blount,B.A. and Ellis,T. (2014) Total synthesis of a eukaryotic chromosome: Redesigning and SCRaMbLE-ing yeast. *Bioessays*, **36**, 855–860.

68. Dymond,J.S., Richardson,S.M., Coombes,C.E., Babatz,T., Muller,H., Annaluru,N., Blake,W.J., Schwerzmann,J.W., Dai,J., Lindstrom,D.L., *et al.* (2011) Synthetic chromosome arms function in yeast and generate phenotypic diversity by design. *Nature*, **477**, 471–476.

69. Blount,B.A., Gowers,G.-O.F., Ho,J.C.H., Ledesma-Amaro,R., Jovicevic,D., McKiernan,R.M., Xie,Z.X., Li,B.Z., Yuan,Y.J. and Ellis,T. (2018) Rapid host strain improvement by in vivo rearrangement of a synthetic yeast chromosome. *Nat Commun*, **9**, 1932.

70. Isalan,M., Lemerle,C., Michalodimitrakis,K., Horn,C., Beltrao,P., Raineri,E., Garriga-Canut,M. and Serrano,L. (2008) Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, **452**, 840–845.

71. Windram,O.P.F., Rodrigues,R.T.L., Lee,S., Haines,M. and Bayer,T.S. (2017) Engineering microbial phenotypes through rewiring of genetic networks. *Nucleic Acids Res.*, **45**, 4984–4993.

72. Yang,T.H., Kim,T.W., Kang,H.O., Lee,S.-H., Lee,E.J., Lim,S.-C., Oh,S.O., Song,A.-J., Park,S.J. and Lee,S.Y. (2010) Biosynthesis of polylactic acid and its

copolymers using evolved propionate CoA transferase and PHA synthase. *Biotechnol. Bioeng.*, **105**, 150–160.

73. Bao,Z., Xiao,H., Liang,J., Zhang,L., Xiong,X., Sun,N., Si,T. and Zhao,H. (2015) Homology-integrated CRISPR-Cas (HI-CRISPR) system for one-step multigene disruption in Saccharomyces cerevisiae. *ACS Synth. Biol.*, **4**, 585–594.

74. Cobb,R.E., Wang,Y. and Zhao,H. (2015) High-efficiency multiplex genome editing of Streptomyces species using an engineered CRISPR/Cas system. *ACS Synth. Biol.*, **4**, 723–728.

75. DiCarlo,J.E., Norville,J.E., Mali,P., Rios,X., Aach,J. and Church,G.M. (2013) Genome engineering in Saccharomyces cerevisiae using CRISPR-Cas systems. *Nucleic Acids Res.*, **41**, 4336–4343.

76. Jiang,Y., Chen,B., Duan,C., Sun,B., Yang,J. and Yang,S. (2015) Multigene editing in the Escherichia coli genome via the CRISPR-Cas9 system. *Appl. Environ. Microbiol.*, **81**, 2506–2514.

77. Jinek,M., Chylinski,K., Fonfara,I., Hauer,M., Doudna,J.A. and Charpentier,E. (2012) A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science*, **337**, 816–821.

78. Li,H., Shen,C.R., Huang,C.-H., Sung,L.-Y., Wu,M.-Y. and Hu,Y.-C. (2016) CRISPR-Cas9 for the genome engineering of cyanobacteria and succinate production. *Metab. Eng.*, **38**, 293–302.

79. Jakočiūnas,T., Bonde,I., Herrgård,M., Harrison,S.J., Kristensen,M., Pedersen,L.E., Jensen,M.K. and Keasling,J.D. (2015) Multiplex metabolic pathway engineering using CRISPR/Cas9 in Saccharomyces cerevisiae. *Metab. Eng.*, **28**, 213–222.

80. Li,Y., Lin,Z., Huang,C., Zhang,Y., Wang,Z., Tang,Y.-J., Chen,T. and Zhao,X. (2015) Metabolic engineering of Escherichia coli using CRISPR-Cas9 meditated genome editing. *Metab. Eng.*, **31**, 13–21.

81. Browning,D.F. and Busby,S.J. (2004) The regulation of bacterial transcription initiation. *Nat. Rev. Microbiol.*, **2**, 57–65.

82. Jensen,P.R. and Hammer,K. (1998) The sequence of spacers between the consensus sequences modulates the strength of prokaryotic promoters. *Appl. Environ. Microbiol.*, **64**, 82–87.

83. Ciampi,M.S. (2006) Rho-dependent terminators and transcription termination. *Microbiology*, **152**, 2515–2528.

84. Adhya,S. and Gottesman,M. (1982) Promoter occlusion: transcription through a promoter may inhibit its activity. *Cell*, **29**, 939–944.

85. Salis,H.M., Mirsky,E.A. and Voigt,C.A. (2009) Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.*, **27**, 946–950.

86. Laursen,B.S., Sørensen,H.P., Mortensen,K.K. and Sperling-Petersen,H.U. (2005) Initiation of protein synthesis in bacteria. *Microbiol. Mol. Biol. Rev.*, **69**, 101–123.

87. de Smit,M.H. and van Duin,J. (1990) Secondary structure of the ribosome binding site determines translational efficiency: a quantitative analysis. *Proceedings of the National Academy of Sciences*, **87**, 7668–7672.

88. Na,D., Lee,S. and Lee,D. (2010) Mathematical modeling of translation initiation for the estimation of its efficiency to computationally design mRNA sequences with desired expression levels in prokaryotes. *BMC Syst Biol*, **4**, 71.

89. Welch,M., Govindarajan,S., Ness,J.E., Villalobos,A., Gurney,A., Minshull,J. and Gustafsson,C. (2009) Design parameters to control synthetic gene expression in Escherichia coli. *PLoS ONE*, **4**, e7002.

90. Sharp,P.M. and Li,W.H. (1986) An evolutionary perspective on synonymous codon usage in unicellular organisms. *J. Mol. Evol.*, **24**, 28–38.

91. Ikemura,T. (1985) Codon usage and tRNA content in unicellular and multicellular organisms. *Mol. Biol. Evol.*, **2**, 13–34.

92. Gustafsson,C., Govindarajan,S. and Minshull,J. (2004) Codon bias and heterologous protein expression. *Trends Biotechnol.*, **22**, 346–353.

93. Sharp,P.M. and Li,W.H. (1987) The codon Adaptation Index--a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res.*, **15**, 1281–1295.

94. Keasling,J.D. (1999) Gene-expression tools for the metabolic engineering of bacteria. *Trends Biotechnol.*, **17**, 452–460.

95. Reznikoff,W.S. (1992) The lactose operon-controlling elements: a complex paradigm. *Mol. Microbiol.*, **6**, 2419–2422.

96. Noel,R.J. and Reznikoff,W.S. (2000) Structural studies of lacUV5-RNA polymerase interactions in vitro. Ethylation interference and missing nucleoside analysis. *J. Biol. Chem.*, **275**, 7708–7712.

97. Guzman,L.M., Belin,D., Carson,M.J. and Beckwith,J. (1995) Tight regulation, modulation, and high-level expression by vectors containing the arabinose PBAD promoter. *Journal of Bacteriology*, **177**, 4121–4130.

98. Yansura,D.G. and Henner,D.J. (1990) Use of Escherichia coli trp promoter for direct expression of proteins. *Meth. Enzymol.*, **185**, 54–60.

99. KAJIWARA,S., FRASER,P.D., KONDO,K. and MISAWA,N. (1997) Expression of an exogenous isopentenyl diphosphate isomerase gene enhances isoprenoid biosynthesis in Escherichia coli. *Biochemical Journal*, **324**, 421–426.

100. Berry,A. (1996) Improving production of aromatic compounds in Escherichia coli by metabolic engineering. *Trends Biotechnol.*, **14**, 250–256.

101. Floras,N., Xiao,J., Berry,A., Bolívar,F. and Valle,F. (1996) Pathway engineering for the production of aromatic compounds in Escherichia coli. *Nat. Biotechnol.*, **14**, 620–623.

102. Skraly,F.A., Lytle,B.L. and Cameron,D.C. (1998) Construction and characterization of a 1,3-propanediol operon. *Appl. Environ. Microbiol.*, **64**, 98–105.

103. Altaras,N.E. and Cameron,D.C. (1999) Metabolic engineering of a 1,2-propanediol pathway in Escherichia coli. *Appl. Environ. Microbiol.*, **65**, 1180–1185.

104. Ruohonen,L., Penttilä,M. and Keränen,S. (1991) Optimization of Bacillus α-amylase production by Saccharomyces cerevisiae. *Yeast*, **7**, 337–346.

105. Cantwell1,B.A., Brazil,G., Murphy,N. and McConnell,D.J. (1986) Comparison of expression of the endo-?-1,3-1,4-glucanase gene from Bacillus subtilis in Saccharomyces cerevisiae from the CYC1 and ADH1 promoters. *Current Genetics*, **11**, 65–70.

106. Kumar,V., Ramakrishnan,S., Teeri,T.T., Knowles,J.K.C. and Hartley,B.S. (1992) Saccharomyces Cerevisiae Cells Secreting an Aspergillus Niger β-Galactosidase Grow on Whey Permeate. *Nat. Biotechnol.*, **10**, 82–85.

107. Yamano,S., Ishii,T., Nakagawa,M., Ikenaga,H. and Misawa,N. (1994) Metabolic engineering for production of beta-carotene and lycopene in Saccharomyces cerevisiae. *Biosci. Biotechnol. Biochem.*, **58**, 1112–1114.

108. Cameron,D.E., Bashor,C.J. and Collins,J.J. (2014) A brief history of synthetic biology. *Nat. Rev. Microbiol.*, **12**, 381–390.

109. Khalil,A.S. and Collins,J.J. (2010) Synthetic biology: applications come of age. *Nat. Rev. Genet.*, **11**, 367–379.

110. Stephanopoulos,G. (2012) Synthetic biology and metabolic engineering. *ACS Synth. Biol.*, **1**, 514–525.

111. Scott,M., Gunderson,C.W., Mateescu,E.M., Zhang,Z. and Hwa,T. (2010) Interdependence of Cell Growth and Gene Expression: Origins and Consequences. *Science*, **330**, 1099–1102.

112. Fischbach,M.A., Walsh,C.T. and Clardy,J. (2008) The evolution of gene collectives: How natural selection drives chemical innovation. *Proc. Natl. Acad. Sci. U.S.A.*, **105**, 4601–4608.

113. Glick,B.R. (1995) Metabolic load and heterologous gene expression. *Biotechnol. Adv.*, **13**, 247–261.

114. Ceroni,F., Algar,R., Stan,G.-B. and Ellis,T. (2015) Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nature Methods*, **12**, 415–418.

115. Zelcbuch,L., Antonovsky,N., Bar-Even,A., Levin-Karp,A., Barenholz,U., Dayagi,M., Liebermeister,W., Flamholz,A., Noor,E., Amram,S., *et al.* (2013) Spanning high-dimensional expression space using ribosome-binding site combinatorics. *Nucleic Acids Res.*, **41**, e98–e98.

116. Pitera,D.J., Paddon,C.J., Newman,J.D. and Keasling,J.D. (2007) Balancing a heterologous mevalonate pathway for improved isoprenoid production in Escherichia coli. *Metab. Eng.*, **9**, 193–207.

117. Shachrai,I., Zaslaver,A., Alon,U. and Dekel,E. (2010) Cost of unneeded proteins in E. coli is reduced after several generations in exponential growth. *Mol. Cell*, **38**, 758–767.

118. Jones,P.R. (2014) Genetic instability in cyanobacteria - an elephant in the room? *Front Bioeng Biotechnol*, **2**, 12.

119. Jeschek,M., Gerngross,D. and Panke,S. (2016) Rationally reduced libraries for combinatorial pathway optimization minimizing experimental effort. *Nat Commun*, **7**, 11163.

120. Jensen,P.R. and Hammer,K. (1998) Artificial promoters for metabolic optimization. *Biotechnol. Bioeng.*, **58**, 191–195.

121. Hammer,K., Mijakovic,I. and Jensen,P.R. (2006) Synthetic promoter libraries--tuning of gene expression. *Trends Biotechnol.*, **24**, 53–55.

122. Anderson,J.C. (2009) Registry of Standard Biological Parts.

123. Kelly,C.L., Liu,Z., Yoshihara,A., Jenkinson,S.F., Wormald,M.R., Otero,J., Estévez,A., Kato,A., Marqvorsen,M.H.S., Fleet,G.W.J., *et al.* (2016) Synthetic Chemical Inducers and Genetic Decoupling Enable Orthogonal Control of the rhaBAD Promoter. *ACS Synth. Biol.*, **5**, 1136–1145.

124. Kelly,C.L., Taylor,G.M., Hitchcock,A., Torres-Méndez,A. and Heap,J.T. (2018) A rhamnose-inducible system for precise and temporal control of gene expression in cyanobacteria. *ACS Synth. Biol.*, 10.1021/acssynbio.7b00435.

125. Cox,R.S., Surette,M.G. and Elowitz,M.B. (2007) Programming gene expression with combinatorial promoters. *Mol. Syst. Biol.*, **3**, 145.

126. Murphy,K.F., Balázsi,G. and Collins,J.J. (2007) Combinatorial promoter design for engineering noisy gene expression. *Proceedings of the National Academy of Sciences*, **104**, 12726–12731.

127. Chen,Y.-J., Liu,P., Nielsen,A.A.K., Brophy,J.A.N., Clancy,K., Peterson,T. and Voigt,C.A. (2013) Characterization of 582 natural and synthetic terminators and quantification of their design constraints. *Nature Methods*, **10**, 659–664.

128. Seo,S.W., Yang,J.-S., Kim,I., Yang,J., Min,B.E., Kim,S. and Jung,G.Y. (2013) Predictive design of mRNA translation initiation region to control prokaryotic translation efficiency. *Metab. Eng.*, **15**, 67–74.

129. Na,D. and Lee,D. (2010) RBSDesigner: software for designing synthetic ribosome binding sites that yields a desired level of protein expression. *Bioinformatics*, **26**, 2633–2634.

130. Kosuri,S. and Church,G.M. (2014) Large-scale de novo DNA synthesis: technologies and applications. *Nature Methods*, **11**, 499–507.

131. Puigbò,P., Guzmán,E., Romeu,A. and Garcia-Vallvé,S. (2007) OPTIMIZER: a web server for optimizing the codon usage of DNA sequences. *Nucleic Acids Res.*, **35**, W126–31.

132. Andersen,J.B., Sternberg,C., Poulsen,L.K., Bjorn,S.P., Givskov,M. and Molin,S. (1998) New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria. *Appl. Environ. Microbiol.*, **64**, 2240–2246.

133. Mirza,N., Crocoll,C., Erik Olsen,C. and Ann Halkier,B. (2016) Engineering of methionine chain elongation part of glucoraphanin pathway in E. coli. *Metab. Eng.*, **35**, 31–37.

134. Zhang,J., Kao,E., Wang,G., Baidoo,E.E.K., Chen,M. and Keasling,J.D. (2016) Metabolic engineering of Escherichia coli for the biosynthesis of 2-pyrrolidone. *Metab Eng Commun*, **3**, 1–7.

135. Alonso-Gutierrez,J., Chan,R., Batth,T.S., Adams,P.D., Keasling,J.D., Petzold,C.J. and Lee,T.S. (2013) Metabolic engineering of Escherichia coli for limonene and perillyl alcohol production. *Metab. Eng.*, **19**, 33–41.

136. Alper,H., Fischer,C., Nevoigt,E. and Stephanopoulos,G. (2005) Tuning genetic control through promoter engineering. *Proc. Natl. Acad. Sci. U.S.A.*, **102**, 12678–12683.

137. Moore,S.J., Lai,H.-E., Kelwick,R.J.R., Chee,S.M., Bell,D.J., Polizzi,K.M. and Freemont,P.S. (2016) EcoFlex: A Multifunctional MoClo Kit for E. coli Synthetic Biology. *ACS Synth. Biol.*, **5**, 1059–1069.

138. Liu,R., Bassalo,M.C., Zeitoun,R.I. and Gill,R.T. (2015) Genome scale engineering techniques for metabolic engineering. *Metab. Eng.*, **32**, 143–154.

139. Borkowski,O., Bricio,C., Murgiano,M., Rothschild-Mancinelli,B., Stan,G.-B. and Ellis,T. (2018) Cell-free prediction of protein expression costs for growing cells. *Nat Commun*, **9**, 519.

140. Fowler,M., Beck,K., Brant,J., Opdyke,W. and Roberts,D. (2012) Refactoring Addison-Wesley.

141. Chan,L.Y., Kosuri,S. and Endy,D. (2005) Refactoring bacteriophage T7. *Mol. Syst. Biol.*, **1**, 2005.0018–E10.

142. Temme,K., Zhao,D. and Voigt,C.A. (2012) Refactoring the nitrogen fixation gene cluster from Klebsiella oxytoca. *Proc. Natl. Acad. Sci. U.S.A.*, **109**, 7085–7090.

143. Shao,Z., Rao,G., Li,C., Abil,Z., Luo,Y. and Zhao,H. (2013) Refactoring the Silent Spectinabilin Gene Cluster Using a Plug-and-Play Scaffold. *ACS Synth. Biol.*, **2**, 662–669.

144. Gimpel,J.A., Nour-Eldin,H.H., Scranton,M.A., Li,D. and Mayfield,S.P. (2016) Refactoring the Six-Gene Photosystem II Core in the Chloroplast of the Green Algae Chlamydomonas reinhardtii. *ACS Synth. Biol.*, **5**, 589–596.

145. Sánchez-Pascuala,A., de Lorenzo,V. and Nikel,P.I. (2017) Refactoring the Embden-Meyerhof-Parnas Pathway as a Whole of Portable GlucoBricks for

Implantation of Glycolytic Modules in Gram-Negative Bacteria. *ACS Synth. Biol.*, **6**, 793–805.

146. Schmidl,S.R., Sheth,R.U., Wu,A. and Tabor,J.J. (2014) Refactoring and optimization of light-switchable Escherichia coli two-component systems. *ACS Synth. Biol.*, **3**, 820–831.

147. Christen,M., Deutsch,S. and Christen,B. (2015) Genome Calligrapher: A Web Tool for Refactoring Bacterial Genome Sequences for de Novo DNA Synthesis. *ACS Synth. Biol.*, **4**, 927–934.

148. Ren,H., Hu,P. and Zhao,H. (2017) A plug-and-play pathway refactoring workflow for natural product research in Escherichia coliand Saccharomyces cerevisiae. *Biotechnol. Bioeng.*, **114**, 1847–1854.

149. Smanski,M.J., Bhatia,S., Zhao,D., Park,Y., B A Woodruff,L., Giannoukos,G., Ciulla,D., Busby,M., Calderon,J., Nicol,R., *et al.* (2014) Functional optimization of gene clusters by combinatorial design and assembly. *Nat. Biotechnol.*, **32**, 1241–1249.

150. Coussement,P., Bauwens,D., Maertens,J. and De Mey,M. (2017) Direct Combinatorial Pathway Optimization. *ACS Synth. Biol.*, **6**, 224–232.

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

152. Young,E.M., Zhao,Z., Gielesen,B.E.M., Wu,L., Benjamin Gordon,D., Roubos,J.A. and Voigt,C.A. (2018) Iterative algorithm-guided design of massive strain libraries, applied to itaconic acid production in yeast. *Metab. Eng.*, 10.1016/j.ymben.2018.05.002.

153. Kim,B., Du,J., Eriksen,D.T. and Zhao,H. (2013) Combinatorial design of a highly efficient xylose-utilizing pathway in Saccharomyces cerevisiae for the production of cellulosic biofuels. *Appl. Environ. Microbiol.*, **79**, 931–941.

154. Fehér,T., Planson,A.-G., Carbonell,P., Fernández-Castané,A., Grigoras,I., Dariy,E., Perret,A. and Faulon,J.-L. (2014) Validation of RetroPath, a computer-aided design tool for metabolic pathway engineering. *Biotechnol J*, **9**, 1446–1457.

155. Xu,P., Rizzoni,E.A., Sul,S.-Y. and Stephanopoulos,G. (2017) Improving Metabolic Pathway Efficiency by Statistical Model-Based Multivariate Regulatory Metabolic Engineering. *ACS Synth. Biol.*, **6**, 148–158.

156. Farasat,I., Kushwaha,M., Collens,J., Easterbrook,M., Guido,M. and Salis,H.M. (2014) Efficient search, mapping, and optimization of multi-protein genetic systems in diverse bacteria. *Mol. Syst. Biol.*, **10**, 731–731.

157. Xu,P., Gu,Q., Wang,W., Wong,L., Bower,A.G.W., Collins,C.H. and Koffas,M.A.G. (2013) Modular optimization of multi-gene pathways for fatty acids production in E. coli. *Nat Commun*, **4**, 493.

158. Gibson,D.G., Glass,J.I., Lartigue,C., Noskov,V.N., Chuang,R.-Y., Algire,M.A., Benders,G.A., Montague,M.G., Ma,L., Moodie,M.M., *et al.* (2010) Creation of a

bacterial cell controlled by a chemically synthesized genome. *Science*, **329**, 52–56.

159. Cohen,S.N., Chang,A.C.Y., Boyer,H.W. and Helling,R.B. (1973) Construction of Biologically Functional Bacterial Plasmids In Vitro. *Proceedings of the National Academy of Sciences*, **70**, 3240–3244.

160. Casini,A., Storch,M., Baldwin,G.S. and Ellis,T. (2015) Bricks and blueprints: methods and standards for DNA assembly. *Nature Reviews Molecular Cell Biology 2015 16:9*, **16**, 568–576.

161. Chang,A.C. and Cohen,S.N. (1978) Construction and characterization of amplifiable multicopy DNA cloning vehicles derived from the P15A cryptic miniplasmid. *Journal of Bacteriology*, **134**, 1141–1156.

162. Florea,M., Hagemann,H., Santosa,G., Abbott,J., Micklem,C.N., Spencer-Milnes,X., de Arroyo Garcia,L., Paschou,D., Lazenbatt,C., Kong,D., *et al.* (2016) Engineering control of bacterial cellulose production using a genetic toolkit and a new cellulose-producing strain. *Proc. Natl. Acad. Sci. U.S.A.*, **113**, E3431–40.

163. Engler,C., Kandzia,R. and Marillonnet,S. (2008) A one pot, one step, precision cloning method with high throughput capability. *PLoS ONE*, **3**, e3647.

164. Engler,C., Gruetzner,R., Kandzia,R. and Marillonnet,S. (2009) Golden gate shuffling: a one-pot DNA shuffling method based on type IIs restriction enzymes. *PLoS ONE*, **4**, e5553.

165. Weber,E., Engler,C., Gruetzner,R., Werner,S. and Marillonnet,S. (2011) A modular cloning system for standardized assembly of multigene constructs. *PLoS ONE*, **6**, e16765.

166. Sarrion-Perdigones,A., Falconi,E.E., Zandalinas,S.I., Juárez,P., Fernández-del-Carmen,A., Granell,A. and Orzaez,D. (2011) GoldenBraid: an iterative cloning system for standardized assembly of reusable genetic modules. *PLoS ONE*, **6**, e21622.

167. Sarrion-Perdigones,A., Vazquez-Vilar,M., Palací,J., Castelijns,B., Forment,J., Ziarsolo,P., Blanca,J., Granell,A. and Orzaez,D. (2013) GoldenBraid 2.0: a comprehensive DNA assembly framework for plant synthetic biology. *Plant Physiol.*, **162**, 1618–1631.

168. Engler,C., Youles,M., Gruetzner,R., Ehnert,T.-M., Werner,S., Jones,J.D.G., Patron,N.J. and Marillonnet,S. (2014) A golden gate modular cloning toolbox for plants. *ACS Synth. Biol.*, **3**, 839–843.

169. Lee,M.E., DeLoache,W.C., Cervantes,B. and Dueber,J.E. (2015) A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly. *ACS Synth. Biol.*, **4**, 975–986.

170. Agmon,N., Mitchell,L.A., Cai,Y., Ikushima,S., Chuang,J., Zheng,A., Choi,W.-J., Martin,J.A., Caravelli,K., Stracquadanio,G., *et al.* (2015) Yeast Golden Gate (yGG) for the Efficient Assembly of S. cerevisiae Transcription Units. *ACS Synth. Biol.*, **4**, 853–859.

171. Mitchell,L.A., Chuang,J., Agmon,N., Khunsriraksakul,C., Phillips,N.A., Cai,Y., Truong,D.M., Veerakumar,A., Wang,Y., Mayorga,M., *et al.* (2015) Versatile genetic assembly system (VEGAS) to assemble pathways for expression in S. cerevisiae. *Nucleic Acids Res.*, **43**, 6620–6630.

172. Kakui,Y., Sunaga,T., Arai,K., Dodgson,J., Ji,L., Csikász-Nagy,A., Carazo-Salas,R. and Sato,M. (2015) Module-based construction of plasmids for chromosomal integration of the fission yeast Schizosaccharomyces pombe. *Open Biol*, **5**, 150054.

173. Pérez-González,A., Kniewel,R., Veldhuizen,M., Verma,H.K., Navarro-Rodríguez,M., Rubio,L.M. and Caro,E. (2017) Adaptation of the GoldenBraid modular cloning system and creation of a toolkit for the expression of heterologous proteins in yeast mitochondria. *BMC Biotechnol.*, **17**, 80.

174. Iverson,S.V., Haddock,T.L., Beal,J. and Densmore,D.M. (2016) CIDAR MoClo: Improved MoClo Assembly Standard and New E. coli Part Library Enable Rapid Combinatorial Design for Synthetic and Traditional Biology. *ACS Synth. Biol.*, **5**, 99–103.

175. Storch,M., Casini,A., Mackrow,B., Fleming,T., Trewhitt,H., Ellis,T. and Baldwin,G.S. (2015) BASIC: A New Biopart Assembly Standard for Idempotent Cloning Provides Accurate, Single-Tier DNA Assembly for Synthetic Biology. *ACS Synth. Biol.*, **4**, 781–787.

176. Storch,M., Casini,A., Mackrow,B., Ellis,T. and Baldwin,G.S. (2017) BASIC: A Simple and Accurate Modular DNA Assembly Method. *Methods Mol. Biol.*, **1472**, 79–91.

177. Shetty,R.P., Endy,D. and Knight,T.F. (2008) Engineering BioBrick vectors from BioBrick parts. *J Biol Eng*, **2**, 5.

178. Anderson,J.C., Dueber,J.E., Leguia,M., Wu,G.C., Goler,J.A., Arkin,A.P. and Keasling,J.D. (2010) BglBricks: A flexible standard for biological part assembly. *J Biol Eng*, **4**, 1.

179. Lee,T.S., Krupa,R.A., Zhang,F., Hajimorad,M., Holtz,W.J., Prasad,N., Lee,S.K. and Keasling,J.D. (2011) BglBrick vectors and datasheets: A synthetic biology platform for gene expression. *J Biol Eng*, **5**, 12.

180. Hartley,J.L., Temple,G.F. and Brasch,M.A. (2000) DNA cloning using in vitro site-specific recombination. *Genome Res.*, **10**, 1788–1795.

181. Walhout,A.J., Temple,G.F., Brasch,M.A., Hartley,J.L., Lorson,M.A., van den Heuvel,S. and Vidal,M. (2000) GATEWAY recombinational cloning: application to the cloning of large numbers of open reading frames or ORFeomes. *Meth. Enzymol.*, **328**, 575–592.

182. Reboul,J., Vaglio,P., Tzellas,N., Thierry-Mieg,N., Moore,T., Jackson,C., Shin-i,T., Kohara,Y., Thierry-Mieg,D., Thierry-Mieg,J., *et al.* (2001) Open-reading-frame sequence tags (OSTs) support the existence of at least 17,300 genes in C. elegans. *Nat. Genet.*, **27**, 332–336.

183. Gibson,D.G., Young,L., Chuang,R.-Y., Venter,J.C., Hutchison,C.A. and Smith,H.O. (2009) Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods*, **6**, 343–345.

184. Beyer,H.M., Gonschorek,P., Samodelov,S.L., Meier,M., Weber,W. and Zurbriggen,M.D. (2015) AQUA Cloning: A Versatile and Simple Enzyme-Free Cloning Approach. *PLoS ONE*, **10**, e0137652.

185. Liang,J., Liu,Z., Low,X.Z., Ang,E.L. and Zhao,H. (2017) Twin-primer non-enzymatic DNA assembly: an efficient and accurate multi-part DNA assembly method. *Nucleic Acids Res.*, **45**, e94–e94.

186. de Kok,S., Stanton,L.H., Slaby,T., Durot,M., Holmes,V.F., Patel,K.G., Platt,D., Shapland,E.B., Serber,Z., Dean,J., *et al.* (2014) Rapid and reliable DNA assembly via ligase cycling reaction. *ACS Synth. Biol.*, **3**, 97–106.

187. Li,M.Z. and Elledge,S.J. (2007) Harnessing homologous recombination in vitro to generate recombinant DNA via SLIC. *Nature Methods*, **4**, 251–256.

188. Zhang,Y., Werling,U. and Edelmann,W. (2012) SLiCE: a novel bacterial cell extract-based DNA cloning method. *Nucleic Acids Res.*, **40**, e55–e55.

189. Horton,R.M., Hunt,H.D., Ho,S.N., Pullen,J.K. and Pease,L.R. (1989) Engineering hybrid genes without the use of restriction enzymes: gene splicing by overlap extension. *Gene*, **77**, 61–68.

190. Bitinaite,J., Rubino,M., Varma,K.H., Schildkraut,I., Vaisvila,R. and Vaiskunaite,R. (2007) USER friendly DNA engineering and cloning method by uracil excision. *Nucleic Acids Res.*, **35**, 1992–2002.

191. Zou,R., Zhou,K., Stephanopoulos,G. and Too,H.P. (2013) Combinatorial engineering of 1-deoxy-D-xylulose 5-phosphate pathway using cross-lapping in vitro assembly (CLIVA) method. *PLoS ONE*, **8**, e79557.

192. Quan,J. and Tian,J. (2009) Circular polymerase extension cloning of complex gene libraries and pathways. *PLoS ONE*, **4**, e6441.

193. Manual,B.A. (2002) BAM: Enumeration of Escherichia coli and the Coliform Bacteria.

194. Clark,D.P. (1989) The fermentation pathways of Escherichia coli. *FEMS Microbiology Reviews*, **5**, 223–234.

195. Causey,T.B., Shanmugam,K.T., Yomano,L.P. and Ingram,L.O. (2004) Engineering Escherichia coli for efficient conversion of glucose to pyruvate. *Proc. Natl. Acad. Sci. U.S.A.*, **101**, 2235–2240.

196. Markovich,G.M., Georgievna,R.Y., Yurievich,K.M. and Yurienva,R.A. (2016) A Method for Producing an L-Amino Acid Using a Bacterium of the Family Enterobacteriaceae Having Disrupted Putrescine Degradation Pathway.

197. Shen,C.R., Lan,E.I., Dekishima,Y., Baez,A., Cho,K.M. and Liao,J.C. (2011) Driving forces enable high-titer anaerobic 1-butanol synthesis in Escherichia coli. *Appl. Environ. Microbiol.*, **77**, 2905–2915.

198. Ohtake,T., Pontrelli,S., Laviña,W.A., Liao,J.C., Putri,S.P. and Fukusaki,E. (2017) Metabolomics-driven approach to solving a CoA imbalance for improved 1-butanol production in Escherichia coli. *Metab. Eng.*, **41**, 135–143.

199. Atsumi,S., Wu,T.-Y., Eckl,E.-M., Hawkins,S.D., Buelter,T. and Liao,J.C. (2010) Engineering the isobutanol biosynthetic pathway in Escherichia coli by comparison of three aldehyde reductase/alcohol dehydrogenase genes. *Appl. Microbiol. Biotechnol.*, **85**, 651–657.

200. Yang,J., Nie,Q., Liu,H., Xian,M. and Liu,H. (2016) A novel MVA-mediated pathway for isoprene production in engineered E. coli. *BMC Biotechnol.*, **16**, 5.

201. Lee,S.Y., Kim,H.M. and Cheon,S. (2015) Metabolic engineering for the production of hydrocarbon fuels. *Current Opinion in Biotechnology*, **33**, 15–22.

202. Yang,J.E., Kim,J.W., Oh,Y.H., Choi,S.Y., Lee,H., Park,A.-R., Shin,J., Park,S.J. and Lee,S.Y. (2016) Biosynthesis of poly(2-hydroxyisovalerate-co-lactate) by metabolically engineered Escherichia coli. *Biotechnol J*, **11**, 1572–1585.

203. Rad,S.A., Zahiri,H.S., Noghabi,K.A., Rajaei,S., Heidari,R. and Mojallali,L. (2012) Type 2 IDI performs better than type 1 for improving lycopene production in metabolically engineered E. coli strains. *World J. Microbiol. Biotechnol.*, **28**, 313–321.

204. Xu,J., Xu,X., Xu,Q., Zhang,Z., Jiang,L. and Huang,H. (2018) Efficient production of lycopene by engineered E. coli strains harboring different types of plasmids. *Bioprocess Biosyst Eng*, **41**, 489–499.

205. Yoon,S.-H., Park,H.-M., Kim,J.-E., Lee,S.-H., Choi,M.-S., Kim,J.-Y., Oh,D.-K., Keasling,J.D. and Kim,S.-W. (2007) Increased beta-carotene production in recombinant Escherichia coli harboring an engineered isoprenoid precursor pathway with mevalonate addition. *Biotechnology Progress*, **23**, 599–605.

206. Yang,J. and Guo,L. (2014) Biosynthesis of β-carotene in engineered E. coli using the MEP and MVA pathways. *Microb. Cell Fact.*, **13**, 160.

207. Lu,Q., Bu,Y.-F. and Liu,J.-Z. (2017) Metabolic Engineering of Escherichia coli for Producing Astaxanthin as the Predominant Carotenoid. *Marine Drugs*, **15**, 296.

208. Pontrelli,S., Chiu,T.-Y., Lan,E.I., Chen,F.Y.-H., Chang,P. and Liao,J.C. (2018) Escherichia coli as a host for metabolic engineering. *Metab. Eng.*, 10.1016/j.ymben.2018.04.008.

209. Gottschalk,G. (2012) Bacterial Metabolism Springer Science & Business Media, New York, NY.

210. Poliquin,K., Ershov,Y.V., Cunningham,F.X., Woreta,T.T., Gantt,R.R. and Gantt,E. (2004) Inactivation of sll1556 in Synechocystis strain PCC 6803 impairs isoprenoid biosynthesis from pentose phosphate cycle substrates in vitro. *Journal of Bacteriology*, **186**, 4685–4693.

211. Boulton,C. and Quain,D. (2013) Brewing Yeast and Fermentation John Wiley & Sons.

212. Ma,M. and Liu,Z.L. (2010) Mechanisms of ethanol tolerance in Saccharomyces cerevisiae. *Appl. Microbiol. Biotechnol.*, **87**, 829–845.

213. Nielsen,J. and Jewett,M.C. (2008) Impact of systems biology on metabolic engineering of Saccharomyces cerevisiae. *FEMS Yeast Res.*, **8**, 122–131.

214. Festel,G.W. (2008) Biofuels – Economic Aspects. *Chemical Engineering & Technology*, **31**, 715–720.

215. Steen,E.J., Chan,R., Prasad,N., Myers,S., Petzold,C.J., Redding,A., Ouellet,M. and Keasling,J.D. (2008) Metabolic engineering of Saccharomyces cerevisiae for the production of n-butanol. *Microb. Cell Fact.*, **7**, 36.

216. Matsuda,F., Ishii,J., Kondo,T., Ida,K., Tezuka,H. and Kondo,A. (2013) Increased isobutanol production in Saccharomyces cerevisiae by eliminating competing pathways and resolving cofactor imbalance. *Microb. Cell Fact.*, **12**, 119.

217. Tokuhiro,K., Ishida,N., Kondo,A. and Takahashi,H. (2008) Lactic fermentation of cellobiose by a yeast strain displaying β-glucosidase on the cell surface. *Appl. Microbiol. Biotechnol.*, **79**, 481–488.

218. Jansen,M.L.A., Van De Graaf,M.J. and Verwaal,R. (2012) Dicarboxylic Acid Production Process.

219. Weber,C., Brückner,C., Weinreb,S., Lehr,C., Essl,C. and Boles,E. (2012) Biosynthesis of cis, cis-Muconic Acid and Its Aromatic Precursors, Catechol and Protocatechuic Acid, from Renewable Feedstocks by Saccharomyces cerevisiae. *Appl. Environ. Microbiol.*, **78**, 8421–8430.

220. Curran,K.A., Leavitt,J.M., Karim,A.S. and Alper,H.S. (2013) Metabolic engineering of muconic acid production in Saccharomyces cerevisiae. *Metab. Eng.*, **15**, 55–66.

221. Ro,D.-K., Paradise,E.M., Ouellet,M., Fisher,K.J., Newman,K.L., Ndungu,J.M., Ho,K.A., Eachus,R.A., Ham,T.S., Kirby,J., *et al.* (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, **440**, 940–943.

222. E4tech (2015) From the Sugar Platform to Biofuels and Biochemicals.

223. Rittmann,B.E. (2008) Opportunities for renewable bioenergy using microorganisms. *Biotechnol. Bioeng.*, **100**, 203–212.

224. Sanderson,K. (2011) Lignocellulose: A chewy problem. *Nature*, **474**, S12–4.

225. US EPA,O.A. (2016) Global Greenhouse Gas Emissions Data. 10.3334/CDIAC/00001_V2017.

226. Keasling,J.D. (2010) Manufacturing molecules through metabolic engineering. *Science*, **330**, 1355–1358.

227. Ducat,D.C., Way,J.C. and Silver,P.A. (2011) Engineering cyanobacteria to generate high-value products. *Trends Biotechnol.*, **29**, 95–103.

228. Bao,H., Liu,R., Liang,L., Jiang,Y., Jiang,M., Ma,J., Chen,K., Jia,H., Wei,P. and Ouyang,P. (2014) Succinic acid production from hemicellulose hydrolysate by an Escherichia coli mutant obtained by atmospheric and room temperature plasma and adaptive evolution. *Enzyme Microb. Technol.*, **66**, 10–15.

229. Saini,M., Lin,L.-J., Chiang,C.-J. and Chao,Y.-P. (2017) Synthetic Consortium of Escherichia colifor n-Butanol Production by Fermentation of the Glucose–Xylose Mixture. *J. Agric. Food Chem.*, **65**, 10040–10047.

230. Gong,F., Liu,G., Zhai,X., Zhou,J., Cai,Z. and Li,Y. (2015) Quantitative analysis of an engineered CO2-fixing Escherichia coli reveals great potential of heterotrophic CO2 fixation. *Biotechnology for Biofuels*, **8**, 86.

231. Bonacci,W., Teng,P.K., Afonso,B., Niederholtmeyer,H., Grob,P., Silver,P.A. and Savage,D.F. (2012) Modularity of a carbon-fixing protein organelle. *Proc. Natl. Acad. Sci. U.S.A.*, **109**, 478–483.

232. Murarka,A., Dharmadi,Y., Yazdani,S.S. and Gonzalez,R. (2008) Fermentative utilization of glycerol by Escherichia coli and its implications for the production of fuels and chemicals. *Appl. Environ. Microbiol.*, **74**, 1124–1135.

233. Canganella,F. and Wiegel,J. (2014) Anaerobic thermophiles. *Life (Basel)*, **4**, 77–104.

234. Ma,Y., Galinski,E.A., Grant,W.D., Oren,A. and Ventosa,A. (2010) Halophiles 2010: life in saline environments. *Appl. Environ. Microbiol.*, **76**, 6971–6981.

235. Reeve,B., Martinez-Klimova,E., de Jonghe,J., Leak,D.J. and Ellis,T. (2016) The Geobacillus Plasmid Set: A Modular Toolkit for Thermophile Engineering. *ACS Synth. Biol.*, **5**, 1342–1347.

236. Hanson,R.S. and Hanson,T.E. (1996) Methanotrophic bacteria. *Microbiol. Rev.*, **60**, 439–471.

237. Angermayr,S.A., Paszota,M. and Hellingwerf,K.J. (2012) Engineering a cyanobacterial cell factory for production of lactic acid. *Appl. Environ. Microbiol.*, **78**, 7098–7106.

238. De Tissera,S., Köpke,M., Simpson,S.D., Humphreys,C., Minton,N.P. and Dürre,P. (2017) Syngas Biorefinery and Syngas Utilization. *Adv. Biochem. Eng. Biotechnol.*, **245**, 2–34.

239. Springer (2006) The Prokaryotes (Vol-2), Handbook of the Biology of Bacteria,3rd Ed, 2006 Bukupedia.

240. Köpke,M., Mihalcea,C., Liew,F., Tizard,J.H., Ali,M.S., Conolly,J.J., Al-Sinawi,B. and Simpson,S.D. (2011) 2,3-butanediol production by acetogenic bacteria, an alternative route to chemical synthesis, using industrial waste gas. *Appl. Environ. Microbiol.*, **77**, 5467–5475.

241. Liew,F., Henstra,A.M., Köpke,M., Winzer,K., Simpson,S.D. and Minton,N.P. (2017) Metabolic engineering of Clostridium autoethanogenum for selective alcohol production. *Metab. Eng.*, **40**, 104–114.

242. Yu,X., Jin,H., Liu,W., Wang,Q. and Qi,Q. (2015) Engineering Corynebacterium glutamicum to produce 5-aminolevulinic acid from glucose. *Microb. Cell Fact.*, **14**, 183.

243. Tan,D., Xue,Y.-S., Aibaidula,G. and Chen,G.-Q. (2011) Unsterile and continuous production of polyhydroxybutyrate by Halomonas TD01. *Bioresour. Technol.*, **102**, 8130–8136.

244. Tan,D., Wu,Q., Chen,J.-C. and Chen,G.-Q. (2014) Engineering Halomonas TD01 for the low-cost production of polyhydroxyalkanoates. *Metab. Eng.*, **26**, 34–47.

245. Tao,W., Lv,L. and Chen,G.-Q. (2017) Engineering Halomonas species TD01 for enhanced polyhydroxyalkanoates synthesis via CRISPRi. *Microb. Cell Fact.*, **16**, 48.

246. Zhao,H., Zhang,H.M., Chen,X., Li,T., Wu,Q., Ouyang,Q. and Chen,G.-Q. (2017) Novel T7-like expression systems used for Halomonas. *Metab. Eng.*, **39**, 128–140.

247. Ishizaki,A., Tanaka,K. and Taga,N. (2001) Microbial production of poly-D-3-hydroxybutyrate from CO2. *Appl. Microbiol. Biotechnol.*, **57**, 6–12.

248. Crépin,L., Lombard,E. and Guillouet,S.E. (2016) Metabolic engineering of Cupriavidus necator for heterotrophic and autotrophic alka(e)ne production. *Metab. Eng.*, **37**, 92–101.

249. Marc,J., Grousseau,E., Lombard,E., Sinskey,A.J., Gorret,N. and Guillouet,S.E. (2017) Over expression of GroESL in Cupriavidus necator for heterotrophic and autotrophic isopropanol production. *Metab. Eng.*, **42**, 74–84.

250. Sharpe,P.L., Dicosimo,D., Bosak,M.D., Knoke,K., Tao,L., Cheng,Q. and Ye,R.W. (2007) Use of transposon promoter-probe vectors in the metabolic engineering of the obligate methanotroph Methylomonas sp. strain 16a for enhanced C40 carotenoid synthesis. *Appl. Environ. Microbiol.*, **73**, 1721–1728.

251. Silverman,J. and Joseph,T. (2014) Propylene synthesis using engineered enzymes.

252. Intrexon Pilot Plant for its Methanotroph Bioconversion Platform Now Operational (2016) Intrexon Pilot Plant for its Methanotroph Bioconversion Platform Now Operational.

253. Intrexon's Industrial Products Division Achieves Bioconversion of Methane to Farnesene (2014) Intrexon's Industrial Products Division Achieves Bioconversion of Methane to Farnesene.

254. Romero,S., Merino,E., Bolívar,F., Gosset,G. and Martinez,A. (2007) Metabolic engineering of Bacillus subtilis for ethanol production: lactate dehydrogenase plays a key role in fermentative metabolism. *Appl. Environ. Microbiol.*, **73**, 5190–5198.

255. Shi,S., Chen,T., Zhang,Z., Chen,X. and Zhao,X. (2009) Transcriptome analysis guided metabolic engineering of Bacillus subtilis for riboflavin production. *Metab. Eng.*, **11**, 243–252.

256. Yoshida,K., Ueda,S. and Maeda,I. (2009) Carotenoid production in Bacillus subtilis achieved by metabolic engineering. *Biotechnol. Lett.*, **31**, 1789–1793.

257. Ledesma-Amaro,R., Dulermo,T. and Nicaud,J.-M. (2015) Engineering Yarrowia lipolytica to produce biodiesel from raw starch. *Biotechnology for Biofuels*, **8**, 148.

258. Ledesma-Amaro,R., Lazar,Z., Rakicka,M., Guo,Z., Fouchard,F., Coq,A.-M.C.-L. and Nicaud,J.-M. (2016) Metabolic engineering of Yarrowia lipolytica to produce chemicals and fuels from xylose. *Metab. Eng.*, **38**, 115–124.

259. Larroude,M., Celinska,E., Back,A., Thomas,S., Nicaud,J.-M. and Ledesma-Amaro,R. (2018) A synthetic biology approach to transform Yarrowia lipolytica into a competitive biotechnological producer of β-carotene. *Biotechnol. Bioeng.*, **115**, 464–472.

260. Niehus,X., Crutz-Le Coq,A.-M., Sandoval,G., Nicaud,J.-M. and Ledesma-Amaro,R. (2018) Engineering Yarrowia lipolytica to enhance lipid production from lignocellulosic materials. *Biotechnology for Biofuels*, **11**, 11.

261. Walsh,G. (2014) Biopharmaceutical benchmarks 2014. *Nat. Biotechnol.*, **32**, 992–1000.

262. Cripps,R.E., Eley,K., Leak,D.J., Rudd,B., Taylor,M., Todd,M., Boakes,S., Martin,S. and Atkinson,T. (2009) Metabolic engineering of Geobacillus thermoglucosidasius for high yield ethanol production. *Metab. Eng.*, **11**, 398–408.

263. Angermayr,S.A., Gorchs Rovira,A. and Hellingwerf,K.J. (2015) Metabolic engineering of cyanobacteria for the synthesis of commodity products. *Trends Biotechnol.*, **33**, 352–361.

264. European Environment Agency. (2018) Atmospheric greenhouse gas concentrations Atmospheric greenhouse gas concentrations. Retrieved from https://www.eea.europa.eu/data-and-maps/indicators/atmospheric-greenhouse-gas-concentrations-10/assessment.

265. Witcover,J., Yeh,S. and Sperling,D. (2013) Policy options to address global land use change from biofuels. *Energy Policy*, **56**, 63–74.

266. Heidorn,T., Camsund,D., Huang,H.-H., Lindberg,P., Oliveira,P., Stensjö,K. and Lindblad,P. (2011) Synthetic biology in cyanobacteria engineering and analyzing novel functions. *Meth. Enzymol.*, **497**, 539–579.

267. Pulz,O. and Gross,W. (2004) Valuable products from biotechnology of microalgae. *Appl. Microbiol. Biotechnol.*, **65**, 635–648.

268. Spolaore,P., Joannis-Cassan,C., Duran,E. and Isambert,A. (2006) Commercial applications of microalgae. *J. Biosci. Bioeng.*, **101**, 87–96.

269. Wang,W., Liu,X. and Lu,X. (2013) Engineering cyanobacteria to improve photosynthetic production of alka(e)nes. *Biotechnology for Biofuels*, **6**, 69.

270. Tyo,K.E.J., Jin,Y.-S., Espinoza,F.A. and Stephanopoulos,G. (2009) Identification of gene disruptions for increased poly-3-hydroxybutyrate

accumulation in Synechocystis PCC 6803. *Biotechnology Progress*, **25**, 1236–1243.

271. Kamravamanesh,D., Pflügl,S., Nischkauer,W., Limbeck,A., Lackner,M. and Herwig,C. (2017) Photosynthetic poly-β-hydroxybutyrate accumulation in unicellular cyanobacterium Synechocystis sp. PCC 6714. *AMB Express*, **7**, 143.

272. Kallio,P., Pásztor,A., Akhtar,M.K. and Jones,P.R. (2014) Renewable jet fuel. *Current Opinion in Biotechnology*, **26**, 50–55.

273. Jagadevan,S., Banerjee,A., Banerjee,C., Guria,C., Tiwari,R., Baweja,M. and Shukla,P. (2018) Recent developments in synthetic biology and metabolic engineering in microalgae towards biofuel production. *Biotechnology for Biofuels*, **11**, 743.

274. Lauersen,K.J., Wichmann,J., Baier,T., Kampranis,S.C., Pateraki,I., Møller,B.L. and Kruse,O. (2018) Phototrophic production of heterologous diterpenoids and a hydroxy-functionalized derivative from Chlamydomonas reinhardtii. *Metab. Eng.*, **49**, 116–127.

275. Chang,W.-C., Zheng,H.-Q. and Chen,C.-N.N. (2016) Comparative transcriptome analysis reveals a potential photosynthate partitioning mechanism between lipid and starch biosynthetic pathways in green microalgae. *Algal Research*, **16**, 54–62.

276. Liang,M.-H. and Jiang,J.-G. (2013) Advancing oleaginous microorganisms to produce lipid via metabolic engineering technology. *Progress in Lipid Research*, **52**, 395–408.

277. Yunus,I.S., Wichmann,J., Wördenweber,R., Lauersen,K.J., Kruse,O. and Jones,P.R. (2018) Synthetic metabolic pathways for photobiological conversion of CO2 into hydrocarbon fuel. *Metab. Eng.*, **49**, 201–211.

278. Aikawa,S., Joseph,A., Yamada,R., Izumi,Y., Yamagishi,T., Matsuda,F., Kawai,H., Chang,J.-S., Hasunuma,T. and Kondo,A. (2013) Direct conversion of Spirulina to ethanol without pretreatment or enzymatic hydrolysis processes. *Energy & Environmental Science*, **6**, 1844–1849.

279. John,R.P., Anisha,G.S., Nampoothiri,K.M. and Pandey,A. (2011) Micro and macroalgal biomass: A renewable source for bioethanol. *Bioresour. Technol.*, **102**, 186–193.

280. Savakis,P. and Hellingwerf,K.J. (2015) Engineering cyanobacteria for direct biofuel production from CO2. *Current Opinion in Biotechnology*, **33**, 8–14.

281. Gudmundsson,S. and Nogales,J. (2015) Cyanobacteria as photosynthetic biocatalysts: a systems biology perspective. *Mol Biosyst*, **11**, 60–70.

282. Deng,M.D. and Coleman,J.R. (1999) Ethanol synthesis by genetic engineering in cyanobacteria. *Appl. Environ. Microbiol.*, **65**, 523–528.

283. Dexter,J. and Fu,P. (2009) Metabolic engineering of cyanobacteria for ethanol production. *Energy & Environmental Science*, **2**, 857–864.

284. Gao,Z., Zhao,H., Li,Z., Tan,X. and Lu,X. (2012) Photosynthetic production of ethanol from carbon dioxide in genetically engineered cyanobacteria. *Energy & Environmental Science*, **5**, 9857–9865.

285. Choi,Y.-N. and Park,J.M. (2016) Enhancing biomass and ethanol production by increasing NADPH production in Synechocystis sp. PCC 6803. *Bioresour. Technol.*, **213**, 54–57.

286. Namakoshi,K., Nakajima,T., Yoshikawa,K., Toya,Y. and Shimizu,H. (2016) Combinatorial deletions of glgC and phaCE enhance ethanol production in Synechocystis sp. PCC 6803. *Journal of Biotechnology*, **239**, 13–19.

287. Lan,E.I. and Liao,J.C. (2011) Metabolic engineering of cyanobacteria for 1-butanol production from carbon dioxide. *Metab. Eng.*, **13**, 353–363.

288. Lan,E.I. and Liao,J.C. (2012) ATP drives direct photosynthetic production of 1-butanol in cyanobacteria. *Proc. Natl. Acad. Sci. U.S.A.*, **109**, 6018–6023.

289. Lan,E.I., Ro,S.Y. and Liao,J.C. (2013) Oxygen-tolerant coenzyme A-acylating aldehyde dehydrogenase facilitates efficient photosynthetic n-butanol biosynthesis in cyanobacteria. *Energy & Environmental Science*, **6**, 2672–2681.

290. Varman,A.M., Xiao,Y., Pakrasi,H.B. and Tang,Y.J. (2013) Metabolic engineering of Synechocystis sp. strain PCC 6803 for isobutanol production. *Appl. Environ. Microbiol.*, **79**, 908–914.

291. Atsumi,S., Higashide,W. and Liao,J.C. (2009) Direct photosynthetic recycling of carbon dioxide to isobutyraldehyde. *Nat. Biotechnol.*, **27**, 1177–1180.

292. Miao,R., Xie,H., M Ho,F. and Lindblad,P. (2018) Protein engineering of α-ketoisovalerate decarboxylase for improved isobutanol production in Synechocystis PCC 6803. *Metab. Eng.*, **47**, 42–48.

293. Shen,C.R. and Liao,J.C. (2012) Photosynthetic production of 2-methyl-1-butanol from CO2 in cyanobacterium Synechococcus elongatus PCC7942 and characterization of the native acetohydroxyacid synthase. *Energy & Environmental Science*, **5**, 9574.

294. Savakis,P.E., Angermayr,S.A. and Hellingwerf,K.J. (2013) Synthesis of 2,3-butanediol by Synechocystis sp. PCC6803 via heterologous expression of a catabolic pathway from lactic acid- and enterobacteria. *Metab. Eng.*, **20**, 121–130.

295. Nozzi,N.E. and Atsumi,S. (2015) Genome Engineering of the 2,3-Butanediol Biosynthetic Pathway for Tight Regulation in Cyanobacteria. *ACS Synth. Biol.*, **4**, 1197–1204.

296. Nozzi,N.E., Case,A.E., Carroll,A.L. and Atsumi,S. (2017) Systematic Approaches to Efficiently Produce 2,3-Butanediol in a Marine Cyanobacterium. *ACS Synth. Biol.*, **6**, 2136–2144.

297. Oliver,J.W.K., Machado,I.M.P., Yoneda,H. and Atsumi,S. (2013) Cyanobacterial conversion of carbon dioxide to 2,3-butanediol. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 1249–1254.

298. Oliver,J.W.K., Machado,I.M.P., Yoneda,H. and Atsumi,S. (2014) Combinatorial optimization of cyanobacterial 2,3-butanediol production. *Metab. Eng.*, **22**, 76–82.

299. Li,H. and Liao,J.C. (2013) Engineering a cyanobacterium as the catalyst for the photosynthetic conversion of CO2 to 1,2-propanediol. *Microb. Cell Fact.*, **12**, 4.

300. Yunus,I.S. and Jones,P.R. (2018) Photosynthesis-dependent biosynthesis of medium chain-length fatty acids and alcohols. *Metab. Eng.*, 10.1016/j.ymben.2018.07.015.

301. Angermayr,S., van der Woude,A.D., Correddu,D., Vreugdenhil,A., Verrone,V. and Hellingwerf,K.J. (2014) Exploring metabolic engineering design principles for the photosynthetic production of lactic acid by Synechocystis sp. PCC6803. *Biotechnology for Biofuels*, **7**, 99.

302. Angermayr,S.A. and Hellingwerf,K.J. (2013) On the use of metabolic control analysis in the optimization of cyanobacterial biosolar cell factories. *J Phys Chem B*, **117**, 11169–11175.

303. Joseph,A., Aikawa,S., Sasaki,K., Tsuge,Y., Matsuda,F., Tanaka,T. and Kondo,A. (2013) Utilization of lactic acid bacterial genes in Synechocystis sp. PCC 6803 in the production of lactic acid. *Biosci. Biotechnol. Biochem.*, **77**, 966–970.

304. Xue,Y., Zhang,Y., Grace,S. and He,Q. (2013) Functional expression of an Arabidopsis p450 enzyme, p-coumarate-3-hydroxylase, in the cyanobacterium Synechocystis PCC 6803 for the biosynthesis of caffeic acid. *Journal of Applied Phycology*, **26**, 219–226.

305. Varman,A.M., Yu,Y., You,L. and Tang,Y.J. (2013) Photoautotrophic production of D-lactic acid in an engineered cyanobacterium. *Microb. Cell Fact.*, **12**, 117.

306. Niederholtmeyer,H., Wolfstadter,B.T., Savage,D.F., Silver,P.A. and Way,J.C. (2010) Engineering Cyanobacteria To Synthesize and Export Hydrophilic Products. *Appl. Environ. Microbiol.*, **76**, 6023–6023.

307. Englund,E., Andersen-Ranberg,J., Miao,R., Hamberger,B. and Lindberg,P. (2015) Metabolic Engineering of Synechocystis sp. PCC 6803 for Production of the Plant Diterpenoid Manoyl Oxide. *ACS Synth. Biol.*, **4**, 1270–1278.

308. Englund,E., Shabestary,K., Hudson,E.P. and Lindberg,P. (2018) Systematic overexpression study to find target enzymes enhancing production of terpenes in Synechocystis PCC 6803, using isoprene as a model compound. *Metab. Eng.*, 10.1016/j.ymben.2018.07.004.

309. Formighieri,C. and Melis,A. (2017) Heterologous synthesis of geranyllinalool, a diterpenol plant product, in the cyanobacterium Synechocystis. *Appl. Microbiol. Biotechnol.*, **101**, 2791–2800.

310. Halfmann,C., Gu,L. and Zhou,R. (2014) Engineering cyanobacteria for the production of a cyclic hydrocarbon fuel from CO 2and H 2O. *Green Chem.*, **16**, 3175–3185.

311. Davies,F.K., Work,V.H., Beliaev,A.S. and Posewitz,M.C. (2014) Engineering Limonene and Bisabolene Production in Wild Type and a Glycogen-Deficient Mutant of Synechococcus sp. PCC 7002. *Front Bioeng Biotechnol*, **2**, 21.

312. Kiyota,H., Okuda,Y., Ito,M., Hirai,M.Y. and Ikeuchi,M. (2014) Engineering of cyanobacteria for the photosynthetic production of limonene from CO2. *Journal of Biotechnology*, **185**, 1–7.

313. Bentley,F.K., García-Cerdán,J.G., Chen,H.-C. and Melis,A. (2013) Paradigm of Monoterpene (β-phellandrene) Hydrocarbons Production via Photosynthesis in Cyanobacteria. *BioEnergy Research*, **6**, 917–929.

314. Formighieri,C. and Melis,A. (2014) Regulation of β-phellandrene synthase gene expression, recombinant protein accumulation, and monoterpene hydrocarbons production in Synechocystis transformants. *Planta*, **240**, 309–324.

315. Reinsvold,R.E., Jinkerson,R.E., Radakovits,R., Posewitz,M.C. and Basu,C. (2011) The production of the sesquiterpene β-caryophyllene in a transgenic strain of the cyanobacterium Synechocystis. *Journal of Plant Physiology*, **168**, 848–852.

316. Englund,E., Pattanaik,B., Ubhayasekera,S.J.K., Stensjö,K., Bergquist,J. and Lindberg,P. (2014) Production of squalene in Synechocystis sp. PCC 6803. *PLoS ONE*, **9**, e90270.

317. Du,W., Liang,F., Duan,Y., Tan,X. and Lu,X. (2013) Exploring the photosynthetic production capacity of sucrose by cyanobacteria. *Metab. Eng.*, **19**, 17–25.

318. Ducat,D.C., Avelar-Rivas,J.A., Way,J.C. and Silver,P.A. (2012) Rerouting carbon flux to enhance photosynthetic productivity. *Appl. Environ. Microbiol.*, **78**, 2660–2668.

319. Savakis,P., Tan,X., Du,W., Branco dos Santos,F., Lu,X. and Hellingwerf,K.J. (2015) Photosynthetic production of glycerol by a recombinant cyanobacterium. *Journal of Biotechnology*, **195**, 46–51.

320. Jacobsen,J.H. and Frigaard,N.-U. (2014) Engineering of photosynthetic mannitol biosynthesis from CO2 in a cyanobacterium. *Metab. Eng.*, **21**, 60–70.

321. Zhou,J., Zhang,H., Zhang,Y., Li,Y. and Ma,Y. (2012) Designing and creating a modularized synthetic pathway in cyanobacterium Synechocystis enables production of acetone from carbon dioxide. *Metab. Eng.*, **14**, 394–400.

322. Liu,X., Sheng,J. and Curtiss,R. (2011) Fatty acid production in genetically modified cyanobacteria. *Proc. Natl. Acad. Sci. U.S.A.*, **108**, 6899–6904.

323. Kawahara,A., Sato,Y., Saito,Y., Kaneko,Y., Takimura,Y., Hagihara,H. and Hihara,Y. (2016) Free fatty acid production in the cyanobacterium Synechocystis sp. PCC 6803 is enhanced by deletion of the cyAbrB2 transcriptional regulator. *Journal of Biotechnology*, **220**, 1–11.

324. Lee,H.J., Choi,J., Lee,S.-M., Um,Y., Sim,S.J., Kim,Y. and Woo,H.M. (2017) Photosynthetic CO2 Conversion to Fatty Acid Ethyl Esters (FAEEs) Using Engineered Cyanobacteria. *J. Agric. Food Chem.*, **65**, 1087–1092.

325. Sakai,M., Ogawa,T., Matsuoka,M. and Fukuda,H. (1997) Photosynthetic conversion of carbon dioxide to ethylene by the recombinant cyanobacterium, Synechococcus sp. PCC 7942, which harbors a gene for the ethylene-forming enzyme of Pseudomonas syringae. *Journal of Fermentation and Bioengineering*, **84**, 434–443.

326. Ungerer,J., Tao,L., Davis,M., Ghirardi,M., Maness,P.-C. and Yu,J. (2012) Sustained photosynthetic conversion of CO2 to ethylene in recombinant cyanobacterium Synechocystis 6803. *Energy & Environmental Science*, **5**, 8998.

327. Takahama,K., Matsuoka,M., Nagahama,K. and Ogawa,T. (2003) Construction and analysis of a recombinant cyanobacterium expressing a chromosomally inserted gene for an ethylene-forming enzyme at the psbAI locus. *J. Biosci. Bioeng.*, **95**, 302–305.

328. Ranade,S., Zhang,Y., Kaplan,M., Majeed,W. and He,Q. (2015) Metabolic Engineering and Comparative Performance Studies of Synechocystis sp. PCC 6803 Strains for Effective Utilization of Xylose. *Front Microbiol*, **6**, 1484.

329. Kanno,M. and Atsumi,S. (2017) Engineering an Obligate Photoautotrophic Cyanobacterium to Utilize Glycerol for Growth and Chemical Production. *ACS Synth. Biol.*, **6**, 69–75.

330. Knoll,A.H. (2008) Cyanobacteria and Earth History.

331. Schirrmeister,B.E., Gugger,M. and Donoghue,P.C.J. (2015) Cyanobacteria and the Great Oxidation Event: evidence from genes and fossils. *Palaeontology*, **58**, 769–785.

332. Whitton,B.A. and Potts,M. (2007) The Ecology of Cyanobacteria Whitton,B.A., Potts,M. (eds) Springer Science & Business Media, Dordrecht.

333. Bhattacharya,D. (2012) Origins of Algae and their Plastids Bhattacharya,D. (ed) Springer Science & Business Media, Vienna.

334. Liu,H., Nolla,H.A. and Campbell,L. (1997) Prochlorococcus growth rate and contribution to primary production in the equatorial and subtropical North Pacific Ocean. *Aquatic Microbial Ecology*, **12**, 39–47.

335. Capone,D.G. (1997) Trichodesmium, a Globally Significant Marine Cyanobacterium. *Science*, **276**, 1221–1229.

336. Berman-Frank,I., Lundgren,P. and Falkowski,P. (2003) Nitrogen fixation and photosynthetic oxygen evolution in cyanobacteria. *Res. Microbiol.*, **154**, 157–164.

337. Irisarri,P., Gonnet,S. and Monza,J. (2001) Cyanobacteria in Uruguayan rice fields: diversity, nitrogen fixing ability and tolerance to herbicides and combined nitrogen. *Journal of Biotechnology*, **91**, 95–103.

338. Stanier,R.Y., Deruelles,J., Rippka,R., Herdman,M. and Waterbury,J.B. (1979) Generic Assignments, Strain Histories and Properties of Pure Cultures of Cyanobacteria. *Microbiology*, **111**, 1–61.

339. Bryant,D.A. (2006) The Molecular Biology of Cyanobacteria Bryant,D.A. (ed) Springer Science & Business Media, Dordrecht.

340. Peschek,G.A., Löffelhardt,W. and Schmetterer,G. (2012) The Phototrophic Prokaryotes Peschek,G.A., Löffelhardt,W., Schmetterer,G. (eds) Springer Science & Business Media, Boston, MA.

341. Tomitani,A., Knoll,A.H., Cavanaugh,C.M. and Ohno,T. (2006) The evolutionary diversification of cyanobacteria: molecular-phylogenetic and paleontological perspectives. *Proc. Natl. Acad. Sci. U.S.A.*, **103**, 5442–5447.

342. Sanchez-Baracaldo,P., Hayes,P.K. and BlanK,C.E. (2005) Morphological and habitat evolution in the Cyanobacteria using a compartmentalization approach. *Geobiology*, **3**, 145–165.

343. Renger,G. and Renger,T. (2008) Photosystem II: The machinery of photosynthetic water splitting. *Photosynthesis Research*, **98**, 53–80.

344. Nelson,N. (2011) Photosystems and global effects of oxygenic photosynthesis. *Biochim. Biophys. Acta*, **1807**, 856–863.

345. Rae,B.D., Long,B.M., Badger,M.R. and Price,G.D. (2013) Functions, Compositions, and Evolution of the Two Types of Carboxysomes: Polyhedral Microcompartments That Facilitate CO2 Fixation in Cyanobacteria and Some Proteobacteria. *Bacteriol. Rev.*, **77**, 357–379.

346. Kupriyanova,E.V., Sinetova,M.A., Cho,S.M., Park,Y.-I., Los,D.A. and Pronina,N.A. (2013) CO2-concentrating mechanism in cyanobacterial photosynthesis: organization, physiological role, and evolutionary origin. *Photosynthesis Research*, **117**, 133–146.

347. Husic,D.W., Husic,H.D., Tolbert,N.E. and Black,C.C.,Jr. (1987) The oxidative photosynthetic carbon cycle or C 2cycle. *Critical Reviews in Plant Sciences*, **5**, 45–100.

348. Eisenhut,M., Kahlon,S., Hasse,D., Ewald,R., Lieman-Hurwitz,J., Ogawa,T., Ruth,W., Bauwe,H., Kaplan,A. and Hagemann,M. (2006) The plant-like C2 glycolate cycle and the bacterial-like glycerate pathway cooperate in phosphoglycolate metabolism in cyanobacteria. *Plant Physiol.*, **142**, 333–342.

349. Knoop,H., Gründel,M., Zilliges,Y., Lehmann,R., Hoffmann,S., Lockau,W. and Steuer,R. (2013) Flux balance analysis of cyanobacterial metabolism: the metabolic network of Synechocystis sp. PCC 6803. *PLoS Comput Biol*, **9**, e1003081.

350. Dempo,Y., Ohta,E., Nakayama,Y., Bamba,T. and Fukusaki,E. (2014) Molar-based targeted metabolic profiling of cyanobacterial strains with potential for biological production. *Metabolites*, **4**, 499–516.

351. Berla,B.M., Saha,R., Immethun,C.M., Maranas,C.D., Moon,T.S. and Pakrasi,H.B. (2013) Synthetic biology of cyanobacteria: unique challenges and opportunities. *Front Microbiol*, **4**, 246.

352. Markley,A.L., Begemann,M.B., Clarke,R.E., Gordon,G.C. and Pfleger,B.F. (2015) Synthetic biology toolbox for controlling gene expression in the

cyanobacterium Synechococcus sp. strain PCC 7002. *ACS Synth. Biol.*, **4**, 595–603.

353. Yu,J., Liberton,M., Cliften,P.F., Head,R.D., Jacobs,J.M., Smith,R.D., Koppenaal,D.W., Brand,J.J. and Pakrasi,H.B. (2015) Synechococcus elongatus UTEX 2973, a fast growing cyanobacterial chassis for biosynthesis using light and CO₂. *Scientific Reports*, **5**, 8132.

354. Redinbo,M.R., Yeates,T.O. and Merchant,S. (1994) Plastocyanin: structural and functional analysis. *J. Bioenerg. Biomembr.*, **26**, 49–66.

355. Grotjohann,I. and Fromme,P. (2005) Structure of cyanobacterial Photosystem I. *Photosynthesis Research*, **85**, 51–72.

356. Nickelsen,J. and Rengstl,B. (2013) Photosystem II assembly: from cyanobacteria to plants. *Annu Rev Plant Biol*, **64**, 609–635.

357. Mulkidjanian,A.Y., Koonin,E.V., Makarova,K.S., Mekhedov,S.L., Sorokin,A., Wolf,Y.I., Dufresne,A., Partensky,F., Burd,H., Kaznadzey,D., *et al.* (2006) The cyanobacterial genome core and the origin of photosynthesis. *Proc. Natl. Acad. Sci. U.S.A.*, **103**, 13126–13131.

358. Golbeck,J.H. (1994) Photosystem I in Cyanobacteria. In *The Molecular Biology of Cyanobacteria*. Springer Netherlands, Dordrecht, pp. 319–360.

359. Heinz,S., Liauw,P., Nickelsen,J. and Nowaczyk,M. (2016) Analysis of photosystem II biogenesis in cyanobacteria. *Biochim. Biophys. Acta*, **1857**, 274–287.

360. Golden,S.S., Ishiura,M., Johnson,C.H. and Kondo,T. (1997) CYANOBACTERIAL CIRCADIAN RHYTHMS. *Annu. Rev. Plant Physiol. Plant Mol. Biol.*, **48**, 327–354.

361. Yeates,T.O., Kerfeld,C.A., Heinhorst,S., Cannon,G.C. and Shively,J.M. (2008) Protein-based organelles in bacteria: carboxysomes and related microcompartments. *Nat. Rev. Microbiol.*, **6**, 681–691.

362. Shively,J.M. and English,R.S. (1991) The carboxysome, a prokaryotic organelle: a mini-review. *Canadian Journal of Botany*, **69**, 957–962.

363. Tanaka,S., Kerfeld,C.A., Sawaya,M.R., Cai,F., Heinhorst,S., Cannon,G.C. and Yeates,T.O. (2008) Atomic-level models of the bacterial carboxysome shell. *Science*, **319**, 1083–1086.

364. Williams,J.G.K. (1988) [85] Construction of specific mutations in photosystem II photosynthetic reaction center by genetic engineering methods in Synechocystis 6803. In *Cyanobacteria*, Methods in Enzymology. Elsevier, Vol. 167, pp. 766–778.

365. Anderson,S.L. and McIntosh,L. (1991) Light-activated heterotrophic growth of the cyanobacterium Synechocystis sp. strain PCC 6803: a blue-light-requiring process. *Journal of Bacteriology*, **173**, 2761–2767.

366. Kaneko,T., Sato,S., Kotani,H., Tanaka,A., Asamizu,E., Nakamura,Y., Miyajima,N., Hirosawa,M., Sugiura,M., Sasamoto,S., *et al.* (1996) Sequence

Analysis of the Genome of the Unicellular Cyanobacterium Synechocystis sp. Strain PCC6803. II. Sequence Determination of the Entire Genome and Assignment of Potential Protein-coding Regions. *DNA Research*, **3**, 109–136.

367. Barten,R. and Lill,H. (1995) DNA-uptake in the naturally competent cyanobacterium, Synechocystis sp.PCC 6803. *FEMS Microbiology Letters*, **129**, 83–87.

368. Cohen,S.E. and Golden,S.S. (2015) Circadian Rhythms in Cyanobacteria. *Microbiol. Mol. Biol. Rev.*, **79**, 373–385.

369. Kondo,T. and Ishiura,M. (2000) The circadian clock of cyanobacteria. *Bioessays*, **22**, 10–15.

370. Golden,S.S., Brusslan,J. and Haselkorn,R. (1987) [12] Genetic engineering of the cyanobacterial chromosome. *Meth. Enzymol.*, **153**, 215–231.

371. Batterton,J.C. and Van Baalen,C. (1971) Growth responses of blue-green algae to sodium chloride concentration. *Archiv. Mikrobiol.*, **76**, 151–165.

372. Xu,Y., Alvey,R.M., Byrne,P.O., Graham,J.E., Shen,G. and Bryant,D.A. (2011) Expression of Genes in Cyanobacteria: Adaptation of Endogenous Plasmids as Platforms for High-Level Gene Expression in <Emphasis Type='Italic'>Synechococcus</Emphasis> sp. PCC 7002. In *Photosynthesis Research Protocols*, Methods in Molecular Biology. Humana Press, Totowa, NJ, Totowa, NJ, Vol. 684, pp. 273–293.

373. Frigaard,N.-U., Sakuragi,Y. and Bryant,D.A. (2004) Gene inactivation in the cyanobacterium Synechococcus sp. PCC 7002 and the green sulfur bacterium Chlorobium tepidum using in vitro-made DNA constructs and natural transformation. *Methods Mol. Biol.*, **274**, 325–340.

374. Mueller,T.J., Ungerer,J.L., Pakrasi,H.B. and Maranas,C.D. (2017) Identifying the Metabolic Differences of a Fast-Growth Phenotype in Synechococcus UTEX 2973. *Scientific Reports*, **7**, 41569.

375. Carroll,A.L., Case,A.E., Zhang,A. and Atsumi,S. (2018) Metabolic engineering tools in model cyanobacteria. *Metab. Eng.*, 10.1016/j.ymben.2018.03.014.

376. Griese,M., Lange,C. and Soppa,J. (2011) Ploidy in cyanobacteria. *FEMS Microbiology Letters*, **323**, 124–131.

377. Zerulla,K., Ludt,K. and Soppa,J. (2016) The ploidy level of Synechocystis sp. PCC 6803 is highly variable and is influenced by growth phase and by chemical and physical external parameters. *Microbiology (Reading, Engl.)*, **162**, 730–739.

378. Guerrero,F., Carbonell,V., Cossu,M., Correddu,D. and Jones,P.R. (2012) Ethylene synthesis and regulated expression of recombinant protein in Synechocystis sp. PCC 6803. *PLoS ONE*, **7**, e50470.

379. Du,W., Angermayr,S.A., Jongbloets,J.A., Molenaar,D., Bachmann,H., Hellingwerf,K.J. and Branco dos Santos,F. (2017) Nonhierarchical Flux Regulation Exposes the Fitness Burden Associated with Lactate Production in Synechocystis sp. PCC6803. *ACS Synth. Biol.*, **6**, 395–401.

380. Kusakabe,T., Tatsuke,T., Tsuruno,K., Hirokawa,Y., Atsumi,S., Liao,J.C. and Hanai,T. (2013) Engineering a synthetic pathway in cyanobacteria for isopropanol production directly from carbon dioxide and light. *Metab. Eng.*, **20**, 101–108.

381. Tyo,K.E.J., Ajikumar,P.K. and Stephanopoulos,G. (2009) Stabilized gene duplication enables long-term selection-free heterologous pathway expression. *Nat. Biotechnol.*, **27**, 760–765.

382. Minda,R., Ramchandani,J., Joshi,V.P. and Bhattacharjee,S.K. (2005) A homozygous recA mutant of Synechocystis PCC6803: construction strategy and characteristics eliciting a novel RecA independent UVC resistance in dark. *Mol. Genet. Genomics*, **274**, 616–624.

383. Wang,B., Eckert,C., Maness,P.-C. and Yu,J. (2018) A Genetic Toolbox for Modulating the Expression of Heterologous Genes in the Cyanobacterium Synechocystis sp. PCC 6803. *ACS Synth. Biol.*, **7**, 276–286.

384. Ruffing,A.M., Jensen,T.J. and Strickland,L.M. (2016) Genetic tools for advancement of Synechococcus sp. PCC 7002 as a cyanobacterial chassis. *Microb. Cell Fact.*, **15**, 190.

385. Liu,D. and Pakrasi,H.B. (2018) Exploring native genetic elements as plug-in tools for synthetic biology in the cyanobacterium Synechocystis sp. PCC 6803. *Microb. Cell Fact.*, **17**, 48.

386. Sun,T., Li,S., Song,X., Diao,J., Chen,L. and Zhang,W. (2018) Toolboxes for cyanobacteria: Recent advances and future direction. *Biotechnol. Adv.*, **36**, 1293–1307.

387. Yan,X., Chu,F., Puri,A.W., Fu,Y. and Lidstrom,M.E. (2016) Electroporation-Based Genetic Manipulation in Type I Methanotrophs. *Appl. Environ. Microbiol.*, **82**, 2062–2069.

388. Herman,N.A., Li,J., Bedi,R., Turchi,B., Liu,X., Miller,M.J. and Zhang,W. (2016) Development of a High-Efficiency Transformation Method and Implementation of Rational Metabolic Engineering for the Industrial Butanol Hyperproducer Clostridium saccharoperbutylacetonicum Strain N1-4. *Appl. Environ. Microbiol.*, **83**, e02942–16.

389. Chua,C.K., Leong,K.F. and Lim,C.S. (2010) Rapid Prototyping World Scientific.

390. Karim,A.S. and Jewett,M.C. (2016) A cell-free framework for rapid biosynthetic pathway prototyping and enzyme discovery. *Metab. Eng.*, **36**, 116–126.

391. Kelwick,R., Ricci,L., Chee,S.M., Bell,D., Webb,A.J. and Freemont,P.S. (2017) Cell-free prototyping strategies for enhancing the sustainable production of polyhydroxyalkanoates bioplastics. *bioRxiv*, 10.1101/225144.

392. Sun,Z.Z., Yeung,E., Hayes,C.A., Noireaux,V. and Murray,R.M. (2014) Linear DNA for rapid prototyping of synthetic biological circuits in an Escherichia coli based TX-TL cell-free system. *ACS Synth. Biol.*, **3**, 387–397.

393. Chappell,J., Jensen,K. and Freemont,P.S. (2013) Validation of an entirely in vitro approach for rapid prototyping of DNA regulatory elements for synthetic biology. *Nucleic Acids Res.*, **41**, 3471–3481.

394. Cardinale,S. and Arkin,A.P. (2012) Contextualizing context for synthetic biology--identifying causes of failure of synthetic biological systems. *Biotechnol J*, **7**, 856–866.

395. Kittleson,J.T., Wu,G.C. and Anderson,J.C. (2012) Successes and failures in modular genetic engineering. *Curr Opin Chem Biol*, **16**, 329–336.

396. Nevozhay,D., Zal,T. and Balázsi,G. (2013) Transferring a synthetic gene circuit from yeast to mammalian cells. *Nat Commun*, **4**, 1451.

397. Curran,K.A., Crook,N.C., Karim,A.S., Gupta,A., Wagman,A.M. and Alper,H.S. (2014) Design of synthetic yeast promoters via tuning of nucleosome architecture. *Nat Commun*, **5**, 2.

398. Imamura,S. and Asayama,M. (2009) Sigma factors for cyanobacterial transcription. *Gene Regul Syst Bio*, **3**, 65–87.

399. Artsimovitch,I., Svetlov,V., Anthony,L., Burgess,R.R. and Landick,R. (2000) RNA polymerases from Bacillus subtilis and Escherichia coli differ in recognition of regulatory signals in vitro. *Journal of Bacteriology*, **182**, 6027–6035.

400. Meysman,P., Collado-Vides,J., Morett,E., Viola,R., Engelen,K. and Laukens,K. (2014) Structural properties of prokaryotic promoter regions correlate with functional features. *PLoS ONE*, **9**, e88717.

401. Greener,A., Lehman,S.M. and Helinski,D.R. (1992) Promoters of the broad host range plasmid RK2: analysis of transcription (initiation) in five species of gram-negative bacteria. *Genetics*, **130**, 27–36.

402. Leavitt,J.M. and Alper,H.S. (2015) Advances and current limitations in transcript-level control of gene expression. *Current Opinion in Biotechnology*, **34**, 98–104.

403. Radeck,J., Kraft,K., Bartels,J., Cikovic,T., Dürr,F., Emenegger,J., Kelterborn,S., Sauer,C., Fritz,G., Gebhard,S., *et al.* (2013) The Bacillus BioBrick Box: generation and evaluation of essential genetic building blocks for standardized work with Bacillus subtilis. *J Biol Eng*, **7**, 29.

404. Blazeck,J., Liu,L., Redden,H. and Alper,H. (2011) Tuning gene expression in Yarrowia lipolytica by a hybrid promoter approach. *Appl. Environ. Microbiol.*, **77**, 7905–7914.

405. McBride,K.E., Schaaf,D.J., Daley,M. and Stalker,D.M. (1994) Controlled expression of plastid transgenes in plants based on a nuclear DNA-encoded and plastid-targeted T7 RNA polymerase. *Proceedings of the National Academy of Sciences*, **91**, 7301–7305.

406. Studier,F.W. and Moffatt,B.A. (1986) Use of bacteriophage T7 RNA polymerase to direct selective high-level expression of cloned genes. *J. Mol. Biol.*, **189**, 113–130.

407. Wyatt,L.S., Moss,B. and Rozenblatt,S. (1995) Replication-Deficient Vaccinia Virus Encoding Bacteriophage T7 RNA Polymerase for Transient Gene Expression in Mammalian Cells. *Virology*, **210**, 202–205.

408. Mordaka,P.M. and Heap,J.T. (2018) Stringency of Synthetic Promoter Sequences in Clostridium Revealed and Circumvented by Tuning Promoter Library Mutation Rates. *ACS Synth. Biol.*, **7**, 672–681.

409. Kushwaha,M. and Salis,H.M. (2015) A portable expression resource for engineering cross-species genetic circuits and pathways. *Nat Commun*, **6**, 7832.

410. Rippka,R. (1988) [1] Isolation and purification of cyanobacteria. In *Cyanobacteria*, Methods in Enzymology. Elsevier, Vol. 167, pp. 3–27.

411. Geissmann,Q. (2013) OpenCFU, a new free and open-source software to count cell colonies and other circular objects. *PLoS ONE*, **8**, e54072.

412. Huang,H.-H., Camsund,D., Lindblad,P. and Heidorn,T. (2010) Design and characterization of molecular tools for a Synthetic Biology approach towards developing cyanobacterial biotechnology. *Nucleic Acids Res.*, **38**, 2577–2593.

413. Chaves,J.E., Rueda-Romero,P., Kirst,H. and Melis,A. (2017) Engineering Isoprene Synthase Expression and Activity in Cyanobacteria. *ACS Synth. Biol.*, **6**, 2281–2292.

414. Boyanapalli,R., Bullerjahn,G.S., Pohl,C., Croot,P.L., Boyd,P.W. and McKay,R.M.L. (2007) Luminescent whole-cell cyanobacterial bioreporter for measuring Fe availability in diverse marine environments. *Appl. Environ. Microbiol.*, **73**, 1019–1024.

415. Zhou,J., Zhang,H., Meng,H., Zhu,Y., Bao,G., Zhang,Y., Li,Y. and Ma,Y. (2014) Discovery of a super-strong promoter enables efficient production of heterologous proteins in cyanobacteria. *Scientific Reports*, **4**, 235.

416. Xu,Y., Alvey,R.M., Byrne,P.O., Graham,J.E., Shen,G. and Bryant,D.A. (2010) Expression of Genes in Cyanobacteria: Adaptation of Endogenous Plasmids as Platforms for High-Level Gene Expression in Synechococcus sp. PCC 7002. In Carpentier,R. (ed), *Photosynthesis Research Protocols*, Methods in Molecular Biology. Humana Press, Totowa, NJ, Vol. 684, pp. 273–293.

417. Englund,E., Liang,F. and Lindberg,P. (2016) Evaluation of promoters and ribosome binding sites for biotechnological applications in the unicellular cyanobacterium Synechocystis sp. PCC 6803. *Scientific Reports*, **6**, 36640.

418. Blasi,B. (2012) Characterization of Stress Responses of Heavy Metal and Metalloid Inducible Promoters in Synechocystis PCC6803. *Journal of Microbiology and Biotechnology*, **22**, 166–169.

419. Čelešnik,H., Tanšek,A., Tahirović,A., Vižintin,A., Mustar,J., Vidmar,V. and Dolinar,M. (2016) Biosafety of biotechnologically important microalgae: intrinsic suicide switch implementation in cyanobacterium Synechocystissp .PCC 6803. *Biology Open*, **5**, 519–528.

420. Peca,L., KÃ³s,P.T.B., MÃ tÃ,Z., Farsang,A. and Vass,I. (2008) Construction of bioluminescent cyanobacterial reporter strains for detection of nickel, cobalt and zinc. *FEMS Microbiology Letters*, **289**, 258–264.

421. Pérez,A.A., Gajewski,J.P., Ferlez,B.H., Ludwig,M., Baker,C.S., Golbeck,J.H. and Bryant,D.A. (2017) Zn2+-Inducible Expression Platform for Synechococcus sp. Strain PCC 7002 Based on the smtA Promoter/Operator and smtB Repressor. *Appl. Environ. Microbiol.*, **83**, e02491–16.

422. Huang,H.-H. and Lindblad,P. (2013) Wide-dynamic-range promoters engineered for cyanobacteria. *J Biol Eng*, **7**, 10.

423. Zess,E.K., Begemann,M.B. and Pfleger,B.F. (2015) Construction of new synthetic biology tools for the control of gene expression in the cyanobacterium Synechococcussp. strain PCC 7002. *Biotechnol. Bioeng.*, **113**, 424–432.

424. Camsund,D., Heidorn,T. and Lindblad,P. (2014) Design and analysis of LacI-repressed promoters and DNA-looping in a cyanobacterium. *J Biol Eng*, **8**, 4.

425. Albers,S.C., Gallegos,V.A. and Peebles,C.A.M. (2015) Engineering of genetic control tools in Synechocystis sp. PCC 6803 using rational design techniques. *Journal of Biotechnology*, **216**, 36–46.

426. Isaacs,F.J., Dwyer,D.J., Ding,C., Pervouchine,D.D., Cantor,C.R. and Collins,J.J. (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.*, **22**, 841–847.

427. Carrier,T.A. and Keasling,J.D. (1999) Library of Synthetic 5' Secondary Structures To Manipulate mRNA Stability in Escherichia coli. *Biotechnology Progress*, **15**, 58–64.

428. Pfleger,B.F., Pitera,D.J., Smolke,C.D. and Keasling,J.D. (2006) Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes. *Nat. Biotechnol.*, **24**, 1027–1032.

429. Chubiz,L.M. and Rao,C.V. (2008) Computational design of orthogonal ribosomes. *Nucleic Acids Res.*, **36**, 4038–4046.

430. Takeshima,Y. (1994) A Novel Expression Vector for the Cyanobacterium, Synechococcus PCC 6301. *DNA Research*, **1**, 181–189.

431. Geerts,D., Bovy,A., de Vrieze,G., Borrias,M. and Weisbeek,P. (1995) Inducible expression of heterologous genes targeted to a chromosomal platform in the cyanobacterium Synechococcus sp. PCC 7942. *Microbiology*, **141**, 831–841.

432. Brahamsha,B. (1996) A genetic manipulation system for oceanic cyanobacteria of the genus Synechococcus. *Appl. Environ. Microbiol.*, **62**, 1747–1751.

433. Marraccini,P., Bulteau,S., Cassier-Chauvat,C., Mermet-Bouvier,P. and Chauvat,F. (1993) A conjugative plasmid vector for promoter analysis in several cyanobacteria of the genera Synechococcus and Synechocystis. *Plant Mol. Biol.*, **23**, 905–909.

434. Iwaki,T., Haranoh,K., Inoue,N., Kojima,K., Satoh,R., Nishino,T., Wada,S., Ihara,H., Tsuyama,S., Kobayashi,H., *et al.* (2006) Expression of foreign type I

ribulose-1,5-bisphosphate carboxylase/ oxygenase (EC 4.1.1.39) stimulates photosynthesis in cyanobacterium Synechococcus PCC7942 cells. *Photosynthesis Research*, **88**, 287–297.

435. De Gelder,L., Ponciano,J.M., Joyce,P. and Top,E.M. (2007) Stability of a promiscuous plasmid in different hosts: no guarantee for a long-term relationship. *Microbiology*, **153**, 452–463.

436. Meyer,R. (2009) Replication and conjugative mobilization of broad host-range IncQ plasmids. *Plasmid*, **62**, 57–70.

437. Burnap,R.L., Qian,M., Shen,J.R., Inoue,Y. and Sherman,L.A. (2002) Role of Disulfide Linkage and Putative Intermolecular Binding Residues in the Stability and Binding of the Extrinsic Manganese-Stabilizing Protein to the Photosystem II Reaction Center. *Biochemistry*, **33**, 13712–13718.

438. Aoki,R., Goto,T. and Fujita,Y. (2011) A Heme Oxygenase Isoform is Essential for Aerobic Growth in the Cyanobacterium Synechocystis sp. PCC 6803: Modes of Differential Operation of Two Isoforms/Enzymes to Adapt to Low Oxygen Environments in Cyanobacteria. *Plant and Cell Physiology*, **52**, 1744–1756.

439. Aoki,S., Kondo,T. and Ishiura,M. (1995) Circadian expression of the dnaK gene in the cyanobacterium Synechocystis sp. strain PCC 6803. *Journal of Bacteriology*, **177**, 5606–5611.

440. Pinto,F., Pacheco,C.C., Oliveira,P., Montagud,A., Landels,A., Couto,N., Wright,P.C., Urchueguía,J.F. and Tamagnini,P. (2015) Improving a Synechocystis-based photoautotrophic chassis through systematic genome mapping and validation of neutral sites. *DNA Research*, **22**, 425–437.

441. Ng,A.H., Berla,B.M. and Pakrasi,H.B. (2015) Fine-Tuning of Photoautotrophic Protein Production by Combining Promoters and Neutral Sites in the Cyanobacterium Synechocystis sp. Strain PCC 6803. *Appl. Environ. Microbiol.*, **81**, 6857–6863.

442. Vogel,A.I.M., Lale,R. and Hohmann-Marriott,M.F. (2017) Streamlining recombination-mediated genetic engineering by validating three neutral integration sites in Synechococcus sp. PCC 7002. *J Biol Eng*, **11**, 425.

443. Landry,B.P., Stöckel,J. and Pakrasi,H.B. (2013) Use of Degradation Tags To Control Protein Levels in the Cyanobacterium Synechocystis sp. Strain PCC 6803. *Appl. Environ. Microbiol.*, **79**, 2833–2835.

444. De Mey,M., Maertens,J., Lequeux,G.J., Soetaert,W.K. and Vandamme,E.J. (2007) Construction and model-based analysis of a promoter library for E. coli: an indispensable tool for metabolic engineering. *BMC Biotechnol.*, **7**, 34.

445. Rud,I., Jensen,P.R., Naterstad,K. and Axelsson,L. (2006) A synthetic promoter library for constitutive gene expression in Lactobacillus plantarum. *Microbiology (Reading, Engl.)*, **152**, 1011–1019.

446. Solem,C. and Jensen,P.R. (2002) Modulation of gene expression made easy. *Appl. Environ. Microbiol.*, **68**, 2397–2403.

447. Davis,J.H., Rubin,A.J. and Sauer,R.T. (2011) Design, construction and characterization of a set of insulated bacterial promoters. *Nucleic Acids Res.*, **39**, 1131–1141.

448. Camsund,D. and Lindblad,P. (2014) Engineered transcriptional systems for cyanobacterial biotechnology. *Front Bioeng Biotechnol*, **2**, 40.

449. Harley,C.B. and Reynolds,R.P. (1987) Analysis of E. coli promoter sequences. *Nucleic Acids Res.*, **15**, 2343–2361.

450. Lisser,S. and Margalit,H. (1993) Compilation of E. coli mRNA promoter sequences. *Nucleic Acids Res.*, **21**, 1507–1516.

451. Stothard,P. (2000) The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences. *BioTechniques*, **28**, 1102–1104.

452. Mohamed,A., Eriksson,J., Osiewacz,H.D. and Jansson,C. (1993) Differential expression of the psbA genes in the cyanobacterium Synechocystis 6803. *Mol. Gen. Genet.*, **238**, 161–168.

453. Pribnow,D. (1975) Bacteriophage T7 early promoters: nucleotide sequences of two RNA polymerase binding sites. *J. Mol. Biol.*, **99**, 419–443.

454. Rothmel,R.K. and LeClerc,J.E. (1989) Mutational analysis of the lac regulatory region: second-site changes that activate mutant promoters. *Nucleic Acids Res.*, **17**, 3909–3925.

455. Bakke,I., Berg,L., Aune,T.E.V., Brautaset,T., Sletta,H., Tøndervik,A. and Valla,S. (2009) Random mutagenesis of the PM promoter as a powerful strategy for improvement of recombinant-gene expression. *Appl. Environ. Microbiol.*, **75**, 2002–2011.

456. Wang,H.H., Isaacs,F.J., Carr,P.A., Sun,Z.Z., Xu,G., Forest,C.R. and Church,G.M. (2009) Programming cells by multiplex genome engineering and accelerated evolution. *Nature*, **460**, 894–898.

457. Oesterle,S., Gerngross,D., Schmitt,S., Roberts,T.M. and Panke,S. (2017) Efficient engineering of chromosomal ribosome binding site libraries in mismatch repair proficient Escherichia coli. *Scientific Reports*, **7**, 1027.

458. Anderson,J.C., Voigt,C.A. and Arkin,A.P. (2007) Environmental signal integration by a modular AND gate. *Mol. Syst. Biol.*, **3**, 133.

459. Guiziou,S., Sauveplane,V., Chang,H.-J., Clerté,C., Declerck,N., Jules,M. and Bonnet,J. (2016) A part toolbox to tune genetic expression in Bacillus subtilis. *Nucleic Acids Res.*, **44**, 7495–7508.

460. Ma,J., Campbell,A. and Karlin,S. (2002) Correlations between Shine-Dalgarno sequences and gene features such as predicted expression levels and operon structures. *Journal of Bacteriology*, **184**, 5733–5745.

461. Estrem,S.T., Gaal,T., Ross,W. and Gourse,R.L. (1998) Identification of an UP element consensus sequence for bacterial promoters. *Proceedings of the National Academy of Sciences*, **95**, 9761–9766.

462. de Smit,M.H. and van Duin,J. (2003) Translational standby sites: how ribosomes may deal with the rapid folding kinetics of mRNA. *J. Mol. Biol.*, **331**, 737–743.

463. Studer,S.M. and Joseph,S. (2006) Unfolding of mRNA secondary structure by the bacterial translation initiation complex. *Mol. Cell*, **22**, 105–115.

464. Levy,L., Anavy,L., Solomon,O., Cohen,R., Brunwasser-Meirom,M., Ohayon,S., Atar,O., Goldberg,S., Yakhini,Z. and Amit,R. (2017) A Synthetic Oligo Library and Sequencing Approach Reveals an Insulation Mechanism Encoded within Bacterial σ54 Promoters. *Cell Reports*, **21**, 845–858.

465. Schilling,D.L. and Belove,C. Electronic Circuits: Discrete and Integrated McGraw-Hill Inc. (ed).

466. Yarchuk,O., Jacques,N., Guillerez,J. and Dreyfus,M. (1992) Interdependence of translation, transcription and mRNA degradation in the lacZ gene. *J. Mol. Biol.*, **226**, 581–596.

467. Cho,K.-O. and Yanofsky,C. (1988) Sequence changes preceding a Shine-Dalgarno region influence trpE mRNA translation and decay. *J. Mol. Biol.*, **204**, 51–60.

468. Telesnitsky,A.P. and Chamberlin,M.J. (1989) Sequences linked to prokaryotic promoters can affect the efficiency of downstream termination sites. *J. Mol. Biol.*, **205**, 315–330.

469. Ellinger,T., Behnke,D., Knaus,R., Bujard,H. and Gralla,J.D. (1994) Context-dependent effects of upstream A-tracts. Stimulation or inhibition of Escherichia coli promoter function. *J. Mol. Biol.*, **239**, 466–475.

470. Stueber,D. and Bujard,H. (1982) Transcription from efficient promoters can interfere with plasmid replication and diminish expression of plasmid specified genes. *EMBO J.*, **1**, 1399–1404.

471. Mutalik,V.K., Guimaraes,J.C., Cambray,G., Mai,Q.-A., Christoffersen,M.J., Martin,L., Yu,A., Lam,C., Rodriguez,C., Bennett,G., *et al.* (2013) Quantitative estimation of activity and quality for collections of functional genetic elements. *Nature Methods*, **10**, 347–353.

472. Heyduk,E. and Heyduk,T. (2018) DNA template sequence control of bacterial RNA polymerase escape from the promoter. *Nucleic Acids Res.*, **46**, 4469–4486.

473. Hsu,L.M., Cobb,I.M., Ozmore,J.R., Khoo,M., Nahm,G., Xia,L., Bao,Y. and Ahn,C. (2006) Initial transcribed sequence mutations specifically affect promoter escape properties. *Biochemistry*, **45**, 8841–8854.

474. Hsu,L.M. (2002) Promoter clearance and escape in prokaryotes. *Biochim. Biophys. Acta*, **1577**, 191–207.

475. Vo,N.V., Hsu,L.M., Kane,C.M. and Chamberlin,M.J. (2003) In vitro studies of transcript initiation by Escherichia coli RNA polymerase. 3. Influences of individual DNA elements within the promoter recognition region on abortive initiation and promoter escape. *Biochemistry*, **42**, 3798–3811.

476. Mutalik,V.K., Guimaraes,J.C., Cambray,G., Lam,C., Christoffersen,M.J., Mai,Q.-A., Tran,A.B., Paull,M., Keasling,J.D., Arkin,A.P., *et al.* (2013) Precise and reliable gene expression via standard transcription and translation initiation elements. *Nature Methods*, **10**, 354–360.

477. Qi,L., Haurwitz,R.E., Shao,W., Doudna,J.A. and Arkin,A.P. (2012) RNA processing enables predictable programming of gene expression. *Nat. Biotechnol.*, **30**, 1002–1006.

478. Lou,C., Stanton,B., Chen,Y.-J., Munsky,B. and Voigt,C.A. (2012) Ribozyme-based insulator parts buffer synthetic circuits from genetic context. *Nat. Biotechnol.*, **30**, 1137–1142.

479. Henry,I. and Sharp,P.M. (2007) Predicting gene expression level from codon usage bias. *Mol. Biol. Evol.*, **24**, 10–12.

480. Roymondal,U., Das,S. and Sahoo,S. (2009) Predicting gene expression level from relative codon usage bias: an application to Escherichia coli genome. *DNA Res.*, **16**, 13–30.

481. Rhodius,V.A., Mutalik,V.K. and Gross,C.A. (2012) Predicting the strength of UP-elements and full-length E. coli σE promoters. *Nucleic Acids Res.*, **40**, 2907–2924.

482. Andreou,A.I. and Nakayama,N. (2018) Mobius Assembly: A versatile Golden-Gate framework towards universal DNA assembly. *PLoS ONE*, **13**, e0189892.

483. Denby,C.M., Li,R.A., Vu,V.T., Costello,Z., Lin,W., Chan,L.J.G., Williams,J., Donaldson,B., Bamforth,C.W., Petzold,C.J., *et al.* (2018) Industrial brewing yeast engineered for the production of primary flavor determinants in hopped beer. *Nat Commun*, **9**, 965.

484. Mandal,M. and Breaker,R.R. (2004) Gene regulation by riboswitches. *Nature Reviews Molecular Cell Biology 2015 16:9*, **5**, 451–463.

485. Chappell,J., Takahashi,M.K. and Lucks,J.B. (2015) Creating small transcription activating RNAs. *Nat. Chem. Biol.*, **11**, 214–220.

486. Blattner,F.R., Plunkett,G., Bloch,C.A., Perna,N.T., Burland,V., Riley,M., Collado-Vides,J., Glasner,J.D., Rode,C.K., Mayhew,G.F., *et al.* (1997) The complete genome sequence of Escherichia coli K-12. *Science*, **277**, 1453–1462.

487. Villegas,A. and Kropinski,A.M. (2008) An analysis of initiation codon utilization in the Domain Bacteria - concerns about the quality of bacterial genome annotation. *Microbiology (Reading, Engl.)*, **154**, 2559–2661.

488. Sussman,J.K., Simons,E.L. and Simons,R.W. (1996) Escherichia coli translation initiation factor 3 discriminates the initiation codon in vivo. *Mol. Microbiol.*, **21**, 347–360.

489. Cridge,A.G., Major,L.L., Mahagaonkar,A.A., Poole,E.S., Isaksson,L.A. and Tate,W.P. (2006) Comparison of characteristics and function of translation termination signals between and within prokaryotic and eukaryotic organisms. *Nucleic Acids Res.*, **34**, 1959–1973.

490. Kim,D., Hong,J.S.-J., Qiu,Y., Nagarajan,H., Seo,J.-H., Cho,B.-K., Tsai,S.-F. and Palsson,B.Ø. (2012) Comparative analysis of regulatory elements between Escherichia coli and Klebsiella pneumoniae by genome-wide transcription start site profiling. *PLoS Genet.*, **8**, e1002867.

491. Crooks,G.E., Hon,G., Chandonia,J.-M. and Brenner,S.E. (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–1190.

492. Emami,S., Yee,M.-C. and Dinneny,J.R. (2013) A robust family of Golden Gate Agrobacterium vectors for plant synthetic biology. *Front Plant Sci*, **4**, 339.

493. Werner,S., Engler,C., Weber,E., Gruetzner,R. and Marillonnet,S. (2012) Fast track assembly of multigene constructs using Golden Gate cloning and the MoClo system. *Bioeng Bugs*, **3**, 38–43.

494. Lutz,R. and Bujard,H. (1997) Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. *Nucleic Acids Res.*, **25**, 1203–1210.

495. Jacob,F. and Monod,J. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, **3**, 318–356.

496. Osbourn,A.E. and Field,B. (2009) Operons. *Cellular and Molecular Life Sciences*, **66**, 3755–3775.

497. Casini,A., Christodoulou,G., Freemont,P.S., Baldwin,G.S., Ellis,T. and MacDonald,J.T. (2014) R2oDNA designer: computational design of biologically neutral synthetic DNA sequences. *ACS Synth. Biol.*, **3**, 525–528.

498. Niu,F.-X., Lu,Q., Bu,Y.-F. and Liu,J.-Z. (2017) Metabolic engineering for the microbial production of isoprenoids: Carotenoids and isoprenoid-based biofuels. *Synth Syst Biotechnol*, **2**, 167–175.

499. Kim,S.W. and Keasling,J.D. (2001) Metabolic engineering of the nonmevalonate isopentenyl diphosphate synthesis pathway in Escherichia coli enhances lycopene production. *Biotechnol. Bioeng.*, **72**, 408–415.

500. Matthews,P.D. and Wurtzel,E.T. (2000) Metabolic engineering of carotenoid accumulation in Escherichia coli by modulation of the isoprenoid precursor pool with expression of deoxyxylulose phosphate synthase. *Appl. Microbiol. Biotechnol.*, **53**, 396–400.

501. Bongers,M., Chrysanthopoulos,P.K., Behrendorff,J.B.Y.H., Hodson,M.P., Vickers,C.E. and Nielsen,L.K. (2015) Systems analysis of methylerythritol-phosphate pathway flux in E. coli: insights into the role of oxidative stress and the validity of lycopene as an isoprenoid reporter metabolite. *Microb. Cell Fact.*, **14**, 193.

502. Weiße,A.Y., Oyarzún,D.A., Danos,V. and Swain,P.S. (2015) Mechanistic links between cellular trade-offs, gene expression, and growth. *Proc. Natl. Acad. Sci. U.S.A.*, **112**, E1038–47.

503. Lindberg,P., Park,S. and Melis,A. (2010) Engineering a platform for photosynthetic isoprene production in cyanobacteria, using Synechocystis as the model organism. *Metab. Eng.*, **12**, 70–79.

504. Bentley,F.K., Zurbriggen,A. and Melis,A. (2014) Heterologous expression of the mevalonic acid pathway in cyanobacteria enhances endogenous carbon partitioning to isoprene. *Mol Plant*, **7**, 71–86.

505. Takahashi,H., Uchimiya,H. and Hihara,Y. (2008) Difference in metabolite levels between photoautotrophic and photomixotrophic cultures of Synechocystis sp. PCC 6803 examined by capillary electrophoresis electrospray ionization mass spectrometry. *Journal of Experimental Botany*, **59**, 3009–3018.

506. Kudoh,K., Kawano,Y., Hotta,S., Sekine,M., Watanabe,T. and Ihara,M. (2014) Prerequisite for highly efficient isoprenoid production by cyanobacteria discovered through the over-expression of 1-deoxy-d-xylulose 5-phosphate synthase and carbon allocation analysis. *J. Biosci. Bioeng.*, **118**, 20–28.

507. Du,W., Jongbloets,J.A., van Boxtel,C., Pineda Hernández,H., Lips,D., Oliver,B.G., Hellingwerf,K.J. and Branco dos Santos,F. (2018) Alignment of microbial fitness with engineered product formation: obligatory coupling between acetate production and photoautotrophic growth. *Biotechnology for Biofuels*, **11**, 38.

508. Jantama,K., Haupt,M.J., Svoronos,S.A., Zhang,X., Moore,J.C., Shanmugam,K.T. and Ingram,L.O. (2008) Combining metabolic engineering and metabolic evolution to develop nonrecombinant strains of Escherichia coli C that produce succinate and malate. *Biotechnol. Bioeng.*, **99**, 1140–1153.

509. Jiang,L.-Y., Chen,S.-G., Zhang,Y.-Y. and Liu,J.-Z. (2013) Metabolic evolution of Corynebacterium glutamicum for increased production of L-ornithine. *BMC Biotechnol.*, **13**, 47.

510. Layton,D.S. and Trinh,C.T. (2014) Engineering modular ester fermentative pathways in Escherichia coli. *Metab. Eng.*, **26**, 77–88.

511. Shastri,A.A. and Morgan,J.A. (2005) Flux Balance Analysis of Photoautotrophic Metabolism. *Biotechnology Progress*, **21**, 1617–1626.

512. Yoshikawa,K., Toya,Y. and Shimizu,H. (2017) Metabolic engineering of Synechocystis sp. PCC 6803 for enhanced ethanol production based on flux balance analysis. *Bioprocess Biosyst Eng*, **40**, 791–796.

513. Hirokawa,Y., Matsuo,S., Hamada,H., Matsuda,F. and Hanai,T. (2017) Metabolic engineering of Synechococcus elongatus PCC 7942 for improvement of 1,3-propanediol and glycerol production based on in silico simulation of metabolic flux distribution. *Microb. Cell Fact.*, **16**, 352.

514. Zhou,Y., Nambou,K., Wei,L., Cao,J., Imanaka,T. and Hua,Q. (2013) Lycopene production in recombinant strains of Escherichia coli is improved by knockout of the central carbon metabolism gene coding for glucose-6-phosphate dehydrogenase. *Biotechnol. Lett.*, **35**, 2137–2145.

515. Alper,H., Miyaoku,K. and Stephanopoulos,G. (2005) Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knockout targets. *Nat. Biotechnol.*, **23**, 612–616.

516. Li,R. and Golden,S.S. (1993) Enhancer activity of light-responsive regulatory elements in the untranslated leader regions of cyanobacterial psbA genes. *Proceedings of the National Academy of Sciences*, **90**, 11678–11682.

517. Dühring,U., Axmann,I.M., Hess,W.R. and Wilde,A. (2006) An internal antisense RNA regulates expression of the photosynthesis gene isiA. *Proceedings of the National Academy of Sciences*, **103**, 7054–7058.

518. Cameron,J.C., Gordon,G.C. and Pfleger,B.F. (2015) Genetic and genomic analysis of RNases in model cyanobacteria. *Photosynthesis Research*, **126**, 171–183.

519. Liu,C.C., Jewett,M.C., Chin,J.W. and Voigt,C.A. (2018) Toward an orthogonal central dogma. *Nat. Chem. Biol.*, **14**, 103–106.

520. Ravikumar,A., Arrieta,A. and Liu,C.C. (2014) An orthogonal DNA replication system in yeast. *Nat. Chem. Biol.*, **10**, 175–177.

521. Chin,J.W. (2014) Expanding and reprogramming the genetic code of cells and animals. *Annu. Rev. Biochem.*, **83**, 379–408.

522. Wang,K., Sachdeva,A., Cox,D.J., Wilf,N.M., Lang,K., Wallace,S., Mehl,R.A. and Chin,J.W. (2014) Optimized orthogonal translation of unnatural amino acids enables spontaneous protein double-labelling and FRET. *Nat Chem*, **6**, 393–403.

523. Ostrov,N., Landon,M., Guell,M., Kuznetsov,G., Teramoto,J., Cervantes,N., Zhou,M., Singh,K., Napolitano,M.G., Moosburner,M., *et al.* (2016) Design, synthesis, and testing toward a 57-codon genome. *Science*, **353**, 819–822.

524. Wang,K., Fredens,J., Brunner,S.F., Kim,S.H., Chia,T. and Chin,J.W. (2016) Defining synonymous codon compression schemes by genome recoding. *Nature*, **539**, 59–64.

525. Sezonov,G., Joseleau-Petit,D. and D'Ari,R. (2007) Escherichia coli Physiology in Luria-Bertani Broth. *Journal of Bacteriology*, **189**, 8746–8749.

526. Tamoi,M., Miyazaki,T., Fukamizo,T. and Shigeoka,S. (2005) The Calvin cycle in cyanobacteria is regulated by CP12 via the NAD(H)/NADP(H) ratio under light/dark conditions. *The Plant Journal*, **42**, 504–513.

527. Cooley,J.W. and Vermaas,W.F. (2001) Succinate dehydrogenase and other respiratory pathways in thylakoid membranes of Synechocystis sp. strain PCC 6803: capacity comparisons and physiological function. *Journal of Bacteriology*, **183**, 4251–4258.

528. Holm,A.K., Blank,L.M., Oldiges,M., Schmid,A., Solem,C., Jensen,P.R. and Vemuri,G.N. (2010) Metabolic and transcriptional response to cofactor perturbations in Escherichia coli. *J. Biol. Chem.*, **285**, 17498–17506.

529. Xiang,S., Usunow,G., Lange,G., Busch,M. and Tong,L. (2007) Crystal structure of 1-deoxy-D-xylulose 5-phosphate synthase, a crucial enzyme for isoprenoids biosynthesis. *J. Biol. Chem.*, **282**, 2676–2682.

530. Wang,Y., San,K.-Y. and Bennett,G.N. (2013) Improvement of NADPH bioavailability in Escherichia coli by replacing NAD(+)-dependent

glyceraldehyde-3-phosphate dehydrogenase GapA with NADP (+)-dependent GapB from Bacillus subtilis and addition of NAD kinase. *J. Ind. Microbiol. Biotechnol.*, **40**, 1449–1460.

531. Moore,S.J., Lai,H.-E., Needham,H., Polizzi,K.M. and Freemont,P.S. (2017) Streptomyces venezuelae TX-TL - a next generation cell-free synthetic biology tool. *Biotechnol J*, **12**, 1600678.

532. Moore,S.J., MacDonald,J.T., Wienecke,S., Ishwarbhai,A., Tsipa,A., Aw,R., Kylilis,N., Bell,D.J., McClymont,D.W., Jensen,K., *et al.* (2018) Rapid acquisition and model-based analysis of cell-free transcription-translation reactions from nonmodel bacteria. *Proc. Natl. Acad. Sci. U.S.A.*, **115**, E4340–E4349.

533. Li,S., Sun,T., Xu,C., Chen,L. and Zhang,W. (2018) Development and optimization of genetic toolboxes for a fast-growing cyanobacterium Synechococcus elongatus UTEX 2973. *Metab. Eng.*, **48**, 163–174.

534. Liang,F. and Lindblad,P. (2017) Synechocystis PCC 6803 overexpressing RuBisCO grow faster with increased photosynthesis. *Metab Eng Commun*, **4**, 29–36.

535. Segall-Shapiro,T.H., Sontag,E.D. and Voigt,C.A. (2018) Engineered promoters enable constant gene expression at any copy number in bacteria. *Nat. Biotechnol.*, **36**, 352–358.

536. Tomazou,M. and Stan,G.-B. (2018) Portable gene expression guaranteed. *Nat. Biotechnol.*, **36**, 313–314.

537. Chen,Y., Xiao,W., Wang,Y., Liu,H., Li,X. and Yuan,Y. (2016) Lycopene overproduction in Saccharomyces cerevisiae through combining pathway engineering with host engineering. *Microb. Cell Fact.*, **15**, 113.

538. Johnson,T.J., Katuwal,S., Anderson,G.A., Gu,L., Zhou,R. and Gibbons,W.R. (2018) Photobioreactor cultivation strategies for microalgae and cyanobacteria. *Biotechnology Progress*, **32**, 583.

539. Schmied,W.H., Tnimov,Z., Uttamapinant,C., Rae,C.D., Fried,S.D. and Chin,J.W. (2018) Controlling orthogonal ribosome subunit interactions enables evolution of new function. *Nature*, **564**, 444–448.

APPENDIX

Appendix Tables

Table A1. Sequences of promoter clones generated in the Synthetic Promoter Library. -35 and -10 regions are shown in **bold**.

SPL Library promoter clone	Sequence -35 region	-10 region
1	CATCGCCCGA**TTGACA**CCTCGCGCCACGCCTGA**TATAAT**GGGGGG
2	TCAAGGGAAG**TTGACA**CGACACCGCTAATTTAT**TATAAT**GAAGAG
3	CCAGATGACC**TTGACA**AGTAACTGCCCCAACAG**TATAAT**ATACCT
5	CAGTGTGGCG**TTGACA**TTCAGTCATGCACCCCG**TATAAT**TGGCGA
6	GCGGGGTTGC**TTGACA**GGATTTACTCGGCAGGG**TATAAT**GGGTAC
10	CGCAGCGGGC**TTGACA**CATCGGCCCAAACTACA**TATAAT**GTGTAT
13	CCCGCAGCTT**TTGACA**CACCGCCACTGACAGTC**TATAAT**TTGGAT
14	TAGTATAAGA**TTGACA**TAATGTAGTCCGTGGAG**TATAAT**GTAATT
15	CACCAGAGGA**TTGACA**GTCCTAGCAGACACGTG**TATAAT**AGTCCC
17	CCCCTAAGCT**TTGACA**CGCAGAGAACACAACCC**TATAAT**GATTCG
19	GCGGCTCGTA**TTGACA**ATTTAACTAAATATCAG**TATAAT**TTAGTT
22	ATAGGCACCG**TTGACA**TACTAGCCAGAAGAAAA**TATAAT**TGGGGG
23	CGGGTGGCAA**TTGACA**TGGCACGAGTCCGTCTA**TATAAT**AACAGA
25	CCACGAAAAA**TTGACA**AGCACTCTGCGCTCTGA**TATAAT**TAAGGT
26	CCCCCACCCC**TTGACA**GCCCGTGGCTCGATGCA**TATAAT**GAATGT
28	CCCACATAAC**TTGACA**GAAGTCCGCAGCGACTT**TATAAT**ACGGGG
31	TTTTATTATA**TTGACA**TCTTTCATACACCTGCA**TATAAT**TCGAGT
32	CGGTCGGCCT**TTGACA**GCGGTTGATTAAGAGAC**TATAAT**GTCTAT
33	CTCGCGGCAA**TTGACA**AAATTGGATCTCGATAT**TATAAT**GGGGGT
34	CCGACATCCT**TTGACA**CGAACTCGCTGGCCCGG**TATAAT**TCTTTA
35	ACACGCCGCC**TTGACA**TCACTTTACGGCACCCA**TATAAT**GGGGGG
36	GCCCCCCGCG**TTGACA**CCCGCCCCGTACTCGCC**TATAAT**TAAGGG
40	CCAAATTATA**TTGACA**TTACCGGGGTCGTGAAC**TATAAT**TCTTTC
41	TGCCCCCCCC**TTGACA**ACCGCTGTCCCGCATCC**TATAAT**TGTTCA
45	CTCGCCACCT**TTGACA**CAGCGGCCCGCTAACTA**TATAAT**AAACGT
46	AGCGCGGACC**TTGACA**CTCCTAGTGACTTACAC**TATAAT**AGCTGG
47	GTGACGTCTA**TTGACA**GCCAGGAGTTACCGAGA**TATAAT**GGAATA
52	CCCCCTCCCC**TTGACA**CCCTCCCATCCCGATGT**TATAAT**GTGTTT
61	TCTAGCCAGA**TTGACA**CTGACATTAAGTACACT**TATAAT**CGGAGG
69	GTCGCCCCCA**TTGACA**AATGTAAGTCGCGGAGG**TATAAT**AGGTGC
75	CTCACGGACC**TTGACA**GACGGGGGGCCCCCAGG**TATAAT**GAGGGG
98	ACCATCTAGT**TTGACA**TCGAAAAGCCAACAACG**TATAAT**CAGCTT
107	CGCGCTACCA**TTGACA**TCCCGTTCCCGCCGCTG**TATAAT**TGATTT
112	CGAAAACACA**TTGACA**CCCTTTTATAATCATGA**TATAAT**TGGATC
115	CCGGACCCCC**TTGACA**CTACGGCCCGCTATGGA**TATAAT**TCACCG
157	CACCGGGCCT**TTGACA**ATACGTATCAAGTACCT**TATAAT**AAGGGG
192	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TATAAT**GTGTAA

Table A2. Sequences of promoter clones generated in the Promoter Library 2. The -35 and -10 regions of the promoter sequence are highlighted, the consensus sequences are highlighted in yellow. -35 and -10 sequences different to the consensus sequences are highlighted in blue.

Promoter library 2 clones	Sequence -35 region	-10 region
SPLc192	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TATAAT**GTGTAA	
1	CATTCCGTGC**TTGCCT**CATTATTAAACTATGAA**TATACT**GTGTAA	
2	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TACACT**GTGTAA	
3	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TATACT**GTGTAA	
4	CATTCCGTGC**TTGACG**CATTATTAAACTATGAA**TACCAT**GTGTAA	
5	CATTCCGTGC**TTGAAA**CATTATTAAACTATGAA**TACAAT**GTGTAA	
6	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TAATAT**GTGTAA	
8	CATTCCGTGC**TTGCAT**CATTATTAAACTATGAA**TACAAT**GTGTAA	
9	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TAAAAT**GTGTAA	
11	CATTCCGTGC**TTGACG**CATTATTAAACTATGAA**TACAAT**GTGTAA	
12	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TACCAT**GTGTAA	
13	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TATCTT**GTGTAA	
14	CATTCCGTGC**TTGACT**CATTATTAAACTATGAA**TACCAT**GTGTAA	
15	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TACAGT**GTGTAA	
16	CATTCCGTGC**TTGTTG**CATTATTAAACTATGAA**TACCAT**GTGTAA	
17	CATTCCGTGC**TTGACT**CATTATTAAACTATGAA**TACTGT**GTGTAA	
19	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TATGAT**GTGTAA	
20	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TATACT**GTGTAA	
22	CATTCCGTGC**TTGAGG**CATTATTAAACTATGAA**TATAGT**GTGTAA	
24	CATTCCGTGC**TTGCGG**CATTATTAAACTATGAA**TAGACT**GTGTAA	
25	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TAATAT**GTGTAA	
28	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TACAAT**GTGTAA	
30	CATTCCGTGC**TTGCAC**CATTATTAAACTATGAA**TAGCTT**GTGTAA	
31	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TAACAT**GTGTAA	
32	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TATAAT**GTGTAA	
34	CATTCCGTGC**TTGACG**CATTATTAAACTATGAA**TAACAT**GTGTAA	
35	CATTCCGTGC**TTGCGA**CATTATTAAACTATGAA**TACACT**GTGTAA	
36	CATTCCGTGC**TTGACT**CATTATTAAACTATGAA**TACTTT**GTGTAA	
37	CATTCCGTGC**TTGCCC**CATTATTAAACTATGAA**TAACAT**GTGTAA	
39	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TAGTCT**GTGTAA	
40	CATTCCGTGC**TTGACT**CATTATTAAACTATGAA**TACACT**GTGTAA	
41	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TAACAT**GTGTAA	
42	CATTCCGTGC**TTGCCA**CATTATTAAACTATGAA**TAGTCT**GTGTAA	
43	CATTCCGTGC**TTGCTC**CATTATTAAACTATGAA**TAGGCT**GTGTAA	
46	CATTCCGTGC**TTGACG**CATTATTAAACTATGAA**TACTAT**GTGTAA	
47	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TATATT**GTGTAA	
48	CATTCCGTGC**TTGATT**CATTATTAAACTATGAA**TATTCT**GTGTAA	
49	CATTCCGTGC**TTGGTC**CATTATTAAACTATGAA**TATAAT**GTGTAA	
51	CATTCCGTGC**TTGCAA**CATTATTAAACTATGAA**TAACCT**GTGTAA	
52	CATTCCGTGC**TTGCCT**CATTATTAAACTATGAA**TAGAGT**GTGTAA	
53	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TAGAGT**GTGTAA	
54	CATTCCGTGC**TTGCAG**CATTATTAAACTATGAA**TATACT**GTGTAA	

55	CATTCCGTGC**TTGTAG**CATTATTAAACTATGAA**TACACT**GTGTAA
56	CATTCCGTGC**TTGCGT**CATTATTAAACTATGAA**TACCTT**GTGTAA
57	CATTCCGTGC**TTGGGA**CATTATTAAACTATGAA**TAAACT**GTGTAA
58	CATTCCGTGC**TTGCAC**CATTATTAAACTATGAA**TACTAT**GTGTAA
60	CATTCCGTGC**TTGCCA**CATTATTAAACTATGAA**TAATGT**GTGTAA
61	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TAACAT**GTGTAA
63	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TATAAT**GTGTAA
64	CATTCCGTGC**TTGATC**CATTATTAAACTATGAA**TAAGAT**GTGTAA
65	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TACATT**GTGTAA
68	CATTCCGTGC**TTGCAT**CATTATTAAACTATGAA**TATAAT**GTGTAA
69	CATTCCGTGC**TTGCCA**CATTATTAAACTATGAA**TATCAT**GTGTAA
70	CATTCCGTGC**TTGCTC**CATTATTAAACTATGAA**TAACGT**GTGTAA
71	CATTCCGTGC**TTGCCA**CATTATTAAACTATGAA**TATCAT**GTGTAA
72	CATTCCGTGC**TTGCGA**CATTATTAAACTATGAA**TAGCGT**GTGTAA
74	CATTCCGTGC**TTGACC**CATTATTAAACTATGAA**TAAACT**GTGTAA
77	CATTCCGTGC**TTGACA**CATTATTAAACTATGAA**TAGCGT**GTGTAA
78	CATTCCGTGC**TTGACT**CATTATTAAACTATGAA**TACCAT**GTGTAA
79	CATTCCGTGC**TTGCCC**CATTATTAAACTATGAA**TATAGT**GTGTAA
80	CATTCCGTGC**TTGGAT**CATTATTAAACTATGAA**TAATGT**GTGTAA
82	CATTCCGTGC**TTGTAA**CATTATTAAACTATGAA**TAAAAT**GTGTAA
83	CATTCCGTGC**TTGCCG**CATTATTAAACTATGAA**TATCAT**GTGTAA
86	CATTCCGTGC**TTGGAA**CATTATTAAACTATGAA**TAGGAT**GTGTAA
87	CATTCCGTGC**TTGCCG**CATTATTAAACTATGAA**TATCAT**GTGTAA

Table A3. Sequences of RBS clones generated in the RBS Library.

RBS Library Design	RBS Library Clone	Sequence
RBS Library Design 1	1	CTCGATTTCGCGAACAAAGGAGGTGAGGT
	4	TATGTCAGTAAAAAAAAAGGAGGTTCGTT
	12	TAAAGTTAAGCGTCCAAAGGAGGTTTAAC
	13	GCGGATAGTGAGGAGAAAGGAGGTGGGGG
	19	ACAAACTCCTCACGGAAAGGAGGTACGGA
	20	GCAAATTCTCGTGGGAAAGGAGGTCGCGG
	21	TACGAAAGATTTACGAAAGGAGGTTGGTT
	30	ATTTCGTATGAAGGTAAAGGAGGTTGGCA
	36	CTCTAGGGTTTTCTAAAGGAGGTTGAAT
	37	AGTTGACCGTAACGCAAAGGAGGTAGAGC
	45	TCGGCGGCGTAAACTAAAGGAGGTGGAAG
	48	AGTGTGTTCCAAGAGAAAGGAGGTATCTG
	49	AATAAAAGCCAGTTTAAAGGAGGTCTTTG
	52	ATTAGACGACGATCCAAAGGAGGTCCCAG
	57	TGGACTCCTACGAGGAAAGGAGGTGGTA
	66	CTTATGGGTGTTTGAAAGGAGGTGGACG
	74	AGTACTAACGGGTTTAAAGGAGGTCGCGG

	83	GCGGATAGTGAGGAGAAAGGAGGTGGGGG
	92	AAACCGAACAGGCCAAAGGAGGTCAGCT
	102	TTTAGTATTGCTACTAAAGGAGGTTGCAC
	107	TTGGGGCGATTTGGGAAAGGAGGTGCGGG
	110	TAGGTATATGCAATCAAAGGAGGTATATG
	120	ATACTGCATCAATGAAAGGAGGTAGGGG
	123	TGTGTAGGTCCCACCAAAGGAGGTCTGTA
RBS Library Design 4	133	TTACTATAATAACGTTTCGGAGGGAATAT
	141	TTCAAATAAGCATTGATCGGAGGGTTTAC
	142	GGGATATAACGAAGAGGCGGAGGAGTGCG
	150	TGTTAAGAGGGGGACTGTGAGGGCTGAG
	155	TTACCGGTCAAGGCAAGAGGAGGCTAGTC
	157	TATTATGTGATAATTAGAGGCGTGCA
	167	GTATATGGTGGAGGAGGTAGCGT
	170	CTGATTTAGGGGCCATTTTGAGGAATATG
	177	GTAGGTTGGATATCTGGAGGTGAAGC
	184	CTGCGTGACAACAGATGTCGAGGTGAAAG
	187	CTCTATTGTGTACTATGAGGAGGCAGCGT
	193	GTTGCGTAACAATAGACGGAGGATGCTT
	197	CTGGTATGGTTCAACGAGGATCGTT
	201	CTATCGGTTATGTGCTAGCGAGGTTTAAT
	207	GTGCAGACGATGAGAAGGGAGGAAAAGA
	212	TTTCGAGGAGGATAGCG
	213	TTTCGAGGAGGATAGCG
	216	TTTCGAGGAGGATAGCG
	219	GAGGGTATAGTGCGTAACGGAGGGGTACG
	225	GTAACGGGGAGTTTAAGTAGAGGGGGTGT
	229	CGGTGTCTTTATTCTAATTGAGGGGGCTT
	230	TTGAAAGGGACCCCACAGTGAGGTAGAGC
	232	TAGGCGTAAGGATCTATCGAGGAAGTGT
	234	TAATGGTCAGCGTGTGCGGAGGAGCGTG
	238	TACGGGTTAGTTTGTTGTGGAGGTTACTT
	240	TCAATTGCTTTGGCGGTGAGAGGCGTGTT
	244	TGGACATTCAAATATTGAAGAGGGGTGTA
	246	CCCTATTAAAAACAGACGGAGGCAGAAT
RBS Library Design 9	249	GGCGCGAGTCAACGTTCCTGAGGAGTGGC
	250	TCTTTGAGACTTTATTGTGGAGGTTTGGC
	257	CGGAGCAATCAACGAAAGGGGGAAGAGGG
	266	CGGAGCAATCAACGAAAGGGGGAAGAGGG
	267	GTTTCTTAGGTGTTGGAGGTTGGTCTA
	274	AGAGCATTAGGTGATAGGAGGTAAGTA

Table A4. Oligonucleotides used in the study. 5'-phosphorylated oligonucleotides denoted /5Phos/. Fw denotates forward primer. Rv denotes reverse primer.

Primer	Sequence (5' – 3')	Comments
oligoATM3	/5phos/NNNNNNNNNNNNNNNNTGTCAANNNNNNNNNNCTGCAGTCTCCTTTAAATCGG	Rv primer for creating SPL library in pATM2
oligoATM5	/5phos/TATAATNNNNNNCCGGCTTCGGCGGTAAGGTAAGAGCAGGATTAGAGGGAGGTCAGAGAATGGTGAGCAAGGGCGAGG	Fw primer for creating SPL library in pATM2
oligoATM51	/5phos/AGGTATATTATACTGTTGGGGCAG	Rv primer for creating RBS library in pATM2
oligoATM52	NNNNNNNNNNNNNNNAAAGGAGGTNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 1 in pATM2
oligoATM53	NNNNNNNNNNNNNNNNAAGGAGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 2 in pATM2
oligoATM54	NNNNNNNNNNNNNNNNNAGGAGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 3 in pATM2
oligoATM55	NNNNNNNNNNNNNNNNNNGGAGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 4 in pATM2
oligoATM56	NNNNNNNNNNNNNNNNNNNGAGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 5 in pATM2
oligoATM57	NNNNNNNNNNNNNNNNNNNNAGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 6 in pATM2
oligoATM58	NNNNNNNNNNNNNNNNNNNNNGGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 7 in pATM2
oligoATM59	NNNNNNNNNNNNNNNNNNNNNNGNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 8 in pATM2
oligoATM60	NNNNNNNNNNNNNNNNNNNNNNNNNNNNNATGGTGAGCAAGGGCGAGG	Fw primer for creating RBS library design 9 in pATM2
oligoGT22	/5phos/GTGTAACCGGCTTCGGCGGTAAGGTAAGAGCAGGATTAGAGGGAGGTCAGAGAATGGTGAGCAAGGGCGAGG	Rv primer for creating Promoter Library 2 in SPLc192
oligoGT23	NNNNNNTTCATAGTTTAATAATGNNNNNNGCACGGAATGCTGCAGTCTCCTTTAAATCGG	Fw primer for creating Promoter Library 2 design 1 in SPLc192
oligoGT24	ANNNTATTCATAGTTTAATAATGNNNCAAGCACGGAATGCTGCAGTCTCCTTTAAATCGG	Fw primer for creating Promoter Library 2 design 1 in SPLc192
oligoGT25	ANNNTATTCATAGTTTAATAATGNNNNNNGCACGGAATGCTGCAGTCTCCTTTAAATCGG	Fw primer for creating Promoter Library 2 design 1 in SPLc192
oligoGT26	WWWWWTTCATAGTTTAATAATGNNNCAAGCACGGAATGCTGCAGTCTCCTTTAAATCGG	Fw primer for creating Promoter Library 2 design 1 in SPLc192

oligoGT114	/5Phos/GGCGCGAGTCAACGTTCCTGAGGAGTGGCATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc249
oligoGT115	/5Phos/CTGGTATGGTTCAACGAGGATCGTTATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc197
oligoGT116	/5Phos/ATTAGACGACGATCCAAAGGAGGTCCCAGATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc52
oligoGT117	/5Phos/ATGATACCTGCAATTCAAGGAGGTCTTATATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc266
oligoGT118	/5Phos/TTTAGTATTGCTACTAAAGGAGGTTGCACATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc102
oligoGT119	/5Phos/CGGAGCAATCAACGAAAGGGGGAAGAGGGATGGTGAGCAAGGGCGAGG	Fw primer for EYFP combinatorial build, primer tail includes RBSc257
oligoGT120	TTACACAAGGTATTCATAGTTTAATAATGACGCAAGCACGGAATGCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes PL2c56
oligoGT121	GCACCTATTATACCTCCGCGACTTACATTTGTCAATGGGGGCGACCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes SPLc69
oligoGT122	CCAGCTATTATAGTGTAAGTCACTAGGAGTGTCAAGGTCCGCGCTCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes SPLc46
oligoGT123	CCCCGTATTATAAAGTCGCTGCGGACTTCTGTCAAGTTATGTGGGCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes SPLc28
oligoGT124	AAATCAATTATACAGCGGCGGGAACGGGATGTCAATGGTAGGGCGCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes SPLc107
oligoGT125	CCCCTTATTATAAGGTACTTGATACGTATTGTCAAAGGCCCGGTGCTGCAGTCTCCTTTAAATCGG	RV primer for EYFP combinatorial build, primer tail includes SPLc157
oligoGT126	ATGGTTAGTAAAGGTGAAGAACTGTTCACTGGTGTGGTGCCCATCCTGGTCGAGC	Fw primer for construct pGT12
oligoGT127	/5Phos/CTGGGACCTCCTTTGGATCGTCGTC	Rv primer for constructing pGT12
oligoGT128	TTTAGTATTGCTACTAAAGGAGGTTGCACATGGTTAGTAAAGGTGAAGAACTG	Fw primer for constructing pGT55 and pGT60
oligoGT129	CTGGTATGGTTCAACGAGGATCGTTATGGTTAGTAAAGGTGAAGAACTG	Fw primer for constructing pGT53 and pGT58
oligoGT130	GGCGCGAGTCAACGTTCCTGAGGAGTGGCATGGTTAGTAAAGGTGAAGAACTG	Fw primer for constructing pGT52 and pGT57

oligoGT131	CGGAGCAATCAACGAAAGGGGGAAGAG GGATGGTTAGTAAAGGTGAAGAACTG	Fw primer for constructing pGT56 and pGT61
oligoGT132	ATGATACCTGCAATTCAAGGAGGTCTT ATATGGTTAGTAAAGGTGAAGAACTG	Fw primer for constructing pGT54 and pGT59
oligoGT133	/5Phos/CCAGCTATTATAGTGTAAGT CACTAGGAGTGTCAAGGTCCGCGCTCT GCAGTCTCCTTTAAATCGG	Rv primer for constructing pGT52 – 72
oligoGT134	ATTAGACGACGATCCAAAGGAGGTCCC AGATGACGGCATTGACGGAAG	Fw primer for constructing pGT62
oligoGT135	TTTAGTATTGCTACTAAAGGAGGTTGC ACATGACGGCATTGACGGAAG	Fw primer for constructing pGT66
oligoGT136	CTGGTATGGTTCAACGAGGATCGTTAT GACGGCATTGACGGAAG	Fw primer for constructing pGT64
oligoGT137	GGCGCGAGTCAACGTTCCTGAGGAGTG GCATGACGGCATTGACGGAAG	Fw primer for constructing pGT63
oligoGT138	CGGAGCAATCAACGAAAGGGGGAAGAG GGATGACGGCATTGACGGAAG	Fw primer for constructing pGT67
oligoGT139	ATGATACCTGCAATTCAAGGAGGTCTT ATATGACGGCATTGACGGAAG	Fw primer for constructing pGT65
oligoGT140	GGATCCAAAGCCACGTTGTGTCTC	Fw primer for amplifying pGT30 backbone
oligoGT141	CTGGGACCTCCTTTGGATCGTCGTC	Fw primer for amplifying pGT30 backbone
oligoGT142	/5Phos/atgttacgtcctgtagaaacc	Fw primer for amplifying *gusA*
oligoGT143	/5Phos/tcattgtttgcctccctgc	Rv primer for amplifying *gusA*
oligoGT144	TTTAGTATTGCTACTAAAGGAGGTTGC ACatgttacgtcctgtagaaaccc	Fw primer for constructing pGT71
oligoGT145	CTGGTATGGTTCAACGAGGATCGTTat gttacgtcctgtagaaaccc	Fw primer for constructing pGT69
oligoGT146	GGCGCGAGTCAACGTTCCTGAGGAGTG GCatgttacgtcctgtagaaaccc	Fw primer for constructing pGT68
oligoGT147	CGGAGCAATCAACGAAAGGGGGAAGAG GGatgttacgtcctgtagaaaccc	Fw primer for constructing pGT72
oligoGT148	ATGATACCTGCAATTCAAGGAGGTCTT ATatgttacgtcctgtagaaaccc	Fw primer for constructing pGT70
oligoGT234	GGGGAAACGCCTGGTATCT	pStA0 Fw sequencing primer
oligoGT235	AGCAAAAACAGGAAGGCAAA	pStA0 Rv sequencing primer
oligoGT249	TGGAGTTCTGAGGTCATTACTGGATCT ATCAACAGGAGTCCAAGGCGCTCGGTC GTTCGGCT	Rv primer for amplifying pGT421 backbone fragment 1
oligoGT250	AAAACGAAAGGCCCAGTCTTTCGACTG AGCCTTTCGTTTTATTTGATGCCCGAA AGGGCCTCGTGATACGCC	Fw primer for amplifying pGT421 backbone fragment 2
oligoGT251	CCGGTGAGCGTGGGTCCCGCGGTATCA TTGCAGCAC	Fw primer for amplifying pGT421 backbone fragment 1
oligoGT252	GTGCTGCAATGATACCGCGGGACCCAC GCTCACCGG	Rv primer for amplifying pGT421 backbone fragment 2

oligoGT253	TTGGTCTCTCACACTGGATTCTCACC	Fw primer for amplifying pStA0 backbone from pGT421
oligoGT254	GGTCTCAGACCTCTAGGGCG	Rv primer for amplifying pStA0 backbone from pGT421
oligoGT255	CTATGCGGCATCAGAGCAGATTG	Fw primer for amplifying *laczα* cassette from pUC19
oligoGT256	GGGCAGTGAGCGCAACGC	Rv primer for amplifying *laczα* cassette from pUC19
oligoGT257	CAAAAGTTGGCCCAGGGCT	Fw primer for amplifying pGT422 backbone fragment 1
oligoGT258	GCTCAGGTCGCAGACGT	Rv primer for amplifying pGT422 backbone fragment 2
oligoGT259	GTTGCATGATAAAGAAAACAGTCATAAGTGC	Rv primer for amplifying pGT422 backbone fragment 1
oligoGT260	CGCACTTATGACTGTTTTCTTTATCATGCAA	Fw primer for amplifying pGT422 backbone fragment 2
oligoGT261	/5Phos/GCCGCAGCCGAACGAC	Fw primer for amplifying *laczα* cassette from pStA0
oligoGT262	/5Phos/GCGTATCACGAGGCCCT	Rv primer for amplifying *laczα* cassette from pStA0
oligoGT265	/5Phos/ATTTGTCCTACTCAGGAGAGCGT	Rv primer for amplifying pStA1AZ, pGT411, pGT412 and pGT413 backbones
oligoGT266	TACTGGCTCTTCTCTGCTCCTGAGACCATTCTCACCAATAAAAAACGCCCG	Fw primer for amplifying pStA1AZ backbone from pGT422
oligoGT268	/5Phos/AGAGACCATTTGTCCTACTCAGGAGAG	Rv primer for construct pStA1BC, pStA1CD, pStA1DE, pStA1AB
oligoGT270	TACTGGCTCTTCTCTGCATTTGAGACCATTCTCACCAATAAAAAACGCC	Fw primer for constructing pStA1BC
oligoGT273	TACTGGCTCTTCTCTGACCTTGAGACCATTCTCACCAATAAAAAACGCC	Fw primer for constructing pStA1CD
oligoGT276	TACTGGCTCTTCTCTGAAGCTGAGACCATTCTCACCAATAAAAAACGCC	Fw primer for constructing pStA1DE
oligoGT279	/5Phos/TGAGACCATTCTCACCAATAAAAAACGCC	Rv primer for constructing pStA1BZ, pStA1CZ, pStA1DZ, pStA1EZ
oligoGT289	TACTGGGTCTCTCTCCGGCATTGTCTTCATTCTCACCAATAAAAAACGCCCG	Fw primer for amplifying pGT411 backbone
oligoGT297	CTTTATCATGCAACTCGTAGGACAGG	Fw primer for amplifying pGT414 backbone
oligoGT298	AATTTAACTGTGATAAACTACCGCATTAAAGCT	Rv primer for amplifying pGT414 backbone

oligoGT299	TACTGGAAGACTTGGCAATTCTCACCA ATAAAAAACGCCCG	Fw primer for constructing pStA313
oligoGT317	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGCACCGGGCCTTTGAC	Clone SPLc157 into pStA0 as a composite part
oligoGT318	GGTCTCTGGTCTTACGCTCTTCACATC CCTCTTCCCCCTTTCGTTG	Clone RBSc257 into pStA0 as a composite part
oligoGT319	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGAGCGCGGACCTTGACAC	Clone SPLc46 into pStA0 as a composite part
oligoGT320	GGTCTCTGGTCTTACGCTCTTCACATC TGGGACCTCCTTTGGATCG	Clone RBSc52 into pStA0 as a composite part
oligoGT321	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGCGCCCTACCATTGACATCCCG	Clone SPLc107 into pStA0 as a composite part
oligoGT322	GGTCTCTGGTCTTACGCTCTTCACATG TGCAACCTCCTTTAGTAGCAATAC	Clone RBSc102 into pStA0 as a composite part
oligoGT323	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGCCCACATAACTTGACAGAAGTCCG	Clone SPLc28 into pStA0 as a composite part
oligoGT324	GGTCTCTGGTCTTACGCTCTTCACATA TAAGACCTCCTTGAATTGCAGG	Clone RBSc266 into pStA0 as a composite part
oligoGT325	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGGTCGCCCCCATTGACAAATG	Clone SPLc69 into pStA0 as a composite part
oligoGT326	GGTCTCTGGTCTTACGCTCTTCACATA ACGATCCTCGTTGAACCATAC	Clone RBSc197 into pStA0 as a composite part
oligoGT327	AAGGGGTTGGTCTCATGTGGCTCTTCG CAGCATTCCGTGCTTGCGTC	Clone PL2c56 into pStA0 as a composite part
oligoGT328	GGTCTCTGGTCTTACGCTCTTCACATG CCACTCCTCAGGAACG	Clone RBSc249 into pStA0 as a composite part
oligoGT329	ACGCTGAAAAGCGTCTTTTTTCGTTTT GGTCCGGACGAAGAGCGACCTCTAGGG CGGCGG	Fw primer for cloning L3S2P55 into pStA0
oligoGT330	CTTATTGTTCGTCTTTGGTACCGAGTT ATGAAGAGCCACACTGGATTCTCACCA ATAAAAAACGC	Rv primer for cloning L3S2P55 into pStA0
oligoGT331	CCCGAAAGGGGGGCCTTTTTTCGTTTT GGTCCGGACGAAGAGCGACCTCTAGGG CGGCGG	Fw primer for cloning L3S2P21 into pStA0
oligoGT332	AGGCCTCTTTTCTGGAATTTGGTACCG AGTTATGAAGAGCCACACTGGATTCTC ACCAATAAAAAACGC	Rv primer for cloning L3S2P21 into pStA0
oligoGT333	TGACAGTGCGGGCTTTTTTTTTcgacc aaaggGGACGAAGAGCGACCTCTAGGG CGGCGG	Fw primer for cloning ECK120033737 into pStA0
oligoGT334	GGTGCGGGCTTTTTTctgtgtttccTT ATGAAGAGCCACACTGGATTCTCACCA ATAAAAAACGC	Rv primer for cloning ECK120033737 into pStA0
oligoGT335	GTAATGCGGTGGACAGGATCGGCGGTT TTCTTTTCTCTTCTCAAGGACGAAGAG CGACCTCTAGGGCGGCGG	Fw primer for cloning ECK120029600 into pStA0

oligoGT336	TGCAAGGTAGTGGACAAGACCGGCGGT CTTAAGTTTTTTGGCTGAATTATGAAG AGCCACACTGGATTCTCACCAATAAAA AACGC	Rv primer for cloning ECK120029600 into pStA0
oligoGT339	GTTGAGGACCCGGCTAGG	pStA1 Fw sequencing primer
oligoGT340	TGTGACGGAAGATCACTTCG	pStA1 Rv sequencing primer
oligoGT356	GGTCTCAGGAGCAGAGAAGAGCCAGTA GGGCAGTGAGCGCAACG	Fw primer for amplifying pStA1AZ *lacZα* cassette
oligoGT357	GGTCTCTAGTATCCTGAAGAGCGGAAA CTATGCGGCATCAGAGCAGATTG	Rv primer for amplifying pStA1AZ *lacZα* cassette
oligoGT358	AATGCAGAGAAGAGCCAGTAGGGC	Fw primer for constructing pStA1BZ
oligoGT359	AGGTCAGAGAAGAGCCAGTAGGGC	Fw primer for constructing pStA1CZ
oligoGT360	GCTTCAGAGAAGAGCCAGTAGGGC	Fw primer for constructing pStA1DZ
oligoGT361	CGCTCAGAGAAGAGCCAGTAGGGC	Fw primer for constructing pStA1EZ
oligoGT362	CGCTCAGAGAAGAGCCAGTAGGGC	Fw primer for constructing pStA1AB
oligoGT363	ACCTTCCTGAAGAGCGGAAACTATGCG	Rv primer for constructing pStA1BC
oligoGT364	AAGCTCCTGAAGAGCGGAAACTATGCG	Rv primer for constructing pStA1CD
oligoGT365	AGCGTCCTGAAGAGCGGAAACTATGCG	Rv primer for constructing pStA1DE
oligoGT366	GGTCTCAAATGCAGAGAAGAGCCAGTA GGGCAGTGAGCGCAAC	Fw primer for constructing pStA1BC
oligoGT367	GGTCTCAAGGTCAGAGAAGAGCCAGTA GGGCAGTGAGCGCAAC	Fw primer for constructing pStA1CD
oligoGT368	GGTCTCAGCTTCAGAGAAGAGCCAGTA GGGCAGTGAGCGCAAC	Fw primer for constructing pStA1DE
oligoGT373	TACTGGGTCTCTCTCCGGCATTGTCTT Ccataactgcagtctcctttaaatcgg tttc	Fw primer for constructing pGT270
oligoGT374	/5Phos/ctccaggatccaaagccacg	Rv primer for constructing pGT270
oligoGT375	GAAGACAATGCCGGAGAGAGACCCAGT ACCAGTAGGGCAGTGAGCGCAAC	Fw primer for constructing pGT270
oligoGT376	GAAGACTTTAGTAGTATGAGACCGGAA AGGAAACTATGCGGCATCAGAGC	Rv primer for constructing pGT270
oligoGT377	GAAGACAATGCCGGAGAGAGACCCAGT AGGGCAGTGAGCGCAACGC	Fw primer for constructing pGT411
oligoGT378	GAAGACTTTAGTAGTATGAGACCGGAA ACTATGCGGCATCAGAGCAGATTG	Rv primer for constructing pGT411
oligoGT379	GAAGACAAACTAGGAGAGAGACCCAGT AGGGCAGTGAGCGCAACGC	Fw primer for constructing pGT412

oligoGT380	TACTGGGTCTCTCTCCTAGTTTGTCTTCATTCTCACCAATAAAAAACGCCCG	Fw primer for constructing pGT412
oligoGT381	GAAGACTTGTAAAGTATGAGACCGGAAACTATGCGGCATCAGAGCAGATTG	Rv primer for constructing pGT412
oligoGT382	GAAGACAATTACGGAGAGAGACCCAGTAGGGCAGTGAGCGCAACGC	Fw primer for constructing pGT413
oligoGT383	TACTGGGTCTCTCTCCGTAATTGTCTTCATTCTCACCAATAAAAAACGCCCG	Fw primer for constructing pGT413
oligoGT384	GAAGACTTCTCGAGTATGAGACCGGAAACTATGCGGCATCAGAGCAGATTG	Rv primer for constructing pGT413
oligoGT385	TGCCAAGTCTTCCAGTAGGGCAGTGAGCGCAACGC	Fw primer for constructing pStA313
oligoGT386	GTAATTGTCTTCGGAAACTATGCGGCATCAGAGCAGATTG	Rv primer for constructing pStA313
oligoGT387	CTCGTTGTCTTCGGAAACTATGCGGCATCAGAGCAGATTG	Fw primer for constructing pStA314
oligoGT436	TATGTCAGTAAAAAAAAAGGAGGTTCGTTATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc4 into pStA0 as a composite part
oligoGT437	TACGAAAGATTTACGAAAGGAGGTTGGTTATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc21 into pStA0 as a composite part
oligoGT438	AGTGTGTTCCAAGAGAAAGGAGGTATCTGATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc48 into pStA0 as a composite part
oligoGT439	TAGGTATATGCAATCAAAGGAGGTATATGATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc110 into pStA0 as a composite part
oligoGT440	TGTGTAGGTCCCACCAAAGGAGGTCTGTAATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc123 into pStA0 as a composite part
oligoGT441	GGCGCGAGTCAACGTTCCTGAGGAGTGGCATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for cloning RBSc249 into pStA0 as a composite part
oligoGT442	AGGTATATTATACTGTTGGGGCAGTTACTTGTCAAGGTCATCTGGCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc3 into pStA0 as a composite part
oligoGT443	AACTAAATTATACTGATATTTAGTTAAATTGTCAATACGAGCCGCCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc19 into pStA0 as a composite part
oligoGT444	ACCTTAATTATATCAGAGCGCAGAGTGCTTGTCAATTTTTCGTGGCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc25 into pStA0 as a composite part

oligoGT445	ACGTTTATTATATAGTTAGCGGGCCGCTGTGTCAAAGGTGGCGAGCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc45 into pStA0 as a composite part
oligoGT446	TATTCCATTATATCTCGGTAACTCCTGGCTGTCAATAGACGTCACCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc47 into pStA0 as a composite part
oligoGT447	CGGTGAATTATATCCATAGCGGGCCGTAGTGTCAAGGGGGTCCGGCTGTGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning SPLc115 into pStA0 as a composite part
oligoGT468	AAAANNNatgcgtaaaggcgaagagctg	Fw primer for constructing pGT274
oligoGT472	acctcctaaaagttaaacaaaattatttctagaggg	Rv primer for constructing pGT274
oligoGT482	AAATCTGGAGCCGGTGAGCGT	Fw primer for amplifying pGT410 backbone
oligoGT483	TAATTTCCCCCAAGATTAGAAAAACTCATCGAGCATCAAATG	Rv primer for amplifying pGT410 backbone
oligoGT486	GATTACGCGCAGACCAAAAC	pStA2 Rv sequencing primer pStA3 Fw sequencing primer
oligoGT487	AAACGGTTAGCGCTTCGTTA	pStA3 Rv sequencing primer
oligoGT503	/5Phos/CTGCGAAGAGCCACACTGGATTCTCACCAATAAAAAACG	Rv primer for cloning promoters into pStA0
oligoGT504	ttgacggctagctcagtcctaggtacagtgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23100
oligoGT505	ttgacagctagctcagtcctaggtactgtgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23102
oligoGT507	tttacggctagctcagccctaggtattatgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23107
oligoGT508	ttgacagctagctcagtcctagggactatgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23116
oligoGT509	ctgatggctagctcagtcctagggattatgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23113
oligoGT510	/5Phos/TGGCGAAGAGCCACACTGGATTCTCACCAATAAAAAACG	Rv primer for cloning RBSs into pStA0
oligoGT512	CTACGTTTTTTAGAAAAAGGAGGTATGCGAGATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc13

oligoGT513	AAAACACTAGACTGGAAAGGAGGTAGAGAATATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc33
oligoGT514	ATCGGATTGGATCCAAAGGAGGTTATACCGATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc44
oligoGT515	CATGATCGAATGATTAAAGGAGGTTGGAGGTATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc58
oligoGT516	AGCTCCTTAGCTCCTAAAGGAGGTAGTACATATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc36
oligoGT517	ACAGGATACATCTGTAAAGGAGGTAACGATGATGTGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::RBSc42
oligoGT538	CAGTGGTCAGCGACT	Spacer 1 alpha (Fw primer)
oligoGT539	TGGAGTCGCTGACCA	Spacer 1 beta (Rv primer)
oligoGT541	TAATGGTCAGCGACT	Spacer 1 delta (Fw primer)
oligoGT542	TCCAGTCGCTGACCA	Spacer 1 epsilon (Rv primer)
oligoGT543	CAGGCTGCCGTGAAT	Spacer 20 alpha (Fw primer)
oligoGT544	TGGATTCACGGCAGC	Spacer 20 beta (Rv primer)
oligoGT545	TAAGCTGCCGTGAAT	Spacer 20 delta (Fw primer)
oligoGT546	TCCATTCACGGCAGC	Spacer 20 epsilon (Rv primer)
oligoGT547	CAGGGCACGCTCAAT	Spacer 23 alpha (Fw primer)
oligoGT548	TGGATTGAGCGTGCC	Spacer 23 beta (Rv primer)
oligoGT549	TAAGGCACGCTCAAT	Spacer 23 delta (Fw primer)
oligoGT550	TCCATTGAGCGTGCC	Spacer 23 epsilon (Rv primer)
oligoGT551	CAGAGTCCGTGCTCA	Spacer 46 alpha (Fw primer)
oligoGT552	TGGTGAGCACGGACT	Spacer 46 beta (Rv primer)
oligoGT553	TAAAGTCCGTGCTCA	Spacer 46 delta (Fw primer)
oligoGT554	TCCTGAGCACGGACT	Spacer 46 epsilon (Rv primer)
oligoGT555	CAGATTCTGTGCCGC	Spacer 48 alpha (Fw primer)
oligoGT556	TGGGCGGCACAGAAT	Spacer 48 beta (Rv primer)
oligoGT557	TAAATTCTGTGCCGC	Spacer 48 delta (Fw primer)
oligoGT558	TCCGCGGCACAGAAT	Spacer 48 epsilon (Rv primer)
oligoGT559	CAGATCAACGCCTGC	Spacer 98 alpha (Fw primer)
oligoGT560	TGGGCAGGCGTTGAT	Spacer 98 beta (Rv primer)
oligoGT561	TAAATCAACGCCTGC	Spacer 98 delta (Fw primer)
oligoGT562	TCCGCAGGCGTTGAT	Spacer 98 epsilon (Rv primer)
oligoGT563	CAGATCTGCGGCAAC	Spacer 120 alpha (Fw v)
oligoGT564	TGGGTTGCCGCAGAT	Spacer 120 beta (Rv primer)
oligoGT565	TAAATCTGCGGCAAC	Spacer 120 delta (Fw primer)
oligoGT566	TCCGTTGCCGCAGAT	Spacer 120 epsilon (Rv primer)
oligoGT567	CAGTGCGACCTGACT	Spacer 137 alpha (Fw primer)
oligoGT568	ttgacggctagctcagtcctaggtattgtgctagcCCATGAAGAGCGTAAGACCTCTAGGGCGGCG	Fw primer for constructing pStA0::J23118

oligoGT569	/5Phos/CGGTATCATTGCAGCACTGGGG	Fw primer for constructing pStA212, pStA223 and pStA234
oligoGT570	CGCTCACCGGCTCCAGATTTG	Rv primer for constructing pStA212, pStA223 and pStA234
oligoGT573	CCTCGGTGAGTTTTCTCCTTC	pStA2 Fw sequencing primer
oligoGT580	TGCGGCCTTTTTTCTTTTCACTGGACGAAGAGCGACCTCTAGGGCGGCGG	Fw primer for cloning ECK120017009 into pStA0
oligoGT581	GAGCGGCCTTTTTAGTTAGATCTTATGAAGAGCCACACTGGATTCTCACCAATAAAAAACGC	Rv primer for cloning ECK120017009 into pStA0
oligoGT603	CCATAAAGCCAACCACACCT	Rv primer to check for presence of lycopene pathway in *ssl0410* locus
oligoGT605	TTGATCGAACGGGGTGTTAT	Fw primer to check for presence of lycopene pathway in *ssl0410* locus
oligoGT614	AAGGGGTTGGTCTCATGTGGCTCTTCGATGGTGAGCAAGGGCGAG	Fw primer for amplifying *eyfp* with pStA0
oligoGT615	AAGGGGTTGGTCTCTGGTCTTACGCTCTTCATTACTTGTACAGCTCGTCCATGCC	Rv primer for amplifying *eyfp* with pStA0
oligoGT615	TAATGCGACCTGACT	spacer 8 delta (Fw primer)
oligoGT616	TCCAGTCAGGTCGCA	spacer 8 epsilon (Rv primer)
oligoGT617	CAGAGGTGTCTCGCA	spacer 9 alpha (Fw primer)
oligoGT618	TGGTGCGAGACACCT	spacer 9 beta (Rv primer)
oligoGT619	TAAAGGTGTCTCGCA	spacer 9 delta (Fw primer)
oligoGT620	TCCTGCGAGACACCT	spacer 9 epsilon (Rv primer)
oligoGT621	CAGGCTACAGGCTGC	spacer 10 alpha (Fw primer)
oligoGT622	TGGGCAGCCTGTAGC	spacer 10 beta (Rv primer)
oligoGT623	TAAGCTACAGGCTGC	spacer 10 delta (Fw primer)
oligoGT624	TCCGCAGCCTGTAGC	spacer 10 epsilon (Rv primer)
oligoGT625	CAGTCAGACGGCACT	spacer 11 alpha (Fw primer)
oligoGT626	TGGAGTGCCGTCTGA	spacer 11 beta (Rv primer)
oligoGT627	TAATCAGACGGCACT	spacer 11 delta (Fw primer)
oligoGT628	TCCAGTGCCGTCTGA	spacer 11 epsilon (Rv primer)
oligoGT629	CAGATCGCAACTGGC	spacer 12 alpha (Fw primer)
oligoGT630	TGGGCCAGTTGCGAT	spacer 12 beta (Rv primer)
oligoGT631	TAAATCGCAACTGGC	spacer 12 delta (Fw primer)
oligoGT632	TCCGCCAGTTGCGAT	spacer 12 epsilon (Rv primer)
oligoGT633	CAGGGCAATCGTGCT	spacer 13 alpha (Fw primer)
oligoGT634	TGGAGCACGATTGCC	spacer 13 beta (Rv primer)
oligoGT635	TAAGGCAATCGTGCT	spacer 13 delta (Fw primer)
oligoGT636	TCCAGCACGATTGCC	spacer 13 epsilon (Rv primer)
oligoGT637	CAGATTGCCTGCGTC	spacer 14 alpha (Fw primer)
oligoGT638	TGGGACGCAGGCAAT	spacer 14 beta (Rv primer)
oligoGT639	TTAATTGCCTGCGTC	spacer 14 delta (Fw primer)

oligoGT640	TCCGACGCAGGCAAT	spacer 14 epsilon (Rv primer)
oligoGT641	CAGGCACCAATCGCT	spacer 15 alpha (Fw primer)
oligoGT642	TGGAGCGATTGGTGC	spacer 15 beta (Rv primer)
oligoGT643	TAAGCACCAATCGCT	spacer 15 delta (Fw primer)
oligoGT644	TCCAGCGATTGGTGC	spacer 15 epsilon (Rv primer)
oligoGT645	CAGAGCAATCCACGC	spacer 16 alpha (Fw primer)
oligoGT646	TGGGCGTGGATTGCT	spacer 16 beta (Rv primer)
oligoGT647	TAAAGCAATCCACGC	spacer 16 delta (Fw primer)
oligoGT648	TCCGCGTGGATTGCT	spacer 16 epsilon (Rv primer)
oligoGT649	TAATGCGACCTGACT	spacer 8 delta (Fw primer)
oligoGT653	GATCTGGCATGGGTGGAG	Rv primer to check for presence of lycopene pathway in *ssl0410* locus
oligoGT654	ACCCTGAATGATGCTCTCCA	Fw primer to check for presence of lycopene pathway in *ssl0410* locus
oligoGT657	ATACTAGatgcgtaaaggcgaagagctg	Fw primer for constructing pGT550
oligoGT658	AAAGTCCatgcgtaaaggcgaagagctg	Fw primer for constructing pGT551
oligoGT659	AAAAAAAatgcgtaaaggcgaagagctg	Fw primer for constructing pGT552
oligoGT660	/5Phos/acctcctaaaagttaaacaaaattatttctagaggg	Fw primer for constructing pGT550 – 552
oligoCK102	gctcagtcgaaagactgggcctttcgc	Fw sequencing primer for pCK302 and derivative plasmids
oligoCK332	ttcatctttccctggttgccaatgg	Fw sequencing primer for pCK302 and derivative plasmids
oligoAH34	CAGGAAATATTGCTTTGCAGG	Fw sequencing primer for pATM2 and all derivative plasmids
oligoAH35	ACACCCCTTGTATTACTGTTTATGT	Rv sequencing primer for pATM2 and all derivative plasmids
oligoAH48	ACTTTGGTTGCTACCTCGCT	Fw primer to check for integration into *ssl0410* locus
oligoAH49	AGGGAAAATGCCAAAACGCA	Rv primer to check for integration into *ssl0410* locus

Table A5. Table of synthetic DNA sequences used in the study

Synthetic DNA sequence name	Sequence
crtI recoded	atgacatcagctctcccccgcccggcaccaagtccgtacgcacgccgtaaa acggcgttggttattggcgcaggttttggtgggttggccctgggcattcgt ctacagtcgttaggttttgatacaacaattttggaacgtctggatggtcct ggtggtcgcgcgtatcaaaaacgtaccccagatggctatgtctttgacatg ggtccgactgtgctgacggtgccgcattttatcgaagaactgtttgcgctt gaacgtgatcgtgccggcctggatgcccccgattatcctcctgaagtgttg tcgggcgagcgcgttaaggaaggcgtttctggtggcccgcatacgagccgg tatgtcaccttagtgccgattctgccctttttaccgcattgtttttcacgat ggcacgtattttgattatgatggcgaccctgaaagtactcggcggcagatt gctgaattggcccctggcgacttagccgggtatgaacgctttcatgccgat gccgaggccatctttcgtcggggcttcctggaactgggctacacgcacttt ggtgacgtcccgacgatgctgcgggttgtacctgatctgctcaaactggac gccgttcggaccctgttctcctttacgagtaagtactttcagagcgacaaa ctgcgccaagtgttctcttttgaaaccctttttggtgggtgggaatcctctg agtgtgccggcgatctatgcaatgattcacttcgttgaaaagacttggggg atccactatgccatgggcggcacaggcgcactggtgcgcggcctagtccaa aaatttgaggagctgggtggcgccattcgttatggcgccggcgtcgatgaa gtactggtggatggcaatctgcctggtaaacgcacagcgcggggtgtgcgc ctggaaagcggcgaagaactgcgcgccgacctggtggcgtccaatggcgat tgggctaacacgtatctgaaacgcgtccggccatcggcacgtctggtcaac tctgatttacgcgtgaaagccgcatctgaaagtatgagcctgctcgtggtt tatttcgggtttcgcggcggtgatgacctgcccctcaaacatcataatatt ttattaggcccacgctacgaggctctgctgagcgaaatctttggcacaaaa cggttgggcgaagattttagccagtacctgcacgtcccaacgctcaccgat ccggctctggcacccgcgggtcatcatgcggcctatacacttgtcccggtg ccgcataatggctcgggtattgattgggacgtggaaggtccaaagcttgcc gaagcagccctggcagatatcgagcgccgcggtttgattccgggcctccgt gaacggctcacacattttgaatttattacgccagattatttcgcaggcact ctcgattcctatctggggaacgcgtttggtccggagccgcgtctggtccag tcggcatttttccgcccgcacaaccgcagcgaggatctccacaacttttac ttagtcggggcgggcgcgcagccaggcgcaggcacaccgagcgttatgatg tccgcgaaaatgacagcgcgcctaatcgctgaagatttcggtatccatgct gatatccggcgctaa
crtB recoded	atgcgtagtcgcgctggtctgagcttacggttacccacgcgtaccttgacc gtgaccgattactcccccgccctgccctgcaccgaactgcgccgtcctcca ctggctcaggcggttcgctactgtcgggatttgacccgccagcactcaaag accttctatctgggttcacagctcttttcgcctccggaacgcgcggcagtt tgggcagtgtatgcggcgtgccgcgcgggcgatgacatcgtcgatgaagcc ggcaacggcgaccgcgaacgcgaattgcgggaatggcgcagccgtattgat gccgcgtttgctggccaaccagcggatgatcccatctcaaccgcgctggcc tgggcggcaggtcggtacgccatcccgcactcagctttcgcggaactgcat gaaggcctcaacatggatttacgcggtcatgaataccgtgatatggatgac ttgttactgtattgccgccgtgtggcaggtgtggttggctttatggtggca ccgatttctggctaccgtgggggggctgctaccctgaatgatgctctccaa ctagggcaggcgatgcaactgacgaatattctgcgcgatgtcggtgaagat ctgacccgcggccgcgtatacctgccacagtctctgcttgatgaatatggc ctgtctcgcgccgcgttagagcgctggggtcagggtgagcccctgtcaccg gcctaccgtgctctcatgactcatcttggcggccttgcacgtgaatggtat

gcagcaggtcgtgctggtattcctcaacttgatggacgcggtcctctcgcc
gttctgactgccgcccgtgcgtatgagggtattctggacgatttggaacgg
gccggctacgacaacttcggtcggcgcgcgtacgtgtcaggtcgtcgtaaa
cttctgatgttaccgcaggcctggtgggaactgcgtagtctgggcgctgtc
cacggctaa

crtE recoded
atgcgcccggaattactcgcacgcgtgttaagcctgttaccggaaacctcc
gcgacgccggaattggcacgcttttacgcgctcctgcgcgactatcctcaa
cgtggtggcaagggcattcggtcagaattactgcttgcctctgctcgtgcg
cacggcctgtccgagtcagataccggttgggagtcagcattatggctggcg
gcagccttagaactgtttcagaactgggtgctggtgcacgatgatattgaa
gatgattcggaagaacgccgtggtcgtccggccctgcaccacttgtgtggt
atgccggtcgctcttaacgtgggggacgcgctgcacgcttacatgtgggct
gctgttgggaaagccaatgttccgggagcgtttgaagagtttctgcagatg
gtgtaccgcacggcggaaggccagcatctggatctggcatgggtggagggt
cgtgaatggggcctgcgtcccgcggattatctccagatggttggcctgaaa
accgcacactacacggttatcgtgccgttacgtctgggggccctggcggca
ggcatggcaccgcaggacgcgttcaccccagcgggtctggcgctgggtacc
gcgtttcagattcgtgacgatgtcctcaatctggcaggtgatccggtgaag
tatggtaaagaaattggtggcgatctgttggaaggtaaacgtactctgatt
gtcctggactggttgactacggcgccggatgatcgcaaagccatcttcctg
gaccagatgcgtcggcaccgcgcagataaagaccctgcggtgatcgatgaa
attcaccgctggctgcttgaaagcggctctgtggaagcggcgcaggactac
gcgcaggcacaagccgcggaaggtctggacttgcttgaaaaagcattggca
gacgcgccggatgcccaggccgccgctgccttactcgcttctgttcgggaa
ctggccacccgcgaaaaataa

dxs recoded
atgtcttttgatattgcgaaatatccaaccctggccctagttgactcgact
caggaattacgcctgctgccgaaagagagccttccaaagctgtgcgatgag
ttacgccgctacctcctggattctgttagtcgtagctccggccacttcgca
tcgggactaggtaccgtcgaactgacggttgcactgcattatgtgtataac
accccgttcgatcaattaatttgggacgtcggtcatcaagcatatccgcat
aaaattctgaccggtcgtcgcgacaagatcggcacaattcgtcaaaaaggt
ggattgcatcctttcccgtggcgcggcgagtccgaatacgatgtgttgagc
gttggccattcgtcaacttctatcagcgctggtattggtattgcggtcgcc
gcggagaaagaaggcaaaaatcggcgtacggtttgcgtgatcggggatggt
gcaatcaccgcaggcatggcattcgaagcgatgaaccatgcgggggacatt
cgtccggatatgctagtgattctgaacgataacgaaatgagtatttccgag
aacgtgggggctcttaataaccacttagcgcagctgctgagcggtaaactt
tattctagcctgcgcgaaggcggtaagaaagtgttctcaggcgtccctccg
attaaagagctccttaagcgtactgaggaacacattaagggtatggtggtt
ccaggcaccctgttcgaagaactgggttttaattacattggtcctgtggac
ggccatgacgtgttaggcttaattaccacgttaaaaaacatgcgcgatctg
aaaggacctcagtttctccacatcatgaccaagaaaggtcgcggctatgaa
ccggccgagaaagatccgattaccttccacgcagttccgaaattcgaccca
tcctccggctgtctgccgaaaagcagcggaggcctgccgagctattccaaa
atctttggtgactggctgtgcgaaactgcggcaaaagataacaaactgatg
gccatcaccccggcaatgcgcgaaggttcgggaatggtggaattctctcgc
aagttcccagaccgttattttgacgttgctatcgccgagcaacatgcagta
acctttgcggctggtctggcaattggtggttataagccaattgtggccatc
tattcaacgttcttacaacgtgcatatgaccaggtgctacatgatgtggcc
attcagaaactgccggtcttatttgccatcgatcgtgcggggatcgttggg
gcagatggtcagacgcatcagggcgccttcgacctaagttatctgcgctgt
atcccagaaatggtgattatgacccctagcgatgagaatgaatgtcgtcag
atgttatacaccggctatcattacaacgacggaccatccgctgtgcgctac

ccgcgcggaaacgcggtgggcgttgaactgaccccgttagaaaaattaccg
atcggaaaagggattgttaaacgtcgcggagaaaaactcgctattctgaac
ttcggcacacttatgcccgaagcggctaaagtagcggaaagcctaaacgca
accttggtggatatgcgcttcgtaaagccactggacgaagccctaatctta
gagatggccgcgtcacatgaggcgttagttacggtagaggaaaacgctatt
atgggcggagcgggtagtggtgtgaacgaagttctgatggctcaccgcaaa
ccggtgccggtactgaacatcggcctcccggatttttcattccccaaggt
actcaagaggaaatgcgcgccgagctcggcttagatgctgcaggaatggaa
gccaaaattaaggcctggcttgcataa

crtW recoded

atggtccagtgccagccgtcgtcgcttcattctgaaaaactcgttcttctc
tcgagcactattcgcgatgataaaaacatcaacaagggcattttcatcgcc
tgctttatttgtttctctgggccatttctctgattcttctgctgtccatt
gatacttcaattattcataaatctctgctgggtatcgcaatgctttggcag
acatttttatataccggcctcttcattacagcgcacgatgctatgcatggc
gttgtttatccgaaaaacccgcgcattaataatttcattggaaaactgacg
ctgattctgtatggtctgttaccgtataaagatctgttaaaaaagcattgg
cttcatcatggtcatccgggtaccgatctggatccggattactacaatggc
catccccagaatttttcttatggtacttgcactttatgaaaagctattgg
cggtggacccaaattttcggccttgtgatgattttccacggactgaaaaat
ctggtacatattccggagaataatttgatcatcttttggatgattccgtcc
attttgtccagtgtgcaactcttttactttgggacattcttaccgcataaa
aaactggaaggcggctataccaatccccactgcgctcgctccattccgctc
ccgctgttctggagctttgtgacatgttatcacttcggctatcataaagaa
catcacgaatatcctcaactcccgtggtggaaactgcctgaagcacacaaa
atctcgctgtaa

crtZ recoded

atgctgtggatttggaatgctctgatcgtgtttgttaccgtcattggcatg
gaagttatcgcggctctggcccacaaatatattatgcatggttggggatgg
ggttggcatttgtcgcaccatgagccgcgcaagggagcgtttgaagtgaac
gatctgtatgcagtagtttttgcggcccttagcattcttctgatttatctg
ggctcgaccggcatgtggcccttgcaatggatcggtgcgggcatgacggct
tacggcttgctttatttttatggttcacgatggattagttcaccagcgctgg
ccattccgttatatcccacgcaaaggctatttaaaacgtctttacatggca
catcgtatgcatcatgcggttcgggggaaagaaggctgcgtgagctttggg
tttctctatgcgccgcctctttctaaactgcaagctaccctgcgtgaacgg
catggcgcgcgcgcaggcgcggctcgtgatgcgcagggtggcgaagatgaa
cccgcatctggcaaataa

lcyB recoded

atggatacgcttctgaaaacgcctaataaccttgagtttctgaatccccat
catggttttgcagtgaaagcgagcaccttccgttccgagaaacatcataat
ttcgggagtcgcaaattctgtgaaacacttggccgctcggtgtgtgtgaag
ggctcatcgtcggccttattggaacttgtgccggagactaaaaaagaaaac
ttggattttgaactgcccatgtatgacccttctaaaggagtggtggttgat
cttgccgtagtaggaggggcccggcgggcctggccgtagcccagcaggtg
agcgaagctggtctgagcgtctgctctattgacccgaatccaaaactgatc
tggccaaacaactacggcgtatgggtggatgagtttgaggcgatggatctg
ctggattgcttagatgcgacctggtcgggcgctgcagtgtatattgatgac
aatactgcaaaagaccttcaccgcccttacggtcgtgttaaccgtaaacag
ttaaaatccaaaatgatgcagaaatgtatcatgaacggagtcaaattccac
caggcaaaagttattaaagtgattcacgaagaaagcaaaagcatgctgatt
tgtaatgacggcatcaccattcaggcgactgtcgtattagatgcgacaggc
tttagccgctctcttgtacaatatgataagccatataacccaggctatcaa
gttgcatatggcatccttgcagaggtggaagaacacccatttgacgttaat
aaaatggtttttatggattggcgtgacagccacttaaagaacaataccgat

ctgaaagagcgtaatagtcgcattccgactttcctctatgccatgccgttt
agtagcaaccgcatttttctggaggaaacaagcctggtcgcccgtccgggc
cttcgcatcgatgacattcaggaacgcatggtggcccgcctgaaccatctg
ggcatcaaagtcaaaagtattgaagaagatgaacactgcctgattccgatg
ggcggtcccctgccggtgcttccgcaacgcgtcgtgggcattggcgggact
gcgggcatggttcacccttcgaccggatatatggtggcgcggacccttgct
gccgcccctgtggtggctaacgctattattcaatacctgggcagtgaacgt
agtcactccggtaatgaactgtcaactgcagtctggaaggatctgtggcca
attgagcgccgtcgccagcgtgaatttttttgcttcggtatggatattctg
ctgaaactggatctgccggctacccgtcgcttcttcgatgctttttcgat
cttgagccccgctattggcatggctttctgtccagccgtttattcctgccg
gaactcatcgttttcggtttgtccttgttttcacatgccagcaacacttct
cgttttgaaatcatgacgaaagggacggtacctctggttaatatgattaac
aacttgttgcaagataaagaataa

idi recoded

Atgctgcgctcgcttctgcgcggtttaactcacatcccgcgcgtcaactcg
gcccaacagccttcatgcgcacacgcgcgcttacaatttaagctgcgttcg
atgcagatgactctgatgcaaccgagtatctcagctaacctgtcgcgcgcc
gaagatcgtactgaccacatgcgcggtgcaagtacgtgggctggtggtcaa
tcccaagatgaactgatgttaaaggacgaatgcattttggttgatgttgaa
gataacattactggacatgcatcaaagttggaatgtcataaatttctgccg
caccaacccgcaggtttgttgcaccgcgccttcagtgtatttctgtttgat
gaccaaggtcgtctgttgcttcaacaacgggcacgctcgaaaattaccttt
ccgagtgtatggaccaatacttgttgcagtcatccattgcacggtcagacg
ccagatgaagtggatcaactgagccaggtcgcggacgggaccgtcccgggg
gctaaggcggccgccatccggaaactggaacatgaactgggtatcccggcc
catcaactgccagcttcagcgtttcgtttcctcacccgtttgcattactgt
gcggctgacgtgcagccggcggctacgcagtcagcactttgggggggaacat
gaaatggattatatcctgttcattcgtgcgaatgtaacactcgccccgaat
cctgacgaagtggacgaggtccgttatgtaacgcaggaagaactgcgccag
atgatgcaaccggataatggccttcagtggagtccatggttccgtattatt
gccgcccggtttcttgaacgttggtgggccgatctggacgcagcactgaac
acggataaacatgaagattggggtaccgtgcatcatattaatgaggcataa

eyfp

atggtgagcaagggcgaggagctgttcaccggggtggtgcccatcctggtc
gagctggacggcgacgtaaacggccacaagttcagcgtgtccggcgagggc
gagggcgatgccacctacggcaagctgaccctgaagttcatctgcaccacc
ggcaagctgcccgtgccctggcccaccctcgtgaccaccttcggctacggc
ctgcaatgcttcgcccgctaccccgaccacatgaagctgcacgacttcttc
aagtccgccatgcccgaaggctacgtccaggagcgcaccatcttcttcaag
gacgacggcaactacaagacccgcgccgaggtgaagttcgagggcgacacc
ctggtgaaccgcatcgagctgaagggcatcgacttcaaggaggacggcaac
atcctggggcacaagctggagtacaactacaacagccacaacgtctatatc
atggccgacaagcagaagaacggcatcaaggtgaacttcaagatccgccac
aacatcgaggacggcagcgtgcagctcgccgaccactaccagcagaacacc
cccatcggcgacggccccgtgctgctgcccgacaaccactacctgagctac
cagtccgccctgagcaaagaccccaacgagaagcgcgatcacatggtcctg
ctggagttcgtgaccgccgccgggatcactctcggcatggacgagctgtac
aagtaa

yetiYFP

atgacggcattgacggaaggtgcaaaactgtttgagaaagagatcccgtat
atcaccgaactggaaggcgacgtcgaaggtatgaaatttatcattaaaggc
gagggtaccggtgacgcgagcgttggtaagattgacgcgcagtttatctgc
accacgggtgacgtcccggtgccgtggagcacgttggtgacgacgctgact
tacggtgcccaatgtttcgcgaaatatccgcgtcacatcgcagacttcttc

aaatcctgtatgccggaaggttatgtgcaagagcgcactattacctttgag
ggtgacggtgtctttaagacccgtgccgaagtgaccttcgaaaacggtagc
gtttacaaccgtgtcaagctgaatggccagggtttcaaaaaggatggtcac
gttctgggtaagaatctggaattcaacttcaccccacactgcctgtacatc
tggggcgatcaagcgaatcatggtctgaaaagcgcatttaagatcatgcac
gagattaccggctccaaagaagatttcatcgtggctgatcacacccagatg
aataccccgattggcggtggccctgtgcacgttccggaatatcatcacatt
acctaccatgtcacgctgagcaaggatgttaccgaccatcgcgaccacctg
aacatcgtagaggttatcaaagcagtggacctggagacgtatcgctaa

eyfp recoded (dnaGT2)

Ttaaaggagactgcagagcgcggaccttgacactcctagtgacttacacta
taatagctggattagacgacgatccaaaggaggtcccagatggttagtaaa
ggtgaagaactgttcactggtgttgtgccgatcctggtggaactggatggc
gacgtaaacggtcataaattctctgttagcggtgaaggcgaaggcgatgcg
acttatggaaaactgaccctgaaattcatctgtacgacgggtaaactgccg
gtgccgtggccgacgctggtaaccacgtttggttacggcttgcagtgcttt
gcgcgttacccggatcacatgaaactgcacgacttcttcaagagcgcgatg
ccggaaggttatgtgcaggaacgtaccatctttttcaaagatgatggtaat
tacaaaacccgtgctgaggtcaaattcgaaggcgacactctggttaaccgt
attgaactgaaagggatcgattttaaggaggacggcaatatcctgggccat
aaactggagtataactacaacagtcataacgtctatatcatggctgacaaa
cagaaaaacggcatcaaagtaaacttcaaaatccgtcataacatcgaggac
ggtagcgttcagctggccgaccactaccagcagaacactcctatcggcgac
ggcccggttctcctgccggacaaccactacctgagctaccagagcgcactg
agcaaagacccgaatgaaaaacgcgaccacatggtgctcctggaattcgtt
accgctgcgggtatcaccttgggtatggacgagctgtacaaataaggatcc
aaagccacgt

ccdB (dnaGT6)

tctggatttgttcagaacgctcggttgccgccgggcgtttttattggtga
gaatccagtgtgagagaccaaactgcagttgacaacataaaaactttgtgt
tatacttgtaacgtaaggaggtaatgattcagtttaaggtttacacctata
aaagagagagccgttatcgtctgtttgtggatgtacagagtgatattattg
acacgcccgggcgacggatggtgatccccctggccagtgcacgtctgctgt
cagataaagtctcccgtgaactttacccggtggtgcatatcggggatgaaa
gctggcgcatgatgaccaccgatatggccagtgtgccggtgtccgttatcg
gggaagaagtggctgatctcagccaccgcgaaaatgacatcaaaaacgcca
ttaacctgatgttctggggaatataaggtctcagacctctagggcggcgga
tttgtcctactcaggagagcgttcaccgacaaacaacagat

Table A6. Alignment of the five strongest promoter clones against non-functional clones that differed by a single nucleotide from Promoter Library 2 against. The sequence of the five strongest promoters from Promoter Library 2 and the parental promoter SPLc192 from the original SPL library are shown in **bold**. The non-functional promoter clone that differs by a single nucleotide (shown in **red**) is shown directly below the promoter. The -35 and -10 consensus sequences are highlighted in yellow.

Promoter Library 2 Promoter clone	Sequence of -35 region	Sequence of -10 region
SPLc192 (Parental promoter)	**TTGACA**	**TATAAT**
Promoter Library 2 c47	TTGACA	TATA**T**T
Promoter Library 2 c11	**TTGACG**	**TACAAT**
Promoter Library 2 c46	TTGACG	TAC**T**AT
Promoter Library 2 c28	**TTGACA**	**TACAAT**
Promoter Library 2 c5	TTGA**A**A	TACAAT
Promoter Library 2 c12	TTGACA	TAC**C**AT
Promoter Library 2 c15	TTGACA	TACA**C**T
Promoter Library 2 c32	**TTGCAA**	**TATAAT**
Promoter Library 2 c9	TTGCAA	TA**A**AAT
Promoter Library 2 c19	TTGCAA	TAT**G**AT
Promoter Library 2 c63	**TTGACC**	**TATAAT**
Promoter Library 2 c3	TTGACC	TATA**C**T
Promoter Library 2 c68	**TTGCAT**	**TATAAT**
Promoter Library 2 c8	TTGCAT	TA**C**AAT

Table A7. Spacers implemented as double-stranded linkers. Spacers can be used in either α-β format in place of a promoter, or δ-ε format in place of a terminator. Here each of the 16 spacers are shown as double-stranded linkers in both the α-β and δ-ε configurations. Fusion site cohesive ends are shown in **bold**. Spacers are obtained as two single-stranded oligonucleotides and then mixed and annealed together (described in section **2.4.8**) to generate the spacer as a linker part that can be used directly in Level 1 assembly reactions.

Spacer	Spacer sequence in double-stranded linker	Forward oligonucleotide	Reverse Oligonucleotide
Spacer 1 α-β format	5′-**CAG**TGGTCAGCGACT-3′ 3′-ACCAGTCGCTGA**GGT**-5′	oligoGT538: CAGTGGTCAGCGACT	oligoGT539: TGGAGTCGCTGACCA
Spacer 1 δ-ε format	5′-**TAA**TGGTCAGCGACT-3′ 3′-ACCAGTCGCTGA**CCT**-5′	oligoGT540: TAATGGTCAGCGACT	oligoGT541: TCCAGTCGCTGACCA
Spacer 2 α-β format	5′-**CAG**GCTGCCGTGAAT-3′ 3′-CGACGGCACTTA**GGT**-5′	oligoGT542: CAGGCTGCCGTGAAT	oligoGT543: TGGATTCACGGCAGC

Spacer 2	5′-**TAA**GCTGCCGTGAAT-3′
δ-ε	||||||||||||
format	3′-CGACGGCACTTA**CCT**-5′
Spacer 3	5′-**CAG**GGCACGCTCAAT-3′
α-β	||||||||||||
format	3′-CCGTGCGAGTTA**GGT**-5′
Spacer 3	5′-**TAA**GGCACGCTCAAT-3′
δ-ε	||||||||||||
format	3′-CCGTGCGAGTTA**CCT**-5′
Spacer 4	5′-**CAG**AGTCCGTGCTCA-3′
α-β	||||||||||||
format	3′-TCAGGCACGAGT**GGT**-5′
Spacer 4	5′-**TAA**AGTCCGTGCTCA-3′
δ-ε	||||||||||||
format	3′-TCAGGCACGAGT**CCT**-5′
Spacer 5	5′-**CAG**ATTCTGTGCCGC-3′
α-β	||||||||||||
format	3′-TAAGACACGGCG**GGT**-5′
Spacer 5	5′-**TAA**ATTCTGTGCCGC-3′
δ-ε	||||||||||||
format	3′-TAAGACACGGCG**CCT**-5′
Spacer 6	5′-**CAG**ATCAACGCCTGC-3′
α-β	||||||||||||
format	3′-TAGTTGCGGACG**GGT**-5′
Spacer 6	5′-**TAA**ATCAACGCCTGC-3′
δ-ε	||||||||||||
format	3′-TAGTTGCGGACG**CCT**-5′
Spacer 7	5′-**CAG**ATCTGCGGCAAC-3′
α-β	||||||||||||
format	3′-TAGACGCCGTTG**GGT**-5′
Spacer 7	5′-**TAA**ATCTGCGGCAAC-3′
δ-ε	||||||||||||
format	3′-TAGACGCCGTTG**CCT**-5′
Spacer 8	5′-**CAG**TGCGACCTGACT-3′
α-β	||||||||||||
format	3′-ACGCTGGACTGA**GGT**-5′
Spacer 8	5′-**TAA**TGCGACCTGACT-3′
δ-ε	||||||||||||
format	3′-ACGCTGGACTGA**CCT**-5′
Spacer 9	5′-**CAG**AGGTGTCTCGCA-3′
α-β	||||||||||||
format	3′-TCCACAGAGCGT**GGT**-5′
Spacer 9	5′-**TAA**AGGTGTCTCGCA-3′
δ-ε	||||||||||||
format	3′-TCCACAGAGCGT**CCT**-5′
Spacer 10 α-β	5′-**CAG**GCTACAGGCTGC-3′
	||||||||||||
format	3′-CGATGTCCGACG**GGT**-5′
Spacer 10 δ-ε	5′-**TAA**GCTACAGGCTGC-3′
	||||||||||||
format	3′-CGATGTCCGACG**CCT**-5′

oligoGT544:	oligoGT545:
TAAGCTGCCGTGAAT	TCCATTCACGGCAGC
oligoGT546:	oligoGT547:
CAGGGCACGCTCAAT	TGGATTGAGCGTGCC
oligoGT548:	oligoGT549:
TAAGGCACGCTCAAT	TCCATTGAGCGTGCC
oligoGT550:	oligoGT551:
CAGAGTCCGTGCTCA	TGGTGAGCACGGACT
oligoGT552:	oligoGT553:
TAAAGTCCGTGCTCA	TCCTGAGCACGGACT
oligoGT554:	oligoGT555:
CAGATTCTGTGCCGC	TGGGCGGCACAGAAT
oligoGT556:	oligoGT557:
TAAATTCTGTGCCGC	TCCGCGGCACAGAAT
oligoGT558:	oligoGT559:
CAGATCAACGCCTGC	TGGGCAGGCGTTGAT
oligoGT560:	oligoGT561:
TAAATCAACGCCTGC	TCCGCAGGCGTTGAT
oligoGT562:	oligoGT563:
CAGATCTGCGGCAAC	TGGGTTGCCGCAGAT
oligoGT564:	oligoGT565:
TAAATCTGCGGCAAC	TCCGTTGCCGCAGAT
oligoGT566:	oligoGT567:
CAGTGCGACCTGACT	TGGAGTCAGGTCGCA
oligoGT616:	oligoGT617:
TAATGCGACCTGACT	TCCAGTCAGGTCGCA
oligoGT618:	oligoGT619:
CAGAGGTGTCTCGCA	TGGTGCGAGACACCT
oligoGT620:	oligoGT621:
TAAAGGTGTCTCGCA	TCCTGCGAGACACCT
oligoGT622:	oligoGT623:
CAGGCTACAGGCTGC	TGGGCAGCCTGTAGC
oligoGT624:	oligoGT625:
TAAGCTACAGGCTGC	TCCGCAGCCTGTAGC

Spacer 11 α-β format	5′-**CAG**TCAGACGGCACT-3′ |||||||||||| 3′-AGTCTGCCGTGA**GGT**-5′	oligoGT626: CAGTCAGACGGCACT	oligoGT627: TGGAGTGCCGTCTGA
Spacer 11 δ-ε format	5′-**TAA**TCAGACGGCACT-3′ |||||||||||| 3′-AGTCTGCCGTGA**CCT**-5′	oligoGT628: TAATCAGACGGCACT	oligoGT629: TCCAGTGCCGTCTGA
Spacer 12 α-β format	5′-**CAG**ATCGCAACTGGC-3′ |||||||||||| 3′-TAGCGTTGACCG**GGT**-5′	oligoGT630: CAGATCGCAACTGGC	oligoGT631: TGGGCCAGTTGCGAT
Spacer 12 δ-ε format	5′-**TAA**ATCGCAACTGGC-3′ |||||||||||| 3′-TAGCGTTGACCG**CCT**-5′	oligoGT632: TAAATCGCAACTGGC	oligoGT633: TCCGCCAGTTGCGAT
Spacer 13 α-β format	5′-**CAG**GGCAATCGTGCT-3′ |||||||||||| 3′-CCGTTAGCACGA**GGT**-5′	oligoGT634: CAGGGCAATCGTGCT	oligoGT635: TGGAGCACGATTGCC
Spacer 13 δ-ε format	5′-**TAA**GGCAATCGTGCT-3′ |||||||||||| 3′-CCGTTAGCACGA**CCT**-5′	oligoGT636: TAAGGCAATCGTGCT	oligoGT637: TCCAGCACGATTGCC
Spacer 14 α-β format	5′-**CAG**ATTGCCTGCGTC-3′ |||||||||||| 3′-TAACGGACGCAG**GGT**-5′	oligoGT638: CAGATTGCCTGCGTC	oligoGT639: TGGGACGCAGGCAAT
Spacer 14 δ-ε format	5′-**TAA**ATTGCCTGCGTC-3′ |||||||||||| 3′-TAACGGACGCAG**CCT**-5′	oligoGT640: TAAATTGCCTGCGTC	oligoGT641: TCCGACGCAGGCAAT
Spacer 15 α-β format	5′-**CAG**GCACCAATCGCT-3′ |||||||||||| 3′-CGTGGTTAGCGA**GGT**-5′	oligoGT642: CAGGCACCAATCGCT	oligoGT643: TGGAGCGATTGGTGC
Spacer 15 δ-ε format	5′-**TAA**GCACCAATCGCT-3′ |||||||||||| 3′-CGTGGTTAGCGA**CCT**-5′	oligoGT644: TAAGCACCAATCGCT	oligoGT645: TCCAGCGATTGGTGC
Spacer 16 α-β format	5′-**CAG**AGCAATCCACGC-3′ |||||||||||| 3′-TCGTTAGGTGCG**GGT**-5′	oligoGT646: CAGAGCAATCCACGC	oligoGT647: TGGGCGTGGATTGCT
Spacer 16 δ-ε format	5′-**TAA**AGCAATCCACGC-3′ |||||||||||| 3′-TCGTTAGGTGCG**CCT**-5′	oligoGT648: TAAAGCAATCCACGC	oligoGT649: TCCGCGTGGATTGCT

Table A8. Table of plasmids used in this study.

Plasmid	Relevant properties or Description	Source
pUC19	Traditional cloning plasmid	Heap lab
pACYC177	Traditional cloning plasmid	Heap lab
pACYC184	Traditional cloning plasmid	Heap lab
pCK302	P_{rhaBAD} reporter plasmid	(123)
pPM6	Source of *gusA* CDS	(408)
pATM2	pSHUTTLE2 with EYFP reporter	(124)
pATM3	pSHUTTLE2 with YetiYFP reporter	Heap lab
pStA0	Start-Stop Assembly empty storage vector	This project

pStA1AZ	Start-Stop Assembly Level 1 vector (A and Z fusion sites)	This project
pStA1AB	Start-Stop Assembly Level 1 vector (A and B fusion sites)	This project
pStA1BZ	Start-Stop Assembly Level 1 vector (B and Z fusion sites)	This project
pStA1BC	Start-Stop Assembly Level 1 vector (B and C fusion sites)	This project
pStA1CZ	Start-Stop Assembly Level 1 vector (C and Z fusion sites)	This project
pStA1CD	Start-Stop Assembly Level 1 vector (C and D fusion sites)	This project
pStA1DZ	Start-Stop Assembly Level 1 vector (D and Z fusion sites)	This project
pStA1DE	Start-Stop Assembly Level 1 vector (D and E fusion sites)	This project
pStA1DZ	Start-Stop Assembly Level 1 vector (D and Z fusion sites)	This project
pStA1EZ	Start-Stop Assembly Level 1 vector (E and Z fusion sites)	This project
pStA212	Start-Stop Assembly Level 2 vector (1 and 2 fusion sites)	This project
pStA223	Start-Stop Assembly Level 2 vector (2 and 3 fusion sites)	This project
pStA234	Start-Stop Assembly Level 2 vector (3 and 4 fusion sites)	This project
pStA313	Start-Stop Assembly Level 3 vector (1 and 3 fusion sites)	This project
pStA314	Start-Stop Assembly Level 3 vector (1 and 4 fusion sites)	This project
pGT11	pSHUTTLE2 with recoded EYFP (version 3) reporter	This project
pGT12	pSHUTTLE2 with recoded EYFP (version 2) reporter	This project
pGT13	pSHUTTLE2 with GusA reporter	This project
pGT16	pSHUTTLE2 with RBSc249::SPLc56::EYFP reporter	This project
pGT17	pSHUTTLE2 with RBSc249::SPLc69::EYFP reporter	This project
pGT18	pSHUTTLE2 with RBSc249::SPLc46::EYFP reporter	This project
pGT19	pSHUTTLE2 with RBSc249::SPLc28::EYFP reporter	This project
pGT20	pSHUTTLE2 with RBSc249::SPLc107::EYFP reporter	This project
pGT21	pSHUTTLE2 with RBSc249::SPLc157::EYFP reporter	This project
pGT22	pSHUTTLE2 with RBSc197::SPLc56::EYFP reporter	This project
pGT23	pSHUTTLE2 with RBSc197::SPLc69::EYFP reporter	This project
pGT24	pSHUTTLE2 with RBSc197::SPLc46::EYFP reporter	This project
pGT25	pSHUTTLE2 with RBSc197::SPLc28::EYFP reporter	This project
pGT26	pSHUTTLE2 with RBSc197::SPLc107::EYFP reporter	This project
pGT27	pSHUTTLE2 with RBSc197::SPLc157::EYFP reporter	This project
pGT28	pSHUTTLE2 with RBSc52::SPLc56::EYFP reporter	This project
pGT29	pSHUTTLE2 with RBSc52::SPLc69::EYFP reporter	This project
pGT30	pSHUTTLE2 with RBSc52::SPLc46::EYFP reporter	This project
pGT31	pSHUTTLE2 with RBSc52::SPLc28::EYFP reporter	This project
pGT32	pSHUTTLE2 with RBSc52::SPLc107::EYFP reporter	This project
pGT33	pSHUTTLE2 with RBSc52::SPLc157::EYFP reporter	This project

pGT34	pSHUTTLE2 with RBSc266::SPLc56::EYFP reporter	This project
pGT35	pSHUTTLE2 with RBSc266::SPLc69::EYFP reporter	This project
pGT36	pSHUTTLE2 with RBSc266::SPLc46::EYFP reporter	This project
pGT37	pSHUTTLE2 with RBSc266::SPLc28::EYFP reporter	This project
pGT38	pSHUTTLE2 with RBSc266::SPLc107::EYFP reporter	This project
pGT39	pSHUTTLE2 with RBSc266::SPLc157::EYFP reporter	This project
pGT40	pSHUTTLE2 with RBSc102::SPLc56::EYFP reporter	This project
pGT41	pSHUTTLE2 with RBSc102::SPLc69::EYFP reporter	This project
pGT42	pSHUTTLE2 with RBSc102::SPLc46::EYFP reporter	This project
pGT43	pSHUTTLE2 with RBSc102::SPLc28::EYFP reporter	This project
pGT44	pSHUTTLE2 with RBSc102::SPLc107::EYFP reporter	This project
pGT45	pSHUTTLE2 with RBSc102::SPLc157::EYFP reporter	This project
pGT46	pSHUTTLE2 with RBSc257::SPLc56::EYFP reporter	This project
pGT47	pSHUTTLE2 with RBSc257::SPLc69::EYFP reporter	This project
pGT48	pSHUTTLE2 with RBSc257::SPLc46::EYFP reporter	This project
pGT49	pSHUTTLE2 with RBSc257::SPLc28::EYFP reporter	This project
pGT50	pSHUTTLE2 with RBSc257::SPLc107::EYFP reporter	This project
pGT51	pSHUTTLE2 with RBSc257::SPLc157::EYFP reporter	This project
pGT52	pSHUTTLE2 with SPLc46::RBSc249::EYFPv3	This project
pGT53	pSHUTTLE2 with SPLc46::RBSc197::EYFPv3	This project
pGT54	pSHUTTLE2 with SPLc46::RBSc266::EYFPv3	This project
pGT55	pSHUTTLE2 with SPLc46::RBSc102::EYFPv3	This project
pGT56	pSHUTTLE2 with SPLc46::RBSc257::EYFPv3	This project
pGT57	pSHUTTLE2 with SPLc46::RBSc249::EYFPv2	This project
pGT58	pSHUTTLE2 with SPLc46::RBSc197::EYFPv2	This project
pGT59	pSHUTTLE2 with SPLc46::RBSc266::EYFPv2	This project
pGT60	pSHUTTLE2 with SPLc46::RBSc102::EYFPv2	This project
pGT61	pSHUTTLE2 with SPLc46::RBSc257::EYFPv2	This project
pGT62	pSHUTTLE2 with SPLc46::RBSc52::YetiYFP	This project
pGT63	pSHUTTLE2 with SPLc46::RBSc249::YetiYFP	This project
pGT64	pSHUTTLE2 with SPLc46::RBSc197::YetiYFP	This project
pGT65	pSHUTTLE2 with SPLc46::RBSc266::YetiYFP	This project
pGT66	pSHUTTLE2 with SPLc46::RBSc102::YetiYFP	This project
pGT67	pSHUTTLE2 with SPLc46::RBSc257::YetiYFP	This project
pGT68	pSHUTTLE2 with SPLc46::RBSc249::GusA	This project
pGT69	pSHUTTLE2 with SPLc46::RBSc197::GusA	This project

pGT70	pSHUTTLE2 with SPLc46::RBSc266::GusA	This project
pGT71	pSHUTTLE2 with SPLc46::RBSc102::GusA	This project
pGT72	pSHUTTLE2 with SPLc46::RBSc257::GusA	This project
pGT270	pSHUTTLE2 with pStA212 Start-Stop Assembly cassette	This project
pGT274	pCK302 reporter plasmid with no scar	This project
pGT287	pStA0::Conventional_Composite_Part1	This project
pGT288	pStA0::Conventional_Composite_Part2	This project
pGT289	pStA0::Conventional_Composite_Part3	This project
pGT290	pStA0::Conventional_Composite_Part4	This project
pGT291	pStA0::Conventional_Composite_Part5	This project
pGT292	pStA0::Conventional_Composite_Part6	This project
pGT293	pStA0::Conventional_Composite_Part7	This project
pGT294	pStA0::Conventional_Composite_Part8	This project
pGT295	pStA0::Conventional_Composite_Part9	This project
pGT296	pStA0::Conventional_Composite_Part10	This project
pGT297	pStA0::Conventional_Composite_Part11	This project
pGT298	pStA0::Conventional_Composite_Part12	This project
pGT299	pStA0::Conventional_Composite_Part13	This project
pGT300	pStA0::Conventional_Composite_Part14	This project
pGT301	pStA0::Conventional_Composite_Part15	This project
pGT302	pStA0::Conventional_Composite_Part16	This project
pGT303	pStA0::Conventional_Composite_Part17	This project
pGT304	pStA0::Conventional_Composite_Part18	This project
pGT305	pStA0::Conventional_Composite_Part19	This project
pGT306	pStA0::Conventional_Composite_Part20	This project
pGT307	pStA0::Conventional_Composite_Part21	This project
pGT308	pStA0::Conventional_Composite_Part22	This project
pGT309	pStA0::Conventional_Composite_Part23	This project
pGT310	pStA0::Conventional_Composite_Part24	This project
pGT311	pStA0::Conventional_Composite_Part25	This project
pGT312	pStA0::Conventional_Composite_Part26	This project
pGT313	pStA0::Conventional_Composite_Part27	This project
pGT314	pStA0::Conventional_Composite_Part28	This project
pGT315	pStA0::Conventional_Composite_Part29	This project
pGT316	pStA0::Conventional_Composite_Part30	This project
pGT317	pStA0::Conventional_Composite_Part31	This project

pGT318	pStA0::Conventional_Composite_Part32	This project
pGT319	pStA0::Conventional_Composite_Part33	This project
pGT320	pStA0::Conventional_Composite_Part34	This project
pGT321	pStA0::Conventional_Composite_Part35	This project
pGT322	pStA0::Conventional_Composite_Part36	This project
pGT323	pStA0::J23100	This project
pGT324	pStA0::J23102	This project
pGT326	pStA0::J23107	This project
pGT327	pStA0::J23116	This project
pGT328	pStA0::J23113	This project
pGT330	pStA0::RBSc13	This project
pGT331	pStA0::RBSc33	This project
pGT332	pStA0::RBSc44	This project
pGT333	pStA0::RBSc58	This project
pGT334	pStA0::RBSc36	This project
pGT335	pStA0::RBSc42	This project
pGT336	pStA0::J23118	This project
pGT337	pStA0::L3S2P55	This project
pGT338	pStA0::L3S2P21	This project
pGT339	pStA0:: ECK120033737	This project
pGT340	pStA0:: ECK120029600	This project
pGT356	pStA0::dnaGT7 (recoded *crtI*)	This project
pGT357	pStA0::dnaGT8 (recoded *crtB*)	This project
pGT358	pStA0::dnaGT9 (recoded *crtE*)	This project
pGT359	pStA0::dnaGT10 (recoded *dxs*)	This project
pGT360	pStA0::dnaGT11 (recoded *crtW*)	This project
pGT361	pStA0::dnaGT12 (recoded *crtZ*)	This project
pGT362	pStA0::dnaGT13 (recoded *lcyB*)	This project
pGT363	pStA0::dnaGT14 (recoded *idi*)	This project
pGT364	pStA0::Portable_Composite_Part1 (RBSc249+SPLc56)	This project
pGT365	pStA0::Portable_Composite_Part2 (RBSc249+SPLc69)	This project
pGT366	pStA0::Portable_Composite_Part3 (RBSc249+SPLc46)	This project
pGT367	pStA0::Portable_Composite_Part4 (RBSc249+SPLc28)	This project
pGT368	pStA0::Portable_Composite_Part5 (RBSc249+SPLc107)	This project
pGT369	pStA0::Portable_Composite_Part6 (RBSc249+SPLc157)	This project
pGT370	pStA0::Portable_Composite_Part7 (RBSc197+SPLc56)	This project

pGT371	pStA0::Portable_Composite_Part8 (RBSc197+SPLc69)	This project
pGT372	pStA0::Portable_Composite_Part9 (RBSc197+SPLc46)	This project
pGT373	pStA0::Portable_Composite_Part10 (RBSc197+SPLc28)	This project
pGT374	pStA0::Portable_Composite_Part11 (RBSc197+SPLc107)	This project
pGT375	pStA0::Portable_Composite_Part12 (RBSc197+SPLc157)	This project
pGT376	pStA0::Portable_Composite_Part13 (RBSc52+SPLc56)	This project
pGT377	pStA0::Portable_Composite_Part14 (RBSc52+SPLc69)	This project
pGT378	pStA0::Portable_Composite_Part15 (RBSc52+SPLc46)	This project
pGT379	pStA0::Portable_Composite_Part16 (RBSc52+SPLc28)	This project
pGT380	pStA0::Portable_Composite_Part17 (RBSc52+SPLc107)	This project
pGT381	pStA0::Portable_Composite_Part18 (RBSc52+SPLc157)	This project
pGT382	pStA0::Portable_Composite_Part19 (RBSc266+SPLc56)	This project
pGT383	pStA0::Portable_Composite_Part20 (RBSc266+SPLc69)	This project
pGT384	pStA0::Portable_Composite_Part21 (RBSc266+SPLc46)	This project
pGT385	pStA0::Portable_Composite_Part22 (RBSc266+SPLc28)	This project
pGT386	pStA0::Portable_Composite_Part23 (RBSc266+SPLc107)	This project
pGT387	pStA0::Portable_Composite_Part24 (RBSc266+SPLc157)	This project
pGT388	pStA0::Portable_Composite_Part25 (RBSc102+SPLc56)	This project
pGT389	pStA0::Portable_Composite_Part26 (RBSc102+SPLc69)	This project
pGT390	pStA0::Portable_Composite_Part27 (RBSc102+SPLc46)	This project
pGT391	pStA0::Portable_Composite_Part28 (RBSc102+SPLc28)	This project
pGT392	pStA0::Portable_Composite_Part29 (RBSc102+SPLc107)	This project
pGT393	pStA0::Portable_Composite_Part30 (RBSc102+SPLc175)	This project
pGT394	pStA0::Portable_Composite_Part31 (RBSc257+SPLc56)	This project
pGT395	pStA0::Portable_Composite_Part32 (RBSc257+SPLc69)	This project
pGT396	pStA0::Portable_Composite_Part33 (RBSc257+SPLc28)	This project
pGT397	pStA0::Portable_Composite_Part34 (RBSc257+SPLc69)	This project
pGT398	pStA0::Portable_Composite_Part35 (RBSc257+SPLc107)	This project
pGT399	pStA0::Portable_Composite_Part36 (RBSc257+SPLc157)	This project
pGT423	pStA0:: ECK120017009	This project
pGT424	pStA0:: ECK120015170	This project
pGT425	pStA0::Conventional_Composite_Part37	This project
pGT426	pStA0::Conventional_Composite_Part38	This project
pGT427	pStA0::Conventional_Composite_Part39	This project
pGT428	pStA0::Conventional_Composite_Part40	This project
pGT429	pStA0::Conventional_Composite_Part41	This project

pGT430	pStA0::Conventional_Composite_Part42	This project
pGT431	pStA0::*eyfp*	This project
pGT432	pStA212::Lycopene pathway strong promoter+RBS	This project
pGT433	pStA212::Lycopene pathway middle promoter+RBS	This project
pGT434	pStA212::Lycopene pathway burden design	This project
pGT435	pStA212::Lycopene pathway opposite to burden design	This project
pGT436	pStA1AZ::P+R+EYFP	This project
pGT437	Lycopene pathway clone blindly chosen from pGT547	This project
pGT438	Lycopene pathway clone blindly chosen from pGT547	This project
pGT439	Lycopene pathway clone blindly chosen from pGT547	This project
pGT440	Lycopene pathway clone blindly chosen from pGT547	This project
pGT441	Lycopene pathway clone blindly chosen from pGT547	This project
pGT442	Lycopene pathway clone blindly chosen from pGT547	This project
pGT443	Lycopene pathway clone blindly chosen from pGT547	This project
pGT444	Lycopene pathway clone blindly chosen from pGT547	This project
pGT445	Lycopene pathway clone blindly chosen from pGT547	This project
pGT446	Lycopene pathway clone blindly chosen from pGT547	This project
pGT447	Lycopene pathway clone blindly chosen from pGT547	This project
pGT448	Lycopene pathway clone blindly chosen from pGT547	This project
pGT449	Lycopene pathway clone blindly chosen from pGT547	This project
pGT450	Lycopene pathway clone blindly chosen from pGT547	This project
pGT451	Lycopene pathway clone blindly chosen from pGT547	This project
pGT452	Lycopene pathway clone blindly chosen from pGT547	This project
pGT453	Lycopene pathway clone blindly chosen from pGT547	This project
pGT454	Lycopene pathway clone blindly chosen from pGT547	This project
pGT455	Lycopene pathway clone blindly chosen from pGT547	This project
pGT456	Lycopene pathway clone blindly chosen from pGT547	This project
pGT457	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT458	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT459	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT460	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT461	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT462	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT463	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT464	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT465	Lycopene pathway clone from pGT547 with Prescreen	This project

pGT466	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT467	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT468	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT469	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT470	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT471	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT472	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT473	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT474	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT475	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT476	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT477	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT478	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT479	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT480	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT481	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT482	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT483	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT484	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT485	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT486	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT487	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT488	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT489	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT490	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT491	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT492	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT493	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT494	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT495	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT496	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT497	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT498	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT499	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT500	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT501	Lycopene pathway clone from pGT549 with Prescreen	This project

pGT502	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT503	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT504	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT505	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT506	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT507	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT508	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT509	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT510	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT511	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT512	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT513	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT514	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT515	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT516	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT517	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT518	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT519	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT520	Lycopene pathway clone from pGT547 with Prescreen	This project
pGT521	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT522	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT523	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT524	Lycopene pathway clone from pGT548 with Prescreen	This project
pGT525	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT526	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT527	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT528	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT529	Lycopene pathway clone from pGT549 with Prescreen	This project
pGT530	pStA1AZ::Combintorial EYFP	This project
pGT531	StA carotene pathway 1 (Monocistronic) in pStA212	This project
pGT532	StA carotene pathway 2 (operon) in pStA223	This project
pGT533	StA carotene pathway 3 (hybrid) in pStA234	This project
pGT534	StA carotene pathway 4 (hybrid) in pStA313	This project
pGT353	StA Astaxanthin pathway (hybrid) in pStA314	This project
pGT536	StA purified astaxanthin pathway clone from library pGT535	This project
pGT537	Astaxanthin pathway clone 1 from library pGT535	This project

pGT538	Astaxanthin pathway clone 2 from library pGT535	This project
pGT539	Astaxanthin pathway clone 3 from library pGT535	This project
pGT540	Astaxanthin pathway clone 4 from library pGT535	This project
pGT541	Astaxanthin pathway clone 5 from library pGT535	This project
pGT542	Astaxanthin pathway clone 6 from library pGT535	This project
pGT543	Astaxanthin pathway clone 7 from library pGT535	This project
pGT544	Astaxanthin pathway clone 8 from library pGT535	This project
pGT545	Astaxanthin pathway clone 9 from library pGT535	This project
pGT546	Astaxanthin pathway clone 10 from library pGT535	This project
pGT547	conventional subset lycopene combinatorial build	This project
pGT548	portable subset lycopene pathway combinatorial build	This project
pGT549	curated subset lycopene pathway combinatorial build	This project
pGT550	pCK302 reporter plasmid with BioBrick scar	This project
pGT551	pCK302 reporter plasmid with BASIC scar	This project
pGT552	pCK302 reporter plasmid with MoClo scar	This project

Appendix Figures

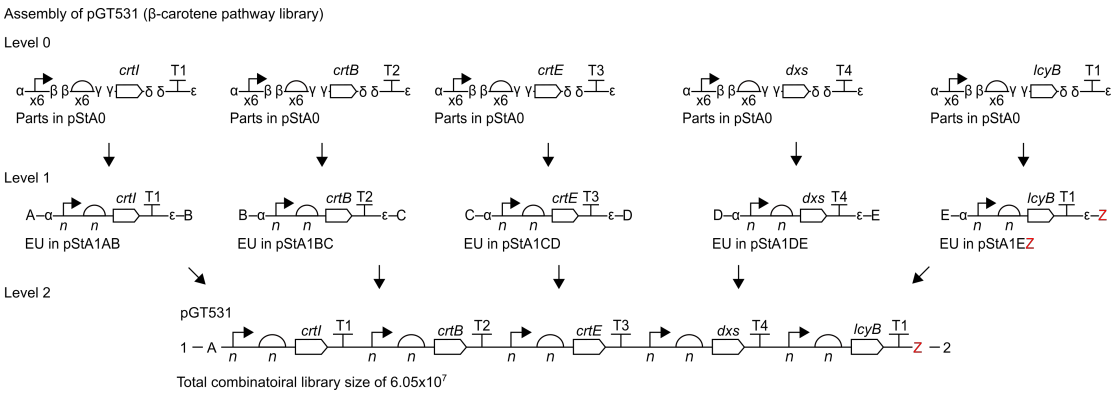


Figure A1. Combinatorial assembly of monocistronic β-carotene pathway library pGT531. A library of constructs each encoding a β-carotene pathway was assembled hierarchically and combinatorially as shown in the figure and as described in section **5.2.6**. The destination vector was pStA212. Equimolar mixtures of six promoter parts or six RBS parts (Figure 5.5) labelled as 'x6' were used in Level 1 assemblies. The uncertain representation of each promoter and RBS at each position following assembly is represented by '*n*'. The total combinatorial design space was 6^{10} = $6.05x10^7$. T1 = terminator L3S2P55, T2 = terminator L3S2P21, T3 = terminator ECK120033737, T4 = terminator ECK120029600.

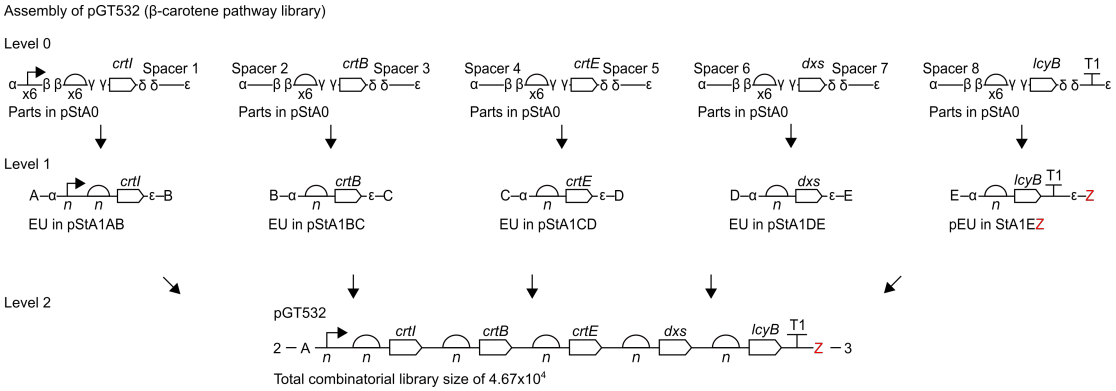


Figure A2. Combinatorial assembly of operon-based β-carotene pathway library pGT532. A library of constructs each encoding a β-carotene pathway was assembled hierarchically and combinatorially as shown in the figure and as described in section **5.2.6**. The destination vector was pStA223. Equimolar mixtures of six promoter parts or six RBS parts (Figure 5.5) labelled as 'x6' were used in Level 1 assemblies where shown. Spacer sequences were used in Level 1 assemblies (as linkers, Table A7) in place of promoters and/or terminators where shown. The uncertain representation of each promoter and RBS at each position following assembly is represented by '*n*'. The total combinatorial design space was 6^6 = $4.6x10^4$. T1 = terminator L3S2P55.

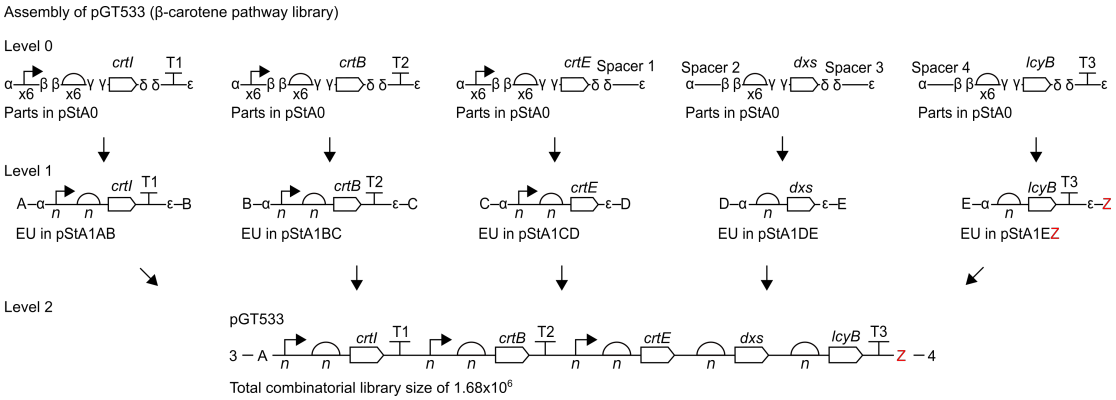


Figure A3. Combinatorial assembly of hybrid β-carotene pathway library pGT533 . A library of constructs each encoding a β-carotene pathway was assembled hierarchically and combinatorially as shown in the figure and as described in section **5.2.6**. The destination vector was pStA234. Equimolar mixtures of six promoter parts or six RBS parts (Figure 5.5) labelled as 'x6' were used in Level 1 assemblies where shown. Spacer sequences were used in Level 1 assemblies (as linkers, Table A7) in place of promoters and/or terminators where shown. The uncertain representation of each promoter and RBS at each position following assembly is represented by '*n*'. The total combinatorial design space was 6^8 = 1.68x10^6. T1 = terminator L3S2P55, T2 = terminator L3S2P21, T3 = terminator ECK120033737.

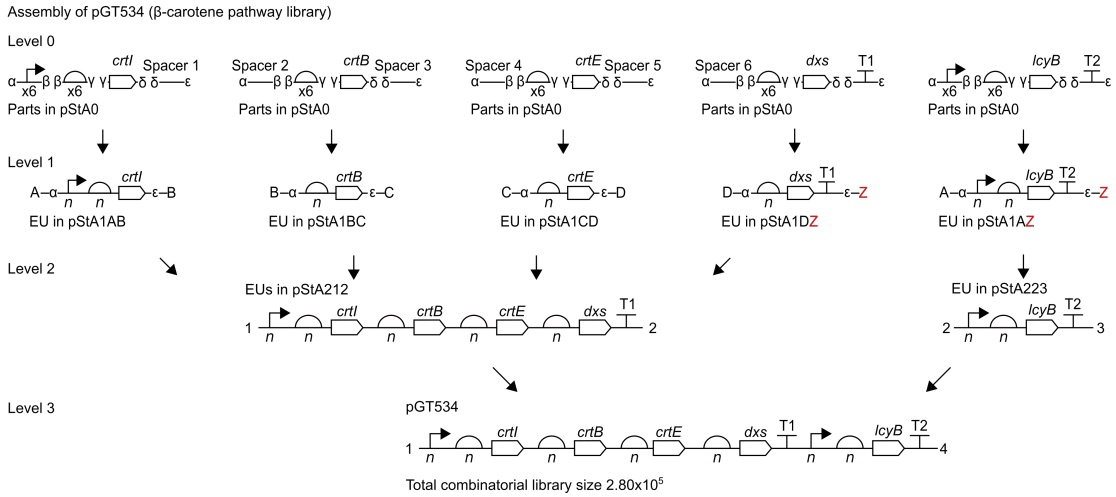


Figure A4. Combinatorial assembly of hybrid β-carotene pathway library pGT534 . A library of constructs each encoding a β-carotene pathway was assembled hierarchically and combinatorially as shown in the figure and as described in section **5.2.6**. The destination vector was pStA313. Equimolar mixtures of six promoter parts or six RBS parts (Figure 5.5) labelled as 'x6' were used in Level 1 assemblies where shown. Spacer sequences were used in Level 1 assemblies (as linkers, Table A7) in place of promoters and/or terminators where shown. The uncertain representation of each promoter and RBS at each position following assembly is represented by '*n*'. The total combinatorial design space was 6^7 = 2.80x10^5. T1 = terminator L3S2P55, T2 = terminator L3S2P21.

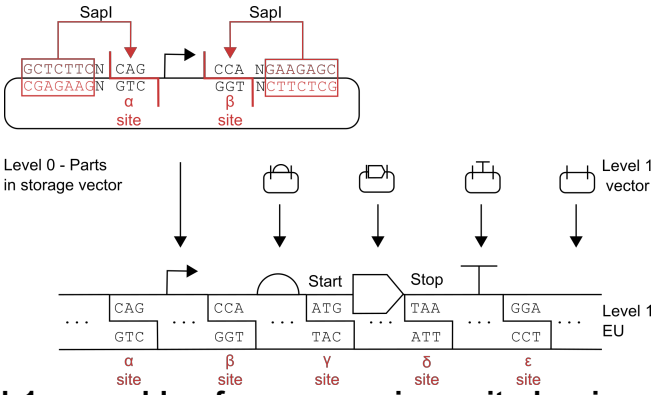


Figure A5. Level 1 assembly of an expression unit showing detail of Level 0 promoter part. Genetic parts stored in pStA0 are flanked by inward-facing SapI recognition sites (red boxes) with corresponding donor fusion sites (staggered red lines) for Level 1 assembly of expression units. Promoters use the α site (CAG) and β site (CCA).

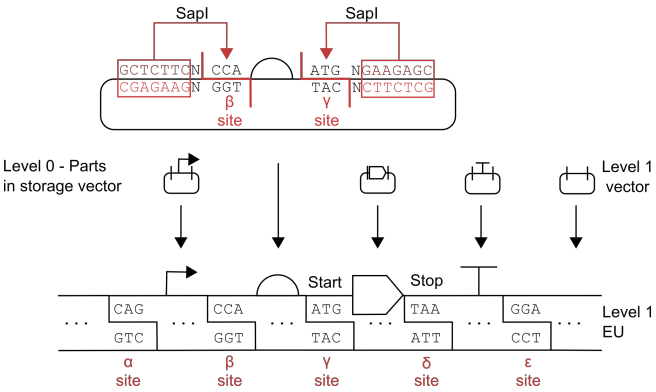


Figure A6. Level 1 assembly of an expression unit showing detail of Level 0 RBS part. Genetic parts stored in pStA0 are flanked by inward-facing SapI recognition sites (red boxes) with corresponding donor fusion sites (staggered red lines) for Level 1 assembly of expression units. RBSs use the β site (CCA) and γ site (ATG, start codon).

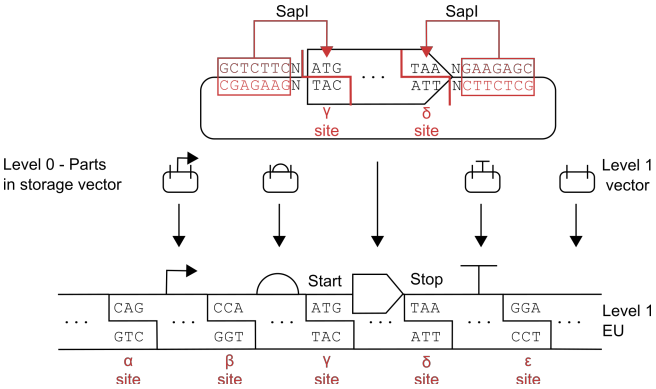


Figure A7. Level 1 assembly of an expression unit showing detail of Level 0 CDS part. Genetic parts stored in pStA0 are flanked by inward-facing SapI recognition sites (red boxes) with corresponding donor fusion sites (staggered red lines) for Level 1 assembly of expression units. CDSs use the γ site (ATG, start codon) and δ site (TAA, stop codon).

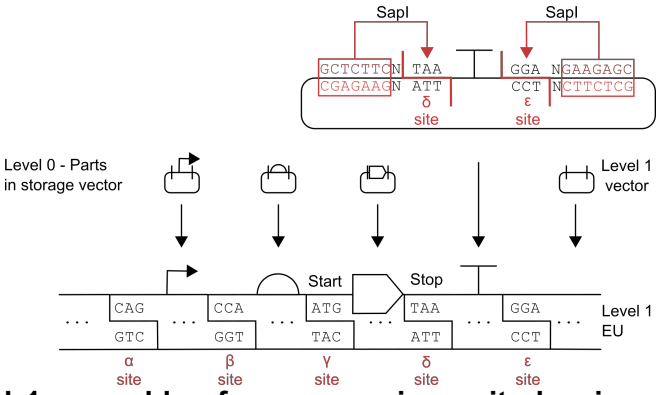


Figure A8. Level 1 assembly of an expression unit showing detail of Level 0 terminator part. Genetic parts stored in pStA0 are flanked by inward-facing SapI recognition sites (red boxes) with corresponding donor fusion sites (staggered red lines) for Level 1 assembly of expression units. Terminators use the δ site (TAA, stop codon) and ε site (GGA).

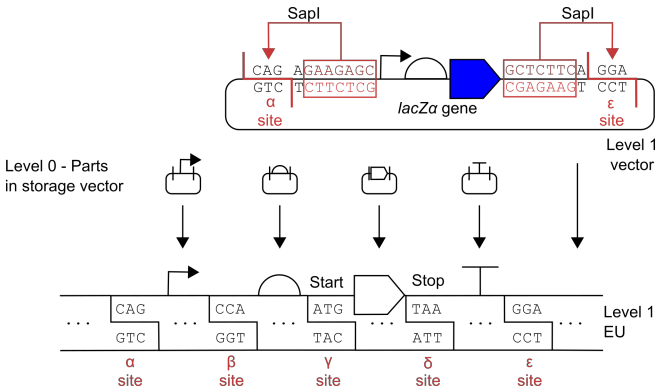


Figure A9. Level 1 assembly of an expression unit showing detail of Level 1 vector. Level 1 vectors contains two outward-facing SapI recognition sites (red boxes) with corresponding α (CAG) and ε (GGA) acceptor fusion sites (staggered red lines) for Level 1 assembly of expression units. Between the SapI sites in a Level 1 empty vector is a *lacZα* gene which can be used for blue/white screening.

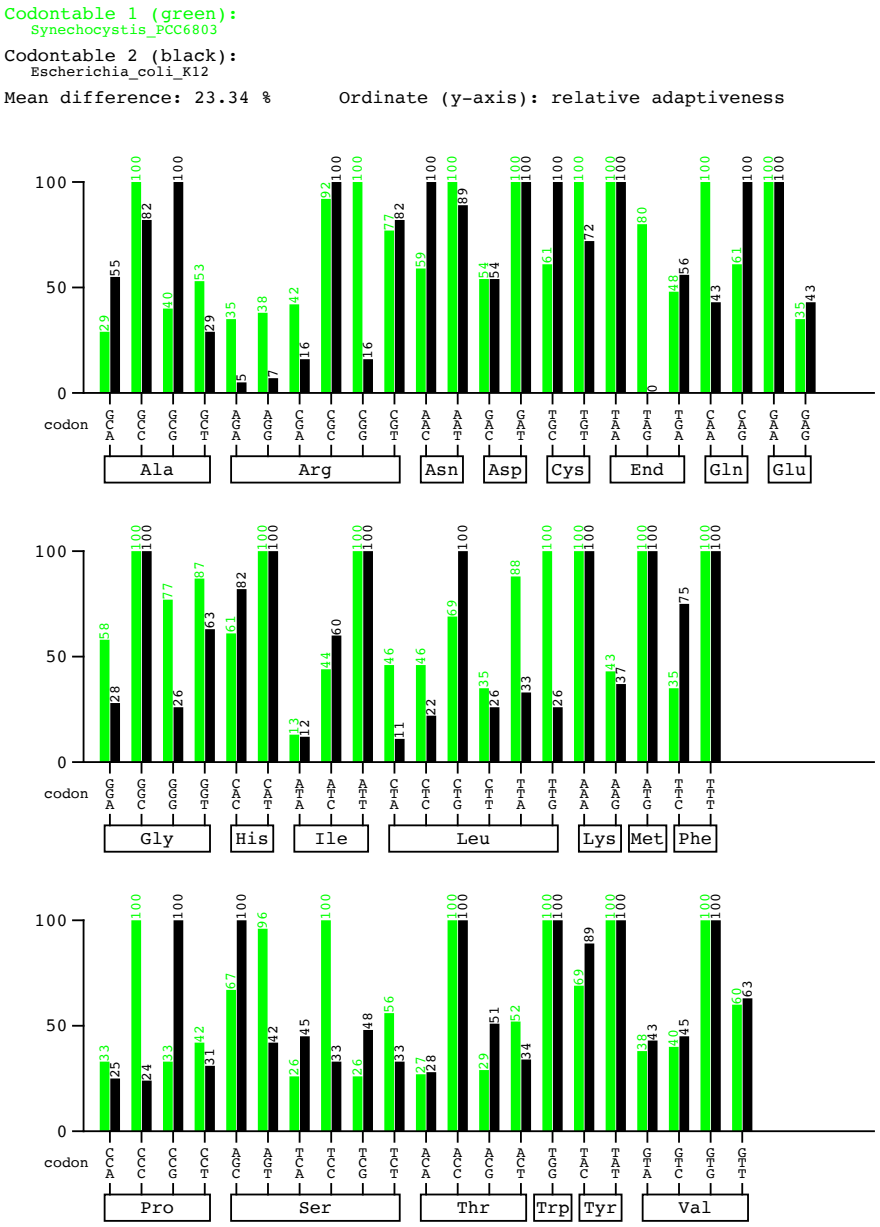


Figure A10. Comparison of codon bias between *E. coli* K-12 and *Synechocystis* sp. PCC 6803. *Synechocystis* sp. PCC 6803 codon bias shown in green. *E. coli* codon bias shown in black. Figure generated using graphical codon usage analyser (GCUA).

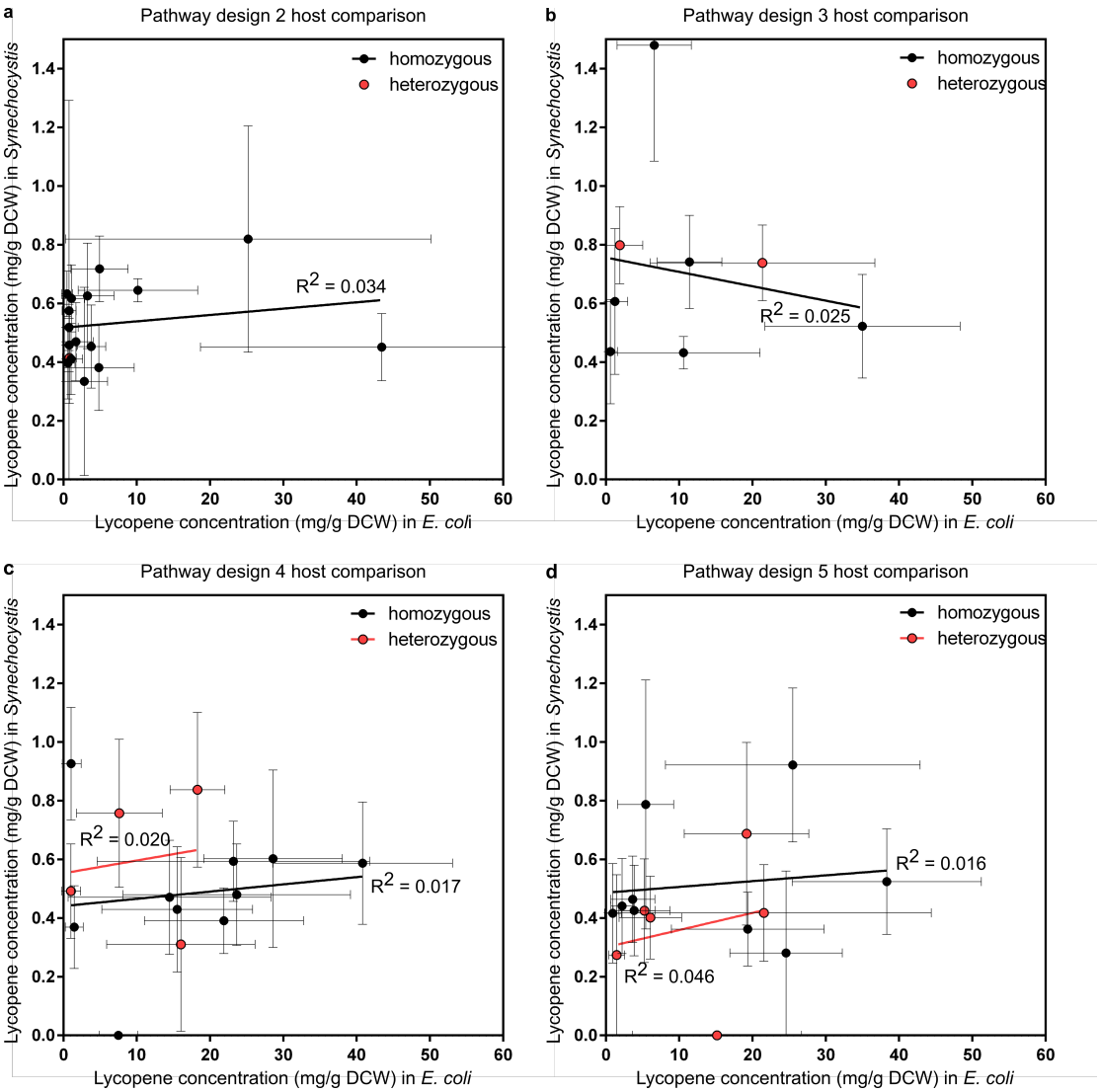


Figure A11. Host comparison of lycopene concentrations for pathways derived from Pathway design 1 – 4. Homozygote strains of *Synechocystis* are shown as black dots, heterozygote strains of *Synechocystis* shown as red dots. (a) Lycopene concentrations from pathways derived from Pathway design 2 (Same as Figure 6.13), but with homozygous and heterozygous strains shown. (b) Lycopene concentrations from pathways derived from Pathway design 3 (Same as Figure 6.14), but with homozygous and heterozygous strains shown. (c) Lycopene concentrations from pathways derived from Pathway design 4 (Same as Figure 6.15), but with homozygous and heterozygous strains shown. (d) Lycopene concentrations from pathways derived from Pathway design 5 (Same as Figure 6.16), but with homozygous and heterozygous strains shown.

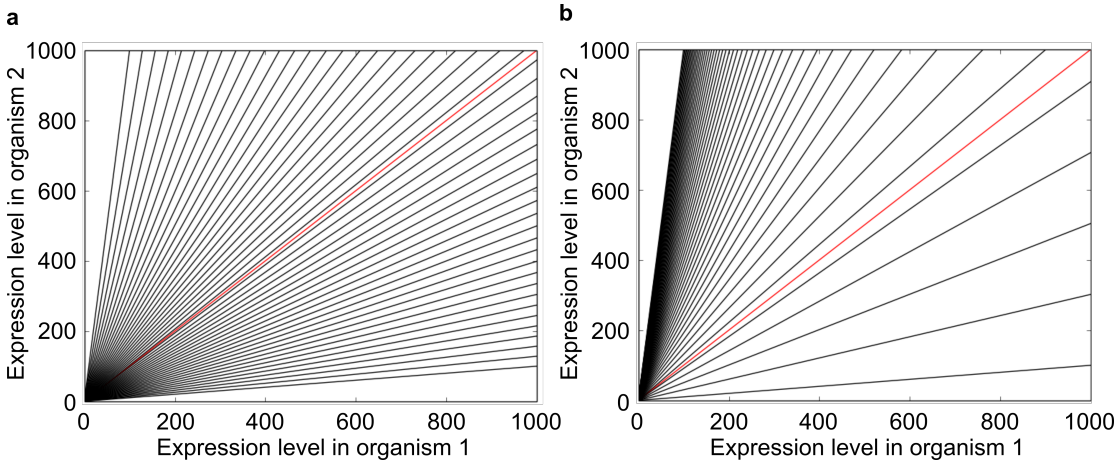


Figure A12. Using angles or gradients as increments to search datasets. (a) Using gradients as the values to increment the search space created a hugely uneven distribution that heavily favoured the bigger gradients. (b) Using angles rather than gradients it was possible to achieve an evenly distributed search.

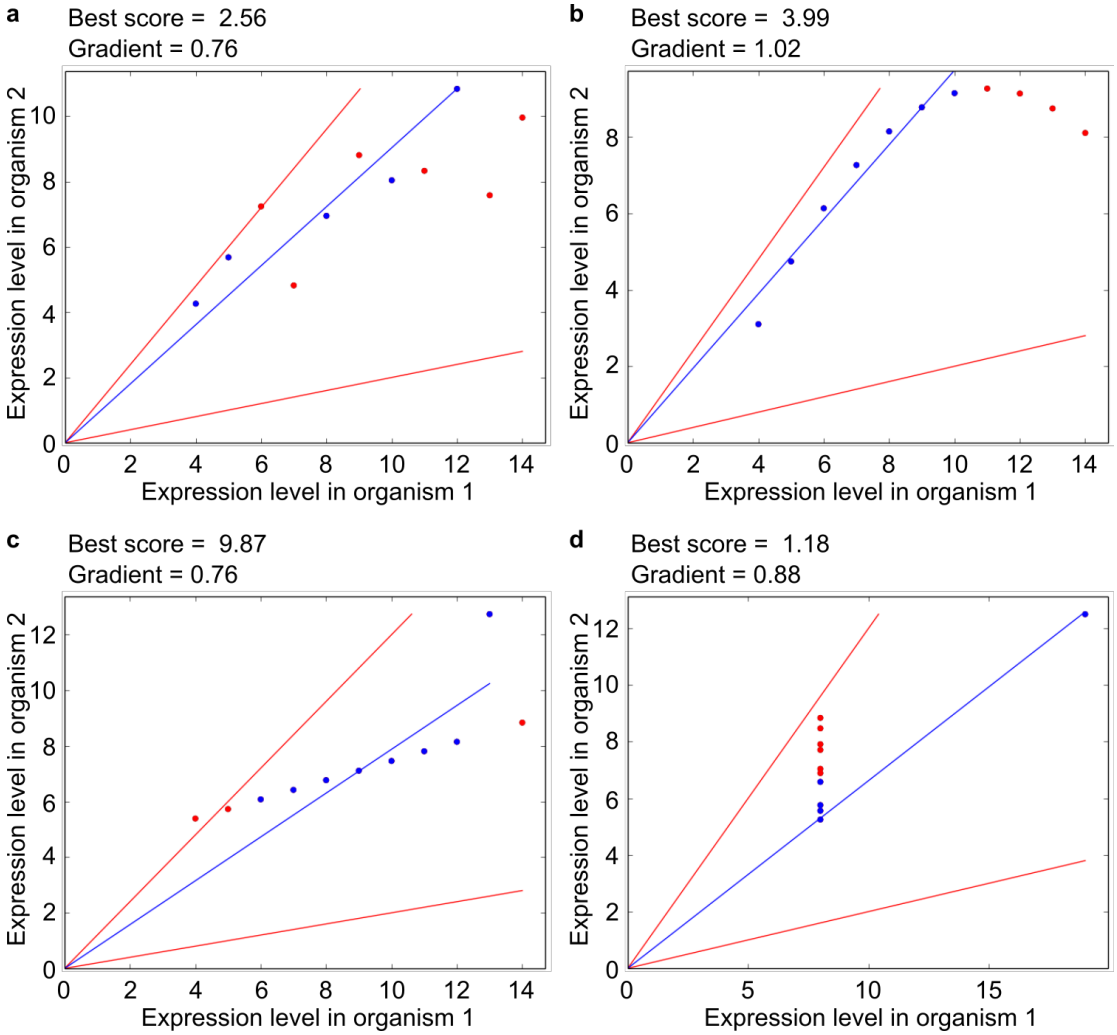


Figure A13. Testing SUBSETTER with Anscombe's Quartet. Anscombe's quartet provided four datasets to test the algorithms performance. For each of the four datasets SUBSETTER was run using the following parameters: gradMin = 0.2, gradMax = 1.2, numberofGradients = 200, lengthofShortlists = 14, minSubsetLength = 5, maxSubsetLength = 10. The data shows the subsets (blue data points) and their corresponding gradient (blue line), which were selected from the shortlisted dataset (red data points) that were found between the maximum and minimum gradient search space (red lines).

Appendix Notes

Appendix Note A1. Order of clones from weakest to strongest in Figure 6.8: clone 34, 67, 93, 10, 70, 86, 5, 14, 46, 51, 74, 89, 37, 22, 41, 49, 61, 91, 78, 94, 58, 44, 82, 39, 64, 54, 50, 27, 15, 53, 19, 32, 38, 56, 59, 33, 43, 85, 52, 3, 88, 30, 28, 25, 47, 48, 57, 62, 9, 83, 8, 11, 68, 18, 80, 26, 2, 31, 65, 71, 84, 40, 36, 96, 6, 73, 55, 60, 1, 81, 66, 63, 35, 72, 13, 42, 16, 23, 12, 45, 7, 21, 79, 87, 17, 75, 4, 24, 90, 95, 77, 92, 20, 69, 76, 29.

Appendix Note A2. Order of clones from weakest to strongest in Figure 6.9: clone 80, 44, 8, 20, 32, 56, 92, 68, 14, 33, 5, 51, 21, 39, 42, 37, 66, 57, 91, 79, 2, 19, 63, 43, 81, 3, 10, 9, 52, 83, 47, 75, 64, 70, 55, 22, 77, 27, 35, 30, 26, 78, 6, 34, 31, 67, 4, 50, 86, 76, 46, 53, 73, 13, 25, 62, 49, 65, 93, 1, 59, 82, 36, 41, 90, 38, 15, 61, 95, 45, 16, 17, 7, 23, 71, 58, 40, 89, 29, 96, 74, 11, 87, 18, 69, 85, 72, 88, 60, 94, 24, 28, 12, 84, 48, 54

Appendix Note A3. Order of clones from weakest to strongest in Figure 6.10: clone 45, 71, 11, 8, 48, 83, 95, 4, 16, 44, 73, 94, 29, 77, 36, 40, 21, 47, 51, 15, 42, 6, 55, 53, 50, 63, 75, 30, 12, 57, 9, 56, 35, 65, 33, 90, 39, 10, 31, 41, 62, 2, 20, 28, 52, 67, 5, 19, 24, 78, 26, 27, 13, 79, 59, 64, 37, 38, 3, 17, 46, 82, 25, 23, 81, 74, 14, 91, 7, 58, 85, 61, 93, 88, 72, 49, 68, 86, 76, 32, 87, 22, 34, 70, 69, 89, 84, 54, 92, 43, 1, 66, 18, 96, 60, 80

Appendix Note A4. Order of clones from weakest to strongest in Figure 3.17: pATM2, PL2c80, PL2c30, PL2c72, PL2c86, PL2c57, PL2c70, PL2ca6, PLc43, PL2c56, PLc60, PL2c42, SPLc112, SPLc1, PL2c52, pATM10, PL2c22, SPLc6, SPLC115, SPLc2, PL2c55, SPLc47, PL2c 49, PL2c77, SPLc75, PL2c37, PL2c51, PL2c36, PL2c87, SPLc33, PL2c83, PL2c24, SPLc25, SPLc14, SPLc3, SPLc69, PL2c48,SPLc46, SPLc52, SPLc19, PL2c79, PL2c17, SPLc28, SPLc17, PL2c65, SPLc23, PL2c35, SPLc35, SPLc107, SPLc45, PL2c58, SPLc13, SPLc15, SPLc157, SPLc32, SPLc36, SPLc10, SPLc22, PL2c82, PL2c69, SPLc34, PL2c71, SPLc26, SPLc40, SPLc5, SPLc41, PL2c64, SPLc61, SPLc98, PL2c4, SPLc192, SPLc31, PL2c11, PL2c61, PL2c34, PL2c41, PL2c6, PL2c54, PL2c1, PL2c2, PL2c74, PL2c39, PL2c5, PL2c46, PL2c63, PL2c28, PL2c25, PL2c19, PL2c9, PL2c8, PL2c13, PL2c14, PL2c32, PL2c3, PL2c20, PL2c40, PL2c68, PL2c53, PL2c78, PL2c15, PL2c12, PL2c31, PL2c47. SPL = Synthetic Promoter Library. PL2 = Promoter library 2.

Appendix Note A5. Order of clones from weakest to strongest in Figure 3.18: clone SPLc115, SPLc17, PL2c48, PL2c43, PL2c56, PL2c60, PL2c55, PL2c30, PL2c41, PL2c53, PL2c13, PL2c61, PL2c58, PL2c65, PL2c36, PL2c46, PL2c57, PL2c42, PL2c37, PL2c64, PL2c17, PL2c52, PL2c79, PL2c77, PL2c54, PL2c2, PL2c16, PL2c24, PL2c86, PL2c49, PL2c51, PL2c15, PL2c14, SPLc98, PL2c1, PL2c74, PL2c22, PL2c35, PL2c39, PL2c25, PL2c83, PL2c6, PL2c34, PL2c4, PL2c72, PL2c78, PL2c87, PL2c19, PL2c70, PL2c40, PL2c71, PL2c47, PL2c80, PL2c20, PL2c82, SPLc31, PL2c3, PL2c5, PL2c31, PL2c9, PL2c12, PL2c8, PL2c69, SPLc5, SPLc32, SPLc45, pATM10, SPLc35, pATM2, SPLc15, SPLc40, SPLc22, PL2c68, PL2c11, PL263, SPLc36, SPLc23, SPLc34, PL2c32, SPLc69, SPLc61, SPLc75, SPLc3, SPLc14, SPLc46, SPLc33, PL2c28, SPLc28, SPLc1, SPLc6, SPLc47, SPLc107, SPLc52, SPLc13, SPLc192, SPLc41, SPLc157, SPLc112, SPLc26, SPLc19, SPLc10, SPLc2, SPLc25. SPL = Synthetic Promoter Library. PL2 = Promoter library 2.

Appendix Algorithms

Algorithm A1. Code for BioPort.py. BioPort.py is the file that contains the code for the search and score functions of SUBSETTER. When SUBSETTER is run it imports the BioPort.py code and runs the function fFindSubset, in accordance with the users defined parameters stated in the SUBSETTER.py code.

```
1.from __future__ import division
2.import numpy as np
3.import itertools
4.import matplotlib.pyplot as plt
5.import math
6.
7.deffFindSubset(inputCsv,gradMin,gradMax,numberOfGradients,lengthOfShortlists,minSubsetLeng
th,maxSubsetLength):
8.      bestScore = 0.0
9.      allGradients = []
10.     bestScoresPerGradient = []
11.     theSet = fCsvToList(inputCsv)
12.     gradMinRadians = np.arctan(gradMin)
13.     gradMaxRadians = np.arctan(gradMax)
14.     for value in np.linspace(gradMinRadians,gradMaxRadians,num=numberOfGradients):
15.             gradient = np.tan(value)
16.             allGradients.append(gradient)
17.             bestScoreForThisGradient = 0.0
18.             shortlist = fShortlister(theSet,gradient,lengthOfShortlists)
19.             for subset in fAllSubsets(shortlist,minSubsetLength,maxSubsetLength):
20.                     scoringResults = fScore(subset,gradient)
21.                     if scoringResults['score'] > bestScoreForThisGradient:
22.                             bestScoreForThisGradient = scoringResults['score']
23.                     if scoringResults['score'] > bestScore:
24.                             bestScore = scoringResults['score']
25.                             bestShortlist = shortlist
26.                             bestSubset = subset
27.                             bestGradient = gradient
28.                             bestxRange = scoringResults['xRange']
29.                             bestyRange = scoringResults['yRange']
30.                             bestrangeBonus = scoringResults['rangeBonus']
31.                             bestsumXGapBadness = scoringResults['sumXGapBadness']
32.                             bestsumYGapBadness = scoringResults['sumYGapBadness']
33.                             bestgapPenalty = scoringResults['gapPenalty']
34.                             bestfitPenalty = scoringResults['fitPenalty']
35.                             print "bestScore = " + str(bestScore)
36.         bestScoresPerGradient.append(bestScoreForThisGradient)
37.     print "bestScore = " + str(bestScore)
38.     print "bestSubset = " + str(bestSubset)
39.     print "bestGradient = " + str(bestGradient)
40.     print ""
41.     print "Sub-scores:"
42.     print "bestxRange = " + str(bestxRange)
43.     print "bestyRange = " + str(bestyRange)
44.     print "bestrangeBonus = " + str(bestrangeBonus)
45.     print "bestsumXGapBadness = " + str(bestsumXGapBadness)
46.     print "bestsumYGapBadness = " + str(bestsumYGapBadness)
47.     print "bestgapPenalty = " + str(bestgapPenalty)
48.     print "bestfitPenalty = " + str(bestfitPenalty)
49.     plt.xlabel('Gradient')
50.     plt.ylabel('Score')
51.     plt.scatter(allGradients,bestScoresPerGradient,color="black")
52.     plt.scatter(bestGradient,bestScore,color="red")
53.     plt.axis([gradMin,gradMax,0,(1.05*bestScore)])
54.     label = "Best score = " + str(bestScore) + "\nGradient = " + str(bestGradient)
55.     plt.text(gradMin+(0.05*(gradMax-gradMin)), (1.08*bestScore), label)
56.     plt.show()
57.     xTheSet = []
58.     yTheSet = []
59.     xTheBestShortlist = []
60.     yTheBestShortlist = []
```

```
    xBestSubset = []
    yBestSubset = []
    yExpectedBestSubset = []
    for point in theSet:
        xTheSet.append (point[0])
        yTheSet.append (point[1])
        setXMax = max(xTheSet)
        setYMax = max(yTheSet)
    for point in bestShortlist:
        xTheBestShortlist.append (point[0])
        yTheBestShortlist.append (point[1])
    for point in bestSubset:
        xBestSubset.append (point[0])
        yBestSubset.append (point[1])
        yExpectedBestSubset.append (point[2])
    plt.scatter(xTheSet,yTheSet,color="black")
    plt.scatter(xTheBestShortlist,yTheBestShortlist,color="red")
    plt.scatter(xBestSubset,yBestSubset,color="blue")
    plt.axis([0,setXMax+(0.05*setXMax),0,setYMax+(0.05*setYMax)])
    plt.xlabel('Expression level in organism 1')
    plt.ylabel('Expression level in organism 2')
    if gradMin*setXMax < setYMax:
        plt.plot([0.0,(setXMax)],[0.0,(gradMin*setXMax)],color="red")
    else:
        plt.plot([0.0,(setYMax/gradMin)],[0.0,(setYMax)],color="red")
    if gradMax*setXMax < setYMax:
        plt.plot([0.0,(setXMax)],[0.0,(gradMax*setXMax)],color="red")
    else:
        plt.plot([0.0,(setYMax/gradMax)],[0.0,(setYMax)],color="red")
    plt.plot([0.0,max(xBestSubset)],[0.0,max(yExpectedBestSubset)],color="blue")
    label = "Gradient = " + str(bestGradient) + "\nScore = " + str(bestScore)
    plt.text(0.05*setXMax, (1.08*setYMax), label)
    plt.show()

def fCsvToList(inputCsv):
   theSet = []
    inputFileHandle = open(inputCsv, "r")
    for line in inputFileHandle:
        if str.isdigit(line[0]) == True:
            lineList = line.split(',')
            theSet.append([float(lineList[0]),float(lineList[1])])
   print "theSet = " + str(theSet)
   return theSet
   inputFileHandle.close()

def fShortlister(setForShortlisting,gradient,lengthofShortlists):
    setBeingShortlised = []
    for dataPoint in setForShortlisting:
            x = dataPoint[0]
            yMeasured = dataPoint[1]
            yExpected = gradient * x
            yResidual = abs(yExpected - yMeasured)
            setBeingShortlisted.append([x,yMeasured,yExpected,yResidual])
    sortedSet = sorted(setBeingShortlisted,key=lambda point:[3])
    shortlist = sortedSet[0:lengthofShortlists]
    return shortlist

def fAllSubsets(theSet,minSubsetLength,maxSubsetLength):
    subsetList = []
    for eachSubsetLength in range(len(theSet)+1[minSubsetLength:maxSubsetLength+1]:
            for subset in list(itertools.combinations(theSet, eachSubsetLength)):
                    subsetList += [list(subset)]
    return subsetList

def fScore(subsetToScore,gradient):
    score = 0.0
    sumXGapBadness = 0.0
    sumYGapBadness = 0.0
    fitPenalty = 0.0
    subsetLength = len(subsetToScore)
    for point in subsetToScore:
```

```
132.            yResidual = point[3]
133.            fitPenalty = fitPenalty + yResidual
134.    fitPenalty = fitPenalty + 0.01
135.    sortedSubset = sorted(subsetToScore,key=lambda point: point[0])
136.    xMin = sortedSubset[0][0]
137.    xMax = sortedSubset[-1][0]
138.    xRange = xMax - xMin
139.    optimumXGap = xRange / (subsetLength - 1)
140.    for index in range(subsetLength)[1:]:
141.            nextX = sortedSubset[index][0]
142.            thisX = sortedSubset[index-1][0]
143.            xGap = abs(nextX - thisX)
144.            xGapBadness = (abs(xGap - optimumXGap) / (optimumXGap + 0.01)
145.            sumXGapBadness = sumXGapBadness + xGapBadness
146.    sortedSubset = sorted(subsetToScore,key=lambda point: point[1])
147.    yMin = sortedSubset[0][1]
148.    yMax = sortedSubset[-1][1]
149.    yRange = yMax - yMin
150.    optimumYGap = yRange / (subsetLength - 1)
151.    for index in range(subsetLength)[1:]:
152.            nextY = sortedSubset[index][1]
153.            thisY = sortedSubset[index-1][1]
154.            yGap = abs(nextY - thisY)
155.            yGapBadness = (abs(yGap - optimumYGap) / (optimumYGap + 0.01))
156.            sumYGapBadness = sumYGapBadness + yGapBadness
157.
158.    gapPenalty = (sumXGapBadness + sumYGapBadness) / 2
159.    gapPenalty = gapPenalty + 0.01
160.    rangeBonus = math.hypot(xRange, yRange)
161.    score = (rangeBonus) / (1 + (gapPenalty * fitPenalty))
162.
163.    return {'xRange':xRange, 'yRange':yRange, 'rangeBonus':rangeBonus,
'sumXGapBadness':sumXGapBadness, 'sumYGapBadness':sumYGapBadness, 'gapPenalty':gapPenalty,
'fitPenalty':fitPenalty, 'score':score}
```