Dynamically controlling metabolic valves to decouple and switch between phenotypic states.

by

Naveen Venayak

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Department of Chemical Engineering and Applied Chemistry
University of Toronto

© Copyright 2018 by Naveen Venayak
Abstract

Dynamically controlling metabolic valves to decouple and switch between phenotypic states.

Naveen Venayak

Doctor of Philosophy

Department of Chemical Engineering and Applied Chemistry

University of Toronto

2018

Microorganisms are well positioned to address many societal concerns in areas such as health, sustainability, and energy. Rapid progress in systems biology and synthetic biology has drastically improved our ability to understand and engineer complex biological systems. Sustainable biochemical production has received significant attention over the past 30 years; however, these processes must often reach near theoretical maximum performance to become competitive with petroleum alternatives. Here, we implement synthetic biological circuits in order to gain finer control over microorganisms and surpass the theoretical limits of organisms lacking such control.

To do so, we aim to identify a set of important reactions which can be dynamically controlled to redirect flux between two phenotypes: growth and production. Then, cells can rapidly accumulate in the growth stage, before a phenotypic switch to production. We have shown that this strategy can significantly improve process rates in two-stage production processes.

Since the process of metabolic engineering is iterative, we tackle diverse challenges through the design-built-test-learn cycle. First, we implement and characterize a bistable transcriptional controller as a genetic memory element to drive gene expression. We show that an optimized controller can effectively decouple growth from lactic acid production and can be implemented for two-stage production. Second, we develop a strain design algorithm (MoVE) which can use a stoichiometric metabolic model to predict genetic engineering strategies using two distinct interventions: static knockouts and dynamically controlled metabolic valves. Using this algorithm, we have identified promising candidates for valves which can be used for a range of products. Finally, we develop liquid handling workflows and a data analysis framework (Impact) to allow for the rapid and thorough study of such complex systems. We anticipate that these tools and techniques will expedite the implementation of efficient dynamic control stra-
tategies. Furthermore, we expect studies which improve the accuracy of metabolic models will reduce the development time to commercialization of these technologies.
Acknowledgements

The support of a number of individuals has contributed substantially to my graduate experience, research, and this thesis.

I would like to thank my supervisor, Professor Mahadevan, for giving me the opportunity to explore my interests and providing direction throughout my research program. I am especially grateful to the members of the Laboratory for Metabolic Systems Engineering (LMSE) and BioZone for providing feedback, engaging in valuable discussions, and providing a source of friendship and support. Particularly, I would like to thank Nikolaos Anesiadis for discussions helping direct my work and career, Pratish Gawand and Aditya Vikram Pandit for cultivating my entrepreneurial nature, Kevin Correia for his curiosity and forward-looking attitude, Kaushik Raj for his persistence and openness to explore new ideas and techniques, Shyam Srinivasan for numerous discussions about computer science and programming, Po-Hsiang (Tommy) Wang for his persistent nature drawing out so many creative ideas and for helping unwind through innumerable lunch breaks, Patrick Hyland for his supportive nature and Christopher Gowen for getting me started with Linux, and supporting the development of numerous lab techniques.

Endang Susilwati (Susie) has been a constant source of support for the entirety of BioZone; I will always be grateful for her open and caring attitude making my work attainable. I would also like to thank Weijun Gao, Dan Tomchynsyn and Dean Robson for discussions and support about everything information technology.

I would like to thank my parents, Narinder and Seema Venayak, who have been a constant source of support through the entirety of my academic pursuits, not only pushing me to excel academically but leaving me the freedom to explore my creativities. I am also grateful to Lance Peters for countless discussions regarding all things mechanical and technological. Finally, I thank my partner, Kayla Nemr, who has been an unwavering source of support through every facet of this degree.

In addition, this work would not have been possible without the support from a number of funding agencies, including the Government of Ontario and the National Science and Engineering Research Council for graduate scholarships, as well as the University of Toronto School of
Graduate Studies for conference bursaries.
## Contents

Acknowledgements ........................................ iv

Contents ........................................ vi

List of Tables ....................................... x

List of Figures ....................................... xii

1 Introduction ........................................ 1

1.1 Motivation ........................................ 1

1.2 Challenges and objectives .......................... 2

1.3 Contributions ...................................... 4
  1.3.1 The potential of dynamic control ............... 4
  1.3.2 Bistable controllers for two-stage production processes .... 5
  1.3.3 Algorithmically determining two-stage genetic intervention strategies .. 6
  1.3.4 Laboratory automation and data analytic workflows to improve throughput 6

2 Literature review ................................... 8

2.1 Abstract ......................................... 8

2.2 Introduction ...................................... 9

2.3 Restoring the Wild-Phenotype In Mutant Strains .......... 13

2.4 Dynamic Control Strategies ........................ 14
  2.4.1 Control of Fermentation Conditions ............. 14
  2.4.2 Genetic Circuits for On-Off Control of Two-Stage Fermentation .... 15
3 An optimized bistable metabolic switch to decouple growth and production states

3.1 Abstract
3.2 Introduction
3.3 Results and Discussion
  3.3.1 Optimization of the bistable circuit in oxygen-limiting conditions
  3.3.2 Dynamic control of metabolic valves to decouple growth from lactate production
  3.3.3 Design, construction, and metabolic optimization of the phenotypic state-controller for metabolic engineering
  3.3.4 Choice of evolved mutant influences phenotype
  3.3.5 Phenotypic stability for scaled production
  3.3.6 Improved production rates using two-stage production
3.4 Conclusions
3.5 Materials and Methods
  3.5.1 Strains and plasmids
  3.5.2 Media
  3.5.3 Automated 96 well plate preparation and evaluation
  3.5.4 Automated preparation of samples for end-point analyte and reporter measurement
  3.5.5 Inducer toxicity and switch rate
  3.5.6 500mL bioreactor trials
  3.5.7 Switch stability
  3.5.8 Analytical methods
# MoVE: An algorithm to identify dynamic intervention strategies to decouple growth and production phenotypes

4.1 Abstract ......................................................... 51
4.2 Introduction ................................................... 52
4.3 Results and Discussion ......................................... 53
  4.3.1 Core-model strategies .................................... 54
  4.3.2 Genome-scale strategies in *E. coli* ..................... 54
  4.3.3 Genome-scale strategies in *S. cerevisiae* ............... 62
4.4 Conclusion .................................................... 62
4.5 Materials and Methods ......................................... 63
  4.5.1 Stoichiometric metabolic models .......................... 63
  4.5.2 Two-state problem formulation ........................... 64
  4.5.3 Algorithm ................................................. 66
  4.5.4 Implementation .......................................... 69
  4.5.5 Model and MILP parameters ............................. 69
  4.5.6 Reaction connectivity .................................... 70
  4.5.7 Clustering of higher-order valve strategies .............. 70
4.6 Acknowledgments ............................................... 71

# Impact: a python framework for data analysis workflows to interpret microbial physiology

5.1 Abstract ......................................................... 72
5.2 Introduction ................................................... 73
5.3 Results and Discussion ......................................... 75
  5.3.1 Design, build, test, learn ............................... 75
  5.3.2 Analyzing raw data ...................................... 76
  5.3.3 Characterizing physiological features ..................... 78
  5.3.4 Translating and interpreting data ........................ 79
  5.3.5 Data storage and retrieval ................................ 82
5.4 Conclusion ......................................................... 83
5.5 Methods ......................................................... 83
  5.5.1 Architecture ............................................... 83
  5.5.2 Data structure ............................................ 85
  5.5.3 Features ................................................... 86
  5.5.4 Availability, continuous integration, and contributing ... 86
5.6 Acknowledgements ............................................. 87

6 Conclusions, Recommendations and Outlook 89
  6.1 Conclusions .................................................. 89
  6.2 Recommendations ........................................... 93
  6.3 Outlook ....................................................... 95

Bibliography ....................................................... 95

A An optimized bistable metabolic switch to decouple growth and production states 114
  A.1 Supplementary figures ..................................... 115
  A.2 Synthesized DNA ........................................... 124
  A.3 Primers ....................................................... 125
  A.4 RBS calculator v1.1 predicted translation initiation rates ........ 131
  A.5 Plasmid sequences .......................................... 132

B MoVE: An algorithm to identify dynamic intervention strategies to decouple growth and production phenotypes 160
  B.1 Supplementary figures ..................................... 160
  B.2 Supplementary tables ...................................... 170

C Impact: a python framework for data analysis workflows to interpret microbial physiology 177
  C.1 Supplementary information ................................. 177
# List of Tables

3.1 Plasmids used in this study, distinguishing features, and RBS strengths. RBS strengths (translation initiation rates, T.I.R) were calculated using the Salis RBS calculator v2.0 (Espah Borujeni et al., 2014). ........................................ 35

3.2 Strains used in this study and their respective genotype. ........................................ 46

3.3 Truth table for implemented state controller. Both inputs would lead to an unstable state in which both repressors are expressed maximally and inactivated by inputs. State 1 is the growth state and state 2 is the production state. A third undesirable state 3 is achieved with both inputs, where both repressors are inactivated, leading to maximal expression of each - this state is unstable. ...... 47

5.1 Example of missing data handling included with Impact through the pandas package. ........................................ 82

A.1 Primers used in this study. ........................................ 125

A.2 RBS strengths of first generation plasmids calculated using the Salis RBS calculator v1.1 (Salis et al., 2009). ........................................ 131

B.1 Targeted products in *E. coli* genome-scale model, associated reactions, and identified valves for both full and partial decoupling. Dash indicates no solution was found. ........................................ 170

B.2 Targeted products in *S. cerevisiae* genome-scale model, associated reactions, and identified valves. Dash indicates no solution was found. ........................................ 174

C.1 Default titers data format, typical of HPLC ........................................ 177
C.2 Spectromax OD data format ........................................... 178
C.3 Default kinetic data format, typical of plate readers. .................. 178
List of Figures

1.1 Overview of metabolic engineering design-built-test-cycle. .......................... 3

2.1 Genetic Circuits for Implementing Dynamic Control ................................. 9

2.2 The Development of Strain Design Algorithms toward Dynamic Control ....... 10

2.3 Substrate partitioning of wild-type and engineered organisms. ................. 12

2.4 Comparison of static and dynamic metabolic engineering strategies to improve productivity. ................................................................. 13

2.5 Genetic Circuits for Implementing Dynamic Control ................................. 17

3.1 A schematic of the bistable transcriptional controller used in this study. The mutually repressive promoter/repressor pairs consist of pTrc2/lacI and pTet/tetR. State A is driven by pL(tet)O, induced by aTc, and expresses lacI and gfpmut3. State B is driven by pTrc2, induced by IPTG, and expresses tetR and mCherry. Pathways for either growth or production phenotypes can be expressed in independent states to decouple, for example, growth and production processes. ..... 30
3.2 Overview of *E. coli* mixed-acid fermentation metabolism, knockout metabolism, and dynamic control strategy. (a) Central metabolic map highlighting anaerobic pyruvate metabolism. Red genes indicate genes which are deleted to improve lactate flux through the ldhA pathway, indicated in green. (b) Growth rate of ∆lacI strains and evolved ∆adh,pta,lacI strains. D1, D28, D59 refer to the duration of evolution, in days. (c) Lactate titer using 2g/L glucose. (d) Metabolic flux profile with valves ON, allowing flux toward acetate and ethanol. (e) Flux profile with valves OFF, redirecting flux toward lactate. Error bars represent s.d. (n ≥ 3).

3.3 Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 1 plasmids, driven by state B (pTSLAC10X). Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. (n ≥ 3). See also Figures A.8,A.9.

3.4 Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 2 plasmids, driven by state A (pTSLAC20X). Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. (n ≥ 3). See also Figures A.10,A.11.
3.5 Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State A. Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. \( n \geq 3 \). See also Figures A.12,A.13.

3.6 Deviation of product titers of evolved variants in the growth state, compared to the wild-type control. (a) Lactate titer. (b) Acetate titer. (c) Ethanol titer. Error bars represent s.d. \( n \geq 3 \).

3.7 Phenotypic stability of final construct, pTSLAC314, in the growth state. Cells were serially passed by diluting 100x every 24 hours into medium without inducer. (a) End-point product and biomass titers. Error bars represent s.d. \( n = 3 \). (b) Cell populations by flow cytometry, mean of three biological replicates is shown.

3.8 Characterization of LAC002-D59+pTSLAC314 strains in pH controlled anaerobic bioreactors. Precultures were grown in corresponding inducers for growth and production controls and inoculated into reactors with matching inducer. Strains to be switched were grown in the growth state, washed, and inoculated into reactors without inducer. These reactors were induced with 2mM IPTG at designated switching times (blue arrows): 0, 2, or 4 hours. (a-d) Product titers. (e-h) Product rates were determined by numerical differentiation. (i-l) The maximum rate is the instantaneous maximum rate obtained from numeric differentiation. (m-p) The maximum overall rate is determined from the final titer (before substrate consumption) divided by the batch time at this point.
4.1 Overview of the metabolic valve enumerator. (a) MoVE identifies intervention strategies to decouple growth and production phenotypes using a set of static genetic knockouts and dynamically regulated metabolic valves. Knockouts prime the network to allow usage of these valves to redirect metabolism between states. (b) These conceptual valves can be implemented through genetic engineering, deleting genes providing undesired network functionality and dynamically regulating a set of valves. (c) The steady-state stoichiometric flux space is mapped on two-dimensions to illustrate the trade-off between metabolic targets. The wild-type has a wide range of potential phenotypes, often achieving maximum biomass yield using natural regulatory systems. In the first state, flux is maintained through valves to achieve a high yield of the first target. In the second state, flux through valves and undesired flux vectors are eliminated (red), enforcing a desired phenotype (green). (d) Native regulatory systems are able to sense some environmental perturbations to exhibit controlled phenotypic responses. (e) Synthetic control logic can be implemented using a typical control architecture and applying genetic circuits to sense metabolites and control flux through targeted metabolic valves toward specified objectives.
4.2 Application of MoVE to a core model for \(\alpha\)-ketoglutarate (AKG) production. Line thickness is proportional to pathway flux, metabolites highlighted in blue are biomass precursors. (a) Wild-type flux distribution of the \textit{E. coli} core model. Aerobically grown wild-type \textit{E. coli} is known to partition flux between glycolysis and the pentose phosphate pathway, with a fully active tricarboxylic acid (TCA) cycle and respiratory chain (Zhao et al., 2004), leading to maximal biomass yield and growth rate (b) Flux-distribution with identified knockouts applied. Following knockout of the predicted reactions and associated genes: pyruvate kinase (PYK, \textit{pykAF}), succinyl-CoA synthetase (SUCOAS, \textit{sucA}), glutamate synthase (GLUSy, \textit{gltB}), and malate dehydrogenase (MDH, \textit{mdh}), flux is partitioned toward the pentose phosphate pathway from glycolysis, and flux through the TCA cycle is redirected through the glyoxylate shunt, marginally impacting the maximum biomass yield. (c) Flux-distribution after eliminating flux through knockouts and valves, achieving maximal AKG yield.

4.3 Valve occurrences in intervention strategies, categorized by metabolic subsystem. (a) Fully decoupled strategies target over 90% theoretical maximum production without cell growth in the second (production) state. (b) Single valves identified for fully decoupled strategies, single valves can independently redirect flux. (c) Partially decoupled strategies achieve over 70% theoretical yield while maintaining a biomass yield of 0.01 gdw/mmol. (d) Single valves identified for partially decoupled strategies. (e) Central metabolic map highlighting identified valves and associated subsystems. Bolded reactions refer to the top five valves for each decoupling strategy. Bolded metabolites refer to metabolites at the center of the bow-tie architecture of metabolism (Zhao et al., 2006).

4.4 Decoupling potential for top five most frequently identified metabolic valves. Columns correspond to strategies employing specified metabolic valves. Rows correspond to specific products. The color scale represents the number of knockouts in the identified strategy for each product and given valve, black indicates no solution was found. (a) Strategies for fully decoupled production (b) Strategies for partially decoupled production.
5.1 Overview of key elements in the design, built, test, learn cycle of metabolic engineering. ................................................................. 76

5.2 Sample data visualization generated with Impact. (a) time course data (b) titer (c) yield. Data adapted from Nemr et al. (2018). ......................... 80

5.3 Sample data visualization generated with Escher with data from Impact. .... 81

5.4 Carbon balance using an anaerobic E. coli simulation with iJO1366 ........ 81

5.5 A description of the software tool stack describing the software tools between users and data. ............................................................... 84

5.6 Overview of the data flow, from raw data to visualization, in Impact. ....... 88

6.1 Synthesis of contributions to the design-build-test-learn cycle used to metabolically engineer microbes for two-stage production processes. .......... 90

A.1 Cell populations of pTAK131 and pTAK132, grown overnight in inducers corresponding to each state. .................................................... 115

A.2 Determination of inducer toxicity and switching rate as a function of cell density and inducer concentration. (a) \(\Delta lacI\) strains were assessed for inducer toxicity by growth assay using IPTG and (b) using aTc. (c) Inducer concentration and induction OD were varied to identify inducer saturation using IPTG and (d) using aTc. The switch rate is defined as the ratio of reporter in the new target state to the previous state, six hours after induction. Error bars represent s.d. \((n \geq 3)\). ................................................................. 115

A.3 Burden assessment of toggle switch and media variants. Three strains were tested: LAC002-D28, LAC002-D28+pKDL071, LAC002-D28+pTB001. (a) Growth profiles in M9 minimal media. (b) Growth profiles in phosphate minimal media (PMM). (c) Growth profile in supplemented PMM (RDM). Error bars represent s.d. \((n \geq 3)\). ................................................................. 116

A.4 Sensitivity of phosphotransacetylase (PTAr) and alcohol dehydrogenase (ALCD2x) flux using the iJO1366 genome-scale metabolic model and anaerobic conditions on lactate export and growth rate. (a) Lactate export flux. (b) Growth rate. ... 116
A.5 Comparison of dynamic range between the two promoters used in this study, pTrc2 and pTet. Strains were grown overnight in the target inducer, to obviate the effect of switching dynamics. Error bars represent s.d. (n ≥ 3).

A.6 A comparison of switching rates between fluorescent reporters and pathway enzymes in a 500mL bioreactor batch. (a) GFP fluorescence. (b) Ethanol titer.

A.7 Simulation of leaky expression with decreasing levels of protease saturation using Michaelis-Menten kinetics. Dynamic range of expression is assumed to be 10x (ON/OFF), degradation rate was assumed to be 15 units, and production rate was solved to ensure an ON expression level of 10. Bar labels indicate the fold-change between ON and OFF states.

A.8 Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 1 (pTSLAC10X) and 1.1 (pTSLAC111) plasmids, driven by State B. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).

A.9 Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 1 (pTSLAC10X) and 1.1 (pTSLAC111) plasmids, driven by State B. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).

A.10 Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 2 (pTSLAC20X) plasmids, driven by State A. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
A.11 Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 2 (pTSLAC20X) plasmids, driven by State A. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).

A.12 Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State B, with the addition of degradation tags. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).

A.13 Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State B, with the addition of degradation tags. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).

B.1 Implementation considerations for MoVE strategies. First, a set of static knockouts can be made using canonical genetic engineering techniques. Second, if the growth rate has suffered due to sub-optimality, the strain can be adaptively evolved to maximize growth rate, accumulating genomic modifications. Finally, genetic circuitry can be applied to effect a switch by eliminating flux through target valves, and achieving at least the minimum specific target (e.g. product yield).

B.2 Overview of distributed mixed-integer linear programming solution. Jobs are created for each metabolite and given simulation parameters, and solved on a general purpose cluster. Each job is solved for a total of 128 core · hours or 256 thread · hours.
B.3 Distribution of reaction connectivities. The connectivity of a given reaction is defined as the total number of reactions connected to all metabolites in that reaction. The reactions are sorted in ascending order.

B.4 Part 1. Production envelopes for fully decoupled strategies using single valves. Log-log axes are used to highlight features at low growth rates. Production states which have near-zero growth rate are displayed as a vertical line. Production states which have a negligible production flux variability are displayed as X. The growth state is shown in red and the production state is shown in green.

B.5 Part 2. Production envelopes for fully decoupled strategies using single valves. Log-log axes are used to highlight features at low growth rates. Production states which have near-zero growth rate are displayed as a vertical line. Production states which have a negligible production flux variability are displayed as X. The growth state is shown in red and the production state is shown in green.

B.6 Part 1. Production envelopes for partially decoupled strategies using single valves. The growth state is shown in red and the production state is shown in green.

B.7 Part 2. Production envelopes for partially decoupled strategies using single valves. The growth state is shown in red and the production state is shown in green. Note: methanol passed the initial screen; however, the low yield generally makes this product infeasible to produce from glucose.

B.8 Clustered heat maps of co-occurring valves for fully decoupled production. Double and triple valves were identified, and an adjacency was built from their co-occurrences. This adjacency matrix was clustered using spectral clustering, and sorted by cluster density. Flux through these valves should be controlled in parallel to shift phenotypes between growth and production states.

B.9 Clustered heat maps of co-occurring valves for partially decoupled production. Double and triple valves were identified, and an adjacency was built from their co-occurrences. This adjacency matrix was clustered using spectral clustering, and sorted by cluster density. Flux through these valves should be controlled in parallel to shift phenotypes between growth and production states.
B.10 Occurrences of each knockout in simulations targeting optimal valves, representing the sum of data from simulations for single, double and triple valve strategies. (a) Knockout occurrences for fully decoupled production. (b) Knockout occurrences for partially decoupled production strategies.

B.11 Single valves identified in iMM904, a genome-scale metabolic model of *S. cerevisiae*. Strategies achieve over 70% of theoretical maximum product yield and a biomass yield of 0.001 gdw/mmol in the production state, and over 90% of theoretical maximum biomass yield in the growth state. (a) Valve occurrences in strategies identified for 61 of 84 naturally producible metabolites from glucose. (b) Core metabolic map highlighting identified valves in red.

C.1 Schematic of a growth curve with death phase highlighted in blue, and exponential highlighted in orange.
Chapter 1

Introduction

1.1 Motivation

The chemical industry has developed over the past 150 years with petroleum as the major feedstock. During this time, significant infrastructure has been developed for refining and converting petroleum into important products for society, such as fuels, plastics, rubbers and other commodity chemicals. However, rising concern over human impact on the environment and increasing volatility in petroleum feedstock pricing has led to the pursuit of alternative chemical production processes.

Microbial production processes are attractive because they rely on renewable feedstocks, require more mild reaction conditions, and exploit the natural diversity of metabolites in biological systems. Despite the numerous benefits of sustainable biochemical production processes, examples of commercialized products are limited. The lack of commercial successes can largely be attributed to difficulties in identifying suitable pathways for non-natural compounds, which would drop directly into the chemical industry supply chain, and optimizing the host strain to reach commercial feasibility.

Despite significant progress in metabolic modeling (Bordbar et al., 2014), strain design algorithms (Maia et al., 2016), and synthetic biology (Cameron et al., 2014), metabolic engineering remains a laborious and iterative process, especially when attempting to reach near theoretical maximum performance (titer, rate and yield). Metabolic engineers typically target a single growth-coupled steady-state phenotype, where biomass and product are produced through the
entire course of a production run. Growth-coupling allows the application of adaptive laboratory evolution, where a growth fitness advantage can be used to improve both growth and production rates. Although these simple operating strategies can be robust, they are often sub-optimal for overall production process performance.

Owing to recent progress in synthetic biology to develop genetic controllers, there is significant interest in controlling these phenotypes dynamically, subject to spatial or temporal cues. These systems can be applied to a range of applications including modifying cell behavior in response to dead zones in a heterogeneous reactor environment, balancing toxic metabolites in pathways, and having more precise control over the time course of a production run.

These two operating strategies are classified as static, where a single phenotype is targeted, and dynamic, where the phenotype is modulated subject to internal or external cues. Here, we focus on dynamic control in metabolic engineering, specifically, the modulation of phenotypes throughout a production run. By decoupling the growth and production phenotypes and switching between them, a production process can be operated with a growth stage, followed by a production stage, to significantly increase process rates (Anesiadis et al., 2008, Gadkar et al., 2005).

Although the potential of two-stage production processes is clear, the tools to implement such a strategy are limited. Recently, metabolic engineers have implemented a range of sensors and actuators to generate microbes which can be dynamically controlled. However, few such examples exist, and none have been demonstrated at scale. Current strategies are often product specific, using strategies intuited from the structure of metabolism, limiting their widespread application. More robust methods for identifying genetic engineering strategies for dynamic control, better characterized genetic components, and tools to explore the large search space of resulting designs will be of great importance in surpassing the complexity associated with dynamic control and synthetic biology.

1.2 Challenges and objectives

Metabolic engineering is often described as an iterative process through the design-build-test-learn cycle. To improve our ability to dynamically control metabolism, we targeted several
bottlenecks in this process. (Figure 1.1).

First, we aimed to identify scenarios where decoupling growth and production phenotypes for a two-stage production process is beneficial. We use a metabolic model to perform dynamic simulations and determine the theoretical potential of dynamically controlling phenotypes, as opposed to choosing single growth-coupled phenotypes (Chapter 2), motivating the implementation of such systems to significantly improve process rates.

Second, we aimed to optimize a bistable transcriptional controller (Gardner et al., 2000) in metabolic engineering, which can offer more precision, robustness, and allow for alternative induction profiles. The implementation of these systems has been limited by the lack of modularity and predictability of many synthetic circuits (Klavins, 2014). These circuits offer an ideal behavior, with two distinct and stable states for controlling pathways to decouple growth and production phenotypes. We optimize and apply this controller to decouple growth from lactate production to improve production rates in *E. coli* and demonstrate the feasibility of this strategy (Chapter 3).

Third, given a promising genetic architecture for decoupling growth and production, we aimed to develop a systematic method to identify genetic strategies to switch between two phenotypic states. This algorithm can help guide the choice of genetic engineering strategies for
dynamic control processes, and considers both growth and production phenotypes, as well as two distinct types of interventions: static knockouts and dynamically controlled valves (Chapter 4). This type of algorithmic approach has been demonstrated using mixed-integer linear programs for static metabolic engineering using growth-coupled production strains (Burgard et al., 2003, Maia et al., 2016, von Kamp and Klamt, 2017), but no algorithms exist which consider multiple phenotypes in a dynamic context.

Finally, given the enormous search space of potential strain variants using dual-state microbes, we aimed to develop a data analysis framework for microbial physiology (Chapter 5). This framework reduces the burden of consolidating and analyzing data from diverse analytical equipment, and reduces the time to begin the redesign of genetic engineering strategies. In unison, high-throughput liquid handling methods were developed to generate sufficient data to characterize the complexity of dynamically controlled microbial systems (Chapter 3).

Using tools and techniques in this work, we anticipate accelerated development of dynamic control for metabolic engineering applications and in other applications requiring phenotypic decoupling.

1.3 Contributions

The challenges and objectives outlined above are addressed in the remaining chapters of this thesis. The specific contributions from each chapter are outlined below.

1.3.1 The potential of dynamic control

To identify the potential of dynamic control for improving important process metrics (titer, rate, yield), we developed a framework to simulate two-stage production processes (Chapter 2). We used a stoichiometric metabolic model to define the bounds of metabolism and describe the trade-off between growth and production. Then, we used dynamic flux balance analysis combined with an optimization function to determine the optimal switching time to maximize the production rate in these simulated batches. Using this framework, we identified that dynamic control could improve rates by nearly 2-fold over the optimal static case, and this trend extends to a multitude of products. Work relating to Chapter 2 has been published or presented in the
journals and conferences listed below:


1.3.2 Bistable controllers for two-stage production processes

Given the importance of dynamic control for developing competitive bioproduction processes, we aimed to explore the feasibility of a relatively complex controller architecture (Chapter 3). We apply a bistable transcriptional toggle switch to control pathways for growth and production states independently, and allow the switch between them. Bistable architectures are present in a range of biological systems, for example the bacteriophage λ switch or the bacterial circadian oscillator, and effective modularization and standardization of these components will be important for widespread application. We applied this system to decouple growth from lactic acid production, the first metabolic engineering application of a bistable transcriptional circuit, to identify limitations and strengths of these systems. This work has been presented at a conference and a manuscript has been submitted for publication based on this chapter:


### 1.3.3 Algorithmically determining two-stage genetic intervention strategies

Despite the successful application of a bistable genetic circuit to decouple growth and production phenotypes, the choice of gene knockouts and dynamically controlled genes remains an open challenge for diverse molecules. Given the success of strain design algorithms to identify genetic intervention strategies for single growth-coupled production strains, we aimed to develop an algorithm with the capability to identify strategies for more complex two-stage production processes (Chapter 4). To do so, we introduce new constraints to capture flux distributions of both states (e.g. growth and production) as well as both types of interventions (e.g. static gene knockouts and dynamically controlled metabolic valves). Using this algorithm, we have identified strategies for over 75% of metabolites producible from glucose in genome-scale metabolic models of *E. coli* and *S. cerevisiae*. We expect that these strategies and this algorithm will accelerate the implementation of dynamic control for diverse applications. This work has been presented at a conference and submitted for publication as follows:


### 1.3.4 Laboratory automation and data analytic workflows to improve throughput

Despite methods to algorithmically design genetic engineering strategies and implement them in a host chassis, the process of metabolic engineering remains iterative. Although metabolic
models and synthetic genetic circuits are promising, they often require tuning on a case-by-case basis. To do so, strains are designed, built and tested, ideally at a reasonably high-throughput and small scale. This workflow typically proceeds iteratively, where designs are continuously refined; however, this process remains laborious and significant bottlenecks exist in implementation. To overcome these limitations, we have developed liquid handling methods to characterize strains in 96-well plates, and these methods have been applied in Chapter 3. In addition, to synthesize the overwhelming amount of data from experiments with this increased throughput, we developed a modular python framework to assist in writing data analysis workflow. As opposed to highly specialized pipelines which often cannot be recycled, this framework is aimed to be minimal, open-source and community driven, to maximize code re-use (http://github.com/nvenayak/impact). In addition to the application of these tools in this thesis, they have also been presented and will be published as follows:

Chapter 2

Literature review

A version of this chapter is published:


Contributions NV wrote the manuscript and performed simulations. NV, NA, WRC and RM reviewed and discussed the manuscript.

2.1 Abstract

Metabolic engineering has proven critical for the microbial production of valuable chemicals. Using new tools in synthetic biology, there has been recent interest in the dynamic regulation of flux through metabolic pathways to overcome some of the issues arising from traditional strategies lacking dynamic control. There are many diverse implementations of dynamic control, with a range of metabolite sensors and inducers being used. Furthermore, control has been implemented at the transcriptional, translational and post-translational levels. Each of these levels have unique sets of engineering tools, and allow for control at different dynamic time-scales. In order to extend the applications of dynamic control, new tools are required to improve the dynamics of regulatory circuits. Further study and characterization of circuit robustness is also needed to improve their applicability to industry. The successful implementation of dynamic
control, using technologies that are amenable to commercialization, will be a fundamental step in advancing metabolic engineering.

![Genetic Circuits for Implementing Dynamic Control](image)

**Figure 2.1:** Genetic Circuits for Implementing Dynamic Control

### 2.2 Introduction

Metabolic engineering of microorganisms has proven critical for improving their ability to produce valuable products, such as pharmaceuticals (Fossati et al., 2014, Paddon et al., 2013), nutraceuticals (Santos et al., 2011), fuels (Atsumi et al., 2008) and commodity chemicals (Nakamura and Whited, 2003, Song and Lee, 2006, Yim et al., 2011). The field of metabolic engineering has been accelerated with the introduction of new computational and experimental methods for the design and construction of plasmids (Cunningham et al., 2009, Engler et al., 2008, Farasat et al., 2014, Gibson et al., 2009, Kok et al., 2014, Lee et al., 2013, Salis et al., 2009, Xu et al., 2012) and strains (Burgard et al., 2003, Datsenko and Wanner, 2000, Song and Lee, 2013, Yang et al., 2011, Zomorrodi et al., 2012). Furthermore, synthetic biology has improved our understanding of genetic circuits (Moon et al., 2012, Olson et al., 2014) and their applicability to industry (Moser et al., 2012), while providing new tools, such as rapid and cost effective gene synthesis and sequencing, to expedite the design-build-test cycle. To produce commercially viable strains, three critical variables must be optimized: yield, titer and productivity. Heuristic targets proposed by Van Dien are 80% of theoretical maximum yield, 50 g/L and 3 g/(L h), respectively, for products near $1.00/lb (Van Dien, 2013). Of these metrics, yield has classically held high priority, and this is particularly exemplified in the development of strain design algorithms (Figure 2.2). These classical metabolic engineering methods have been reviewed at length (Lee et al., 2012).
Figure 2.2: Strain Design Algorithms. Metabolic engineering strategies can be guided by strain design algorithms (Lewis et al., 2012, von Kamp and Klamt, 2014). These algorithms have placed most emphasis on the improvement of yield, with little discussion of productivity. In yield-maximizing strain design algorithms, genetic engineering strategies are proposed to improve product yield, at the expense of growth rate (a, solid arrow). These algorithms suggest manipulations which can be implemented using static techniques, such as static gene overexpression or deletion; therefore, no dynamic control of pathways will be present. More recently, algorithms which consider productivity have been developed (Zhuang et al., 2013). These algorithms use dynamic FBA (Varma and Palsson, 1994) in order to estimate product concentrations throughout a fermentation batch. Using this strategy, many operating points can be established to generate hypothetical strains with a defined product flux and growth rate (a, dots). These strains are then evaluated for productivity, yield and titer, to find an optimal strain maximizing an objective which considers these three metrics (b).
One of the first papers to discuss the merits of titer and productivity, early in the strain design workflow, was presented by Zhuang et al. (Zhuang et al., 2013). They demonstrated that a strain designed for optimal yield results in suboptimal productivity, and discussed the inherent trade-off between the two. To overcome this issue, they developed an algorithm that takes into account productivity and titer, in addition to yield (DySSCo). However, this algorithm, along with most others, assumes that enzymes will not be dynamically controlled. We refer to this as static metabolic engineering.

In most of these strategies, product pathway flux is maximized. Since these pathways can drain metabolites used in biomass synthesis, this leads to a trade-off between growth and the production of the desired compound (Figure 2.3). Hence, most strategies to improve yield will also result in strains with low volumetric productivity, due to impaired growth rates. Slow growth rates can also be a result of poor cofactor balance (Charusanti et al., 2010), the accumulation of toxic intermediates (Kizer et al., 2008), or an inefficient metabolic network resulting from the elimination of byproducts (Fong et al., 2005). In strains with near optimal theoretical yield, optimization of the metabolic network alone is insufficient to significantly improve growth rate, as all substrate is diverted to product; this effect is evident when drawing from an early glycolytic metabolite (Solomon et al., 2013, 2012). Slow growth rates can result in low volumetric productivities and high capital costs for commercial plants, and this must be addressed to ensure commercial viability.

Although growth rates can also be improved using adaptive evolution (Fong et al., 2005), such a strategy cannot be applied in all cases; the growth rate may not improve to acceptable levels or the initial growth rate can be too low for successful adaptive evolution. Another approach to overcoming the deleterious effects of genetic modifications is the use of two-stage dynamic control, whereby fluxes can be restored to wild-type distributions to improve growth rates in a growth stage, followed by a production stage with maximal flux through the product pathway (Anesiadis et al., 2008, Gadkar et al., 2005). To demonstrate the benefit of dynamic control, we have simulated and compared the productivity for static strategies and two-stage dynamic control using dynamic flux balance analysis (Varma and Palsson, 1994). These results show that the use of a dynamic strategy has the potential to further improve the productivity for any hypothetical strain designed using the previously described DySSCo strategy (Figure
Figure 2.3: Substrate partitioning of wild-type and engineered organisms. The relative amount of flux for each product is depicted by the arrow thickness. (a) wild-type organisms mostly produce biomass with minimal or no desired product and a variable amount of byproducts (based on the selected organism and growth conditions). (b) engineered organisms generally produce a low amount of biomass and byproducts, and will mostly produce the desired product. Since the maximum total flux is limited by the substrate uptake rate, there is an inherent trade-off between flux towards biomass, and the desired product.
Figure 2.4: Comparison of static and dynamic metabolic engineering strategies to improve productivity. (a) Production envelope for succinate production. (b) Productivity comparison between the static strategy (solid line) and the dynamic strategy (dashed line) for strains along the production envelope. Hypothetical operating points were defined along the production envelope (a, solid line), to define strains with a specific product flux and growth rate. dFBA (Varma and Palsson, 1994) was used to estimate product profiles. The productivities for static strategies assumed a single-stage fermentation, with no dynamic gene regulation (Zhuang et al., 2013). The dynamic strategy implies two-stage fermentation, with wild-type growth rates in the first stage and a production stage with product fluxes corresponding to each hypothetical strain. Simulations were performed using a genome-scale metabolic model (iJO1366 (Orth et al., 2011)), 90g/L starting glucose, 10 mmol/(gdw h) glucose uptake rate and succinate as the target metabolite.

2.3 Restoring the Wild-Phenotype In Mutant Strains

In this review, we focus on applications of dynamic metabolic engineering strategies and the different approaches for their implementation. We also highlight some challenges in the context of metabolic and regulatory network dynamics.

2.3 Restoring the Wild-Phenotype In Mutant Strains

To successfully implement two-stage fermentation and address low growth rates, we require a wild-type flux distribution in the growth stage, and maximal flux through the product pathway
in the production stage (Anesiadis et al., 2008, Gadkar et al., 2005). In order to take advantage of the benefits of each stage, biomass can be quickly generated in the growth stage, before switching to a production stage. To restore a wild-type flux distribution in a mutant strain, an efficient strategy is to eliminate the effect of any genetic modifications. This restoration can be achieved through wild-type level expression of any native genes that have been deregulated, and repression of any heterologous pathways. Following the growth stage, the genetic manipulations must be restored in order to maximize the production rate.

Alternatively, in some cases, growth can be improved by modifying environmental conditions, including dissolved oxygen concentration, inducer concentration, and pH. In these scenarios it is possible to improve productivity by implementing dynamic control at the process level to control fermentation conditions; most of the early attempts to implement dynamic control utilized process level methods. More recently, owing largely to progress in synthetic biology, fluxes can be regulated dynamically by controlling the expression of key enzymes using genetic circuits. These methods will be described in the following section.

2.4 Dynamic Control Strategies

2.4.1 Control of Fermentation Conditions

One common implementation of two-stage fermentation at the process level is to follow an aerobic growth stage by an anaerobic production stage. This strategy has been applied for lactic acid production, where single-stage anaerobic strategies (using minimal media) have productivities ranging from 0.27 to 0.33 g/(L h) (Zhou et al., 2005), and can be improved approximately 10-fold using a two stage strategy, reaching a productivity of 3.32 g/(L h) (Zhou et al., 2012). More recently, process level control has been applied for 1,4-butanediol production; cells were grown aerobically to an optical density (OD$_{600}$) of approximately 10 before switching to microaerobic conditions and inducing pathway gene expression using isopropyl-1-β-thiogalactopyranoside (IPTG) (Yim et al., 2011). An alternative is to use pH for the control of the growth and production stages. For example, α-ketoglutarate was produced in Yarrowia lipolytica using two-stage fermentation; a pH of 5.0 and 50% air saturation (aerobic conditions) was used for the growth stage, while a switch to pH 3.8 and 10% air saturation (microaerobic conditions) was
used for the production stage (Yovkova et al., 2014). A two-stage strategy is also common in the recombinant protein production industry, where protein production can significantly reduce the growth rate of the production host. Thus, the production stage is induced with IPTG, or other inducers, after reaching optimal cell density (Rosano and Ceccarelli, 2014).

Two-stage fermentation has proven successful for anaerobic products and high-value proteins; however, it has not been thoroughly explored for lower-value products, or for products with pathways which are difficult to link to a process level parameter. In these cases, it is cost prohibitive to use inducers, such as IPTG, and it may not be possible to use oxygen concentration, or pH, as a trigger to switch between states. Furthermore, aerobic growth of strains designed for anaerobic production will not necessarily restore wild-type growth rates, especially when heterologous pathways draw significant flux from biomass precursors. To overcome these issues, dynamic control strategies have been proposed to allow for two-stage fermentation and dynamic control of gene expression (Holtz and Keasling, 2010).

2.4.2 Genetic Circuits for On-Off Control of Two-Stage Fermentation

Dynamic metabolic control relies largely on advances in synthetic biology to create genetic sensors and actuators (Michener et al., 2012); this has led to diverse strategies for its implementation. Constraint-based metabolic models of *Escherichia coli* and dynamic flux balance analysis (dFBA) (Mahadevan et al., 2002) were used to assess the applicability of two-stage fermentation, for the specific cases of ethanol and glycerol production. Using this strategy, it was shown that controlling gene expression in two-stages has the potential to improve productivity (Gadkar et al., 2005). To implement this system, Anesiadis et al. proposed using quorum sensing (Figure 2.5a) (Anesiadis et al., 2008), as described by Kobayashi et al. (Kobayashi et al., 2004), and demonstrated the feasibility of this system *in silico*. Another quorum sensing system, using a pheromone and RNA interference (RNAi), was developed for p-hydroxybenzoic acid (pHBA) production in *Saccharomyces cerevisiae*. Production pathway genes are repressed, while chorismate mutase (ARO7), a gene required for efficient growth, is expressed. When the critical cell density is reached, ARO7 is knocked down using RNAi, while production pathway genes are expressed (Figure 2.5b) (Williams et al., 2014).

On-off genetic control was also implemented using temperature or inducers as triggers (Zhou
et al., 2012). The $P_R$ and $P_L$ promoters from the bacteriophage $\lambda$ were used to control the native lactate dehydrogenase gene ($ldhA$), such that growth at 33°C repressed $ldhA$ and was able to increase biomass yield by 10% (as compared to static strategies), while a switch to 42°C induced expression of $ldhA$ and increased lactate production (Figure 2.5c) (Zhou et al., 2012). This strategy improved productivity by 30%, when compared to two-stage strategies without genetic manipulation (Zhou et al., 2011). A similar switch was designed for isopropanol production; the growth stage was achieved by expressing citrate synthase ($gltA$), subject to a $tetR/lacI$ based inverter, which repressed pathway genes and allowed for high growth rates. Upon addition of IPTG, $gltA$ was repressed and pathway genes were expressed to redirect flux to isopropanol production (Figure 2.5d) (Soma et al., 2014). Another implementation of two-stage dynamic control used engineered antisense RNA (Figure 2.5e), or an inverting gene circuit, to control glucokinase expression for the production of gluconate (Solomon et al., 2012). The latter system reduced growth rate and carbon waste to acetate by 50% by redirecting flux to the heterologous pathway.

2.4.3 Genetic Circuits for Continuous Control using Metabolite Sensors

The explicit two-stage strategies defined above are typically on-off strategies, with a single explicit switch within the batch. Alternatively, continuous control has the benefit of being able to dynamically sense the environment and control metabolism accordingly, thereby allowing cells to respond to variations within each batch and within the non-uniform environment of an industrial fermenter. The earliest such strategy was proposed by Farmer & Liao, who demonstrated the use of dynamic control to improve the yield and productivity for the production of lycopene (Farmer and Liao, 2000). An acetyl phosphate activated sensor was used as a marker for excess glycolytic flux; this sensor was used to drive the expression of genes responsible for lycopene synthesis (Figure 2.5f). More recently, dynamic control has been used to upregulate efflux pumps upon detection of toxic metabolites, improving yield using synthetic feedback loops (Dunlop et al., 2011, 2010, Harrison and Dunlop, 2012). Dynamic sensors have also been discovered using transcriptome analysis, by identifying promoters that are sensitive to pathway intermediates and using them to control biosynthesis pathways (Figure 2.5g, (Dahl et al., 2013); Figure 2.5h, (Zhang et al., 2012)). A similar strategy was applied using malonyl-CoA sensitive
### A. Anesiadis, 2008

**Type of Control:** On-Off  
**Level of control:** Transcriptional  
**Trigger:** Quorum Sensing (AHL)

<table>
<thead>
<tr>
<th>Protein</th>
<th>Expression</th>
<th>Regulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBiC</td>
<td>Induced</td>
<td>P&lt;sub&gt;FUS1J2&lt;/sub&gt;</td>
</tr>
<tr>
<td>ARO4</td>
<td>Induced</td>
<td>P&lt;sub&gt;FUS1J2&lt;/sub&gt;</td>
</tr>
<tr>
<td>TKL1</td>
<td>Induced</td>
<td>P&lt;sub&gt;FUS1J2&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

ARO7 interfering RNA (ARO7i) is also expressed in the presence of this pheromone, knocking down ARO7 and eliminating a competing pathway. This system requires the presence of argonaut (AGO1) and dicer (DCR1).

### B. Williams, 2014

**Type of Control:** On-Off  
**Level of control:** Transcriptional  
**Level of control:** Translational  
**Trigger:** Quorum Sensing (Pheromone)

A genetic toggle switch is used in combination with a quorum sensing circuit to control the expression of a growth-promoting pathway, subject to a cell density sensor. The quorum sensing molecule (AHL) is produced by *E. coli* carrying the luxI gene, and induces the P<sub>Lux</sub> promoter. Upon reaching critical cell density, AHL induces lacI expression from the quorum sensing circuit and represses cIts and pta in the toggle switch circuit, repressing a growth promoting pathway. Since the switch is bistable, the switch should remain in this state, even in the absence of inducers.

### C. Zhou, 2012

**Type of Control:** On-Off  
**Level of control:** Transcriptional  
**Trigger:** Heat-Shock (42°C)

IdhA expression is controlled using the P<sub>l</sub> and P<sub>o</sub> promoters. At 33°C, IdhA is repressed, preventing lactate production and forcing flux towards biomass. Upon induction at 42°C, the P<sub>l</sub> promoter is derepressed, and IdhA expression is increased to allow for lactate production.

**Figure 2.5:** Genetic Circuits for Implementing Dynamic Control
f. Farmer, 2000

**Type of Control:** Continuous

**Level of control:** Transcriptional

**Trigger:** Metabolite Concentration (Acetyl-Phosphate)

---

Acetate (Ac) concentration is sensed using the acetyl-phosphate (AcP)-dependent NR-I protein. ACP phosphorylates NR-I, allowing it to activate the PglnAp2 promoter. This promoter drives the expression of two genes required for the lycopene production pathway, in the presence of excess acetate.

---

**Figure 2.5:** (continued) Genetic Circuits for Implementing Dynamic Control
### h. Zhang, 2012

**Type of Control:**
Continuous  

**Level of control:**
Transcriptional  

**Trigger:**
Metabolite Concentration (Fatty Acid)  

Fatty acid concentration is controlled using the fadR repressor. Fatty acids inhibit fadR and derepress $P_{modB/C}$ promoters. This causes genes required for fatty acid ethyl ester (FAEE) production from fatty acids to be expressed, reducing fatty acid levels.

### i. Xu, 2014

**Type of Control:**
Continuous  

**Level of control:**
Transcriptional  

**Trigger:**
Metabolite Concentration (Malonyl-CoA)  

Malonyl-CoA concentration is controlled by regulating its production and degradation pathways. In the absence of malonyl-CoA, fapR acts as an activator of $P_{GAP}$ and produces ACC (the production pathway), while repressing $P_{T7}$ and FAS (the consumption pathway). Malonyl-CoA represses these effects, reducing ACC expression and increasing FAS expression.

---

**Figure 2.5:** (continued) Genetic Circuits for Implementing Dynamic Control
promoters, which are based on the fapR protein from *B. subtilis* (Xu et al., 2013). In the presence of high malonyl-CoA levels, the malonyl-CoA production pathway is downregulated while the malonyl-CoA consumption pathway is upregulated; the inverse is true for low level of malonyl-CoA (Figure 2.5i) (Xu et al., 2014).

Although these dynamic strategies have proven successful, they are difficult to extend to a multitude of products. In many cases, sensors and actuators will need to be derived for each compound of interest, which can be time consuming. Two-stage fermentation is an alternative approach to feedback control strategies, and both techniques can be combined. Using two-stage fermentation subject to an on-off dynamic controller, we can explicitly define the optimal switching time based on known fermentation characteristics, such as the ideal batch time and growth kinetics of an organism. Then, continuous control can be used during the production stage to finely tune gene expression. However, optimization of the dynamic controller to regulate the expression of metabolic pathways, combined with dynamic control of the external environment, will require sufficiently detailed mathematical models (Anesiadis et al., 2013) and improved genetic circuits. This represents an opportunity for further research and motivates the integration of systems and synthetic biology.

### 2.5 Challenges in the Genetic Implementation of Dynamic Control

The regulation of gene expression in the product and growth related pathways, along with many other dynamic control strategies, can be accomplished at the transcriptional (DNA), translational (mRNA) or post-translational level (protein). Many potential systems are discussed in a recent synthetic biology review (Khalil and Collins, 2010). In the simple case of eliminating flux through competing pathways to produce a compound of interest, the proteins associated with these competing pathways must be either inactive or absent. To control their activity, transcription can be reduced using well studied techniques, including inducible and repressible promoters and translation can be controlled using several RNA regulatory mechanisms (Breaker, 2011, Chang et al., 2012, Dixon et al., 2010, Lucks et al., 2011, Mutalik et al., 2012, Serganov and Nudler, 2013, Winkler and Breaker, 2003). Alternatively, the protein can be
inactivated through targeted degradation using degradation tags (Keiler et al., 1996, Torella et al., 2013), or specific inactivation, such as acetylation (Wang et al., 2010, Zhang et al., 2013) or phosphorylation (Oliveira et al., 2012, Ptacek et al., 2005). A recent review describes many of these strategies in natural systems (Chubukov et al., 2014).

The critical differences between these methods are the rate at which they act, and the tools available to engineer them. When inducing pathway expression, transcription and translation are required and each step is on the order of minutes (Alon, 2006); however, eliminating specific pathway flux requires the degradation of the associated biomacromolecules, which is much slower. Any mRNA or proteins which are present in the cell must be degraded using the natural cell machinery, and typical half-lives for mRNA are on the order of 5 minutes for *E. coli* (Bernstein et al., 2002) and 20 minutes for *S. cerevisiae* (Wang et al., 2002), whereas protein half-lives are on the order of hours (Goldberg and John, 1976, Larrabee and Phillips, 1980, Mosteller et al., 1980). Transcriptional regulation is the best studied of these three targets; however, it will exhibit the slowest response. Techniques to control translation would suffer from similar problems as with transcriptional regulation, but with slightly quicker dynamics, since transcription is not required; however, the tools to implement translational control are not as well characterized. Transcription-based dynamic control circuits are appropriate when the dynamics are on the order of hours (e.g. cell density sensing via quorum sensing circuits); however, the application of these circuits for control based on metabolite sensing can be challenging, given the fast time scales involved in metabolite turnover, on the order of 1 minute (Nielsen, 2002).

Post-translational control would exhibit fast dynamics, since proteins can be rapidly activated and inactivated, on the order of 1 to 100 µsec (Alon, 2006). Thus, they would have immediate effects on flux through associated pathways. The speed of this system is exemplified in signaling cascades and in allosteric regulation of amino acid pathways, where a quick response is required to control the amino acid pools (Jones and Fink, 1982). However, many target enzymes cannot be natively controlled at the protein level, and protein engineering is required to make them amenable to regulation. This is a more challenging task than DNA or mRNA level control since effective tools are lacking. In the case of eliminating flux through a pathway, by inactivating key pathway enzymes of interest, the dynamics can be improved
by introducing degradation tags to the proteins (Keiler et al., 1996). However, this approach increases the burden on the cell due to continuous degradation and lower steady-state protein levels. To maintain steady state protein levels, transcription and translation rates must be increased, leading to a futile cycle in which proteins are continuously produced and degraded. Hence, there is a need for new tools for post-translational control and additional research to understand the metabolic burden imposed by these dynamic control circuits, especially when degradation tags are introduced.

Furthermore, successful implementation of regulatory circuits at the transcriptional, translational or post-translation level, will require robust performance (Bashor and Collins, 2012, Lou et al., 2012, Qi et al., 2012). This requirement for robustness necessitates a clear understanding of circuit characteristics, including their input-output relationships (Olson et al., 2014), and their behavior in the diverse environments common to metabolic engineering (Klumpp and Hwa, 2014, Klumpp et al., 2009, Scott et al., 2010). In addition, tools for the design of biological circuits with faster response times (less than 1 minute) will be needed to obtain more effective dynamic control, especially in continuous control systems. A key component of these circuits will be the construction of biosensors with fast measurement dynamics, along with actuators that implement the flux changes rapidly through protein-level control.

2.6 Outlook

With recent progress in synthetic biology, dynamic metabolic engineering has become increasingly attractive. This technique has the potential to improve critical fermentation metrics, including yield, titer and productivity. Two of the most promising current technologies are using two-stage growth conditions and using dynamic feedback control systems to control pathway flux. Both of these systems have shown potential to improve not only productivity, but also yield and titers. For dynamic control strategies to be more effective and widely applicable, two important developments are required. First, genetic circuits must become more robust, so that their function is preserved across strains and conditions, and second, the dynamics of sensing and actuation have to be rapid enough to enable effective control of metabolic pathway flux, especially in the case of continuous control. In the future, we expect that advances in synthetic
biology can specifically address these issues and enable the widespread application of dynamic control in metabolic engineering and bioprocess development. Ultimately, given that dynamic metabolic engineering combines the dynamics of regulatory circuits, intracellular metabolism and the process environment, additional computational methods that integrate these aspects into a single framework will be needed to design and optimize such dynamic control strategies for the production of valuable products.

2.7 Acknowledgements

We would like to acknowledge Natural Sciences and Engineering Research Council of Canada (NSERC), BioFuelNet Canada, and the University of Toronto for project funding, Mahadevan lab members for revisions on this manuscript, and Pratish Gawand for contributions to artwork. NV would also like to acknowledge funding from the NSERC Postgraduate Scholarship.

2.8 Summary of the literature and outline of the thesis

The process of metabolic engineering has been accelerated by advances in metabolic modeling and genetic engineering. Using metabolic models, genetic engineering strategies can be proposed to achieve desired steady-state growth-coupled flux distributions, referred to as static metabolic engineering strategies. Then, these strategies can be implemented using a range of tools to modulate gene expression in vivo to test these hypotheses. However, these types of genetic engineering strategies have fundamental limitations for overall process performance, due to the strict competition between the production of biomass and product.

Recently, there has been significant interest in dynamically controlling the flux distribution of microorganisms to overcome limitations arising form static strategies. In this case, gene expression is controlled subject to internal or external cues to modify this flux distribution, referred to as dynamic metabolic engineering. These control systems are often thought of analogously to more canonical electronic controllers, with a sensor, controller and actuator.

Several sensors have been developed to either sense inducers (Brockman and Prather, 2015a, Soma et al., 2017), internal metabolites (Dahl et al., 2013, Xu et al., 2014), or quorum sensing molecules (Gupta et al., 2017, Soma and Hanai, 2015) and control has most commonly been
applied at the transcriptional level. However, these systems have been applied to a limited number of compounds and the feasibility of such systems at scale is unclear.

This thesis aims to explore a number of limitations in the implementation of dynamic metabolic engineering strategies. In Chapter 3, we develop a bistable transcriptional controller to decouple growth from lactic acid production. This controller is an addressable unit of genetic memory, allowing cells to maintain their state until perturbed. Using this system, cells can be propagated in a growth state without inducer, and switched using transient induction. In Chapter 4, we implement an algorithm to identify genetic intervention strategies for these two-stage production processes. This algorithm considers two distinct steady-state flux distributions for the growth and production states, and identifies a set of knockouts and dynamically controlled valves to switch between these phenotypes. In Chapter 5, we develop a python framework to write data analysis workflows to interpret microbial physiology, and accelerate data analysis in the highly iterative metabolic engineering cycle.
Chapter 3

An optimized bistable metabolic switch to decouple growth and production states

A version of this chapter has been submitted for publication:


Contributions NV and RM conceived the study. NV designed and constructed the plasmids and strains. NV and KR performed experiments. NV analyzed data and wrote the manuscript. NV, KR, and RM reviewed the manuscript. RJ performed preliminary studies for the manuscript.

3.1 Abstract

Rising concern over human-impact on the environment and increasing volatility in petroleum feedstock pricing has led to the pursuit of microbial chemical production processes. Metabolic engineers aim to genetically modify microorganisms to improve their ability to produce valuable compounds. Growth-coupled production processes have become a center of focus; however, the burden of high-flux production pathways impose limitations on production rates. Instead,
process rates can be improved by decoupling and optimizing growth and production independently, and operating with a growth stage followed by a production stage. Here, we implement a bistable transcriptional controller to decouple and switch between growth and production states. We optimize the transcriptional controller in anaerobic conditions, typical of high-density fermentations, to ensure stability and improve induction characteristics. In addition, we characterize a range of architecture variants to minimize the impact of leaky expression and improve switching dynamics. We identify constructs which restore the optimal growth phenotype from a library of expression variants in a range of adaptively evolved strains, and identify limitations in restoring a wild phenotype after adaptive evolution. Using our optimal construct, we show that state-stability is maintained for a cumulative 100 million-fold dilutions in the growth state, simulating a high growth scale-up from 5mL to 500 000L. Finally, we assess our optimal construct in a two-stage production process to improve the instantaneous rate of lactate production by over 50%, motivating the use of these systems in broad metabolic engineering applications.

3.2 Introduction

Rising concern over human-impact on the environment and increasing volatility in petroleum feedstock pricing has led to the pursuit of microbial chemical production processes. These processes rely on renewable feedstocks, involve more mild reaction conditions, and exploit the natural diversity of compounds producible in metabolic systems. However, microbes often have limited natural performance when applied to producing valuable chemicals (Lee and Kim, 2015). Metabolic engineers aim to improve these microbes by optimizing three key performance indicators of the production process: titer, rate, and yield (TRY). It is well understood that optimization of each of these metrics is crucial to a feasible commercial bioprocess (Stephanopoulos, 2012, Van Dien, 2013), especially when producing low-value and high-volume commodity chemicals.

The negligible yield of most important products by wild microbes often makes this the first production target, with rate and titer improving naturally as a result. If the product is non-natural, a heterologous pathway must first be discovered, expressed and balanced. Then, competing pathways can be eliminated to redirect flux toward this production pathway. Most
commonly, metabolic engineers target a growth-coupled production phenotype, where biomass and product are simultaneously accumulated. The identification of such growth-coupled strategies can be guided by strain design algorithms, which use a metabolic model to predict genetic interventions (Maia et al., 2016) and many of these algorithms have been experimentally validated (Fong et al., 2005, Gawand et al., 2013, Harder et al., 2016, 2017). These strategies are implemented by knocking out genes or balancing pathways to optimize flux toward desired phenotypes. However, growth-coupled production strategies lead to theoretical limits of overall process rates, especially when producing chemicals at high yield (Klamt et al., 2017, Mahadevan et al., 2005, Venayak et al., 2015).

Recently, given the wealth of synthetic biological circuits which offer a new level of control, there has been significant interest in decoupling growth and production into distinct stages (Brockman and Prather, 2015b, Burg et al., 2016, Cress et al., 2015, Holtz and Keasling, 2010, Venayak et al., 2015). In the first stage, flux toward growth can be maximized to rapidly generate biomass, the required catalyst for the production process. Then, the second stage can proceed with maximal flux toward product, achieving significant improvements to overall production rates.

For some products, growth and production can be decoupled using natural regulatory systems, by controlling environmental conditions such as pH, temperature or oxygen availability. However, these natural regulatory systems often do not coincide with engineering objectives such as producing non-natural chemicals. To extend this capability for arbitrary products, synthetic genetic circuits can be applied to effectively control gene expression subject to numerous cues. For example, gene expression can be controlled using inducers (Brockman and Prather, 2015a, Soma et al., 2017), internal metabolites (Dahl et al., 2013, Xu et al., 2014) or quorum sensing molecules (Gupta et al., 2017, Soma and Hanai, 2015). Although such circuits have proven effective in improving biochemical production processes, significant optimization will be required to implement such genetic logic at scale.

Despite the range of implemented triggers and actuators, controllers have not received comparable attention. The complex dynamics of biological regulatory networks offer diverse natural examples of controllers with important properties. For example, the bacteriophage λ switch (Ptashne, 2004) and cyanobacterial circadian oscillator (Ishiura et al., 1998) could be applied
to achieve desirable complex network dynamics. Engineered applications of these systems have been described as some of the seminal works in synthetic biology (Cameron et al., 2014), moving toward the design and application of more robust and modular genetic circuits.

The genetic toggle switch (Gardner et al., 2000), one of the earliest implemented synthetic circuits, provides an addressable unit of cellular memory, which can be switched between states using transient induction. This type of system, which is used by the λ phage to switch between lytic and lysogenic states, could also be applied to switching microbes between phenotypes in the context of metabolic engineering, by controlling the expression of genes responsible for critical pathways (Anesiadis et al., 2008, 2013). As opposed to more typical expression systems which exhibit an analog response to inducer concentration, such a bistable system should be less sensitive to heterogeneous bioreactor environments. Furthermore, these switches can be induced transiently, with a pulse of some cue such as light (Olson et al., 2014) or nutrient starvation (Bothfeld et al., 2017). These systems could also prove effective in decreasing population heterogeneity for improved production performance (Xiao et al., 2016).

Here, we implement a bistable state controller to decouple growth and production phenotypes. These types of systems could be extended to a large number of states for more complex temporal or spatial control of microbial phenotypes (Lo et al., 2016, Roquet et al., 2016). We developed a bistable transcriptional phenotypic controller by applying a well-known mutually repressive design (Litcofsky et al., 2012) and controlled critical pathways using this switch. First, we assess this circuit in oxygen-limited conditions to characterize the burden, stability and switching dynamics in an environment common to bioproduction. Next, we apply this system to decouple growth from lactic acid production in a range of evolved E. coli mutants. This system requires the control of a small number of genes, or metabolic valves, to decouple these phenotypes and thus allows us to study the interplay of growth and production states in detail, as a function of evolutionary time. More complex metabolic engineering strategies would require careful consideration in the choice of such metabolic valves.

We consider a library of switch architecture variants to minimize the negative effect of leaky expression on production phenotypes. To ensure balanced pathway flux, we also considered a library of controllers with a range of expression strengths. With the optimal controller chosen, we assess its stability to understand the feasibility of such circuits for engineered applications,
such as prolonged production processes. Finally, we implement the optimal construct in a production batch to assess the impact of switching time on the trade-off between yield and rate.

The lack of modularity and predictability of synthetic circuits has limited the application of complex architectures for metabolic engineering. Here, we characterize a common bistable controller in conditions relevant to metabolic engineering, motivate the use of promoters with tight expression control, and demonstrate the application of an optimized circuit for lactate production. This work presents a foundation for the application of bistable architectures in a generically applicable context for metabolic engineering.

3.3 Results and Discussion

3.3.1 Optimization of the bistable circuit in oxygen-limiting conditions

As the backbone for our phenotypic controller, we implemented a two-state bistable transcriptional controller based on a seminal work in synthetic biology, the genetic toggle switch. Our first attempts using one of the original toggle switch designs, pTAK132 (Gardner et al., 2000), exhibited monostability in our host (E. coli MG1655) and process conditions (Figure A.1). We then implemented the pKDL071 toggle switch (Litcofsky et al., 2012), built with pL(tet)O/tetR and pTrc2/lacI promoter/repressor pairs, using mCherry and gfpmut3 as reporters, respectively. The two states of this controller will henceforth be referred to as state A (to denote the state with lacI/gfpmut3 expression) and state B (to denote the state with tetR/mCherry expression). State 1 can be induced with isopropyl β-D-1-thiogalactopyranoside (IPTG), and state 2 can be induced with anhydrotetracycline (aTc), analogues to lactose and tetracycline, respectively (Figure 3.1).

We assessed the switch in oxygen limited conditions which are typical of high density industrial production processes. Despite the effectiveness and wide use of GFP, mCherry, and other proteins of the same family, their requirement for molecular-oxygen to exhibit fluorescence (Tsien, 1998) can be limiting in metabolic engineering applications. This can be overcome by sampling and maturing the proteins in the presence of oxygen, before reading fluorescence. However, this process precludes the ability for on-line reads in anaerobic environments and
Figure 3.1: A schematic of the bistable transcriptional controller used in this study. The mutually repressive promoter/repressor pairs consist of pTrc2/lacI and pTet/tetR. State A is driven by pL(tet)O, induced by aTc, and expresses lacI and gfpmut3. State B is driven by pTrc2, induced by IPTG, and expresses tetR and mCherry. Pathways for either growth or production phenotypes can be expressed in independent states to decouple, for example, growth and production processes.

increases measurement variability. In experiments where fluorescent proteins were used, we allowed these proteins to mature in oxygen-rich environment before quantification. Thus, since all subsequent experiments are performed in anaerobic conditions, fluorescent proteins are matured where needed (see materials and methods, 3.5.5). However, as will be discussed, these fluorescent probes have a weak correlation to the expression level of critical metabolic pathways, and thus are not frequently used (section 3.3.3). Improved probes which do not exhibit a strong dependence on oxygen availability for fluorescence could be used to overcome some of these limitations (Drepper et al., 2007).

**Inducer concentration**

First, we determined the effective range of both inducers, IPTG and aTc, by assessing the growth profile of ΔlacI strains with pKDL071 (a control plasmid with only the core bistable controller) between 0 and 100 times of typical inducer concentrations (0-100mM IPTG, 0-10mg/mL aTc). Based on this screen, we established the maximum non-toxic inducer concentrations as 50mM IPTG (Figure A.2a) and 2.5mg/mL aTc (Figure A.2b).

Using these bounds, we determined the optimal inducer concentration to use at specific switching optical densities (OD) in order to maximize the switch rate. To quantify the switch rate, we use the ratio of fluorescent reporters in the new target state to the previous state, 6 hours after induction. After switching has completed, this ratio should be maximal. Cultures
were grown overnight in IPTG or aTc medium, before washing and diluting into fresh medium with the opposing inducer. Cells were grown for six hours before quantifying reporter concentrations. For IPTG, the switching rate had a small dependence on inducer concentration, and the switch rate decreased at higher induction densities (Figure A.2c). For aTc, both the inducer concentration and induction density had an impact on the switch rate (Figure A.2d).

Although protein degradation and production have been reported in the stationary phase (Navarro Llorens et al., 2010, Nyström, 2004, Shaikh et al., 2010), we noticed a distinct decrease in the switch rate at higher induction densities which were not in an exponential growth phase. This is especially apparent when induced with IPTG, where nearly no switching was observed in cultures inoculated and induced at OD$_{600} = 1$. These results would indicate that protein dilution by cell division has a significant impact for the clearing of proteins in this system. These results also indicate that IPTG reaches a saturating regime for LacI binding at relatively low concentrations, while significant concentrations of aTc are required to reach saturation for TetR binding, which was not observed within the non-toxic range. Therefore, rapid switching from state A to B would require relatively high concentrations of aTc ($\geq 2.5$mg/mL), while the switch from state B to A would require more typical concentrations of IPTG ($\sim 1$mM). Hence, in our subsequent experiments we used 2mM of IPTG throughout, 500ng/mL aTc for overnight cultures, and 1mg/mL aTc when rapid switching from state B to state A was desired.

**Plasmid burden**

Although bistable circuits have attractive characteristics, these can oftentimes come at the expense of metabolic burden. To obtain bistability, the expression and degradation rates of each repressor must be balanced. In addition, high expression levels lead to a wider area of bistability (Gardner et al., 2000) and increased growth rates lead to a smaller area of bistability (Klumpp et al., 2009). We characterized the growth of one of our target strains, LAC002-D28 (Table 3.2), with our target transcriptional controllers. This strain has a lower growth rate compared to wild-type, a typical consequence of most metabolic engineering efforts. In this case, the reduced growth rate is attributed to the elimination of pathways for preferred fermentation products, ethanol and acetate, which allow for a higher biomass yield and growth rate.

We characterized three different strains: LAC002-D28 (no plasmid), LAC002-D28+pKDL071,
and LAC002-D28+pTB001 (Table 3.1), in three different media (see materials and methods for recipes): M9 minimal medium, phosphate minimal medium (PMM), and rich defined medium (RDM). pTB001 is a modified version of pKDL071, which has significantly reduced translation initiation rate (TIR, according to RBS calculator v2.0 (Salis et al., 2009)) of lacI (0.71% of pKDL071) and tetR (2.7% of pKDL071). The comparison of burden between pKDL071 and pTB001 is indicative of the burden owed to the expression of tetR and lacI alone, as opposed to other elements in the bistable switch.

In M9 minimal medium, the burden of the pKDL071 plasmid is significant as compared to the no plasmid control and the pTB001 strain (Figure A.3a). This trend is even more evident in PMM, where the no plasmid control has a significant growth advantage, followed by the pTB001 and pKDL071 strains, respectively (Figure A.3b). These results suggest that the plasmid alone has a significant burden, which is substantially increased with higher repressor translation rates. Recently, a single-copy toggle switch was developed which could alleviate this burden (Lee et al., 2016); however, further study is required to understand the burden of these systems in strains which already suffer from a growth defect, such as metabolically engineered strains, as well as their stability in diverse environments. Finally, we were able to eliminate this burden by supplementing PMM with amino acids (RDM) which can directly be used to produce the toggle switch peptides (Figure A.3c).

### 3.3.2 Dynamic control of metabolic valves to decouple growth from lactate production

To implement this bistable architecture for metabolic engineering applications, a set of important reactions should be identified and dynamically controlled; these reactions are referred to as metabolic valves. The choice of such valves in complex engineered strains with a large number of genetic perturbations requires significant consideration. We applied this system to an anaerobic lactic acid production strain of *E. coli*, which required only two knockouts, simplifying the choice of these valves.

We targeted a series of MG1655 Δ*adh,pta,lacI* strains (LAC002), which have approximately half the growth rate of wild-type but reach near 100% lactate yield (Fong et al., 2005) (Figure 3.2a-c). Δ*lacI* will be referred to as a wild-type in the context of this manuscript, since it
exhibits the same phenotype under our growth conditions; *lacI* is deleted from all strains to prevent competition with the toggle switch repressors. The two deleted genes, *adhE* and *pta* are responsible for ethanol and acetate production, respectively, the preferred fermentation products in *E. coli*. By deleting these genes, flux is naturally diverted to produce mostly lactic acid, one of the few remaining fermentation products. This results in a sub-optimal biomass yield, and growth rate. By exploiting adaptive laboratory evolution, the yield of this strain improves by approximately 25% (Figure 3.2c).

![Figure 3.2](image)

**Figure 3.2:** Overview of *E. coli* mixed-acid fermentation metabolism, knockout metabolism, and dynamic control strategy. (a) Central metabolic map highlighting anaerobic pyruvate metabolism. Red genes indicate genes which are deleted to improve lactate flux through the ldhA pathway, indicated in green. (b) Growth rate of Δ*lacI* strains and evolved Δ*adh,pta,lacI* strains. D1, D28, D59 refer to the duration of evolution, in days. (c) Lactate titer using 2g/L glucose. (d) Metabolic flux profile with valves ON, allowing flux toward acetate and ethanol. (e) Flux profile with valves OFF, redirecting flux toward lactate. Error bars represent s.d. (n ≥ 3).

In this case, stoichiometric models suggest that expression of either *adh* or *pta* alone is insufficient to restore the optimal biomass yield (Figure A.4). To realize an optimal growth phenotype, *adh* and *pta* are co-expressed to allow flux to ethanol and acetate, respectively, maximizing biomass yield and growth rate. We explored a library of architecture variants and mutants to optimize expression levels toward achieving a wild-type growth rates.
3.3.3 Design, construction, and metabolic optimization of the phenotypic state-controller for metabolic engineering

The phenotypic state-controller was expressed on a medium-copy plasmid with ColE1 replicon, and to ensure optimal expression in this genetic context we designed a library with varying expression levels of \textit{adh} and \textit{pta}. It is well known that several parameters affect the steady-state levels of proteins, and thus the steady-state flux distribution and phenotype of resulting strains. These include transcription rate, translation rate, degradation rate, and dilution rate (by growth). Using this canonical switch design, we were restricted to the specific promoters used in the state controller, and thus controlled the translation rate. We modified the ribosome binding site (RBS) to build our library and determine optimal expression levels of \textit{adhE} and \textit{pta}. The RBS translation initiation rates were targeted to span a 4-fold range (Table 3.1) and were calculated using the RBS calculator (Espah Borujeni et al., 2014, Salis et al., 2009).

**Toggle switch driven by state B is limited by background expression**

The symmetric nature of this switch leads to two potential architectures for the implementation of the desired growth and production states. Here, state A is implemented for production (empty) and state B for growth. Thus, \textit{adhE} and \textit{pta} were expressed in state B (pTSLAC10X, Figure 3.3b). We characterized a range of TIR variants by modifying the RBS, in order to determine which set optimized growth rate and product yield in their respective states (Figure 3.3c-f).

The strain with the optimal growth phenotype, LAC002-D59+pTSLAC103, restored 89% of the wild growth rate and nearly eliminated lactic acid production. Despite this promising result, none of the constructs achieved high lactic acid titer in the production state, with the exception of pTSLAC104. Although pTSLAC104 achieved favorable lactic acid titer in the production state (Figure 3.3e), this construct exhibited poor growth in the growth state (Figure 3.3f). We anticipate that the high expression of \textit{adhE} and/or \textit{pta} might have led to inactive protein aggregates, preventing optimal expression and making this construct inadequate (Figure 3.3f).

We also assessed the impact of \textit{ldhA} expression in the production state to determine if lactate yield could be improved; however, this did not significantly improve lactate yields, especially
Table 3.1: Plasmids used in this study, distinguishing features, and RBS strengths. RBS strengths (translation initiation rates, T.I.R) were calculated using the Salis RBS calculator v2.0 (Espah Borujeni et al., 2014).

<table>
<thead>
<tr>
<th>plasmid</th>
<th>generation</th>
<th>distinguishing feature</th>
<th>RBS Strength (x10³ T.I.R.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>adhE</td>
</tr>
<tr>
<td>pKDL071</td>
<td>—</td>
<td>control</td>
<td>—</td>
</tr>
<tr>
<td>pTSLAC101</td>
<td>—</td>
<td></td>
<td>1.4</td>
</tr>
<tr>
<td>pTSLAC102</td>
<td>1</td>
<td>state B growth</td>
<td>2.5</td>
</tr>
<tr>
<td>pTSLAC103</td>
<td>—</td>
<td></td>
<td>4.2</td>
</tr>
<tr>
<td>pTSLAC104</td>
<td>—</td>
<td></td>
<td>8.5</td>
</tr>
<tr>
<td>pTSLAC111</td>
<td>1.1</td>
<td>ldhA expression</td>
<td>4.2</td>
</tr>
<tr>
<td>pTSLAC201</td>
<td>—</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>pTSLAC202</td>
<td>2</td>
<td>state A growth</td>
<td>4.6</td>
</tr>
<tr>
<td>pTSLAC203</td>
<td>—</td>
<td></td>
<td>7.6</td>
</tr>
<tr>
<td>pTSLAC204</td>
<td>—</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>pTSLAC301</td>
<td>—</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>pTSLAC302</td>
<td>3</td>
<td>degradation tags</td>
<td>4.6</td>
</tr>
<tr>
<td>pTSLAC303</td>
<td>—</td>
<td></td>
<td>7.6</td>
</tr>
<tr>
<td>pTSLAC304</td>
<td>—</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>pTSLAC312</td>
<td>—</td>
<td></td>
<td>4.6</td>
</tr>
<tr>
<td>pTSLAC313</td>
<td>3.1</td>
<td>low pta expression</td>
<td>7.6</td>
</tr>
<tr>
<td>pTSLAC314</td>
<td>—</td>
<td>low repressor expression</td>
<td>10</td>
</tr>
<tr>
<td>pTB001</td>
<td>—</td>
<td>low repressor expression</td>
<td>—</td>
</tr>
</tbody>
</table>
Figure 3.3: Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 1 plasmids, driven by state B (pTSLAC10X). Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. (n ≥ 3). See also Figures A.8,A.9.
in our most evolved mutant (Figure A.8-A.9)

Generally, the high levels of ethanol and acetate in the production state, where \textit{adhE} and \textit{pta} are anticipated to be OFF, led to low lactate yield. This indicates that leaky expression can be a significant hindrance to decoupling growth and production phenotypes when low levels of enzyme can lead to high pathway flux. This is corroborated by the concentration of fluorescent reporters, where the dynamic range (the ratio of ON to OFF states) of the pTrc2 promoter is 8.3-fold, compared to the pL(tet)O promoter which is 17.5-fold (Figure A.5). We anticipate the use of promoters engineered to reduced leaky expression could alleviate this issue (Lee et al., 2016), and will be of critical importance in the implementation of synthetic genetic circuits for metabolic engineering.

**Toggle switch driven by state A for tighter expression control**

We also assessed an inverted transcriptional controller architecture (compared to generation 1 constructs), where state A drove the growth state, and state B drove the production state (empty). We expected this architecture to significantly alleviate the impact of leaky expression due to the improved dynamic range of the pL(tet)O promoter (Figure A.5). Here, we note that the performance of both states is improved (Figure 3.4). Ethanol production (Figure 3.4d) and growth rate (Figure 3.4f) are restored to 97% of wild-type in the pTSLAC202 variant. Most importantly, pTSLAC202 also shows a significantly improved lactate titer in the production state, 61% of the production target (LAC002-D59+pKDL071) and a 3.2-fold increase compared to LAC002-D59+pTSLAC103 (our optimal construct from the first generation plasmids).

Although the growth state reached performance near the wild-type control and the production state was improved over first generation plasmids, it only achieved 60% of the lactate titer of the production control (LAC002-D59+pKDL071). This can be attributed to the high acetate titers, since ethanol is undetectable in the production state. We anticipate that small amounts of \textit{pta} are responsible for significant flux through this pathway, further necessitating tighter control of gene expression. The significant impact of leaky \textit{pta} expression could be owed to its favorable enzyme kinetics ($k_{\text{cat}}/K_m = 659\text{mM}^{-1}\text{s}^{-1}$ (Campos-Bermudez et al., 2010)).

Using this strain, we performed preliminary pH controlled bioreactor trials to effect a switch
Chapter 3. Implementation of a bistable phenotypic switch

Figure 3.4: Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 2 plasmids, driven by state A (pTSLAC20X). Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. (n ≥ 3). See also Figures A.10,A.11.
between growth and production. We monitored fluorescent proteins and product titers to characterize the switching dynamics. Following induction, we noted that GFP fluorescence (which should be co-expressed with \textit{adhE} and \textit{pta}) reached OFF levels within approximately 3 hours. However, we did not notice a significant shift in metabolite production rates within the entire batch duration (\textbf{Figure A.6}). This is likely due to significant differences in degradation rate between the large native peptides (≥720AA) and the heterologous fluorescent reporters (≤240AA). In order for this switch architecture to be practical in the target batch time scales, significant improvements to degradation rate of AdhE and Pta would be required. Furthermore, this experiment exemplifies the poor correlation between levels of reporters and pathway enzymes, precluding the use of reporters to study metabolic switching dynamics.

**Degradation tags improve switching rate and reduce leaky expression**

To improve switching dynamics, degradation tags were added to second generation plasmids, to generate third generation plasmids (pTSLAC30X, \textbf{Figure 3.5b}). These tags are natively added to peptides which have been incompletely translated, tethering them to the ClpXP protease and leading to their rapid degradation (McGinness et al., 2006). We utilized a degradation tag with a high affinity for the native SsrA protease, referred to as “LAA” for the terminal three amino acids of the tag (Andersen et al., 1998). Due to the significant production state titers of acetate in all generation 2 constructs, we also generated a set of variants with \textit{pta} fixed at its lower limit, while varying \textit{adhE} expression (pTSLAC31X, \textbf{Figure 3.5c}).

For this generation of plasmids, we noted that leaky expression seemed to be eliminated in all constructs for the production state (\textbf{Figure 3.5d-e}). This is demonstrated by consistently low levels of acetate and ethanol in the production state, very similar to the production controls (LAC002-D59+pKDL071), in all tested constructs. These same constructs achieve a growth rate near growth controls (Δ\textit{lacI}+pKDL071) in the corresponding state.

Based on a simple model of transcription, translation and degradation, we would expect leaky expression to be unaffected by the inclusion of degradation tags for a given steady-state protein concentration:
Figure 3.5: Characterization of LAC002-D59 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State A. Purple bars indicate controls, LAC002-D59+pKDL071 for the production state, and WT+pKDL071 for the growth state. Dashed lines indicate the growth and production targets based on these controls. (a) pKDL071 contains empty state A and B. (b) pTSLAC10X contains pta and adhE in state B, and an empty state A. (c) Acetate titer. (d) Ethanol titer. (e) Lactate titer. (f) Growth rate. Error bars represent s.d. (n ≥ 3). See also Figures A.12,A.13.
\[
\frac{dP}{dt} = \alpha - P\gamma
\]  

(3.1)

where \( P \) is the protein concentration (M), \( \alpha \) is the lumped transcription and translation rate (mol/(L s)), and \( \gamma \) is the degradation rate (1/s).

At steady-state, the dynamic range is independent of degradation rate:

\[
\frac{P_{ON}}{P_{OFF}} = \frac{\alpha_{ON}}{\alpha_{OFF}}
\]

(3.2)

However, this model assumes a linear relationship between degradation rate and protein concentration, which is unlikely owing to limited cytoplasmic concentrations of protease. Since ssrA-dependent proteolysis ensures proteins which cannot be fully translated are degraded, and only approximately 0.4% of all peptides are ssrA tagged (Lies and Maurizi, 2008), this is a likely consequence of high ssrA-tagged protein expression. Considering Michaelis-Menten type kinetics for this saturation, the model is refined as:

\[
\frac{dP}{dt} = \alpha - \gamma_{max} \frac{P}{P + K_m}
\]

(3.3)

where \( \gamma_{max} \) is the maximum degradation rate (mol/(L s)) and \( K_m \) is the Michaelis-Menten constant (mol/L).

With this model, assuming that the protease is saturated in the ON case, the relative degradation rate in the OFF state increases, leading to reduced leaky expression (Figure A.7). This result would motivate the use of degradation tags, not only to improve switching dynamics but also to control leaky expression, in cases where extremely tight control is required. These systems could be further improved by using degradation tags which can be tuned using the expression of an adapter protein (McGinness et al., 2006), albeit oftentimes with lower maximal degradation rates.

### 3.3.4 Choice of evolved mutant influences phenotype

We had anticipated that the evolved mutant (LAC002-D59), despite having near 100% lactate yield in the production state, would suffer from suboptimal growth in the growth state. Our
initial hypothesis was that silencing the effect of genetic perturbations (knockouts) would restore the optimal wild-type growth rate. In this case, we expressed these genes (adhE and pta) on a plasmid and screened a library to find the optimal construct with balanced expression. However, since the evolved mutant has accumulated mutations in addition to the gene knockouts, its genotype is more divergent from the parent strain than an unevolved mutant, and all of these mutations may need to be silenced to effectively restore the parent phenotype.

Despite these genotypic variations, we did not observe significant differences in growth state growth rates in any of the evolved mutants, using our final construct (pTSLAC314, Figure 3.6). This discrepancy could be attributed to the difference in medium used in this study (RDM), as compared to the medium used for adaptive laboratory evolution (M9), or because the original wild-type strain had not been evolved to its theoretical maximum growth rate. However, despite the evolved mutant attaining favorable growth rate, the overall phenotype diverged from the wild-type with increased evolutionary time, indicating that adaptively evolved strains require further consideration to be restored to the original parent phenotype (wild-type), especially if several phenotypic traits are of interest.

**Figure 3.6:** Deviation of product titers of evolved variants in the growth state, compared to the wild-type control. (a) Lactate titer. (b) Acetate titer. (c) Ethanol titer. Error bars represent s.d. (n ≥ 3)
3.3.5 Phenotypic stability for scaled production

To determine if cell populations would remain phenotypically stable when cells are propagated before production, we assessed the phenotypic stability of our optimal construct and strain (LAC002-D59+pTSLAC314) in the growth state. Since stable circuits should not require additional inducer, this would allow for the use of minute inducer quantities for very small volume preculture steps, used as inoculum to a seed fermentation train. We simulated a seed train from 0.005L to 500 000L, a typical industrial fermentation volume, by using four 100x dilution steps (\(10^8\times\) total dilution).

We monitored end-point phenotypic traits including product and biomass titers, as well as population distribution using fluorescent reporters (Figure 3.7). We identify that this phenotype is relatively stable through this experiment, demonstrating the effectiveness of these bistable systems for maintaining their state through scale-up processes.

![Figure 3.7](image)

**Figure 3.7:** Phenotypic stability of final construct, pTSLAC314, in the growth state. Cells were serially passed by diluting 100x every 24 hours into medium without inducer. (a) End-point product and biomass titers. Error bars represent s.d. (n = 3). (b) Cell populations by flow cytometry, mean of three biological replicates is shown.

3.3.6 Improved production rates using two-stage production

Finally, we use our optimal strain, LAC002-D59+pTSLAC314 for a two-stage production run in pH controlled 500mL bioreactors sparged with nitrogen. We tested five conditions: growth only, production only, and strains which were switched from growth to production at 0, 2 and 4 hours. The growth and production only strains are grown overnight in their corresponding inducers, and inoculated into reactors also containing these inducers. The strains to be switched are grown overnight in the growth inducer, and inoculated into reactors containing inducer-free medium. This exploits the memory characteristic of this circuit to maintain the growth state,
until the production state is induced. We screen a range of switching times, to identify the optimal switching time for growth and production rates.

The growth and production controls behaved as expected, with the growth control having the highest biomass, acetate and ethanol titers with low lactate titer, and the production control having the highest lactate titer, and lowest ethanol, acetate and biomass titers (Figure 3.8). The switched strains resulted in titers between these controls, becoming more similar to the production strain the earlier the switch was induced.

Figure 3.8: Characterization of LAC002-D59+pTSLAC314 strains in pH controlled anaerobic bioreactors. Precultures were grown in corresponding inducers for growth and production controls and inoculated into reactors with matching inducer. Strains to be switched were grown in the growth state, washed, and inoculated into reactors without inducer. These reactors were induced with 2mM IPTG at designated switching times (blue arrows): 0, 2, or 4 hours. (a-d) Product titers. (e-h) Product rates were determined by numerical differentiation. (i-l) The maximum rate is the instantaneous maximum rate obtained from numeric differentiation. (m-p) The maximum overall rate is determined from the final titer (before substrate consumption) divided by the batch time at this point.

One of the most important benefits of two-stage production is the improvement to process
rates, and the choice of switching time will have a strong influence on this rate. Early switching will leave insufficient time for biomass production, and late switching would result in unacceptably low titers. We have shown that the optimal switching time given our batch conditions was 2 hours, leading to an increase in the maximum instantaneous lactic acid production rate of 55% and an overall improvement of 20%. Given longer batch times, this improvement would be further emphasized owing to larger biomass concentrations.

3.4 Conclusions

We demonstrate here the first application of a bistable state-controller in the context of two-stage production. These advanced controller architectures offer promising characteristics for stability and induction profiles, at the expense of increased complexity. To understand this complexity, we first characterize the state controller in anaerobic conditions to identify optimal induction parameters. Next, we screen a range of architecture variants to identify an optimal architecture to overcome leaky expression and reach theoretical production and growth targets. In addition, we characterize the stability of this metabolic controller to show this controller is stable for a cumulative 100-million fold dilution (comparable to a scale-up from 5mL to 500000L). Finally, we use our optimized strain to improve the instantaneous rate of lactate production by 55% over control strains with no dynamic control. This study and optimized circuit architecture will drive the adoption of more complex and promising synthetic circuits to engineered systems, and motivates the improved characterization of such circuits for engineering applications.
3.5 Materials and Methods

3.5.1 Strains and plasmids

The parent strain, MG 1655 Δadh,pta and the resulting strains from adaptive laboratory evolution, were generously donated by Dr. Stephen Fong (Fong et al., 2005). The transcriptional repressor, lact, was deleted from all strains using the λ-red method to eliminate interference with the toggle switch (Datsenko and Wanner, 2000). The deletion cassette was amplified from pKD4 using primers lacI_RED_F and lacI_RED_R. The resistance marker was removed using the pCP20 plasmid (Datsenko and Wanner, 2000). Strains were transformed with designated plasmids as indicated using electroporation.

Table 3.2: Strains used in this study and their respective genotype.

<table>
<thead>
<tr>
<th>Name</th>
<th>Genotype</th>
<th>ALE time</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT</td>
<td>MG1655 ΔlacI</td>
<td></td>
<td>Litcofsky et al. (2012)</td>
</tr>
<tr>
<td>LAC001-D1</td>
<td>MG1655 Δadh,pta</td>
<td>1</td>
<td>Fong et al. (2005)</td>
</tr>
<tr>
<td>LAC001-D28</td>
<td>MG1655 Δadh,pta</td>
<td>28</td>
<td>Fong et al. (2005)</td>
</tr>
<tr>
<td>LAC001-D59</td>
<td>MG1655 Δadh,pta</td>
<td>59</td>
<td>Fong et al. (2005)</td>
</tr>
<tr>
<td>LAC002-D1</td>
<td>MG1655 Δadh,pta,lacI</td>
<td>1</td>
<td>This study</td>
</tr>
<tr>
<td>LAC002-D28</td>
<td>MG1655 Δadh,pta,lacI</td>
<td>28</td>
<td>This study</td>
</tr>
<tr>
<td>LAC002-D59</td>
<td>MG1655 Δadh,pta,lacI</td>
<td>59</td>
<td>This study</td>
</tr>
</tbody>
</table>

Plasmids were transformed and maintained in DH10β. All enzymes were obtained from New England Biolabs (NEB). Synthetic oligonucleotides were obtained from Integrated DNA Technologies. All PCR amplifications for plasmid construction were performed using Q5 High-Fidelity Polymerase (NEB). Plasmids were constructed using standard molecular biology techniques and ligase cycling reaction (Kok et al., 2014). Plasmid parts were flanked by insulating sequences (UNS) to improve assembly efficiency (Torella et al., 2013). pKDL071 was generously donated by Dr. James Collins and used as the backbone for the state controller plasmids in this study (Litcofsky et al., 2012). pTSLAC plasmids were constructed from pKDL071, MG1655 gDNA.
and gBlocks using primers indicated in Supplemental Table A.1. Ribosome binding sites (RBS) were generated using the RBS Calculator (Salis et al., 2009) v2.0, and were added on primer overhangs.

**Table 3.3:** Truth table for implemented state controller. Both inputs would lead to an unstable state in which both repressors are expressed maximally and inactivated by inputs. State 1 is the growth state and state 2 is the production state. A third undesirable state 3 is achieved with both inputs, where both repressors are inactivated, leading to maximal expression of each - this state is unstable.

<table>
<thead>
<tr>
<th>Input</th>
<th>State</th>
<th>Current</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>aTc</td>
<td>IPTG</td>
<td>Current</td>
<td>Next</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>A or B</td>
<td>unstable</td>
</tr>
</tbody>
</table>

### 3.5.2 Media

Phosphate minimal medium (PMM) was composed of salts (10X, in 1L: 35g KH$_2$PO$_4$, 50g K$_2$HPO$_4$, 35g (NH$_4$)$_2$HPO$_4$) (Causey et al., 2003), supplements (1000X: 1M MgSO$_4$, 20000X: 0.5M CaCl$_2$), and trace metals (1000X: 1.6g/L FeCl$_3$, 0.2g/L CoCl$_2$·6H$_2$O, 0.1g/L CuCl$_2$, 0.05g/L H$_3$BO$_3$, 0.8g/L MnCl$_2$, 0.2g/L NaMoO$_4$, 0.2g/L ZnCl$_2$·4H$_2$O).

M9 minimal medium was composed of salts (5X in 1L: 64g Na$_2$HPO$_4$-7H$_2$O, 15g KH$_2$PO$_4$, 2.5g NaCl, 5.0g NH$_4$Cl), with supplements and trace metals from PMM.

Rich defined medium was composed of PMM, amino acid supplements (5X: 4mM alanine, 26mM arginine, 2mM aspartate, 0.5mM cysteine, 3.3mM glutamate, 3mM glutamine, 4mM glycine, 1mM histidine, 2mM isoleucine, 4mM leucine, 2mM lysine, 1mM methionine, 2mM phenylalanine, 2mM proline, 50mM serine, 2mM threonine, 0.5mM tryptophan, 3mM valine) and nucleotide supplements (10X: 2mM adenine, 2mM cytosine, 2mM uracil, 2mM guanine in 15mM KOH). Glucose was added as the sole carbon source as indicated.
Phosphate buffered saline was obtained from Biobasic (PD8117) and used for cell resuspension when reading fluorescence.

3.5.3 Automated 96 well plate preparation and evaluation

Experiments were prepared using a Tecan EVO 200 liquid handling system, equipped with an 8-channel fixed tip arm and a Tecan M200 plate reader. Cells were grown overnight in 96-well plates in 150µL LB+1% glucose including inducers and antibiotics as required. Plates were sealed with a PCR film (Bio-Rad, MSB1001) in order to limit oxygen availability. Plates were centrifuged at 2000xg for 10 minutes and transferred to liquid handler. The liquid handler removed supernatant from the plate, added fresh medium, and resuspended cells. This wash procedure was repeated twice, before an OD normalization procedure.

Plates were transferred to a Tecan M200 plate reader to begin OD normalization procedure. The OD in each well was read at 600nm, and from this blank subtracted data the required volume of cells, and balance volume of medium for a total of 150µL were calculated using Magellan (v7.2, Tecan) and dispensed by the liquid handler. Plates were cycled 3x with nitrogen in an airlock and transferred to an anaerobic chamber with a 100% N₂ atmosphere. 50µL mineral oil (BioShop, MIN444) was added to each well before being read to eliminate evaporation. Transparent films were unsuitable in our setup due to condensation on the film. Plates were incubated at 37°C in a plate reader (Spectramax M2) with shaking inside the anaerobic chamber and the optical density was monitored at 600nm.

3.5.4 Automated preparation of samples for end-point analyte and reporter measurement

Following incubation, plates were transferred to the liquid handler. Medium was aspirated from below the oil layer to a microtiter plate and centrifuged at 2000xg for 10 minutes. Supernatant was transferred to a 0.2µm filter plate (Millipore MSGVN2210) and filtered at 300 psi for 120 seconds, the flow through was collected for measurement of analyte concentration via HPLC. Cell pellets were resuspended by repeated mixing in phosphate buffered saline and transferred to a microtiter plate (Corning 3904). This plate was incubated at room temperature for thirty minutes, for a total of one hour from oxygen exposure, before fluorescence determination (if
3.5.5 Inducer toxicity and switch rate

Inducer toxicity was assessed by growing cells overnight in 5mL LB+1% glucose+50µg/mL kan with corresponding inducer, either 2mM IPTG or 500ng/mL aTc. Cells were washed with PMM, OD normalized to 0.05 and seeded into 96-well plates containing 150µL of medium with varying inducer concentration. Cells were grown in an anaerobic chamber.

Switching rate was assessed similarly to above, with OD being varied as an additional variable. Cultures was grown for 6 hours before resuspending cells in PBS, incubating at room temperature for 30 minutes, and reading fluorescence.

3.5.6 500mL bioreactor trials

LAC002-D59+pTSLAC314 strains were inoculated from glycerol stocks into 5mL+1% glucose+50µg/mL kan LB and grown overnight. Cultures were transferred to 200mL LB+1% glucose+50µg/mL kan in 250mL sealed baffled flasks, for oxygen limitation. Cells were induced with 500ng/mL aTc for the production state and 2mM IPTG for the growth state. Cells were washed 3x in RDM before being transferred to nitrogen-sparged 0.5L bioreactors controlled at pH 7 with 15M KOH and 37°C. Bioreactor medium was RDM + 5g/L glucose. Growth control reactors contained 2mM IPTG, production control reactors contained 500ng/mL aTc, switched reactors contained no-inducer, and 2mM IPTG was added at given induction points.

3.5.7 Switch stability

Switch stability was determined using our final strain and construct, LAC002-D59+pSLAC314. Cultures were inoculated from glycerol stock into 5mL LB+1% glucose+50µg/mL kan. 50µL of culture was transferred to anaerobic tubes, prepared by sparging nitrogen for 5 minutes. Anaerobic tubes were grown overnight in a shaker incubator at 250RPM and 37°C for 24 hours. This process was repeated 3x for a total 10^8 dilution. Samples were taken daily for HPLC and flow cytometry.
3.5.8 Analytical methods

For plate experiments, absorbance was measured every 15 minutes at 600nm to assess optical density. Following incubation, plates were prepared as needed for downstream HPLC or fluorescence applications. Product concentrations were measured by HPLC using an Aminex HPX-87H cation-exchange column (BioRad, 1250140) at $0.6 \frac{mL}{min}$ with 5mM H$_2$SO$_4$ mobile phase, 60° column temperature and 20µL injection volume. Chromatograms were integrated using Chromeleon v7.

Fluorescence was read in a Tecan M200 plate reader equipped with a monochromater and the following parameters: gfpmut3 (Ex: 488/9, Em: 525/20, K: 110) and mCherry (Ex: 570/9, Em: 610/20, K: 150).

Flow cytometry was performed using a LSRFortessa X-20 (BD Biosciences). GFP was acquired using a 488nm laser and a 525/50nm emission filter, and mCherry was acquired using a 561nm laser and 610/20nm emission filter. At least 25,000 single cell events were collected for each sample. A FSC-A threshold of 1500 and a SSC-A threshold of 1000 were used in combination to improve detection. Data was analyzed in python, using the fcsparser package. Data was gated as follows to eliminate non-cell particles: side scatter 6000-20000, forward scatter 2000-15000. Two dimensional log-histograms were generated using numpy, with 25 bins in each dimension, between 0.1 and $10^6$.

3.5.9 Analyte data analysis

Fermentation data was analyzed in python using the IMPACT framework, an open-source toolbox for data analysis (http://github.com/nvenayak/impact).
Chapter 4

MoVE: An algorithm to identify dynamic intervention strategies to decouple growth and production phenotypes

A version of this chapter has been submitted for publication:


Contributions NV and RM conceived the study. AvK and NV implemented the algorithm. NV performed calculations and analyzed the results. NV, AvK, SK and RM discussed the results and wrote the manuscript.

4.1 Abstract

Metabolism is highly regulated, allowing for robust and complex behavior (Browning and Busby, 2016, Heiden et al., 2009). This behavior can often be achieved by controlling a small number of important metabolic reactions, or metabolic valves. Here, we present a method to identify the
location of such valves: the metabolic valve enumerator (MoVE). MoVE uses a metabolic model to identify genetic intervention strategies which decouple two desired phenotypes. We apply this method to identify valves which can decouple growth and production to systematically improve the rate and yield of biochemical production processes (Brockman and Prather, 2015b, Burg et al., 2016, Cress et al., 2015, Harder et al., 2017, Holtz and Keasling, 2010, Venayak et al., 2015). We used a distributed mixed-integer linear programming approach to identify a set of gene knockouts, and a set of metabolic valves, to switch between growth and production phenotypes for over 70% of natural products in genome-scale models of \textit{E. coli} and \textit{S. cerevisiae}. Despite the diversity of targets, a small number of valves were highly represented. These valves were involved in energy metabolism or near important branchpoints, consistent with the bow-tie structure of metabolic networks (Friedlander et al., 2015, Zhao et al., 2006). MoVE offers a systematic approach to identify metabolic valves using readily available metabolic models, providing insight into the architecture of metabolic networks and accelerating the widespread implementation of dynamic flux redirection in diverse systems.

4.2 Introduction

Biological regulatory networks allow for metabolic transitions which are apparent in a range of biological processes. These regulatory systems are the basis for ubiquitous cellular phenomena including complex cell cycles, robustness to changing environments, and eukaryotic development through stem cell differentiation and tissue morphogenesis. The architecture of these systems can vary tremendously, from regulating short pathways, to having global metabolic effects (Covert and Palsson, 2002, Reznik et al., 2017). Engineered applications of these systems have been developed to understand complex cellular processes (Polstein et al., 2017), implement genetic logic (Nielsen et al., 2016, Roquet et al., 2016), and improve microbes to produce valuable chemicals (Brockman and Prather, 2015b, Burg et al., 2016, Cress et al., 2015, Holtz and Keasling, 2010, Venayak et al., 2015), using control points identified by biochemical assay, or intuited from models of metabolic structure. The choice of such control points is further complicated for non-natural objectives, such as chemical production, where few biological examples exist.

Metabolic network structure has been studied using metabolic models for diverse purposes
including consolidating high-throughput -omics data, identifying drug targets, and predicting metabolic phenotypes (Bordbar et al., 2014). In particular, these models have been used extensively for designing microbial cell factories (Maia et al., 2016). This is commonly accomplished using mixed-integer linear programming (MILP) techniques to identify network modifications, which can then be implemented by modulating gene function. In particular, a large number of algorithms have been developed to identify such interventions to improve microbes for growth-coupled chemical production (Burgard et al., 2003, Maia et al., 2016, Ranganathan et al., 2010, von Kamp and Klamt, 2014). These strategies rely on the simultaneous production of both biomass and product; however, the burden imposed by high-flux production pathways can severely limit this strategy when producing chemicals at high yields, which is particularly evident when considering the trade-offs between yield and productivity (Burg et al., 2016, Klamt et al., 2017, Venayak et al., 2015). Furthermore, growth-coupling can require specific network features, which may not exist in all organisms (Jouhten et al., 2017, Klamt and Mahadevan, 2015, Pandit et al., 2017). Instead, growth and production could be separated, allowing production processes to be operated in two stages, where biomass is accumulated before high-rate production is initiated. A phenotypic shift can be realized using a number of stimuli including inducers (Brockman and Prather, 2015a, Soma et al., 2017), internal metabolites (Dahl et al., 2013, Xu et al., 2014), and cell density (Gupta et al., 2017, Soma and Hanai, 2015). Although many strain design algorithms exist, these algorithms identify static interventions to achieve a single steady-state growth-coupled phenotype. None of these algorithms is suitable for identifying dynamic interventions, where multiple phenotypes must be considered.

### 4.3 Results and Discussion

The efficient transition between phenotypes could be achieved by controlling flux through a set of metabolic valves, effectively decoupling both phenotypes. By designating a target reaction for each phenotype, MoVE uses a constraint-based metabolic model to identify (1) a set of static knockouts and (2) a set of dynamically controlled valves, to enable the transition between high flux for each of these targets (Figure 4.1a). Static knockouts can be implemented using genome editing, and valves controlled using responsive genetic elements (Nielsen et al.,
to enable, for example, the transition between growth and production states. These knockouts minimally impact the first desired target, but prime the network for the activity of valves. This set of valves can then be used to eliminate undesired fluxes, and enforce a high production yield (Figure 4.1c). For some products, the phenotype can be shifted using process conditions such as pH, temperature, or oxygen availability, which trigger native regulatory systems (Chen and Bailey, 1994, Rajgarhia et al., 2002) (Figure 4.1d); however, these systems may not coincide with engineering objectives such as chemical production. Instead, internal metabolites or inducers can trigger sensors and metabolic controllers to manipulate valves and effect a phenotypic shift (Figure 4.1e). Since these strategies enforce a predefined minimum product yield, adaptive evolution can be effectively applied (Figure B.1).

4.3.1 Core-model strategies

We first apply MoVE to identify strategies to decouple growth from chemical production using a core reconstruction of *E. coli* (Orth et al., 2010). We present a strategy for the production of α-ketoglutarate (AKG), an important intermediate in the tricarboxylic acid (TCA) cycle with uses as a dietary supplement, to illustrate the role of valves and knockouts for redirecting flux, and their impact on the phenotypic space (Figure 4.2). This strategy achieves theoretical maximum production of AKG in the production state. Such core model strategies have recently been applied successfully for the production of itaconic acid using an interactive approach (Harder et al., 2016). A more efficient two-stage strategy can be directly identified by MoVE, and this strategy has been shown to be effective for improving the yield and titer of itaconic acid, in addition to overcoming the need for media supplementation due to auxotrophy (Harder et al., 2017).

4.3.2 Genome-scale strategies in *E. coli*

Next, we applied this algorithm to a genome-scale metabolic reconstruction of *E. coli*, iJO1366 (Orth et al., 2011). Despite the high computational demand of many strain design algorithms applied to genome-scale models, recent algorithmic advancements (von Kamp and Klamt, 2014), modern MILP solvers, and high-performance computing clusters can be used to explore a large
Figure 4.1: Overview of the metabolic valve enumerator. (a) MoVE identifies intervention strategies to decouple growth and production phenotypes using a set of static genetic knockouts and dynamically regulated metabolic valves. Knockouts prime the network to allow usage of these valves to redirect metabolism between states. (b) These conceptual valves can be implemented through genetic engineering, deleting genes providing undesired network functionality and dynamically regulating a set of valves. (c) The steady-state stoichiometric flux space is mapped on two-dimensions to illustrate the trade-off between metabolic targets. The wild-type has a wide range of potential phenotypes, often achieving maximum biomass yield using natural regulatory systems. In the first state, flux is maintained through valves to achieve a high yield of the first target. In the second state, flux through valves and undesired flux vectors are eliminated (red), enforcing a desired phenotype (green). (d) Native regulatory systems are able to sense some environmental perturbations to exhibit controlled phenotypic responses. (e) Synthetic control logic can be implemented using a typical control architecture and applying genetic circuits to sense metabolites and control flux through targeted metabolic valves toward specified objectives.
Figure 4.2: Application of MoVE to a core model for α-ketoglutarate (AKG) production. Line thickness is proportional to pathway flux, metabolites highlighted in blue are biomass precursors. (a) Wild-type flux distribution of the *E. coli* core model. Aerobically grown wild-type *E. coli* is known to partition flux between glycolysis and the pentose phosphate pathway, with a fully active tricarboxylic acid (TCA) cycle and respiratory chain (Zhao et al., 2004), leading to maximal biomass yield and growth rate (b) Flux-distribution with identified knockouts applied. Following knockout of the predicted reactions and associated genes: pyruvate kinase (PYK, *pykAF*), succinyl-CoA synthetase (*SUCCOAS, sucA*), glutamate synthase (*GLUSy, gltB*), and malate dehydrogenase (*MDH, mdh*), flux is partitioned toward the pentose phosphate pathway from glycolysis, and flux through the TCA cycle is redirected through the glyoxylate shunt, marginally impacting the maximum biomass yield. (c) Flux-distribution after eliminating flux through knockouts and valves, achieving maximal AKG yield.
range of target metabolites. We use a distributed MILP algorithm to identify intervention strategies which meet our desired growth and production thresholds, identifying the optimal feasible solution within a fixed computational time for each metabolite (Figure B.2).

We searched for intervention strategies for all 87 organic products that can be derived from glucose in our model (Table B.1). In this case, we focus on natural products; however, this analysis can be trivially extended for the case of non-natural products by including heterologous reactions and exchanges in the model. We investigate strategies for two scenarios: full and partial decoupling. We have proposed that an optimal operating strategy will generally require full decoupling, with a switch from maximum growth to maximum production (Vena-yak et al., 2015); however, substrate uptake rate can decrease in resting cells (Klamt et al., 2017) and the inability to produce biomass precursors could lead to difficulties for sustained production. For these reasons, it may be beneficial to sacrifice product yield to maintain some capacity for cell growth, leading to partially decoupled production. For these simulations, fully decoupled strategies achieve over 90% of theoretical maximum product yield at the expense of cell growth, while partially decoupled strategies achieve over 70% of theoretical maximum yield, while maintaining a minimum biomass yield of 0.01 gdw/mmol (approximately 10% of the maximum biomass yield, allowing a growth rate of 0.1 $h^{-1}$). Both strategies achieve over 90% of theoretical maximum biomass yield in the growth state.

**Fully decoupled**

First, we explore the ability of single valves to redirect flux for full decoupling, obtaining over 90% biomass yield and product yield in their respective states (Figure 4.3a). We identified strategies where controlling single valves could meet the desired flux thresholds for 56 products, or 64% of all targets (Figure 4.3b, Figures B.4-B.5). Three metabolic subsystems were highly represented: glycolysis, the TCA cycle, and oxidative phosphorylation. The top five valves included three from glycolysis: glyceraldehyde-3-phosphate dehydrogenase (GAPD, gapA), pyruvate dehydrogenase (PDH, aceE) and phosphoglucomutase (PGM, pgm); citrate synthase (CS, gltA) from the TCA cycle; and oxygen exchange (EX_o2(e), passive) used in oxidative phosphorylation. These valves were relatively evenly distributed amongst the degree of reaction connectivity, indicating their high representation is not solely owed to their bran-
ched nature (Figure B.3). However, we noted a high-representation of reactions proximal to the twelve precursor metabolites related to the naturally evolved bow-tie (hourglass) topology of metabolism (Friedlander et al., 2015, Zhao et al., 2006), indicating this topology may be important to allow for efficient metabolic transitions.

**Figure 4.3:** Valve occurrences in intervention strategies, categorized by metabolic subsystem. (a) Fully decoupled strategies target over 90% theoretical maximum production without cell growth in the second (production) state. (b) Single valves identified for fully decoupled strategies, single valves can independently redirect flux. (c) Partially decoupled strategies achieve over 70% theoretical yield while maintaining a biomass yield of 0.01 gdw/mmol. (d) Single valves identified for partially decoupled strategies. (e) Central metabolic map highlighting identified valves and associated subsystems. Bolded reactions refer to the top five valves for each decoupling strategy. Bolded metabolites refer to metabolites at the center of the bow-tie architecture of metabolism (Zhao et al., 2006).

**Partially decoupled**

Next, we identify strategies for partial decoupling. These strategies maintain a growth rate of $0.1h^{-1}$ in the production state, and target a more modest 70% of maximum product yield (Figure 4.3c). Experimentally determined essential reactions (Baba et al., 2006) were also blacklisted from being used as valves, corresponding to the goal of allowing a minimum biomass
yield throughout. We identified valves from similar metabolic subsystems for both full and partial decoupling, with a few exceptions (Figure 4.3d, Figures B.6-B.7). First, PGM had a notably higher representation for partial decoupling, likely due to the essentiality of many other glycolytic reactions. In addition, more valves from upper glycolysis and the pentose phosphate pathway are identified. Lastly, α-ketoglutarate dehydrogenase (AKGDH, sucA) is the only single valve identified in the TCA cycle, and connected to AKG, a precursor to amino acids (Figure 4.3e). The requirement to maintain a minimum biomass yield is a strong constraint, requiring the production of several metabolites as biomass precursors; thus, more complex strategies are required to ensure the low production of these metabolites while ensuring high product yield.

Identifying platform valve candidates

With the optimal valves for each product identified, we interrogated whether the top five valves from both decoupling strategies could be used for a broader range of products (Figure 4.4). Although some of these valves could be effective for many products, this trend was not universal. For example, employing oxygen exchange as a valve could decouple fewer than 10 products, including known fermentative products such as acetate, ethanol, lactate and succinate. This strategy has been applied for these products by exploiting the natural switch between high yield aerobic growth and low-yield fermentative growth (coupled to high-yield product synthesis), triggered by oxygen availability and controlled at the process-level. Interestingly, these strategies required relatively few knockouts, indicating that metabolic networks are structured to readily allow for such natural transitions. However, these results indicate that employing oxygen availability as a valve may only be applicable for a small subset of relevant products, motivating the implementation of synthetic genetic circuits to control metabolic flux. Similarly, while GAPD was one of the most frequently identified valves, it could only decouple 11 products.

Contrarily, we have identified valves which could be applied to a majority of tested products. For example, CS was identified as an effective valve to decouple 55 products. It lies at an important branchpoint which has been successfully dynamically controlled to improve the production of isopropanol from acetyl-CoA (Soma et al., 2014). It is an intuitive choice
Figure 4.4: Decoupling potential for top five most frequently identified metabolic valves. Columns correspond to strategies employing specified metabolic valves. Rows correspond to specific products. The color scale represents the number of knockouts in the identified strategy for each product and given valve, black indicates no solution was found. (a) Strategies for fully decoupled production (b) Strategies for partially decoupled production.
for eliminating the main pathway for flux into the TCA cycle, leading to overflow production of desired compounds. We have shown that this valve is applicable to more than 60% of tested metabolites, making it a good candidate for modular platform strains. This reaction is also known to be regulated by global regulators, with reduced flux during anaerobic growth to compensate for increased flux from pyruvate to fermentative products (Park et al., 1994). In addition, two closely related valves: PDH and PGM, were also both effective for a wide range of products. They are central to committing phosphoenolpyruvate or pyruvate to the TCA cycle. PDH is known to be downregulated in anaerobic conditions due to oxygen sensitivity of the lpdA subunit, and its activity replaced by pyruvate formate lyase (PFL, pflAB). These reactions are also important for controlling ATP generation through pyruvate kinase (PYK, pykAF) and NADH generation through PDH, allowing alternate routes for entry into the TCA cycle. These results highlight the importance of the pyruvate node for controlling metabolic flux.

AKGDH, which produces succinyl-CoA from α-ketoglutarate, was uniquely identified as a valve for partial decoupling. It is proximal to a highly regulated branch point for the production of amino acids from AKG, which also has implications in nitrogen metabolism and cofactor balance. We identified AKGDH as a suitable valve for AKG production, as well as glutamate, arginine, and proline which are derived from AKG. Another important valve in amino acid metabolism is phosphoglycerate dehydrogenase (PGCD), the committing step into serine, cysteine, and glycine metabolism from 3-phosphoglycerate (3PG). It is regulated through feedback inhibition by serine, to maintain appropriate concentrations of these amino acids. PGCD and PGM share a common metabolite (3PG) and were both identified as top valves. By controlling this node, flux can either be directed through PGM toward pyruvate and downstream products, or through PGCD to produce amino acids such as serine and cysteine.

**Higher-order valve strategies**

We also identified higher-order strategies which required actuating multiple valves simultaneously. By applying two or three valves, we identified strategies for 64 and 68 products, respectively, compared to 56 products using single valves. This indicates that multiple valves can be required to decouple growth and production for some targets. We also identified
clusters of valves which include reactions from a wide range of different subsystems for both fully (Figure B.8) and partially (Figure B.9) decoupled strategies. Additionally, we have shown that knockouts which were commonly identified amongst all simulations are often found in pyruvate metabolism, to eliminate alternative fermentative byproducts, as well as amino acid metabolism, eliminating alternate routes for flux leakage (Figure B.10).

4.3.3 Genome-scale strategies in *S. cerevisiae*

Finally, we applied MoVE to a genome-scale model of *S. cerevisiae*, a common eukaryotic production host, to assess the method’s effectiveness in a more complex multi-compartment model. Using a similar procedure, we searched for partially decoupled strategies for all 84 metabolites producible from glucose in our model, targeting a minimum biomass yield of 0.001 gdw/mmol (achieving a growth rate of 0.01 $h^{-1}$) and a minimum product yield of 70% of theoretical maximum in the production state. Strategies also targeted over 90% of theoretical maximum growth rate in the growth state. We identified solutions for 61 of the 84 targets using single valves (Supplemental Table B.2). Mitochondrial succinate dehydrogenases (SUCD1m, SUCD2_u6m, SUCD3_u6m) were found for over 20 targets, making this an important target in *S. cerevisiae*. In addition, several valves were identified at important branchpoints, similar to *E. coli* (Supplemental Figure B.11).

4.4 Conclusion

Here, we have developed a method that can be applied to identify metabolic valves to redirect metabolism between phenotypic states, using readily available metabolic models. Using this method, we have shown that decoupling of growth and production phenotypes is possible for a majority of natural products in *E. coli* and *S. cerevisiae*. We have identified strategies to achieve near theoretical maximum product and biomass yield by manipulating three or fewer valves, demonstrating the feasibility of two-stage production strategies for diverse targets. We have made this data set fully available to be used as a guide for metabolic engineering endeavors, or to be used as seeds to identify strategies for related compounds. These strategies can be combined with recent methods for strain design prioritization, based on metrics such
as robustness (Yang et al., 2014), to effectively guide experimental implementation of such
dynamic metabolic engineering strategies. Using this comprehensive data set, we identified
a high proportion of valves in energy metabolism and near important metabolic bottlenecks,
indicating that these bottleneck metabolites are important targets for both natural and syn-
thetic control. Furthermore, we have identified valves which can be applied to a wide range of
products, making them strong candidates for modular platform strains. The location of these
valves highlight important architectural traits of metabolism (Zhao et al., 2006), and provide
insight into important control points.

We anticipate the application of this algorithm will drive the development of dynamically
controlled microbial production hosts and allow the design of more efficient genetic engineering
strategies. Furthermore, given the rapidly growing number of curated genome-scale models,
and the improving ability for metabolic model generation from from -omics data (Machado and
Herrgård, 2014, Magnúsdóttir et al., 2017, Markert and Vazquez, 2015), MoVE could be applied
to elucidate important natural regulatory branchpoints in diverse metabolic systems. This will
include more complex microbial and multicellular organisms, such as mammalian systems,
where extremely complex regulatory networks exist (Schwanhäusser et al., 2011, Thiery and
Sleeman, 2006).

4.5 Materials and Methods

4.5.1 Stoichiometric metabolic models

Stoichiometric metabolic models are defined by the reactions present in a given organism,
based on genome sequences and experimental validation. Central to this metabolic model is
the stoichiometric matrix $N$, with $m$ rows representing metabolites and $n$ columns representing
reactions. Steady state is assumed in constraint-based models, demanding that there is no net
accumulation or consumption of internal metabolites:

$$N \cdot r = 0$$ (4.1)
where \( \mathbf{r} \) is the steady-state flux vector. The network is often further constrained by setting (known) flux bounds for certain reactions, \( i \), to define upper bounds on uptake rates (e.g. glucose or oxygen), or fix parameters such as ATP maintenance:

\[
\alpha_i \leq r_i \leq \beta_i
\]

These flux bounds also include flux directionality constraints for irreversible reactions:

\[
r_i \geq 0 \quad \forall \ i \in \text{Irr}
\]

We assume here that all fluxes in the network are explicitly or implicitly bounded by these constraints (Equations 4.1-4.3)

### 4.5.2 Two-state problem formulation

The metabolic valve enumerator (MoVE) algorithm aims to find a minimal set of knockouts and a set of dynamically regulated valves to allow switching between two distinct metabolic phenotypes. Here, we have applied this algorithm to achieve efficient switching between two relevant phenotypes for two-stage bioproduction processes: growth and production. This is accomplished by formulating a mixed-integer linear program, which can then be solved using a range of commercial or open-source solvers. Diverse optimization problems, including those relying on stoichiometric models, have been solved in this fashion.

To consider multiple phenotypes in a dynamic context, we require the specification of flux vectors for both of these states and variables describing both static (knockout) and dynamic (valve) interventions. Furthermore, this method must remain scalable to genome-scale models, given these additional variables and constraints. In MoVE, \( \mathbf{r} \) denotes the production state (state 2) flux vector and a second flux vector, \( \mathbf{f} \), is introduced to represent the growth state (state 1), subject to similar flux bounds: \( \gamma_i \leq f_i \leq \delta_i \), and steady-state constraints: \( \mathbf{N} \cdot \mathbf{f} = \mathbf{0} \).

We introduce the parameters \( \mathbf{T} \ (t \times n) \) and \( \mathbf{t} \ (t \times 1) \) to formulate linear inequality constraints
for undesired flux vectors in the production state (state 2, e.g. low product yield, Figure 4.1c):

$$\mathbf{T} \cdot \mathbf{r} \leq \mathbf{t} \quad (4.4)$$

For convenience, these inequalities can be formulated to eliminate flux vectors below a minimum yield threshold ($Y_{P/S}^{P/S}_{min, state2}$), where $r_P, r_S, r_B$ represent the production or consumption rates of product, substrate and biomass, respectively:

$$\frac{r_P}{r_S} \leq Y_{P/S}^{P/S}_{min, state2} \iff r_P - Y_{P/S}^{P/S}_{min, state2} \cdot r_S \leq 0 \quad (4.5)$$

Hence, the matrix $\mathbf{T}$ has a single row of zeros derived from Equation 4.5, except for a ‘+1’ in the column for the product reaction, and ‘$-Y_{P/S}^{P/S}_{min, state2}$’ in the column for the substrate reaction. The vector $\mathbf{t}$ accordingly contains only one element, ‘0’.

Similarly, the parameters $\mathbf{D}$ ($d \times n$) and $\mathbf{d}$ ($d \times 1$) impose constraints for the desired flux vectors in the production state (state 2, e.g. high product yield, Figure 4.1c):

$$\mathbf{D} \cdot \mathbf{r} \leq \mathbf{d} \quad (4.6)$$

Again, these constraints are formulated to describe desired product yield:

$$\frac{r_P}{r_S} \geq Y_{P/S}^{P/S}_{min, state2} \iff Y_{P/S}^{P/S}_{min, state2} \cdot r_S - r_P \leq 0 \quad (4.7)$$

and biomass yield, in the case of partial decoupling (where some minimum growth rate, $r_B$, is maintained in the production state):

$$\frac{r_B}{r_S} \geq Y_{B/S}^{B/S}_{min, state2} \iff Y_{B/S}^{B/S}_{min, state2} \cdot r_S - r_B \leq 0 \quad (4.8)$$

In this case, $\mathbf{D}$ consists of two rows derived from Equations 4.7-4.8: the first contains zeros except for a ‘-1’ in the column for $r_P$ and ‘$Y_{P/S}^{P/S}_{min, state2}$’ in the column for $r_S$, the second contains non-zero values only for the $r_B$ (-1) and again for the substrate uptake rate $r_S$ ($Y_{B/S}^{B/S}_{min, state2}$). Accordingly, the vector $\mathbf{d}$ is of size 2, and contains ‘0’ elements.
In addition, we also introduce the parameters $\mathbf{G}$ ($g \times n$) and $\mathbf{g}$ ($g \times 1$) to represent desired phenotypes in the growth state (state 1, e.g. high biomass yield, Figure 4.1c), which now relies on the growth state flux vector, $\mathbf{f}$:

$$\mathbf{G} \cdot \mathbf{f} \leq \mathbf{g} \quad (4.9)$$

and is again formulated for yield constraints:

$$\frac{f_B}{f_S} \geq Y_{\min, state1}^{B/S} \Leftrightarrow Y_{\min, state1}^{B/S} \cdot f_S - r_B \leq 0 \quad (4.10)$$

Furthermore, constraints are added to describe desired flux vectors in the production (Equation 4.6) and growth (Equation 4.9) states, to ensure ATP maintenance ($r_{\text{ATPM}} \geq \text{ATPM}_{\min}$ and $f_{\text{ATPM}} \geq \text{ATPM}_{\min}$) and substrate uptake rates ($r_S \leq r_{S,\text{max}}$ and $f_S \leq f_{S,\text{max}}$).

MoVE applies the constraints defining the undesired (Equation 4.4) and desired (Equations 4.6, 4.9) flux spaces, to identify valves and knockouts.

### 4.5.3 Algorithm

To identify interventions allowing an efficient switch between growth and production states, MoVE applies the concept of minimal cut sets (MCS) (Klamt and Gilles, 2004), a minimal set of knockouts to eliminate undesired functionality. For computation of MCS, the primal problem described above (Equations 4.1-4.4) is transformed into its dual (Ballerstein et al., 2012, de Figueiredo et al., 2009, von Kamp and Klamt, 2014, 2017) and constraints for desired functionality (Equations 4.6, 4.9) are applied. Finally, an objective function is used to find solutions requiring a minimal number of interventions ($z_i$). The full formulation of the MoVE optimization problem thus reads:
minimize $\sum z_i$

s.t.

$$\begin{pmatrix}
N^{T}_{Irr} & I_{Irr} & 0 & 0 & T^{T}_{Irr} & 0 & 0 \\
N^{T}_{Rev} & 0 & I_{Rev} & -I_{Rev} & T^{T}_{Rev} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & N & 0 \\
0 & 0 & 0 & 0 & 0 & D & 0 \\
0 & 0 & 0 & 0 & 0 & N & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & G \\
\end{pmatrix}
\begin{pmatrix}
u \\
v_{P_{Irr}} \\
v_{P_{Rev}} \\
w \\
r \\
f \\
\end{pmatrix}
\begin{pmatrix}
\geq \\
= \\
= \\
\leq \\
= \\
\leq \\
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
0 \\
d \\
0 \\
g \\
\end{pmatrix}$$

$$t^T w \leq -c$$

$u \in \mathbb{R}^m$

$r, f \in \mathbb{R}^n$

$w \in \mathbb{R}^t$

$\alpha, \beta, \gamma, \delta \in \mathbb{R}^n$

$zp_i, zn_i, y_i \in \{0, 1\}$

$\forall i \in Rev: z_i = zp_i + zn_i, z_i \leq 1$

$\forall i \in Irr: z_i = zp_i$

$v_{P_{Irr}}, v_{P_{Rev}}, v_{N_{Rev}}, w \geq 0$

$c > 0$

$r_i \geq (1 - z_i) \cdot \alpha_i$

$r_i \leq (1 - z_i) \cdot \beta_i$

$y_i \geq (1 - z_i)$

$f_i \leq y_i \cdot \delta_i$

$f_i \geq y_i \cdot \gamma_i$

$\sum_i y_i - (1 - z_i) \leq max_values$

To enable efficient calculation of MCS, new dual variables are introduced, $u, w, v_{P_{Irr}}, v_{P_{Rev}}, v_{N_{Rev}}$ and the production state variables are further separated into reversible and irreversible com-
ponents. Following, the stoichiometric matrix $\mathbf{N}$, the identity matrix $\mathbf{I}$, and the undesired flux matrix $\mathbf{T}$ are split into two submatrices containing the reversible ($\mathbf{N}_{\text{Rev}}, \mathbf{I}_{\text{Rev}}, \mathbf{T}_{\text{Rev}}$) and irreversible ($\mathbf{N}_{\text{Irr}}, \mathbf{I}_{\text{Irr}}, \mathbf{T}_{\text{Irr}}$) reactions (columns).

This makes it possible to use Boolean indicator variables $z p_i = 0 \Leftrightarrow v p_i = 0$, $z p_i = 1 \Leftrightarrow v p_i \neq 0$ for all reactions, and additionally $z n_i = 0 \Leftrightarrow v n_i = 0$, $z n_i = 1 \Leftrightarrow v n_i \neq 0$ for reversible reactions. If the value of an indicator variable is 1, then its associated reaction is in the cut set and can carry no flux as demanded by the constraints for $r_i$.

Although identified MCSs will successfully eliminate undesired functionality (Equation 4.4), these MCSs do not guarantee any desired functionality will remain feasible. To do so, the additional constraints for the production (Equation 4.6) and growth (Equation 4.9) states are applied. In addition, new constraints are added to ensure that metabolic valve reactions are a subset of the reaction knockouts (i.e. flux through valves is ON in the growth state, and OFF in the production state):

$$y_i \geq (1 - z_i)$$ (4.12)

and to limit the number of possible valves:

$$\sum_i y_i - (1 - z_i) \leq \text{max\_valves}$$ (4.13)

The Boolean variables $y_i$ thus indicate whether a reaction can carry flux ($y_i = 1$) or not ($y_i = 0$) in the growth state. Up to max\_valves reactions of a MCS (as determined by the values of the $z_i$ variables) are allowed to carry flux in the growth state. Hence, the valve reactions are those for which $y_i = 1$ and $z_i = 1$. The reactions for which $y_i = 0$ and $z_i = 1$ are static knockouts and are disabled in both production and growth state, whereas all other reactions are available in both states.

Finally, these variables and constraints are combined to allow the direct identification of optimal combinations of valves and knockouts (with a minimal number of interventions). It is important to note that since any feasible solution will achieve desired functionality (in both states) and eliminate undesired functionality, solving the algorithm to optimality is not absolutely
essential.

### 4.5.4 Implementation

Simulations were performed in MATLAB 2010b using the COBRA toolbox (Schellenberger et al., 2011) and CellNetAnalyzer v2017.4 (CNA) (von Kamp et al., 2017). Mixed-integer linear programs were solved using ILOG CPLEX (IBM, v12.6), via the provided Java virtual machine interface in CNA.

Genome-scale *E. coli* simulations were performed on the SciNet general purpose cluster (Baldwin et al., 2010). The cluster is composed of 3864 nodes using Infiniband interconnect. Each node contains 2x Intel Xeon E5540 processors for a total of 8 cores or 16 threads per node, with 16GB of RAM. Simulations were performed in parallel for each metabolite and set of parameters. Each simulation was performed on 4 nodes, using 16 threads per node, for 8 hours. The MILP was solved in two steps, ramp-up and distributed tree search. In the ramp-up phase, the same problem is solved on each node using different startup parameters for two hours. Following the ramp-up phase, the optimal startup parameters are used to start a distributed search tree for the remaining 6 hours. The optimal feasible solution from this process is returned.

Genome-scale *S. cerevisiae* solutions were solved for two hours using 4x Intel Xeon CPU E7-4830 processors, for a total of 32 cores.

### 4.5.5 Model and MILP parameters

The *E. coli* core model (Orth et al., 2010) was derived from iAF1260 (Feist et al., 2007) and is available from the BiGG database (King et al., 2016) (http://bigg.ucsd.edu/models/e.coli_core). Maximum glucose uptake rate was set at 10 mmol gdw$^{-1}$ h$^{-1}$ and minimum ATP maintenance at $-8.39$ mmol gdw$^{-1}$ h$^{-1}$.

The *E. coli* genome-scale model iJO1366 (Orth et al., 2011) was used for all genome-scale simulations and available from the BiGG database (http://bigg.ucsd.edu/models/iJO1366). Maximum glucose uptake rate was set at 10 mmol gdw$^{-1}$ h$^{-1}$ and minimum ATP maintenance at 3.15 mmol gdw$^{-1}$ h$^{-1}$. Target reactions were chosen from the total set of export reactions by eliminating non-organic molecules and those which were not producible from glucose based on
a flux variability analysis. A set of exchange reactions, excluding the target reaction, common fermentation products, and non-organic molecules, was removed from the model to improve computational feasibility; this list is provided in Supplemental Information.

Strategies for the *E. coli* core model were solved to optimality using an upper bound of 3 valves in negligible computational time. Strategies for the genome-scale model were solved using the distributed MILP search method with an equality constraint on the number of valves. Searches explicitly specified one, two, or three valves. To identify the optimal valve for each metabolite, all reactions were allowed to be used as valves for fully coupled strategies, and only non-essential valves for partially coupled strategies. To identify valves which could be applied to many products, we ran independent optimizations with each valve explicitly specified (no other valve was allowed to be identified as a valve).

Strategies for the *S. cerevisiae* genome-scale model were identified using iMM904 (Mo et al., 2009) which is available from the BiGG database (http://bigg.ucsd.edu/models/iMM904). Maximum glucose uptake rate was set at 10 mmol gdw$^{-1}$ h$^{-1}$ and minimum ATP maintenance at 1 mmol gdw$^{-1}$ h$^{-1}$. A set of exchange reactions, excluding the target reaction and reactions required for wild-type growth, was removed from the model to improve computational feasibility; this list is provided in Supplemental Information.

### 4.5.6 Reaction connectivity

Reaction connectivity was determined in Python 3.5 using the COBRApy package (Ebrahim et al., 2013). Connectivity for each reaction was determined as the sum of the connectivities of all metabolites involved in that reaction. The connectivity of each metabolite is determined as the number of reactions in which it partakes.

### 4.5.7 Clustering of higher-order valve strategies

Intervention strategies were determined using one, two, or three metabolic valves. Higher-order valves which improved the objective (e.g. a solution was only found with a larger number of valves, or the required number of knockouts was decreased) were used to generate an adjacency matrix for clustering, using the co-occurrence frequency as a similarity metric. The resulting sparsely connected matrix was clustered using an iterative spectral clustering approach. Several
iterations of spectral clustering were performed using the scikit-learn package (v0.19.0). Following, a new similarity matrix was generated based on the mutual information available between clustering solutions to identify the most represented solution, this clustering result is returned. The numbers of clusters was chosen to ensure clusters contained at least two members.

4.6 Acknowledgments

We acknowledge Daniel Tomchynsyn and Dean Robson for support in deploying this algorithm, as well as Compute Canada and SciNet for providing compute resources. We would also like to thank Elad Noor for insightful discussions. This work was supported by the Natural Sciences and Engineering Research Council (NSERC), the NSERC Industrial Biocatalysis Network, BioFuelNet Canada, the Ontario Ministry of Research and Innovation, the Alexander von Humboldt foundation, the German Federal Ministry of Education and Research (FKZ: 031L104B) and the European Research Council (ERC Consolidator Grant 721176).
Chapter 5

Impact: a python framework for data analysis workflows to interpret microbial physiology

A version of this chapter is in preparation to be submitted for publication:


Contributions NV conceived and implemented the framework, performed all simulations, and wrote the manuscript. RM discussed and reviewed the manuscript.

5.1 Abstract

Microorganisms can be genetically engineered to solve a range of challenges in areas such as health and sustainability. The natural complexity of biological systems makes this an iterative design-build-test process, perturbing metabolism and making stepwise progress toward a desired phenotype. This process has been rapidly accelerated by robust DNA synthesis and assembly techniques, liquid handling automation, and scale-down characterization platforms, generating large heterogeneous data sets. Here, we present an extensible python framework to interpret, model, and visualize these data sets: Impact (Integrated microbial physiology:
analysis, characterization, translation). Impact aims to ease the development of python-based data analysis workflows for a range of stakeholders in the bioengineering process, offering tools for data analysis, physiology characterization and translation to visualization. Using this framework, biologists and engineers can opt for reproducible and extensible programmatic data analysis workflows, accelerating the throughput of microbial engineering.

5.2 Introduction

Microorganisms serve important roles in diverse areas including health and sustainability. Modern tools in biotechnology have accelerated the characterization and engineering of microbes to face these new challenges. In the past two decades, there have been significant advancements in the field of systems biology, to rapidly characterize and develop models for organisms of interest (King et al., 2015, Metzker, 2009), and in the field of synthetic biology, to design and synthesize biological constructs (Cameron et al., 2014, Hillson et al., 2012). Simultaneously, the throughput of the laboratory has been significantly increased, owing to advanced analytics and automation. These advancements have drastically improved our understanding and ability to engineer biology to solve new challenges.

From cell culture experiments, a microbe’s physiology and metabolic state can be assessed, often studied in batch, semi-batch, or chemostat culture. To engineer these microbes, their metabolism is perturbed based on metabolic hypotheses to be tested. This process is generally iterative, composed of four main stages: design, build, test and learn (DBTL). Although our understanding of microbes is rapidly improving, reaching desired microbial performance requires many cycles through this loop. Metabolic engineering continues to strive for modular and predictable designs common to other engineering disciplines (Nielsen et al., 2016, Olson et al., 2014, Salis et al., 2009), but complexity of metabolism imposes significant challenges.

To overcome this lack of predictability, large libraries of strains can be developed and characterized. To improve the throughput, fermentations have been scaled-down to microtiter plate and even to the droplet scale (Wang et al., 2014), generating significant amounts of features (e.g. analytes) to understand the metabolism of the microorganism(s) involved. The complexity of acquiring key data types can vary significantly, especially at different fermentation scales;
accordingly the measured data types can vary. At the bioreactor scale, on-line pH, dissolved oxygen, and feed additions are commonly monitored. Modern methods also exist to monitor these features at the microplate scale (Unthan et al., 2015), although they generally require specialized equipment. In most experiments, the composition of the fermentation medium can be sampled if there is sufficient volume and analyzed by chromatography, the gas phase can be sampled or monitored continuously using a process mass spectrometer and the biomass concentration can be monitored using optical density. This type of data can be synthesized into some key performance indicators, such as titer, rate and yield (TRY) for measured components, all of which are important to understanding the physiology of the microbe.

Given the quantities of biological data being generated, there has been widespread standardization of data formats, databases and analysis tools (Galdzicki et al., 2014, King et al., 2016, NCBI Resource Coordinators, 2016). Biological systems are often characterized using -omics technologies, including genomics, proteomics, metabolomics and fluxomics. The fluxome is generally determined using fermentations with radiolabelled glucose and mass spectrometry data to estimate fluxes using a least squared algorithm (Quek et al., 2009, Tang et al., 2009, Zamboni et al., 2009); however, this can require complex setups and expensive medium. The predictive power of these constraint based metabolic models has been improving, and they can often be used to predict complex phenotypes (Bordbar et al., 2014). In lieu of the detail required to determine the fluxome, on-line and off-line fermentation features can be used to estimate microbial exchange fluxes. Then, the model can be constrained using these rates and an objective function can be used to predict an internal flux distribution.

Typically, fermentation data is handled in spreadsheet applications using custom data processing templates. This process is cumbersome, non-transparent and it does not lend itself to facile data sharing. Here we present Impact (integrated microbial physiology: analysis, characterization, translation), an integrated framework for analyzing microbial physiology. The Impact framework aims to aid scientists and engineers analyze, characterize and translate raw data describing microbial physiology. To do so, Impact relies on a standard metadata schema to describe experiments, and uses this data to parse it into a logical hierarchical format. From here, features are extracted to provide an augmented view of this data. Finally, this organized data structure can be queried for plotting, or downstream analyses.
Impact is modularly designed to promote contribution from a range of stakeholders in the bioengineering pipeline. To do so, Impact relies on a number of open-source packages, to keep the code base small and agile. Thus, Impact is not aimed to be a “black-box” solution for data analysis, but rather promote contributions to the data warehousing and analysis pipeline in order to arrive to a community-driven consensus on data analysis best practices. Through standards in the software development life cycle (e.g. testing and continuous integration), contributions to the framework remain robust.

5.3 Results and Discussion

5.3.1 Design, build, test, learn

Metabolic engineering typically proceeds iteratively through a design-built-test-learn cycle (Figure 5.1). Impact is a framework to accelerate the learning process, by automating the analysis of raw data. Thus, Impact requires raw quantified analyte data as input, which can be directly parsed from analytical equipment or a laboratory information management system (LIMS).

In brief, quantified raw data is extracted from analytical equipment without significant curation and saved into a spreadsheet (typically .xlsx). These spreadsheets can then be parsed by Impact into the data schema (Fig. 5.6). Using this data schema, features can be extracted and finally plotted as needed. This process is divided into four stages:

1. Analyze: The process of parsing raw data into the data schema

2. Characterize: The process of extracting features or parameters from the data either directly or using a model.

3. Translate: The process of generating visualization or extracting insight from raw data and extracted features.

4. Store & share: The process of saving data for later query by the initial user or others.
5.3.2 Analyzing raw data

Trial identifier (metadata)

Impact divides the trial identifier into three distinct components: (1) the strain, (2) the media, and (3) the environment. The strain describes the organism used and any genetic engineering. To do so, strains are described using a parent genus and species (the wild-type), associated knockouts, and plasmids. Each strain will of course behave differently depending on the medium used, and thus we describe the formulation of each media and associate that to a given dataset. Finally, we describe the environment in which this strain was grown, including temperature, shaking speed and the labware used (microplate, flask, bioreactor, etc.)

Most effectively, this identifier should be used to label samples on primary analytical equipment (e.g. HPLC injections). Then, this data can be directly parsed by Impact and raw data remaining in analytical software can be referenced. In this case, the identifier should be provided in a flat format, for example: `strain:MG1655|media:M9|strain_plasmid:pTrc99a`. The
The entire trial can be described here, or identifiers can be defined externally and referenced by shorthand name.

**Parsing raw data into Impact schema**

Parsing is an essential part of data interpretation, organizing raw data into logical elements which can be queried. To do so, Impact makes a clear distinction between two elements: the identifier, which is responsible for metadata, and the data itself, describing the sample time and concentration. Impact uses metadata to sort raw data into a hierarchical structure based on a typical experimental setup and data analysis workflow (Figure 5.6).

The data structure is built around trials, which can be considered as an independent bioreactor, flask or well in a plate. Each of these trials can be composed of multiple analytes, which in turn are built from raw time and data vectors. Replicate single trials are combined to form a replicate trial, which can then extract statistical information. This organization process is handled solely by the parser, and the user typically will not need to modify this process except to handle new raw data formats.

**Data formats**

Data describing microbial phenotypes are varied, depending on the particular attributes of interest for a given project. Here we provide parsers for two common pieces of equipment: HPLC and plate reader. The HPLC is a workhorse in metabolic engineering, providing the external concentration of diverse metabolites which can be readily parsed by Impact (Table C.1). The plate reader is commonly used to measure growth profiles via optical density, or specific aspects of metabolism using fluorescent probes, and we provide parsers for data from SoftMax Pro (Table C.2) and more typical readers (Table C.3).

Depending on the complexity of the data set, parsers are generally simple to write and register to the framework. Raw data is parsed into a basal data type (e.g. a time point), and then standard parsing functions organize the data into the appropriate data structures. The process of creating a new parser is described in detail in the documentation.
5.3.3 Characterizing physiological features

Impact provides a set of core features, which are derived from the raw quantified data. These features are extracted at different stages, corresponding to the data required. A set of features is provided in the current version, and new features can be created as needed and contributed back to the package.

**Analyte features**

Analyte features are those which can be calculated with only one analyte.

Although state metrics, such as yield and titer, are often used to describe microbial phenotypes, dynamic metrics such as rates and specific productivities can provide further insight. These rates are determined numerically, for each analyte, and used for subsequent calculations. Alternatively, rates can be extracted via parameter fitting.

Typically microbial growth kinetics are described as exponential: \( \frac{dX}{dt} = \mu X \Leftrightarrow X = X_0 e^{\mu t} \). However, since microbial growth is oftentimes characterized through lag, log and death phases, high order models may be relevant. Any model can be added for analysis, and some common models such as the 5-parameter Richard Curve or a generalized logistic function are included.

**Trial features**

Trial features are calculated from a single fermentation volume, but require several analytes. These features are widespread, and examples include product yield (which requires a product and substrate), and OD normalization or specific productivity (which require biomass and an additional analyte).

The specific productivity is the unit most often used in constraint-based models, it is defined as the rate of product export per unit of biomass, usually with the following units \( \text{mmol g dw h}^{-1} \) (in the case of growth rate).

**Replicate features**

Biological replicates are typical to ensure consistency in data and conclusions. Thus, if multiple replicates are available, they are combined to calculate statistics. If no replicates are available,
the mean is calculated from a single replicate. With small data sets, outlier experiments can be easily identified and excluded. However, with large quantities of data this becomes challenging and can drastically affect data interpretation. Thus, outliers can be detected as replicates which deviate significantly from others, and be excluded from analysis. Parameters can be chosen to control the aggressiveness of this process.

**Experiment features**

Experiment features are those which are relevant to many independent trials, such as the different stages in a production batch, or blanks. Oftentimes, there can be a benefit of dividing a fermentation into distinct stages, while optimizing different process parameters at each (Brockman and Prather, 2015b, Cress et al., 2015, Soma et al., 2014, Venayak et al., 2015). Analyzing each of these stages independently is often necessary to extract more relevant metrics. For example, product yields can be more insightful when only considering the production stage.

Blank subtraction is a common process to eliminate background signal for more accurate quantification and can automatically be performed. Typically, specific trials which do not contain cells can be used as blanks, and these trials can automatically be assigned to non-blank trials for subtraction.

**5.3.4 Translating and interpreting data**

**Visualization**

By default, visualization are generated by defining subplots per analyte and plotting the average data for each strain. Using this format, figures ready for interpretation can easily be generated. In lieu of the provided options, figures can be generate directly using any number of python plotting packages (Figure 5.2).

In addition, this data can be used to constrain a metabolic model and plot fluxes using other packages such as Escher, a python based flux visualization package (Figure 5.3)
Data Integrity

In addition to data analysis to extract features, data integrity can be play an important role to identify erroneous analytics or experimental design.

**Mass balance** A carbon balance or mass balance can be important to ensure all significant analytes are accounted for, including gaseous and liquid phases. Oftentimes gaseous products are not analyzed, in which case CO₂ can be estimated using an integrated metabolic model constrained using known production and consumption rates (Figure 5.4).

**Missing data**

Missing data is a significant challenge for many data analysis tasks. Missing data can be relatively common due to experimental or equipment error. To overcome this limitation, we take advantage of the pandas framework, where data input with 'nan' values will not be included in statistical calculations and data sets with different time indices are handled seamlessly (Table 5.1).

**Metabolic model integration** Using the aforementioned calculated specific productivities, basic integration with metabolic models is relatively straightforward. A purely stoichiometric metabolic model can be solved by constraining only the substrate uptake rate, and using an objective function to estimate the flux distribution. With experimental data, we can follow a similar approach but add additional constraints for all measured analytes and estimate internal...
Figure 5.3: Sample data visualization generated with Escher with data from Impact.

Figure 5.4: Carbon balance using an anaerobic *E. coli* simulation with iJO1366.
Table 5.1: Example of missing data handling included with Impact through the pandas package.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicate #1</td>
<td>✓ 0.5</td>
<td>✓ 0.8</td>
<td>✓ 0.9</td>
<td>✓ 1.2</td>
</tr>
<tr>
<td>replicate #2</td>
<td>✓ 0.6</td>
<td>✗ nan</td>
<td>✓ 1.0</td>
<td>✗ nan</td>
</tr>
<tr>
<td>replicate #3</td>
<td>✓ 0.4</td>
<td>✓ 0.7</td>
<td>✓ 1.2</td>
<td>✗ nan</td>
</tr>
<tr>
<td>average</td>
<td>0.5</td>
<td>0.75</td>
<td>1.03</td>
<td>1.2</td>
</tr>
<tr>
<td>standard deviation</td>
<td>0.1</td>
<td>0.07</td>
<td>0.15</td>
<td>n/a</td>
</tr>
</tbody>
</table>

fluxes for additional insight into metabolism.

5.3.5 Data storage and retrieval

The object relational mapping

Although organization of raw data into a logical structure eases data interpretation and visualization, sharing and collaboration are still limited. Since the process of biological engineering is highly collaborative and relies on data from a large number of scientists and engineers, this is of paramount importance. Impact aims to store all this data in a standardized database, so it can be shared and queried.

Relational databases have become the workhorse for most data storage tasks since they are highly structured and can be efficiently queried. To avoid the complexity and domain knowledge associated with such data structures, Impact is built on top of an object relational mapping (ORM) framework, SQLalchemy. This allows developers to focus on adding new functionality using a familiar programming language (python) and programming paradigm (object oriented programming), while still benefiting from a relational database.

Narratives and the Jupyter Notebook

The rate of data collection and the complexity of data analyses continue to increase. Narratives allow scientists to disseminate complex information by providing context and guiding their audience through the experiment, its setup and the relevant conclusions. Jupyter notebooks provide the perfect platform for this type of narrative in a programming context and as such is receiving significant interest (Shen, 2014). In addition to writing and sharing Jupyter note-
books locally, significant progress has been made to deploy this platform on the cloud, allowing scientists and engineers to begin writing narratives without any local installation (KBase - http://kbase.us, JupyterHub - https://github.com/jupyterhub/jupyterhub).

5.4 Conclusion

The increasing application of microbial solutions to diverse challenges has drastically accelerated the rate of data generation. Engineering these microorganisms is highly iterative through the design-build-test-learn cycle, and evolving tools are continuing to reduce the cycle time. Here, we present Impact, a python framework aimed to unify stakeholders in the bioengineering process and provide a set of tools to characterize microbial physiology using programmatic workflows. The architecture of Impact is aimed to make community contribution simple and make complex data workflows more transparent and shareable. We expect that the application of such tools following software development best practices (Yurkovich et al., 2017) to continue to develop and accelerate the process of microbial engineering.

5.5 Methods

5.5.1 Architecture

Impact aims to provide a high-level interface to analyzing data for microbial physiology. This process often involves a range of stakeholders with varying level of coding experience, from biologists to software developers. The complex and iterative nature of biological engineering necessitates communication between many of these stakeholders, accelerated by easy access to underlying datasets. As opposed to many other software packages, Impact does not aim to increase segregation between bench scientists and developers, but rather offer a bridge to programmatic data analysis for all stakeholders (Figure 5.5).

This concept has motivated a majority of design decisions in the Impact framework. First, the most important design decision was the choice of programming language: python. This choice was clear owing to the popularity of python amongst developers and data scientists (https://insights.stackoverflow.com/survey/2018/#technology). In addition, a signifi-
A description of the software tool stack describing the software tools between users and data.
5.5.2 Data structure

Models

Being written entirely in python, Impact is implemented using object oriented programming (OOP). Using inheritance, native data structures can be extended to including new analytes, features, or analysis methodologies. Furthermore, this structure allows the convenient application of an object-relational mapping (ORM) which can translate these python data structures into a range or relational database for facile storing and query.

Schema

The data schema of Impact is based on the logical structure of experimental design. Typically, an experiment is proposed to determine the validity of a hypothesis and this experiment will consist of a number of independent trials. Each of these trials will have a set of analytes of interest (e.g. substrate, product, reporter) and may be performed in replicates.

- **Trial identifier** Every datum has an associated metadata component. Detailed entry of this metadata is important to ensure that data is parsed correctly, and can then be stored and queried efficiently. New formats for trial identifiers can be created to allow more flexibility for parsing from different equipment. Currently, the trial identifier is a flat string which can be used to identify quantification events (e.g. hplc injections) directly on analytical equipment.

- **Time point, time course, and analyte data** The most basic form of analyte data is a time point, each analyte can have one or many time points, depending on the setup of the experiment. If several time points are present, the time points are built into a time course, which can now extract temporal features. In addition, each analyte type (e.g. substrate, product, reporter) inherits from this base time course class in order to implement different models to fit this data and extract unique parameters.

- **Single trial** A single trial is considered as an independent fermentation volume (e.g. a flask, or well in a plate), and thus consists of all of the analytes of interest from one trial.
• **Replicate trial** A replicate trial contains one or more single trials which are considered as replicates, and can be used to extract features from these replicates such as mean and standard deviation, as well as check for consistency between replicates.

• **Experiment** Finally, an experiment contains one or more single trials in order to group relevant experiments together. A replicate trial can belong to multiple experiments, in case one trial can be used to answer multiple hypotheses.

### Object-relational mapping

A number of python-based object-relational mapping packages exist, but the two most popular packages are django (https://www.djangoproject.com/) and SQLAlchemy (https://www.sqlalchemy.org/). Both packages provide similar functionality, although they differ in their implementation. Although the simplicity of django can be powerful, the use of this ORM outside of a backend web application can be cumbersome. Instead, Impact relies on SQLAlchemy because of its pythonic syntax and portability in a range of python environments.

#### 5.5.3 Features

From the raw data stored in the hierarchical data schema, unique features can be extracted. As data moves up the data hierarchy, more complex features can be extracted. Analyte features are those which can be calculated with a single data vector, such as model-fitted parameters or numerically determined rates (gradients). Trial features are those which require several analytes, such as specific productivity or product yield. Replicate features are those which require multiple replicates, such as mean and standard deviation. Experiment features are typically relevant for all trials performed at a given time, such as relevant blanks or specific fermentation stages. The list of features will continue to be expanded, and new features can be added by creating, registering and committing them to the package (see documentation).

#### 5.5.4 Availability, continuous integration, and contributing

Impact is written and tested for python ≥ 3.5, but could be made compatible with older version with some refactoring.
The framework is open-source under the GPL v3 license and available on github (https://github.com/nvenayak/impact). The documentation is available on readthedocs (http://impact.readthedocs.io).

The github repository has hooks connected to Travis CI (https://travis-ci.org/nvenayak/impact) to automatically run tests when new commits are merged with the repository, all new contributions should include relevant tests. The repository is also connected to codecov, to ensure that a majority of the code base is tested (https://codecov.io/gh/nvenayak/impact).

5.6 Acknowledgements

We would like to acknowledge Weijun Gao and Dan Tomchyszyn for help with server deployment, and Kayla Nemr for providing test data and testing.
Figure 5.6: Overview of the data flow, from raw data to visualization, in Impact.
Chapter 6

Conclusions, Recommendations and Outlook

Although microorganisms have the potential to solve numerous challenges, these systems are extremely complex and remain poorly understood. Thus, engineering microorganisms is an iterative process through the design-build-test-learn cycle, where hypotheses are constantly refined. To reach stringent targets for these microorganisms, improved tools are required to reduce the number of iterations, and decrease the time for each. Here, we explore the potential of two-stage biochemical production processes, which can improve process rates and ease commercialization of these technologies. By decoupling the growth and production of microorganisms, each stage can be optimized independently. However, these more complex systems require more detailed characterization in a much larger search space (Figure 6.1).

6.1 Conclusions

Two-stage production processes can significantly improve bioprocess production rates (Chapter 2) Implementing two-stage production processes often requires significantly more complex genetic circuits. To understand the potential of this production strategy, we develop a framework which uses a metabolic model to determine the stoichiometric bounds of metabolism and thus the trade-off between biomass and product yield. In addition, we use an optimization function to identify an optimal switching time to maximize process rate. Using
Chapter 6. Conclusions, Recommendations and Outlook

**Figure 6.1:** Synthesis of contributions to the design-build-test-learn cycle used to metabolically engineer microbes for two-stage production processes.

This framework, we have shown that two-stage production processes are beneficial for nearly all chemicals. These strategies can be implemented using a range of sensors and actuators available in the synthetic biology literature.

**Bistable genetic controllers are effective for two-stage production (Chapter 3)** Given the importance of two-stage production processes, we aimed to implement a generic genetic controller. We iteratively optimized a bistable phenotypic controller to decouple growth and production processes in *E. coli* and applied this switch to demonstrate a two-stage production process to improve process rates by 20%. This bistable circuit allows for propagating cells from small lab-scale cultures to large industrial fermenters without inducer, and can allow for state switching using transient induction such as a period of nutrient starvation or a pulse of light.

**Adaptive laboratory evolution can limit the potential to restore a growth phenotype in genetically engineered organisms. (Chapter 3)** Using our bistable controller, we were able to assess the ability of our circuit to restore a wild-phenotype in a range of adaptively evolved lactate producing mutants. We hypothesized that silencing the effect of...
genetic manipulations would be an effective strategy to restore the wild phenotype. However, adaptive evolution can complicate this process since it is based on accumulating genetic modifications (mutations) which confer a fitness advantage. Thus, restoring the wild phenotype could require the silencing of all these mutations, which can be cumbersome given the lack of spatial homogeneity across the genome. In our study, we have shown that adaptive evolution did not greatly hinder the ability of our strains to achieve a high growth rate; however, we did show that the product profiles were more divergent from the parent wild-type strain as the strain was adaptively evolved. This is an important consideration, especially when multiple phenotypes are of interest.

**Bistable controllers can be effective to maintain their state through scaled seed fermentation processes. (Chapter 3)** Using our optimal controller, we also determined their effectiveness for scale-up procedures. One frequent criticism of synthetic circuits which require inducers is their high cost. These costs are prohibitive when considering production processes on the order of 1 million liters. The memory characteristic of this circuit should allow the induction of a given state at a small scale, and this state should be active until perturbed. We have shown that this circuit is stable for a cumulative 100 million fold dilution, comparable to a scale-up from 5mL to 500 000L.

**Genetic engineering strategies to decouple two phenotypes can be determined algorithmically (Chapter 4)** The complexity of metabolism has led to the application of metabolic models and strain design algorithms to help direct genetic engineering strategies. Here, we developed the first algorithm (MoVE: the metabolic valve enumerator) to identify genetic engineering strategies to consider two phenotypes (e.g. growth and production) and two distinct interventions (static knockouts and dynamically controlled valves). Using this algorithm, we have identified strategies to decouple growth and production phenotypes in genome-scale models of *E. coli* and *S. cerevisiae* for a majority of products using a small number of valves.

**Metabolic valves used to decouple growth and production phenotypes elucidate metabolic topology (Chapter 4)** Based on the genetic engineering strategies we obtained for *E. coli*, we further analyzed the topology of identified valves. We noted that several valves
were highly represented, despite the diversity of target metabolites. We have shown that many of our most identified valves involve a bow-tie metabolite (Zhao et al., 2006). This bow-tie is a description of metabolic structure, where a large number of substrates are funneled into a small number of metabolites (bow-tie metabolites), before being used for diverse biomass precursors. Our results would indicate that these bow-tie metabolites are of paramount importance when considering dynamic control.

**Specific metabolic valves are effective to decouple growth from the production of numerous chemicals (Chapter 4)** Using our most common valves, we explored the ability of each of these valves to be used to decouple growth and production for all products in our metabolic model. We identified some valves, such as oxygen exchange, to be effective for only a small number of products, motivating the implementation of genetic metabolic valves as opposed to process level control strategies. Furthermore, we identified valves, such as citrate synthase and pyruvate dehydrogenase, to be effective for a majority of targets, and thus could be promising candidates for generating platform strains.

**Laboratory automation and data workflows allow the exploration of a large design space (Chapter 5)** The complexity of metabolic systems and the lack of sufficiently thorough models leads to an iterative metabolic engineering cycle. Although the learning and redesign elements of this cycle require significant consideration, several steps of this cycle are highly repetitive and ripe for automation. First, we developed a set of liquid handling protocols to aid the implementation of our bistable phenotypic controller (Chapter 3), generating large volumes of data. To consolidate these large data sets we developed an open-source python-based data analysis framework (Impact).

This framework allows metabolic engineers to rapidly analyze, characterize and translate data from analytical equipment, to maximize the time spent on identifying limitations and redesigning constructs. In addition, this framework parses data into a database-friendly format for facile storage and query. We anticipate the open-source nature of this framework will promote contributions to the code base by end users, to add functionality for parsing data from new pieces of analytical equipment and for extracting features relevant to a range of
experiments.

6.2 Recommendations

**Develop more accurate metabolic models to capture the effect of substrate uptake rate**  One significant assumption in steady-state metabolic modeling is a fixed substrate uptake rate. Based on this constraint, all the remaining fluxes can be solved by means of an objective function. Although this assumption has some biological basis, for example in the case of a membrane occupancy constraint (Zhuang et al., 2011), its applicability to heavily engineered strains is unclear. This is particularly important for two-stage production processes, which target a high product yield and low growth rate (or no growth) for the production state. In these circumstances, particular care must be taken to ensure that the substrate uptake rates remain high (Klamt et al., 2017).

To improve this framework, substrate uptake rate should be estimated more accurately. This can be accomplished by either identifying and modeling fundamental constraints which limit substrate uptake, or by using data from empirical studies. Although significant literature exists for characterizing non-growing or stationary phase microbes, detailing metabolism at low growth rates owed to metabolic engineering remains an open challenge.

**Assess the burden and stability of complex genetic circuits in engineered production strains**  Despite the success of our bistable controller in lab-scale trials, industrial applications of these systems require significant consideration. First, there is significant metabolic burden associated with these circuits which is imposed by the high repressor expression levels required for bistability. This metabolic burden will either necessitate media supplementation, or reduce the yield of product, and must be carefully controlled. Although circuits with reduced burden have very recently been developed (Lee et al., 2016), further study is required when these circuits are implemented in engineered strains, which typically have significant growth defects.

**Develop more modular and predictable genetic circuits**  In this work, we screened a large number of architecture and expression level variants to identify successful constructs. This process is highly iterative and laborious, and will limit the feasibility of such systems. For
this process to become more straightforward, more detailed characterization of these circuits is required to parametrize models in a range of environmental conditions. These modular genetic components could then be effectively applied in more complex architectures combining diverse sensors, controllers and actuators.

**Implement controllers with more finite states to improve process performance**

Despite the success of two-stage production processes, there could be significant potential in exploring circuits which have more finite states. For example, in microbial production systems there could be value in a dedicated protein production stage between the growth and production stages. In fully decoupled production strategies, there is high potential to lose the capacity to produce all amino acids, and thus protein production can be stunted. In such a circumstance, a three stage production process for growth, protein production, and finally production of the desired chemical could be beneficial. Some complex state controllers exist, and could be implemented for this purpose (Roquet et al., 2016). Furthermore, given the potential for a biological system with a large number of states, strain design algorithms could be developed to develop genetic engineering strategies in these complex systems.

**Experimentally validate the MoVE algorithm to identify limitations**

Although we have identified important metabolic valves, many of which have been experimentally implemented, further experimental validation will be important to understand the limitations of the algorithm. Furthermore, since this algorithm is based on a stoichiometric metabolic model, further model refinement will play an important role for MoVE to identify effective strategies.

**Identify metabolic valves in more complex biological systems**

In this thesis, we have shown the application of MoVE for a prokaryotic and eukaryotic organism, which can be applied for two-stage production processes. However, metabolic transitions are also apparent in more complex multi-cellular and mammalian systems. For example, these transitions are found in disease states (Heiden et al., 2009) or processes such as tissue morphogenesis (Thiery and Sleeman, 2006). We suspect regulatory targets could be elucidated algorithmically, using metabolic models specific to the context of each state (Machado and Herrgård, 2014, Magnúsdóttir et al., 2017, Markert and Vazquez, 2015).
Increase the usability and access to complex data analysis workflows  As targets for engineered biological systems become more stringent, biological engineers will require the ability to generate and synthesize large global data sets. These processes are becoming increasingly complex, and the data scientists and bioinformaticians who have the skills required to analyze these data sets are becoming more specialized. The Impact framework for developing data analysis workflows aims to bridge this gap. However, significant challenges remain to ensure this system can be implemented by a range of stakeholders in the biological engineering pipeline, from biologists to software developers.

The development of graphical user interfaces remains of paramount importance to reduce the barrier to many data analysis challenges. KBase (https://kbase.us/), for example, provides widgets which can interact with typical python code. Similarly, JupyterHub (https://github.com/jupyterhub) provides server-deployed notebooks which can be accessed remotely, obviating the need for local installations. Although point and click interfaces will be important for the data analytical process, more complex analyses should be performed programmatically. A consolidated solution for use by a wide-range of users will be critical.

6.3 Outlook

Here, we have shown the potential of two-stage production processes and identified limitations in the implementation of such systems. To overcome some of these limitations, we thoroughly characterized a bistable genetic controller to decouple growth and production states, developed an algorithm to identify genetic engineering strategies suitable to this task, and implemented a set of tools in laboratory automation and data analytics to expedite this process. For this technology to be commercially viable, significant efforts are required in characterizing synthetic genetic circuits to generate more modular designs with low burden, as well as the development of more sophisticated and predictive metabolic models. Given these advances, we expect the application of two-stage production processes to increase the competitiveness of bioproduction processes and allow them to serve a greater role in the chemical industry.
Bibliography


and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide. *EcoSal Plus*, 1(10).


Appendix A

An optimized bistable metabolic switch to decouple growth and production states
A.1 Supplementary figures

**Figure A.1:** Cell populations of pTAK131 and pTAK132, grown overnight in inducers corresponding to each state.

**Figure A.2:** Determination of inducer toxicity and switching rate as a function of cell density and inducer concentration. (a) $\Delta$lacI strains were assessed for inducer toxicity by growth assay using IPTG and (b) using aTc. (c) Inducer concentration and induction OD were varied to identify inducer saturation using IPTG and (d) using aTc. The switch rate is defined as the ratio of reporter in the new target state to the previous state, six hours after induction. Error bars represent s.d. ($n \geq 3$).
Appendix A. Implementation of a bistable phenotypic switch

Figure A.3: Burden assessment of toggle switch and media variants. Three strains were tested: LAC002-D28, LAC002-D28+pKDL071, LAC002-D28+pTB001. (a) Growth profiles in M9 minimal media. (b) Growth profiles in phosphate minimal media (PMM). (c) Growth profile in supplemented PMM (RDM). Error bars represent s.d. (n ≥ 3).

Figure A.4: Sensitivity of phosphotransacetylase (PTAr) and alcohol dehydrogenase (ALCD2x) flux using the iJO1366 genome-scale metabolic model and anaerobic conditions on lactate export and growth rate. (a) Lactate export flux. (b) Growth rate.
Appendix A. Implementation of a bistable phenotypic switch

Figure A.5: Comparison of dynamic range between the two promoters used in this study, pTrc2 and pTet. Strains were grown overnight in the target inducer, to obviate the effect of switching dynamics. Error bars represent s.d. (n ≥ 3).

Figure A.6: A comparison of switching rates between fluorescent reporters and pathway enzymes in a 500mL bioreactor batch. (a) GFP fluorescence. (b) Ethanol titer.

Figure A.7: Simulation of leaky expression with decreasing levels of protease saturation using Michaelis-Menten kinetics. Dynamic range of expression is assumed to be 10x (ON/OFF), degradation rate was assumed to be 15 units, and production rate was solved to ensure an ON expression level of 10. Bar labels indicate the fold-change between ON and OFF states.
Appendix A. Implementation of a bistable phenotypic switch

Figure A.8: Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 1 (pTSLAC10X) and 1.1 (pTSLAC111) plasmids, driven by State B. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
Figure A.9: Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 1 (pTSLAC10X) and 1.1 (pTSLAC111) plasmids, driven by State B. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
Figure A.10: Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 2 (pTSLAC20X) plasmids, driven by State A. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
Figure A.11: Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 2 (pTSLAC20X) plasmids, driven by State A. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
Appendix A. Implementation of a bistable phenotypic switch

Figure A.12: Lactate titer and growth rate of LAC002 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State B, with the addition of degradation tags. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
Appendix A. Implementation of a bistable phenotypic switch

Figure A.13: Acetate and ethanol end-point titers of LAC002 strains in growth and production states using 2g/L glucose and generation 3 (pTSLAC30X) and 3.1 (pTSLAC31X) plasmids, driven by State B, with the addition of degradation tags. Purple bars indicate controls, LAC002+pKDL071 for the production state, and WT+pKDL071 for the growth state. Three different evolutionary states are presented: 1, 28 and 59 days of evolution. Error bars represent s.d. (n ≥ 3).
A.2 Synthesized DNA

**TSLAC001_gBlock_001**
CATTACTCGCATCCATTCTCAGGCTGTCTCGTCTCGTCTCCTGAAATGAGCTGTTGACAATTAATCATCCGGCTCGTATAATGTGTGGAATTGTGAGCGGATAACAATTTCACACGCTGGGAGTTTCGTAGACGGAAACAAACGCAGAATCCAAGC

**TSLAC001_gBlock_001**
GCACTGAAGGTCCTCAATCGCACTGGAAACATCAAGGTCGCTGAAATGAGCTGTTGACAATTAATCATCCGGCTCGTATAATGTGTGGAATTGTGAGCGGATAACAATTTCACACGCTGGGAGTTTCGTAGACGGAAACAAACGCAGAATCCAAGC
## A.3 Primers

<table>
<thead>
<tr>
<th>Name</th>
<th>Fragment</th>
<th>Plasmid</th>
<th>Template</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSLAC101_p001</td>
<td>F backbone</td>
<td>pTSLAC101</td>
<td>pKDL071</td>
<td>ATCTATCGCCCTAGGGACCG</td>
</tr>
<tr>
<td>pTSLAC101_p002</td>
<td>R backbone</td>
<td>pTSLAC101</td>
<td>pKDL071</td>
<td>ATCTTACTTATACAGCTCGTCATGCG</td>
</tr>
<tr>
<td>pTSLAC101_p003</td>
<td>F pta prom</td>
<td>pTSLAC101</td>
<td>TSLAC_gBlock_001</td>
<td>CATTACTCGCATCATTCTCAGGC</td>
</tr>
<tr>
<td>pTSLAC101_p004</td>
<td>R pta prom</td>
<td>pTSLAC101</td>
<td>TSLAC_gBlock_001</td>
<td>TTTGAATATGACCTTGTTTTTGTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TCAGTTGGTTGCTTGGATTCTGCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TTTGTGTTCCG</td>
</tr>
<tr>
<td>pTSLAC101_p005</td>
<td>F pta</td>
<td>pTSLAC101</td>
<td>MG1655 gDNA</td>
<td>GTGTCCTCGATTATTATATGCTGATCC</td>
</tr>
<tr>
<td>pTSLAC101_p006</td>
<td>R pta</td>
<td>pTSLAC101</td>
<td>MG1655 gDNA</td>
<td>TTACTTGCTGCTGGACTGG</td>
</tr>
<tr>
<td>pTSLAC101_p007</td>
<td>R backbone</td>
<td>pTSLAC101</td>
<td>MG1655 gDNA</td>
<td>TTTGAATATGACCTTGTTTTTGTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TCGTTGGTTGCTTGGATTCTGCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TTTGTGTTCCG</td>
</tr>
<tr>
<td>pTSLAC101_p008</td>
<td>F backbone</td>
<td>pTSLAC101</td>
<td>MG1655 gDNA</td>
<td>ATGCGCTGTATTACATTGCTGATGAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CTTAACCG</td>
</tr>
<tr>
<td>pTSLAC101_p009</td>
<td>F adhE promoter</td>
<td>pTSLAC101</td>
<td>TSLAC_gBlock_002</td>
<td>GCACGAGGTGCTCAATACTCG</td>
</tr>
<tr>
<td>pTSLAC101_p010</td>
<td>R adhE promoter</td>
<td>pTSLAC101</td>
<td>TSLAC_gBlock_002</td>
<td>GACTTTGCGTTGCTTACTATTGCTG</td>
</tr>
<tr>
<td>pTSLAC102_p001</td>
<td>F Backbone and adhE</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>CCTCGCTCTCAACAAAGCAATCAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CCCATCTCTCCTCCCAATCAAACCCCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AGGAACATATCATAAATGGCTGATT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CTAATGTCGC</td>
</tr>
</tbody>
</table>
Table A.1 Primers used in this study (continued).

<table>
<thead>
<tr>
<th>Name</th>
<th>Fragment</th>
<th>Plasmid</th>
<th>Template</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSLAC102_p002</td>
<td>R Backbone and adhE</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>CGAGACAGCCTGAGAATGGATGCAGTAATGATCTTTACTTATACAGCTCGTCC</td>
</tr>
<tr>
<td>pTSLAC102_p003</td>
<td>F pta promoter</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>GCTGGGAGGTTGAGACGGAAACAACGCACCTGAAATGAGCT</td>
</tr>
<tr>
<td>pTSLAC102_p004</td>
<td>R pta promoter</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>GTTTCCAGTGCAGTTTAGGGACCTTTAGTGCGTGTGAAATTGTTATCCCGTCC</td>
</tr>
<tr>
<td>pTSLAC102_p005</td>
<td>F pta</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>CTGACCTCTCTGCAATAGTAAAGCAACACCTGAAATGAGCTGTTGAATAATCATC</td>
</tr>
<tr>
<td>pTSLAC102_p006</td>
<td>R pta</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>ACTTGAGTGGAGTTTAAAGGAGTTGGCTTACTGCTGTGCAAGATTAATCATC</td>
</tr>
<tr>
<td>pTSLAC102_p007</td>
<td>F adhE promoter</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>CTCGTTCCGTCGCCACCTAAGAATACTTACGCTGAGCTGTTGAAATGAGCTGTTGATCC</td>
</tr>
<tr>
<td>pTSLAC102_p008</td>
<td>R adhE promoter</td>
<td>pTSLAC102</td>
<td>pTSLAC111</td>
<td>AGTAGCGGAAATGTAGGAGCCAGGCCTTGTGATCTGCTGTGCAAGATTAATCATC</td>
</tr>
<tr>
<td>pTSLAC103_p001</td>
<td>F pta RBS</td>
<td>pTSLAC103</td>
<td>pTSLAC111</td>
<td>CTGACCTCTCTGAGCAATAGTAAAGCAACACTAATCAAGAATTACAGGAGGCCTCTTTGTGCTCCCGTATTATATGCTGATCC</td>
</tr>
<tr>
<td>Name</td>
<td>Fragment</td>
<td>Plasmid</td>
<td>Template</td>
<td>Sequence</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>---------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| pTSLAC103_p002   | F        | adhE RBS| pTSLAC103| pTSLAC111
|                  |          |         |          | CCTCGTCTCAACCAAGCAATCAA
|                  |          |         |          | CCCATCTCCAGAGAATTTAATATA
|                  |          |         |          | ATAAGGAAGAGATTTAATGCTG
|                  |          |         |          | TTACTAATGTCGC |
| pTSLAC104_p001   | F        | pta RBS | pTSLAC104| pTSLAC111
|                  |          |         |          | TCCAATCTAATTTAATGACTAGGT
|                  |          |         |          | AGGTAATCTAA |
| pTSLAC104_p002   | F        | adhE RBS| pTSLAC104| pTSLAC111
|                  |          |         |          | ATATTCAAGAACTAAGGAGGACCC |
|                  |          |         |          | TAT |
| pTSLAC111_p001   | R        | backbone| pTSLAC111| pTSLAC103
|                  |          |         |          | GTTCCAGCAGTCATAGTCAAGTTC
|                  |          |         |          | CACACGGCTGGGTATCGAATTCAAGTTT |
|                  |          |         |          | GTGACTGC |
| pTSLAC111_p002   | F        | backbone| pTSLAC111| pTSLAC103
|                  |          |         |          | AGAGGTTTGGCTTTGGCAAGAAGGTG |
|                  |          |         |          | CGAATGGGTCAATGGCTCAGC |
| pTSLAC111_p003   | F        | ldhA    | pTSLAC111| MG1655 gDNA
|                  |          |         |          | GCTTTGGTGATGATACGGGATGTAG |
|                  |          |         |          | TCGGACTTTAACCCAGTTCGTTGG |
|                  |          |         |          | GCAGG |
| pTSLAC111_p004   | R        | ldhA    | pTSLAC111| MG1655 gDNA
|                  |          |         |          | CAGAGCCCTTTAGTTACCGTCATA |
|                  |          |         |          | CTTCGCCAACAACCAACAGAAGAAGAA |
|                  |          |         |          | TTTAAGTACAGGCCAAATGAAACTC |
|                  |          |         |          | GCCGTTTATAGCAGC |
| pTSLAC111_p005   | F        | ldhA promoter | pTSLAC111| pKDL071
|                  |          |         |          | AGAGGTTTGGCTTTGGCAAGAAGGTG |
|                  |          |         |          | CGAATGGGTCAATGGCTCAGC |
| pTSLAC111_p006   | R        | ldhA promoter | pTSLAC111| pKDL071
|                  |          |         |          | GTCCTTTATCATCTGGCGAATTCG |
|                  |          |         |          | ACCCACTGCCATCTCATGGGATAGAG |
|                  |          |         |          | ATTGACATCC |
**Table A.1** Primers used in this study (continued).

<table>
<thead>
<tr>
<th>Name</th>
<th>Fragment</th>
<th>Plasmid</th>
<th>Template</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSLAC2XX_p001</td>
<td>F</td>
<td>backbone_1</td>
<td>pTSLAC2XX</td>
<td>pTSLAC101 GCTGGGAGTTCGTAGACGGAAACA AACGCAATCTATCGCCCTAGGGACCG</td>
</tr>
<tr>
<td>pTSLAC2XX_p002</td>
<td>F</td>
<td>backbone_2</td>
<td>pTSLAC2XX</td>
<td>pTSLAC101 GTTCCAGCAGTCATAGTCAAGGTTT CAACACGCTGGGTATCGAATTC AAATCC</td>
</tr>
<tr>
<td>pTSLAC2XX_p003</td>
<td>R</td>
<td>backbone_2</td>
<td>pTSLAC2XX</td>
<td>pTSLAC101 CGAGACAGCCTGAGAATGGATGCG AGTAATGATCTTTACTTATACAGGCT CTGTCCATGCC</td>
</tr>
<tr>
<td>pTSLAC2XX_p004</td>
<td>F</td>
<td>pta</td>
<td>pTSLAC2XX</td>
<td>pKDL071 GTTTCAGTGCAGTGGAGACCCTT CAGTGCCTGGTCAGTGGCCTCTGC</td>
</tr>
<tr>
<td>pTSLAC2XX_p005</td>
<td>R</td>
<td>pta</td>
<td>pTSLAC2XX</td>
<td>pKDL071 GTCCTTTATCATCTGGCGAATCGG ACCCACATCCCATACAGTGAAGA ATGGACATCC</td>
</tr>
<tr>
<td>pTSLAC2XX_p006</td>
<td>F</td>
<td>pta + RBS</td>
<td>pTSLAC2XX</td>
<td>pTSLAC101 ACTTAGTGAGGTGTGAAGAAGGAG TTGGCTTTACTGCTTGCTGTGGCA</td>
</tr>
<tr>
<td>pTSLAC2XX_p007</td>
<td>R</td>
<td>adhE</td>
<td>pTSLAC2XX</td>
<td>pKDL071 ACGAGGAGAATGTCAGACGACGC AGTCTGGGTCAGTGGCCTCTGC</td>
</tr>
<tr>
<td>pTSLAC2XX_p008</td>
<td>F</td>
<td>adhE + RBS</td>
<td>pTSLAC2XX</td>
<td>pTSLAC101 GCTTGGGTGGATGTACGTACGAGATT TCGGACTTTAAGCGGATTTTTTCGC TTTTTTCAGCTTTAGC</td>
</tr>
</tbody>
</table>
Appendix A. Implementation of a bistable phenotypic switch

Table A.1 Primers used in this study (continued).

<table>
<thead>
<tr>
<th>Name</th>
<th>Fragment</th>
<th>Plasmid</th>
<th>Template</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSLAC201_p001</td>
<td>R adhE + RBS</td>
<td>pTSLAC201</td>
<td>pTSLAC101</td>
<td>CCTCGTCTCAACCAAGACAATCAA&lt;br&gt;CCCATCAGTTTACCCACAGGAGT&lt;br&gt;TATTAACACATGGCTGTCTAATTAG&lt;br&gt;TCGCTGAACCTTAACGC</td>
</tr>
<tr>
<td>pTSLAC202_p001</td>
<td>R adhE + RBS</td>
<td>pTSLAC202</td>
<td>pTSLAC101</td>
<td>CCCATCATCTCGACAAAGGAAGGT&lt;br&gt;AACAATGGCTGTCTAATTAGTCGC&lt;br&gt;TGAACTTAACGC</td>
</tr>
<tr>
<td>pTSLAC203_p001</td>
<td>R adhE + RBS</td>
<td>pTSLAC203</td>
<td>pTSLAC101</td>
<td>CCAATAATGGCTGTCTAATTAGTCG&lt;br&gt;GCTGAACCTTAACGC</td>
</tr>
<tr>
<td>pTSLAC204_p001</td>
<td>R adhE + RBS</td>
<td>pTSLAC204</td>
<td>pTSLAC101</td>
<td>CCAATAATGGCTGTCTAATTAGTCG&lt;br&gt;GCTGAACCTTAACGC</td>
</tr>
<tr>
<td>pTSLAC201_p002</td>
<td>R pta + RBS</td>
<td>pTSLAC201</td>
<td>pTSLAC101</td>
<td>CTGACCTCTGCGAGCAATAGTAA&lt;br&gt;GACAACAAACCCGGAGGAGTCCCA&lt;br&gt;AGTGTCGCCATTATTATGC</td>
</tr>
<tr>
<td>pTSLAC202_p002</td>
<td>R pta + RBS</td>
<td>pTSLAC202</td>
<td>pTSLAC101</td>
<td>CTGACCTCTGCGAGCAATAGTAA&lt;br&gt;GACAACAAACCCGGAGGAGTCCCA&lt;br&gt;AGTGTCGCCATTATTATGC</td>
</tr>
<tr>
<td>pTSLAC203_p002</td>
<td>R pta + RBS</td>
<td>pTSLAC203</td>
<td>pTSLAC101</td>
<td>CTGACCTCTGCGAGCAATAGTAA&lt;br&gt;GACAACAAACCCGGAGGAGTCCCA&lt;br&gt;AGTGTCGCCATTATTATGC</td>
</tr>
</tbody>
</table>
Table A.1 Primers used in this study (continued).

<table>
<thead>
<tr>
<th>Name</th>
<th>Fragment</th>
<th>Plasmid</th>
<th>Template</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSlac204_p002</td>
<td>R pta + RBS</td>
<td>pTSlac204</td>
<td>pTSlac101</td>
<td>CTGACCTCCTGCCAGCAATAGTAA GACAACATTATATTACGATATCAATACCCAAAATCGGAGGCCAACGTGTC</td>
</tr>
<tr>
<td>pTSlac30X_p001</td>
<td>F LAA + adhE + promoter</td>
<td>pTSlac30X</td>
<td>pTSlac20X</td>
<td>GCTTGGTGATGATACGATAGTCGGACTTAAGCAGCCAGAGCGTAAGTTTCGTCGTTAGCAGCCAGAGCGA</td>
</tr>
<tr>
<td>pTSlac30X_p002</td>
<td>R LAA + adhE + promoter</td>
<td>pTSlac30X</td>
<td>pTSlac20X</td>
<td>CTCGTTCGCTGCCACC</td>
</tr>
<tr>
<td>pTSlac30X_p003</td>
<td>F LAA + pta + backbone</td>
<td>pTSlac30X</td>
<td>pTSlac20X</td>
<td>ACTTGAGTGAGGTTGTAAGGGGAGGATGCCCTTTAAACGAGCCAGAGCGTAGTGGTTCGTCGTTAGCAGCCTGC</td>
</tr>
<tr>
<td>pTSlac30X_p004</td>
<td>R LAA + pta + backbone</td>
<td>pTSlac30X</td>
<td>pTSlac20X</td>
<td>GCTGTCGAGACTGAATCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A.4 RBS calculator v1.1 predicted translation initiation rates

Table A.2: RBS strengths of first generation plasmids calculated using the Salis RBS calculator v1.1 (Salis et al., 2009).

<table>
<thead>
<tr>
<th>Construct</th>
<th>pta</th>
<th>adhE</th>
<th>ldhA</th>
</tr>
</thead>
<tbody>
<tr>
<td>pTSLAC001</td>
<td>4 874</td>
<td>1 358</td>
<td></td>
</tr>
<tr>
<td>pTSLAC002</td>
<td>45 928</td>
<td>46 662</td>
<td></td>
</tr>
<tr>
<td>pTSLAC003</td>
<td>102 695</td>
<td>95 871</td>
<td></td>
</tr>
<tr>
<td>pTSLAC004</td>
<td>253 981</td>
<td>190 020</td>
<td></td>
</tr>
<tr>
<td>pTSLAC101</td>
<td>102 695</td>
<td>95 871</td>
<td>90 287</td>
</tr>
</tbody>
</table>
A.5 Plasmid sequences

LOCUS pTSLAC10X 11080 bp dna circular UNK 09-SEP-2017
DEFINITION pTSLAC10X two-state lactate phenotype controller.
ACCESSION unknown (modified)
KEYWORDS .
SOURCE null
ORGANISM .
REFERENCE 1 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 2 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 3 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 4 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
FEATURES Location/Qualifiers
  misc_feature 64..93
    /note="Geneious type: insulator"
  misc_feature 94..124
    /note="Geneious type: insulator"
  RBS 125..159
  CDS 160..2304
    /locus_tag="b2297"
    /function="enzyme; Degradation of small molecules: Carbon compounds"
    /protein_id="NP_416800.1"
    /gene="pta"
    /transl_table=11
    /note="phosphotransacetylase"
    /db_xref="ASAP:ABE-0007582; EcoGene:EG20173; GI:16130232; GeneID:946778; UniProtKB/Swiss-Prot:P0A9M8"
    /codon_start=1
    /translation="MSRIIMLIPTGTSGVLTSSLGVIRAMERKGVRLSVFKPKIAQPR
TGGDADPQTFTTVRANSSTTTAAEPLMSYVEGLSSNQKDVMEEIVANYHANTKDA
EVVLVEGLVTPTRKHFAQSLNYEIAKTLNAEIVFVMSQGTDTPQELKRIELTRNKSFG
GAKNTNUITGVVKNLAPVDEOQGRTRPLSEIFDDSSKAKVNNVPAPKALQESSPLPV
GAVPWSFDLIAITRAIDMARHNLATIIEENGTINTRVKSVTFCARSIFSHPMLEHFRAGSL
LVTSADRBDVLVAACALMGEIGALLLTGGYEMDARISKLCERAFATGTPVLMVNT
NTWQTSSLQSFNLVQVPDHERIEKQVEYVANYINADWISLETATSERSRRLSPPAF
RYQLTELARKAGRIVLPEGDEPRVTKAAGCAERGIATCVLLGNPANPVEINVAAASQGV
ELGAGIEIVDPEVRESYLNLREKLMKCHTEVAREQLEDNVVLGLTMLEQDEVDG
LVSGAVHTTANTIRPQLIKTPAGSSLVSSVFFMLLPEQVYVYGDCAINPDPTAEQL"
Appendix A. Implementation of a bistable phenotypic switch

AEIAIQSADSAAAFGIEPRVAMLSTYGTSGAGSDVEKVRREATRLAQERPKPMLDG
LQYDAVMDVASKAPNPVAGRATVFIPDLTNTGTYKAVQSRADLISIGPMLQG
MRKVNDLSRGLVDDIVYTIALTAIQSAQQQ

//product="phosphate acetyltransferase"
//EC_number="2.3.1.8"

misc_feature 2305..2335
//note="Geneious type: insulator"

misc_feature 2336..2366
//note="Geneious type: insulator"

promoter 2367..2441
misc_feature 2442..2471
//note="Geneious type: insulator"

misc_feature 2472..2501
//note="Geneious type: insulator"

RBS 2502..2534
CDS 2535..5210
/locus_tag="b1241"
/function="enzyme; Energy metabolism, carbon: Fermentation"
/protein_id="NP_415757.1"
/gene="adhE"
/transl_table=11
//note="CoA-linked acetaldehyde dehydrogenase and iron-dependent alcohol dehydrogenase; pyruvate-formate-lyase deactivase"
/db_xref="ASAP: ABE-0004164; EcoGene: EG10031; GI: 16129202; GeneID: 945837; UniProtKB/Swiss-Prot: P0A9Q7"
/codon_start=1
/translation="MAVTNVAELNALVKKAQREYASFTQEVQVKIFRAAALAAA
ARPIAKMAESAEMGIVEDKVIKNHEASYYAYKDEKTGOVLSEDDTFTITIAE
PIGIIICGVPTNPTSTAIFKSLIELKTNRIIFSPHRPAKATNKAADIVLPAAIAA
GAPKDLGWIDQPSVLESALMHPHDPLILILATGPGMKAAYSSGPAIVGAGN
VIDETEIKRAVAVLSMTFDNGVIASECTIONSVVVDVVYDAVRERFATHGGYLLQQ
KELKAVQDVILKNGALNAIAIVGQPAAYIAELAGFSVPENTKXKILIGVTVDDAPEF
AKLSPVLMYRAKDFPDEVEKAELVAMGGIGHTSCEYTDQDNQPARVSYFQGKMKTA
RILINTPSAQXGIDLYNFKLAPSLLCGSWGNNSEVENPVPKLINKTVKRAEN
MLWHRKLPSIYFFRGSPLALDEVITDHKRALKLVTDRLFNNYADQITSVLKAAVG
ETEVFFEADPASTLSIVRKEAELANSFKPDVIIALGGSSPMDAKIMMVYEHPETHF
EEALRFDIRKRIYKFKMGVKKMXAVTTSSTGSEVTVPAVVTDATGQKPLAD
YALTDMAVDLVMDPGLSCLAFGGLAVTHAMEAVAHSVLSFEDSGQALKLL
KEYLPASYHESKSNPVARERVHASSAITAGIAIFANAFGLVCHSMAHKLQSFHIPHGLA
NALLICNVIRYANDNPTKQATSQDRPSQARRYAIAEHHLQISAPGDRTAAKIEK
LAWLTDLAIKPGKRSRAEGQVQADFDLANVDKSLSADDDQCTGANGPRYPLISELKLQ
ILLDTYGYRVDYVEGETAAKKEAAPPKAEKAKKSA"

//product="fused acetaldehyde-CoA dehydrogenase/iron-dependent alcohol dehydrogenase/pyruvate-formate lyase deactivase"
/EC_number="1.1.1.1; 1.2.1.10"
Appendix A. Implementation of a bistable phenotypic switch

misc_feature 5262..5366
  /gene="T1"
  /product="transcriptional terminator from rrnB operon"
rep_origin 5373..6180
misc_feature 6187..6292
  /gene="T0"
  /product="transcriptional terminator from phage lambda"
CDS complement(6318..7112)
  /gene="KanR"
CDS complement(7303..8019)
  /gene="GFPmut3b_RSR"
promoter complement(8071..8144)
  /gene="PL(tetO)"
CDS complement(8187..9269)
  /gene="lacI-RSR"
promoter complement(9306..9379)
  /gene="PL(tetO)"
promoter 9398..9472
CDS 9511..10134
  /gene="tetR_RSR"
promoter 10177..10251
CDS 10294..11004
  /gene="mCherry"
misc_feature 11008..11038
  /note="Geneious type: insulator"
misc_feature 11039..11068
  /note="Geneious type: insulator"
promoter 11069..63

ORIGIN

1  gttgacaatt aatcatccgg ctcgtataat gtgtggaatt gtgagcggat aacaatttca
  61  caacgcactga aaggtcctcaa tcgcactgga aacctgacct cctgccagca atagtaagac
  121  aacannnnnn nnnnnnnnn nnnnnnnnn nnnnnnnnn nnnnnnnnn tcgcgtctat tattagctg
  181  atccctaccg gaaccagcgt cggtctgacc agcgtcagcc ttggcgtgat ccgtgcaatg
  241  gaacgcaaag gcgttcgtct gagcgttttc aaacctatcg ctcagccgcg taccgggtggc
  301  gatgcgcccg atcagactac gactatcgtg cgtgcgaact cttccaccac cagccgccgc
  361  gaaccgctga aaatgagcta cgttagaaggt ctgcttttcca gcaatcagaa agatgtgcgtg
  421  atgtactgtt cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
  481  aaggtcgtgg tccgacacag taaaacatcg gaccgctgtg cctttactca gcgtgcgtgc
  541  aacagctgga atgcgtcgcg atcagactac gactatcgtg cgtgcgaact cttccaccac cagccgccgc
  601  gaaccgctga aaatgagcta cgttagaaggt ctgcttttcca gcaatcagaa agatgtgcgtg
  661  atcggctcct cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
  721  gatgtactgtt cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
  781  aaggtcgtgg tccgacacag taaaacatcg gaccgctgtg cctttactca gcgtgcgtgc
  841  atcggctcct cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
  901  gatgtactgtt cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
  961  atcggctcct cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
 1021  gatgtactgtt cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
 1081  gatgtactgtt cacgctcgctc agcggccctg gccatccacct tcggtctgttt ctgcttttca gcaatcagaa
Appendix A. Implementation of a bistable phenotypic switch

1141 ccggtattta tggtgaacac caaacctgta cagaccttct tcagcctgta cagcttcaac
1201 ctggaagtcc cggttgacga tcaagcgaat ctgtagaatg tcgtgcatc
1261 tacatcaacg ctagctggtt cgaatccgat actgtccacg ctagctggtt cagcttcaac
1321 ttccgccttg cgtttccgta tcaagcgtta cagcttcaac tcaacatcgc cagcttcaac
1381 gaaagctgga cagacctgta cagcttcaac atgctctcct ctgcttgaac gaaagctgga
1441 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
1501 caggtgttag aactgggtgc cgggtttatc cagcaagtta cagcttcaac cgggtttatc
1561 tatgtttggtc gcttggttttt ccaagccttg cccttggttga cagcttcaac cgggtttatc
1621 gaaagctgga cagacctgta cagcttcaac ctgctctcct ctgcttgaac gaaagctgga
1681 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
1741 atcaagcgtta ctgcttgaac gaaagctgga cagacctgta cagcttcaac cgggtttatc
1801 caggtgttag aactgggtgc cgggtttatc cagcaagtta cagcttcaac cgggtttatc
1861 gaaagctgga cagacctgta cagcttcaac ctgctctcct ctgcttgaac gaaagctgga
1921 gaaagctgga cagacctgta cagcttcaac ctgctctcct ctgcttgaac gaaagctgga
1981 gaaagctgga cagacctgta cagcttcaac ctgctctcct ctgcttgaac gaaagctgga
2041 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2101 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2161 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2221 aacgtggctg cgggtttatc cagcaagtta cagcttcaac cgggtttatc cgggtttatc
2281 attcctggga cagacctgta cagcttcaac ctgctctcct ctgcttgaac gaaagctgga
2341 atcaagcgtta ctgcttgaac gaaagctgga cagacctgta cagcttcaac cgggtttatc
2401 caggtgttag aactgggtgc cgggtttatc cagcaagtta cagcttcaac cgggtttatc
2461 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2521 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2581 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2641 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2701 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2761 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2821 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2881 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
2941 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3001 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3061 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3121 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3181 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3241 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3301 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3361 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3421 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3481 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3541 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3601 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3661 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3721 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3781 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3841 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3901 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
3961 cgggtttatc cagcaagtta cagcttcaac ccctggttga cagcttcaac cgggtttatc
Appendix A. Implementation of a bistable phenotypic switch

4021 ctgatcagat caacttcgta ctgaagacag caggcgttga aactgaagtc ttcttcgaag
4081 tagaagcgga cggacgccctg agcatcgttc gtaaaggtgc agaactggca aactccttca
4141 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4201 ggttcatctg cgagctgtcc gctgggtcag aatatccgctg gcagactatg cgctgactcc ggatatggcg attgtcgacg
4261 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4321 ggatcttcgc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4381 ggttcatctg cgagctgtcc gctgggtcag aatatccgctg gcagactatg cgctgactcc ggatatggcg attgtcgacg
4441 caccacagtc gcagctgtcc gctgggtcag aatatccgctg gcagactatg cgctgactcc ggatatggcg attgtcgacg
4501 ggttcatctg cgagctgtcc gctgggtcag aatatccgctg gcagactatg cgctgactcc ggatatggcg attgtcgacg
4561 ggttcatctg cgagctgtcc gctgggtcag aatatccgctg gcagactatg cgctgactcc ggatatggcg attgtcgacg
4621 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4681 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4741 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4801 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4861 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4921 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
4981 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5041 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5101 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5161 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5221 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5281 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5341 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5401 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5461 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5521 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5581 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5641 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5701 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5761 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5821 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5881 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
5941 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6001 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6061 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6121 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6181 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6241 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6301 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6361 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6421 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6481 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6541 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6601 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6661 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6721 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6781 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
6841 aaccacagtc gattatcgcg ctgggtggtg gttccccgat ggacgccgcg aagatcatgt
Appendix A. Implementation of a bistable phenotypic switch

6901 gccagccacag atagccgcgc tgcctcgtct tgcagttcat tcagggcacc ggacaggtcg
6961 gtcttgacaa aaaaagcgg gcgacccttg gcgacaggt gcgacacgg gcgacaggtcgc
7021 cagccgattg ctctggacat gcagtcaatt cagttcatg ttgtagttccg ggtactgttc
7081 gcagggcatt gtaaccctcc cagcgggttg ggaacaggtc ggtctctcctt
7141 ttgctgacgg atctggagaac ggtgccacag ctaggagttg ttcggctcctt gaagggcattgc
7201 gacagggcct gcgacaggtc accatcctgc gcgtctcatgc ggcgaccgcc gcagggcattgc
7261 atctctcaat gccgtctcgg cctctcttga tacagttcat gcgtctcatgc gcgtctcatgc
7321 ttcgggcctt cctctcttga ttcgggcctt gcgcgtcctt gcgcgtcctt gcgcgtcctt
7381 gacagggcatt gccgtctcgc ggtggaacag ggtggaacag ggtggaacag ggtggaacag
7441 gcagggacag cgcaggtgtt thcaggttgt tcaggttgtt ggtggaacag ggtggaacag
7501 atctctccta gccgtctcgc ggtggaacag ggtggaacag ggtggaacag ggtggaacag
7561 catcttcctt gccgtctcgc ggtggaacag ggtggaacag ggtggaacag ggtggaacag
7621 atctctccta gccgtctcgc ggtggaacag ggtggaacag ggtggaacag ggtggaacag
7681 cgcaccggac cgcaccggac cgcaccggac cgcaccggac cgcaccggac cgcaccggac
7741 cgcaccggac cgcaccggac cgcaccggac cgcaccggac cgcaccggac cgcaccggac
7801 ttcgggcctt cctctcttga gctcattcct gccgtctcgc ggtggaacag ggtggaacag
8041 ctctctctc tgcaggtgtt gcgtgctct tgcaggtgtt gcgtgctct tgcaggtgtt
8101 gcaggtgtt gcgtgctct tgcaggtgtt gcgtgctct tgcaggtgtt gcgtgctct
8161 ttccgacagg taccctctcg gcgcgtgtcc tgcaggtgtt gcgtgctct tgcaggtgtt
8221 gcctggtgca ttataacgc gcgtgctct tgcaggtgtt gcgtgctct tgcaggtgtt
8281 agggtggttg cttctctcttc accaggtgtt gcgtgctct tgcaggtgtt gcgtgctct
8341 cctgagagct gggactggtg gctcattcct gccgtctcgc ggtggaacag ggtggaacag
8401 ttgagttcct ccgctcgttg gctgcaattg gctgcaattg gctgcaattg gctgcaattg
8461 accagagtct ccgctcgttg gctgcaattg gctgcaattg gctgcaattg gctgcaattg
8521 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
8581 atgtgctctt gccccgtcgac ctcacctggt gcagttacct gcagttacct gcagttacct
8641 attaggtgcgt gccttgtgcg gccctgtgcg gccctgtgcg gccctgtgcg gccctgtgcg
8701 ttctatttgg tggcagcttc acgcgacggt gcagttacct gcagttacct gcagttacct
8761 gccatctcgc gcagttacct gcagttacct gcagttacct gcagttacct gcagttacct
8821 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
8881 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
8941 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9001 atgtgctctt gccccgtcgac ctcacctggt gcagttacct gcagttacct gcagttacct
9061 cttctctctc gccccgtcgac ctcacctggt gcagttacct gcagttacct gcagttacct
9121 tcggtgctct gctgcaattg gctgcaattg gctgcaattg gctgcaattg gctgcaattg
9181 agctggtgctct gctgcaattg gctgcaattg gctgcaattg gctgcaattg gctgcaattg
9241 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9301 gacatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9361 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9421 atctcttctc gcagttacct gcagttacct gcagttacct gcagttacct gcagttacct
9481 ctctctctct ctcacctggt gcagttacct gcagttacct gcagttacct gcagttacct
9541 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9601 ggcgccgacg ctcacctggt gcagttacct gcagttacct gcagttacct gcagttacct
9661 gccatctcgag atctcttctc gcacgggacgc gcagttacct gcagttacct gcagttacct
9721 gcagttacct gcagttacct gcagttacct gcagttacct gcagttacct gcagttacct
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDS</td>
<td>complement(1138..1932)</td>
</tr>
<tr>
<td>misc_feature</td>
<td>complement(2120..2150)</td>
</tr>
<tr>
<td>misc_feature</td>
<td>complement(2151..2180)</td>
</tr>
<tr>
<td>CDS</td>
<td>complement(2181..4856)</td>
</tr>
<tr>
<td>RBS</td>
<td>complement(4857..4882)</td>
</tr>
</tbody>
</table>

**Function:**
- **Energy metabolism, carbon:** Fermentation
- **protein_id:** NP_415757.1
- **EC_number:** 1.1.1.1; 1.2.1.10

**Product:**
- fused acetaldehyde-CoA dehydrogenase/iron-dependent alcohol dehydrogenase/pyruvate-formate lyase deactivase

**Note:**
- Geneious type: rep_origin
- Geneious type: KanR
- Geneious type: insulator
- Geneious type: Insulator
- Geneious type: promoter

**Translation:**
```
MAVTNVAELNALVERVKKAQREYASFTQEQVDKIFRAAAALAAAD
ARIPLAKMAVAESMGIVEDKVIKNHFASEYIYNAKYDEKTCGVLSEDDTFTGTTIAE
PIGIGICGIPVPTNPTSTAIFKSLISLKTNRNAAIFSPHPRAKADTNKAADIVLQAIAAA
GAPKDLIGIWIDQPSVLSMALMHPDINLILATGGPGVMKAAAYSSGGKPAIVGVAGNTP
VVIDETADKRAVASVLMSKTFDNGVICASEQSVVVDVSYDAVRERFATHGYYLQK
KELKAVQDVIKLNGALNAIAVQGPAYKIAELAIGFSVPENTKILIGEVTVVDESEPF
EKLSPTLAMYRAKDFDEAVEKAELVAMGGIGHTSCLYTDQNPARVSYFQKMKTA
RILINTPASQGIIDLYNFKLAPSLTLGCGSWGNSEINCENGPKHLINKKTVAKRAEN
MLWHKLPSIYFRGRLPSIALDEVITDGHKRALIVTDRFLFNGYADQITSVLKAA
ETEVEFVEADPTSLISVRKGAELANSFKPDVIIALGGSPMDAKIMWVWYEHPCET
EELALARFMDIRKRIYKFPKGVMKKMAVTTTSTGTSEVTIPAVVTDATGQXPLAD
YALTPTMAIDVANLVDMPKSLCAGGLDLAVTHEAMEAVSLAEGSDFGQALKLL
KEYLPASYEGSKNPVARERVHSATIAGIFANAFGLVCHSMAHKGLSOFHPHGLA
NALLICNVIRYANDNPTKQTAFSPYDRCPQARRRYEAADHLIGLSAPGRTAISKEL
LAWLETLKAEGLPKSIREAVGQEAFLANVDKLSEDAFDDQCTGANGPRYPLIS
ILLDTYGGYRDYEVEETAKKAAPAAKEKAKKSA
```
Appendix A. Implementation of a bistable phenotypic switch

misc_feature complement(5017..5047)
    /note="Geneious type: insulator"

misc_feature complement(5048..5078)
    /note="Geneious type: insulator"

CDS complement(5079..7223)
    /locus_tag="b2297"
    /functions="enzyme; Degradation of small molecules: Carbon compounds"
    /protein_id="NP_416800.1"
    /gene="pta"
    /transl_table=11
    /note="phosphotransacylase"
    /db_xref="ASAP:ABE-0007582; EcoGene:EG20173; GI:16130232; GeneID:946778; UniProtKB/Swiss-Prot:P0A9M8"
    /codon_start=1
    /translation="MSRIIIMLIPTGTSVGLTSVSLGVRAMERKGVRSLSFVKPIAQPR TGGDAPDQTNTTITVRSNNTTTTAEEPLKMSYVEGLSNSQDVDMLVEIVANYHNTKDA EYVLVGEGLVPTRKHQAFMSLNYEIAKTLNAEVFMVMSQGTDTPEQKLKERIELTRNSFG GAKTNITGIVVKNLNAPVDEQGRTRPDLSIFDSSKAKVNVDPAKLQEQSSPLPVGL GAVPWSFDLIATRAIDMARHLNATINEGDIINRRVKVSFTCARSIIPMLEHFRAGSL LVTSADRDPDVLAACALANGVEIGALLTTGGYEMDAKLCLERAFATGLPVMVNT NTWQTSLSLQSFNLELPDHERIEKQERYVANYINADWIESTLTATSERSSRLPAPF RYQLTLEARKAGKIVLPDEPRVKAIAICAERGIACVLGGNPAEINVVAASQGV ELGAGIEIVDPEVREYVGLVRLEKNSKGMETVAREQLEDNVLGLMLQDEVVDV GVSAGVHTTANTIRPPLLQILAKPSSLVSSVFFMPQLPEQVYVGYDCAINPDTAEQL AEIAIQSADSAAAAFGIEPRVAMLTSYGTSGAGSDVEKREATLAEKRDPDMIDGP LQYDAVMDAVAKSKAPNPSVAGRFVTFIFPDNLNTGYTKAVRSADLSIIPMLQG MKPVPDLNLSRAGLDDGYTIALTAIQSAQQ"
    /product="phosphate acetyltransferase"
    /EC_number="2.3.1.8"

RBS complement(7224..7259)

misc_feature complement(7260..7290)
    /note="Geneious type: insulator"

misc_feature complement(7291..7320)
    /note="Geneious type: insulator"

misc_feature complement(7321..7394)
    /gene="PL(tet0)"
    /note="Geneious type: promoter"

misc_feature complement(7395..7425)
    /note="Geneious type: Insulator"

misc_feature complement(7426..7457)
    /note="Geneious type: promoter"

CDS complement(7458..8174)
    /gene="GFPmut3b_RSR"

misc_feature complement(8226..8299)
    /note="Geneious type: promoter"

CDS complement(8342..9424)
Appendix A. Implementation of a bistable phenotypic switch

```
/gene="lacI-RSR"
/misc_feature complement(9461..9534)
/gene="PL(tetO)"
/note="Geneious type: promoter"
/promoter 9553..9627
/CDS 9666..10289
/gene="tetR_RSR"
/promoter 10332..10406
/CDS 10449..11159
/misc_feature 11163..11193
/note="Geneious type: insulator"
/ORIGIN

1 gctgggagtt cgtagacgga aacaaacgca atctatcgcc ctagggaccg tctcgagaga
61 atcaatatta atccaacgcg tggcatcaaa taaaacgaaa ggctcagtcg aaagactggg
121 cctttcgttt tatctgttgt ttgtcggtga acgctctcct gagtaggaca aatccgccgc
181 cctagactta ggccgttcgc gtcggcagac gttacgaagt caatccgagt ctcagttgat
241 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
301 ggcacgagac cgttaaaaag cccggtgctt tcttctcttc cggctcggct cggctctctc
361 ggttaaacag gctcagttgg ttgcggctct tggctctggt tcagggaccg tctcgagaga
421 aatcagttcg ccttttcttc gaagctcttc ttcggtgatt tccagttgat ctcagggcgc
481 tgcagccgat gtaaagcacg ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
541 ggttaaacag gctcagttgg ttgcggctct tggctctggt tcagggaccg tctcgagaga
601 aatcagttcg ccttttcttc gaagctcttc ttcggtgatt tccagttgat ctcagggcgc
661 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
721 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
781 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
841 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
901 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
961 tgcagccgat gtaaagcacg ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1021 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1081 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1141 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1201 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1261 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1321 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1381 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1441 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1501 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1561 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1621 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1681 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1741 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1801 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1861 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1921 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1981 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
2041 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
```

**ORIGIN**

1 gctgggagtt cgtagacgga aacaaacgca atctatcgcc ctagggaccg tctcgagaga
61 atcaatatta atccaacgcg tggcatcaaa taaaacgaaa ggctcagtcg aaagactggg
121 cctttcgttt tatctgttgt ttgtcggtga acgctctcct gagtaggaca aatccgccgc
181 cctagactta ggccgttcgc gtcggcagac gttacgaagt caatccgagt ctcagttgat
241 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
301 ggcacgagac cgttaaaaag cccggtgctt tcttctcttc cggctcggct cggctctctc
361 ggttaaacag gctcagttgg ttgcggctct tggctctggt tcagggaccg tctcgagaga
421 aatcagttcg ccttttcttc gaagctcttc ttcggtgatt tccagttgat ctcagggcgc
481 tgcagccgat gtaaagcacg ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
541 ggttaaacag gctcagttgg ttgcggctct tggctctggt tcagggaccg tctcgagaga
601 aatcagttcg ccttttcttc gaagctcttc ttcggtgatt tccagttgat ctcagggcgc
661 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
721 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
781 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
841 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
901 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
961 tgcagccgat gtaaagcacg ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1021 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1081 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1141 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1201 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1261 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1321 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1381 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1441 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1501 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1561 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1621 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1681 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1741 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1801 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1861 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1921 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
1981 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
2041 gtagcgcaga aatcggcagc ggtgacgcgc ggacacgagt ctcagttgat ctcagggcgc
Appendix A. Implementation of a bistable phenotypic switch

2101 acttggaattc gataccagg cgctgtgaaac ttgactatga ctgctggaac gcttggtgga
d2161 tgtgatacgg tagttcggac ttaagcggat tttttcgcct tttttccagc tttacccagg
2221 agacgatcgc gaaatggaac cttggagacc gcgtctcgtg cgtgcctttt cgtgcctttt
2281 gctcccaacct cagccagcat gctgagtttt tggagctgcg gcatcagcgt cgtgcctttt
2341 tctgttcttg atgttaactg aagctactgc ccggcagccc gctacttcag cgcggcgccc
d2401 gctggtgttg ttttacttcg tagccggtgc actcttggttg aattttactg accgatcgttg
2461 gcgtcttttt ttcttttcttt ctttttttttt ttttttttttt ttttttttttt ttttttttttt
2521 gcatcttctg gctgtagcgt gactgtttcg ctgatcgttt cgtgcccttt cgtgcccttt
2581 atgatttcttc tggattgggt tggaaagatc gtttttaccc tggacccggt ggtgccgtcg
d2641 tgcgcttctg cggatcggcg cttacttcag cggagcggcgg cggagcggcgg cggagcggcgg
2701 tacacttcag gctggttttg tggctgaagc ctttttcttt ctttttcttt ctttttcttt
2761 gcttgtgttg cgtgcttctg gctgtgtttt ctttttcttt ctttttcttt ctttttcttt
2821 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
2881 cgggtggtggt ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
2941 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3001 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3061 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3121 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3181 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3241 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3301 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3361 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3421 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3481 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3541 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3601 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3661 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3721 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3781 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3841 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3901 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
3961 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4021 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4081 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4141 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4201 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4261 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4321 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4381 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4441 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4501 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4561 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4621 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4681 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4741 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4801 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4861 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
4921 ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg ggtggtggtg
Appendix A. Implementation of a bistable phenotypic switch

```plaintext
//
LOCUS pTSLAC30X 11259 bp dna circular UNK 24-JAN-2018
DEFINITION pTSLAC30X two-state lactate phenotype controller.
ACCESSION unknown (modified)
KEYWORDS .
SOURCE null
ORGANISM .
REFERENCE 1 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 2 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 3 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
REFERENCE 4 (bases 1 to 5797)
  AUTHORS Self
  JOURNAL Unpublished.
FEATURES Location/Qualifiers
  misc_feature 1..30
    /note="Geneious type: Insulator"
  misc_feature 193..1000
    /gene="ColE1"
    /product="High Copy Origin of Replication"
    /note="Geneious type: rep_origin"
CDS complement(1138..1932)
    /gene="KanR"
  misc_feature complement(2120..2150)
    /note="Geneious type: Insulator"
  misc_feature complement(2151..2180)
    /note="Geneious type: Insulator"
CDS complement(2181..4889)
    /locus_tag="b1241"
    /function="enzyme; Energy metabolism, carbon: Fermentation"
    /protein_id="NP_415757.1"
    /gene="adhE"
    /transl_table=11
```
Appendix A. Implementation of a bistable phenotypic switch

CoA-linked acetaldehyde dehydrogenase and iron-dependent alcohol dehydrogenase; pyruvate-formate-lyase deactivase

/translation="MAVTNVAELNALVERVKQAQREYASFTQEVDKIFRAAALAAAD ARIPLAKMAVESGNGMEVDKVIKHFASEYIYNAKDEKTCGVLSEDDEDFGTTITIA PIGIICIVPTNPTSTAIKSLISLKTNRNAIFSPHPRAKDATNKAADIVLQAIIAAG GAPYDLGWIDQPSVELSNLMHHPDINLILATGPGMVKAAAYSSGKPAIGVAGNTP VVIDETADIKRAVASVLMSKTFDNAEGASICASEQSVVVDSYDAVRERFATHGYYLQG KELKAVQDVLKNGALNAIVGQPAYKIAELAGFVPTENTKIGELGTVTDESEFPAH EKLSPALTAMYRAKDFEVAKAELVAMGGIGHTSCLYTDQNPQARVSYFCQMKMAT RILINTPASQGIGDLYMNFKLAPSLTGCGSWGNSISENVGPKHLINKKTVAKEREN MLWKLKPKSILYFRGSLPIALDEVIHGKRIALVTDLFINGYADQITSVLAAAGV ETENVFFEVEADPTSLVISRKGAEANSFKPVDIILAGGSPMDAAKLMVWYHPEHTHF EELALRFMDIRKRIYKFPKMVKAKMAVTTTSGTGETVFPAVTDATGQYPLAD YALTPDMAIRDVANLVMMPKSLCAFGGLDAVTHAMEAYVSLASEFDGDQALQALKILL KEYLPASYEHSKNPVARERVHAATIAIFANAFGLVCHSHMAHKLGQFHIPHLGAL NALLICNVIRYNAANDTPKTAFSQLPQARRYAIEADHLGSLAPGRTAIAKIEKL LAWLETLYKELGPIKSFIREAQVQEAQFLANVDKLSDAFDDQCTGANPRLISELK QILDITYGRDYVEGETAAKEAPKAEEKKAKSA"

/product="fused acetaldehyde-CoA dehydrogenase/iron-dependent alcohol dehydrogenase/pyruvate-formate lyase deactivase"

/EC_number="1.1.1.1; 1.2.1.10"

/misc_feature complement(2184..2216)

/misc_feature complement(4916..4945)

/misc_feature complement(4946..4975)

/misc_feature complement(4976..5049)

/gene="PL(tetO)"

/misc_feature complement(5050..5080)

/misc_feature complement(5081..5111)

/CDS complement(5112..7289)

/locus_tag="b2297"

/function="enzyme; Degradation of small molecules: Carbon compounds"

/protein_id="NP_416800.1"

/gene="pta"

/transl_table=11

/note="phosphotransacetylase"

/db_xref="ASAP:ABE-0007582; EcoGene:EG20173; GI:16130232; GeneID:946778; UniProtKB/Swiss-Prot:P0A9M8"
Appendix A. Implementation of a bistable phenotypic switch

/codon_start=1
/translaction="MSRIIMLIPTGVGLTSVGLVIRAMERKGVRLSVFKPIAQPR
TGGDAPDQTIVRANSSTTTAAEPLKMSYVEGLSSNQKDVLMEEIVANYHANTKDA
EVLVEGLVPTRKHSQAQLNYEIAKTLNEAEIVFMSQGTDIPEQLKERIELNRSFG
GAKNTNITGVINKLAPVDEQGTRPGDSEIFDDSSKAKVNVDPAPLQESSPLPVL
GAVPWSDFLIATRAPDMARHNLATIINEGDIINTRVKSVTPCARSPHMLHEFRAGSL
LVTSADRDPVLDVARACLAAMNGVEIGALLLTGGYEMDARISKLCERAFATGLPVFMVNT
NTRQTSLSLQSNIEVPPVDHHERVEKQEYVANYINADWIESLTATERSRSLPPAF
RYQLTELARKGRIVLPEDPEPTVKAIAICAERGIATCVLGNPAAINVAASQQGV
ELGAGEIEVDEPVRESYVGLVELRKNKMTETVAREQLEDNVVLGTMLERQEDVDG
LVSGAVVHTTANTIRPLLQLTGAPSSLVSSVFMLLEPQVYVVYGDCAINPDPDEAQL
AEIAIQSAADSSAAAAFICRVALMSYSTGTSGAGSDVEKVTREATRLAQKEKRPDLMIDGP
LQYDAAVMDAVKSKAPNSPGRATVFIIFPDNLTVNGTGGYKAVQRSADLISIGMQLQG
MRKPVDNLRGLVDDVITYLTIALTAIQSAQQQ"
/product="phosphate acetyltransferase"
/EC_number="2.3.1.8"
misc_feature complement(5115..5147)
misc_feature complement(7326..7356)
  /note="Geneious type: insulator"
misc_feature complement(7357..7386)
  /note="Geneious type: insulator"
misc_feature complement(7387..7460)
  /gene="PL(tetO)"
  /note="Geneious type: promoter"
misc_feature complement(7461..7491)
  /note="Geneious type: promoter"
misc_feature complement(7492..7523)
  /note="Geneious type: insulator"
CDS complement(7524..8240)
  /gene="GFPmut3b_RSR"
misc_feature complement(8292..8365)
  /gene="PL(tetO)"
  /note="Geneious type: promoter"
CDS complement(8408..9490)
  /gene="lacI-RSR"
misc_feature complement(9527..9600)
  /gene="PL(tetO)"
  /note="Geneious type: promoter"
promoter 9619..9693
CDS 9732..10355
  /gene="tetR_RSR"
promoter 10398..10472
CDS 10515..11225
  /gene="mCherry"
misc_feature 11229..11259
  /note="Geneious type: insulator"

ORIGIN

1 gctgggagtt ctagacgga aacaaacgca attctgccc ctagggaccg tctcgagaga
Appendix A. Implementation of a bistable phenotypic switch

A.4.1

2941 ccgaaagcac acagggactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3001 tccggagtcttg ccagttttcga taacgcttgtt ccaggtcctgt tggcgtcgac aatcgccata
3061 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3121 gctttcagcc ccatctggcaga cttctggtgag cgtgacttgt ccaggtcctgt tggcgtcgac aatcgccata
3181 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3241 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3301 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3361 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3421 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3481 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3541 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3601 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3661 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3721 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3781 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3841 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3901 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
3961 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4021 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4081 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4141 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4201 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4261 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4321 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4381 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4441 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4501 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4561 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4621 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4681 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4741 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4801 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4861 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4921 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
4981 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
5041 tccgatcgcag agaggtgactt cggcatgtcc ataaccaggt tggcgtcgac aatcgccata
Appendix A. Implementation of a bistable phenotypic switch

LOCUS pTSLAC111 12351 bp dna circular UNK 24-JAN-2018
DEFINITION pTSLAC111 two-state lactate phenotype controller.
ACCESSION unknown (modified)
KEYWORDS .
SOURCE null

ORGANISM .

REFERENCE 1 (bases 1 to 5797)
AUTHORS Self
JOURNAL Unpublished.

REFERENCE 2 (bases 1 to 5797)
AUTHORS Self
JOURNAL Unpublished.

REFERENCE 3 (bases 1 to 5797)
AUTHORS Self
JOURNAL Unpublished.

REFERENCE 4 (bases 1 to 5797)
AUTHORS Self
JOURNAL Unpublished.

FEATURES Location/Qualifiers
misc_feature 64..93
   /note="Geneious type: insulator"
misc_feature 94..124
   /note="Geneious type: insulator"
RBS 125..151
CDS 152..2296
   /locus_tag="b2297"
   /function="enzyme; Degradation of small molecules: Carbon compounds"
   /protein_id="NP_416800.1"
   /gene="pta"
   /transl_table=11
   /note="phosphotransacetylase"
   /db_xref="ASAP:ABE-0007582; EcoGene:EG20173; GI:16130232; GeneID:946778; UniProtKB/Swiss-Prot:P0A9M8"
   /codon_start=1
   /translation="MSRIIMLIPTGTSVGLTSVSLGVTSMERKGVRLSVFKPIAQPR
TGDDAPQTTTIVRANSSTTTAEPLKMSYVEGLLSSNQKDLVMEEVANYHANTKDA
EVVLVEGLVPTRKHQFAQSLNYEIAKTLNAEIVFVMSQGTDTPQLEKIERELTRNSFG
GAKNTNTGVVNVKNAPVDQGRTRPDLESEIFDSSKAKVNNVPDKLQESSPLVPV
GAVPWSFDLIAATRAIDMRALHNLATINEGDINTRRVSSTFCARSIPHMLE HF RAGSL
LVTSA DRP D DL VAACLAAMNGVEIGALLLTGYYEMDA RISKLCERAFATG LPVFVMNT
NTWQT SLQSFSNLIEPVDDHERIEKVQFY VANYINADWIE S L TA S Rooms
RVTQEL T AKARGR IVL PEGDEPTVKAAIAI CAERGIATCVLGNP A ERVAA SQQV
ELGAGIEIVDPEVRESYVGRVLVELRKNKMGTETVAREQLEDNVVLGTLMELQDEVDG
LVSGAVHTTANTIRPPPLQTIKA PGSSLVSSVVFMLPEQYVYGYDCAINPDTA EQL
AEIAIQSADSAAFGIEPRVAML SYSTGTSAG SDVEKvre ATRLAQEK RDPLMDG
LPQYDAA VMA D VAKSKAPNSPV AGRATV FIFP D NT T Y KAV QRS AD LIS IG PMLQG
MKPVNDLSRGLV DDIVYTIALTAIQSAQQ"
   /EC_number="2.3.1.8"
misc_feature 2297..2327
Appendix A. Implementation of a bistable phenotypic switch

/misc_feature 2328..2358

/promoter 2359..2433

/misc_feature 2434..2463

/misc_feature 2464..2493

/RBS 2494..2528

/CDS 2529..5204

/locus_tag="b1241"

/function="enzyme; Energy metabolism, carbon: Fermentation"

/protein_id="NP_415757.1"

gene="adhE"

/transl_table=11

/translation="MAVTNVAELNALVERVKKAQREYASFTTQEQVDKIFRAAAALAAAAD
ARIPKAKMVAESGMGIVEDKVIKHNFASEYIYAYKDEKTGVLSEDDTFGITITIAE
P1GI1GCIVPTTNSTSAIFKSLISLKTRNAIIIFSHPRAKDATNKAIDIVLQLAAIAA
GAPKDLIGWDQPSVLSNALMHHDDINLILATGGPGMKAAYSSGPAIGVGAGNTP
VVIDETADIKRAVASVLMKSTFDNGVICASEQSVGVDVDYDAVRERFATHGYYLLQG
KELKAVQDVILKVGALANAAIVGQPAYKIAELAGFSVPENTKILIGEVTVDDESEPFH
AELKSPLEMRYAKDFEAVKEAKLVMAGWIGTSCLYTDQNDQPARVSYFGQKMKTA
RLIINTPASQQGIDLNYFKLAPSITLGC pharmacies are not the
MLWKHLKPSTYFRQGSLPIALDEVITGHKRALITVTDRLFNLNGYADQITSVLAAGV
ETEVFEVEADPTLSIRVKGAELANSFKPDEVIIIAGQGGPSMDAAKLMWYMHEPETFH
EELALRFMDIRKRIYKFPKMVGKAKMIAVTTTSGTGSEVTTPFAVVTDDATGQKYPFAD
YALTPDMAIDANLVMMPKSLCAFGGADAVTREAYAVSLASEFSDQGLQALQKLL
KEYLPASYHGKSNPVARERVHAAATIGAFANAFLGVCHSMASHKLGQSQHIPGHA
NLALLCNVIYANDNPTKQTAFSDQPQARRRYEAIADHLGLSAPDRTAIAKIEKL
LAWLETLKAELGIPKSIAREAIGQVEADFLANVDKLSEDAFDDQTGANPYPRIPLSELKQ
ILLDTYYGRDYEVEGAIAKKEAAPPKAKKAKSA"

/product="fused acetaldehyde-CoA
dehydrogenase/iron-dependent alcohol
dehydrogenase/pyruvate-formate lyase deactivase"

/EC_number="1.1.1.1; 1.2.1.10"

/rep_origin 5367..6174

/CDS complement(6312..7106)

/position="KanR"

/misc_feature complement(7294..7324)

/note="Geneious type: insulator"

/misc_feature complement(7325..7354)
Appendix A. Implementation of a bistable phenotypic switch

CDS complement(7355..8344)
/gene="ldhA"
/locus_tag="b1380"
/gene_synonym="ECK1377; hslF; hslI; htpH; JW1375"
/EC_number="1.1.1.28"
/function="enzyme; Energy metabolism, carbon: Fermentation"
/codon_start=1
/transl_table=11
/product="fermentative D-lactate dehydrogenase, NAD-dependent"
/protein_id="NP_415898.1"
/db_xref="GI:16129341"
/db_xref="ASAP:ABE-0004619"
/db_xref="UniProtKB/Swiss-Prot:P52643"
/db_xref="EcoGene:EG13186"
/db_xref="GeneID:946315"
/translation="MKLAVYSTKQYDKKYLQQVNESFGFELEFFDFLLTEKTAKTANG CEAVICFVNDGSRPVLEELKHKHGVKYLARCAGFNNVDLDAKEKLGLKVRVVPAYDP EAVAHAIGMMTTLNRRHAYQRTDANFSLEGTLTFTMYGKTACVIGTKIGVAML RILKFGMRLLLAFDPYPSAAAELGVEYVLPTLFSESDVISLHCPLTCPENYHLLNEA AFEMQMKNGVMVNTSRGALIDSQAIAEALKNQIGSLGMVYENERDLFFEDKSDVI QDDVFRRLSACHNVLFTGHAFLTAEALTSISQTTLQNLSNEKGETCPNELV"

RBS complement(8345..8376)
misc_feature complement(8377..8407)
/notes="Geneious type: insulator"
misc_feature complement(8408..8437)
/notes="Geneious type: insulator"
promoter complement(8438..8511)
/genes="PL(tetO)"
misc_feature complement(8512..8541)
/notes="Geneious type: insulator"
misc_feature complement(8542..8573)
/notes="Geneious type: insulator"
CDS complement(8574..9290)
/genes="GFPmut3b_RSR"
promoter complement(9342..9415)
/genes="PL(tetO)"
CDS complement(9458..10540)
/genes="lacI-RSR"
promoter complement(10577..10650)
/genes="PL(tetO)"
promoter 10669..10743
CDS 10782..11405
/genes="tetR_RSR"
promoter 11448..11522
CDS 11565..12275
Appendix A. Implementation of a bistable phenotypic switch

/gene="mCherry"
/misc_feature 12279..12309
/note="Geneious type: insulator"
/misc_feature 12310..12339
/note="Geneious type: insulator"
/promoter 12340..63

ORIGIN
1 gttgacaatt aatcatcggg ctcgtataat gttgtggaatt gttgacggat aacaatättca
61 cagccagctga cagccagctga cagccagctga cagccagctga cagccagctga cagccagctga
121 aacactaatc aaggaatctg aaggaatctg aaggaatctg aaggaatctg aaggaatctg
181 agggttaggg ctcgaggtgt ctcgaggtgt ctcgaggtgt ctcgaggtgt ctcgaggtgt
241 cgaagcttctg cgaagcttctg cgaagcttctg cgaagcttctg cgaagcttctg cgaagcttctg
301 cgagatgctg ctcgaggtgt ctcgaggtgt ctcgaggtgt ctcgaggtgt ctcgaggtgt
361 gaaatgtggt gtcgaggtgt gtcgaggtgt gtcgaggtgt gtcgaggtgt gtcgaggtgt
421 gatcgcagct gatcgcagct gatcgcagct gatcgcagct gatcgcagct gatcgcagct
481 gttcggagat cgcgtttcgg ctcggttttc cgcgtttcgg ctcggttttc cgcgtttcgg
541 gatcgcagct gatcgcagct gatcgcagct gatcgcagct gatcgcagct gatcgcagct
601 ggtgcctggc gtcggcgtgg gtcggcgtgg gtcggcgtgg gtcggcgtgg gtcggcgtgg
661 tgttctgtcc gatgtaagat gttgtggtgt gttgtggtgt gttgtggtgt gttgtggtgt
721 cagcagctgc tggggggtt ggggggtggt ggggggtggt ggggggtggt ggggggtggt
781 gcagtcgagc gttcgggggg ttcggtgggt ttcggtgggt ttcggtgggt ttcggtgggt
841 tgcgttctgc ctcggttttc ctcggttttc ctcggttttc ctcggttttc ctcggttttc
901 tcggcgcagct ggttttgcgc gccgcgtggc gcgtgtggtt gcgtgtggtt gcgtgtggtt
961 acgtggcgtt ctcggttttc ctcggttttc ctcggttttc ctcggttttc ctcggttttc
1021 gttcggggtt ctttcgcggg cgggcgcggg cgggcgcggg cgggcgcggg cgggcgcggg
1081 aatgtggtgt gtcggcgtgg gtcggcgtgg gtcggcgtgg gtcggcgtgg gtcggcgtgg
1141 tgttctgtcc gatgtaagat gttgtggtgt gttgtggtgt gttgtggtgt gttgtggtgt
1201 gtttctgggc gggggttggt gggggttggt gggggttggt gggggttggt gggggttggt
1261 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1321 tgtgcggggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1381 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1441 ctcggtgggt gtcgcgtggg gtcgcgtggg gtcgcgtggg gtcgcgtggg gtcgcgtggg
1501 tgtgcggggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1561 tgtgcggggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1621 gagaagccgct cgggggtggt cgggggtggt cgggggtggt cgggggtggt cgggggtggt
1681 ctcggtgggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1741 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1801 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1861 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1921 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
1981 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2041 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2101 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2161 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2221 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2281 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2341 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
2401 gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt gcgtggtggt
Appendix A. Implementation of a bistable phenotypic switch

2461 `actcctcgtc tcaaccaaaag caaccaacc atctccagag aacttaatat aataaggaga`
2521 `ggtccttact tattgcctgc tatttcaccg taagacgaag tagaacgaa`
2581 `gaagagcgag ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
2641 `tgctcctgccg tgcctcctgccg tgcctcctgccg tgcctcctgccg tgcctcctgccg`
2701 `acttattgcg tattgcctgc tatttcaccg taagacgaag tagaacgaa`
2761 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
2821 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
2881 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
2941 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3001 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3061 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3121 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3181 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3241 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3301 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3361 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3421 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3481 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3541 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3601 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3661 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3721 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3781 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3841 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3901 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
3961 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4021 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4081 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4141 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4201 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4261 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4321 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4381 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4441 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4501 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4561 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4621 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4681 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4741 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4801 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4861 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4921 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
4981 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
5041 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
5101 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
5161 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
5221 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
5281 `ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa ttgacatgaa`
Appendix A. Implementation of a bistable phenotypic switch
Appendix A. Implementation of a bistable phenotypic switch

11101 aaaaacagta tgaaactctc gaaaatcaat tagccttttt atgccaacaa ggtttttcact
11161 tagagaatgc attatatgca ctcagcgtct ggggcatatt tacttttaggt tcgctattggt
11221 aagatcaaga gcataacagtc gctaaagaag aaagggaaac acctactact gatagtatgc
11281 cgccattatt atcgaatatt ctcagcactt atgatcacc gatagccctt ctacgctctt
11341 tattcggcct tgaattgtac ctcctcggat tagaaatctc cgcttcacaa acttactatt gaaagtggggt
11401 cttgagagtct cggactgctct aatgctctt atatgctctgt tccccggtctg aatgagctg
11461 tttgacaatatt atcatctcggc tcgtaattag tcgacggata acaattatcact
11521 accaatcggag tattaactatat cgttaaactgt atagagagc gggatgtggt aggcaagggcg
11581 aggagtagaa cataggccatc atcatgatgct tcagtgcttc atgggtcagct atggagttgct
11641 cctgtaagcg ccacgagttc gagatcaggg gcggagggccag gggcgcccccc tacgagggca
11701 cccagaccgc caagctcagag tcgacggtatg cggcgggctt gcctctgcctt cggagacatcc
11761 tgggcccctca ggctctgtac ggcctcaagg cctatggcttg gcaccccggg cccatccccc
11821 actacttgtgaa gctgctccct cccgaggggt tcagtgcttcct ccgggttgcagct atggagttgct
11881 acggccgctgt tttgacgccg ctgcacagact cctcctggca agacggtgcc tctatctaca
11941 aggtagaagct cggccggacc aactccccct cggacggttcc cggatatgcag aagaagacta
12001 tgggcttgga ggctctccctc gacctggttt acggcggagc cggccggctgc gggcgggaga
12061 tcaagctcaac gctgaagctt cggcgagggc gcgccactcgc cggctggtact cgggacgacct
12121 acaaggaacca gaagccgccggt ctgcagtaag cctgctctgtaa cggctcagat ccggagacta
12181 tcaagctcagca cagcagttgc atcagagagc tcagtgatgc gcaagcggtgcc gaggccggcct
12241 actcagcacc gcctcggagag cggctgcttc aatgacagatc tccattctccgc gggcgggaga
12301 gctgctctgg ctgacgactt gcaggtcagcc acaacgcgtc tgaatgagctct
Appendix B

MoVE: An algorithm to identify dynamic intervention strategies to decouple growth and production phenotypes

B.1 Supplementary figures

Figure B.1: Implementation considerations for MoVE strategies. First, a set of static knockouts can be made using canonical genetic engineering techniques. Second, if the growth rate has suffered due to sub-optimality, the strain can be adaptively evolved to maximize growth rate, accumulating genomic modifications. Finally, genetic circuitry can be applied to effect a switch by eliminating flux through target valves, and achieving at least the minimum specific target (e.g. product yield).
Figure B.2: Overview of distributed mixed-integer linear programming solution. Jobs are created for each metabolite and given simulation parameters, and solved on a general purpose cluster. Each job is solved for a total of 128 core · hours or 256 thread · hours.

Figure B.3: Distribution of reaction connectivities. The connectivity of a given reaction is defined as the total number of reactions connected to all metabolites in that reaction. The reactions are sorted in ascending order.
Figure B.4: Part 1. Production envelopes for fully decoupled strategies using single valves. Log-log axes are used to highlight features at low growth rates. Production states which have near-zero growth rate are displayed as a vertical line. Production states which have a negligible production flux variability are display as X. The growth state is shown in red and the production state is shown in green.
**Figure B.5:** Part 2. Production envelopes for fully decoupled strategies using single valves. Log-log axes are used to highlight features at low growth rates. Production states which have near-zero growth rate are displayed as a vertical line. Production states which have a negligible production flux variability are display as X. The growth state is shown in red and the production state is shown in green.
Figure B.6: Part 1. Production envelopes for partially decoupled strategies using single valves. The growth state is shown in red and the production state is shown in green.
Figure B.7: Part 2. Production envelopes for partially decoupled strategies using single valves. The growth state is shown in red and the production state is shown in green. Note: methanol passed the initial screen; however, the low yield generally makes this product infeasible to produce from glucose.
Figure B.8: Clustered heat maps of co-occurring valves for fully decoupled production. Double and triple valves were identified, and an adjacency was built from their co-occurrences. This adjacency matrix was clustered using spectral clustering, and sorted by cluster density. Flux through these valves should be controlled in parallel to shift phenotypes between growth and production states.
Figure B.9: Clustered heat maps of co-occurring valves for partially decoupled production. Double and triple valves were identified, and an adjacency was built from their co-occurences. This adjacency matrix was clustered using spectral clustering, and sorted by cluster density. Flux through these valves should be controlled in parallel to shift phenotypes between growth and production states.
Figure B.10:Occurrences of each knockout in simulations targeting optimal valves, representing the sum of data from simulations for single, double and triple valve strategies. (a) Knockout occurrences for fully decoupled production. (b) Knockout occurrences for partially decoupled production strategies.
Figure B.11: Single valves identified in iMM904, a genome-scale metabolic model of *S. cerevisiae*. Strategies achieve over 70% of theoretical maximum product yield and a biomass yield of 0.001 gdw/mmol in the production state, and over 90% of theoretical maximum biomass yield in the growth state. (a) Valve occurrences in strategies identified for 61 of 84 naturally producible metabolites from glucose. (b) Core metabolic map highlighting identified valves in red.
### B.2 Supplementary tables

**Table B.1:** Targeted products in *E. coli* genome-scale model, associated reactions, and identified valves for both full and partial decoupling. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Full decoupling</th>
<th>Partial decoupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R)-Propane-1,2-diol</td>
<td>EX$_{12}$ppd-R(e)</td>
<td>FRD3</td>
<td></td>
</tr>
<tr>
<td>(S)-Propane-1,2-diol</td>
<td>EX$_{12}$ppd-S(e)</td>
<td>CS</td>
<td>PGM</td>
</tr>
<tr>
<td>1,5-Diaminopentane</td>
<td>EX$_{15}$dap(e)</td>
<td>PFL</td>
<td></td>
</tr>
<tr>
<td>3-Hydroxypropanoate</td>
<td>EX$_{3}$hpp(e)</td>
<td>—</td>
<td>PGI</td>
</tr>
<tr>
<td>4-Aminobutanoate</td>
<td>EX$_{4}$abut(e)</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>5-Dehydro-D-gluconate</td>
<td>EX$_{5}$dgln(e)</td>
<td>GAPD</td>
<td>PGM</td>
</tr>
<tr>
<td>L-alanine-D-glutamate-meso-2,6-diaminoheptanedioate</td>
<td>EX$_{LalaDgluMdap}$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L-alanine-D-glutamate-meso-2,6-diaminoheptanedioate-D-alanine</td>
<td>EX$_{LalaDgluMdapDala}$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Acetate</td>
<td>EX$_{ac}$e</td>
<td>EX$_{o2}$e</td>
<td></td>
</tr>
<tr>
<td>Acetaldehyde</td>
<td>EX$_{acald}$e</td>
<td></td>
<td>EX$_{o2}$e</td>
</tr>
<tr>
<td>4-Amino-4-deoxy-L-arabinose modified core oligosaccharide lipid A</td>
<td>EX$_{acolipa}$e</td>
<td>CS</td>
<td>PPC</td>
</tr>
<tr>
<td>O-Acetyl-L-serine</td>
<td>EX$_{acser}$e</td>
<td></td>
<td>DSERDHr</td>
</tr>
<tr>
<td>Adenine</td>
<td>EX$_{ade}$e</td>
<td>XYL12</td>
<td>PGM</td>
</tr>
<tr>
<td>Adenosine</td>
<td>EX$_{adn}$e</td>
<td>PGM</td>
<td>G6PDH2r</td>
</tr>
<tr>
<td>Agmatine</td>
<td>EX$_{agm}$e</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-Oxoglutarate</td>
<td>EX$_{akg}$e</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>D-Alanine</td>
<td>EX$_{ala-D}$e</td>
<td>R15BPK</td>
<td></td>
</tr>
<tr>
<td>L-Alanine</td>
<td>EX$_{ala-L}$e</td>
<td>EX$_{o2}$e</td>
<td></td>
</tr>
<tr>
<td>D-Alanyl-D-alanine</td>
<td>EX$_{alaala}$e</td>
<td>NDPK1</td>
<td></td>
</tr>
<tr>
<td>Allantoin</td>
<td>EX$_{alltn}$e</td>
<td></td>
<td>PDH</td>
</tr>
</tbody>
</table>
Table B.1 (continued): Targeted products in *E. coli* genome-scale model, associated reactions, and identified valves for both full and partial decoupling. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Identified valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-Acetyl-D-glucosamine (anhydrous)</td>
<td>EX_anhg(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Arginine</td>
<td>EX_arg-L(e)</td>
<td>—</td>
</tr>
<tr>
<td>L-Asparagine</td>
<td>EX_asn-L(e)</td>
<td>CS</td>
</tr>
<tr>
<td>L-Aspartate</td>
<td>EX_asp-L(e)</td>
<td>NDPK1</td>
</tr>
<tr>
<td>Cys-Gly</td>
<td>EX_cgly(e)</td>
<td>—</td>
</tr>
<tr>
<td>Citrate</td>
<td>EX_cit(e)</td>
<td>—</td>
</tr>
<tr>
<td>CO2</td>
<td>EX_co2(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>Core oligosaccharide lipid A</td>
<td>EX_colipa(e)</td>
<td>CS</td>
</tr>
<tr>
<td>L-Cysteine</td>
<td>EX_cys-L(e)</td>
<td>PGM</td>
</tr>
<tr>
<td>Cytidine</td>
<td>EX_cytd(e)</td>
<td>PDH</td>
</tr>
<tr>
<td>Dihydroxyacetone</td>
<td>EX_dha(e)</td>
<td>GLUDy</td>
</tr>
<tr>
<td>(enterobacterial common antigen)x4 core oligosaccharide lipid A</td>
<td>EX_eca4colipa(e)</td>
<td>PPC</td>
</tr>
<tr>
<td>Phosphoethanolamine KDO(2)-lipid (A)</td>
<td>EX_enlipa(e)</td>
<td>ATPS4rpp</td>
</tr>
<tr>
<td>Enterochelin</td>
<td>EX_enter(e)</td>
<td>TKT2</td>
</tr>
<tr>
<td>Ethanolamine</td>
<td>EX_etha(e)</td>
<td>—</td>
</tr>
<tr>
<td>Ethanol</td>
<td>EX_etoh(e)</td>
<td>EX_o2_e</td>
</tr>
<tr>
<td>Fe-enterobactin</td>
<td>EX_feenter(e)</td>
<td>EDA</td>
</tr>
<tr>
<td>Formate</td>
<td>EX_for(e)</td>
<td>—</td>
</tr>
<tr>
<td>Sn-Glycero-3-phosphoethanolamine</td>
<td>EX_g3pe(e)</td>
<td>—</td>
</tr>
<tr>
<td>Glycerophosphoglycerol</td>
<td>EX_g3pg(e)</td>
<td>—</td>
</tr>
<tr>
<td>D-Gluconate</td>
<td>EX_glen(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Glutamate</td>
<td>EX_glu-L(e)</td>
<td>—</td>
</tr>
</tbody>
</table>
Table B.1 (continued): Targeted products in *E. coli* genome-scale model, associated reactions, and identified valves for both full and partial decoupling. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Identified valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glycine</td>
<td>EX_gly(e)</td>
<td>—</td>
</tr>
<tr>
<td>D-Glyceraldehyde</td>
<td>EX_glyald(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>Glycerol</td>
<td>EX_glyc(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>(R)-Glycerate</td>
<td>EX_glyc-R(e)</td>
<td>—</td>
</tr>
<tr>
<td>Glycerol 3-phosphate</td>
<td>EX_glyc3p(e)</td>
<td>—</td>
</tr>
<tr>
<td>Glycolate</td>
<td>EX_glyctl(e)</td>
<td>—</td>
</tr>
<tr>
<td>Reduced glutathione</td>
<td>EX_gthrd(e)</td>
<td>—</td>
</tr>
<tr>
<td>Guanine</td>
<td>EX_gua(e)</td>
<td>ENO</td>
</tr>
<tr>
<td>L-Histidine</td>
<td>EX_his-L(e)</td>
<td>PDH</td>
</tr>
<tr>
<td>L-Homoserine</td>
<td>EX_hom-L(e)</td>
<td>—</td>
</tr>
<tr>
<td>Hexanoate (n-C6:0)</td>
<td>EX_hxa(e)</td>
<td>EX_o2-e</td>
</tr>
<tr>
<td>Hypoxanthine</td>
<td>EX_hxan(e)</td>
<td>DRPA</td>
</tr>
<tr>
<td>L-Idonate</td>
<td>EX_idon-L(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Isoleucine</td>
<td>EX_ile-L(e)</td>
<td>PDH</td>
</tr>
<tr>
<td>Indole</td>
<td>EX_indole(e)</td>
<td>—</td>
</tr>
<tr>
<td>Inosine</td>
<td>EX_ins(e)</td>
<td>PDH</td>
</tr>
<tr>
<td>KDO(2)-lipid IV(A)</td>
<td>EX_kdo2lipid4(e)</td>
<td>CS</td>
</tr>
<tr>
<td>D-Lactate</td>
<td>EX_lac-D(e)</td>
<td>EX_o2-e</td>
</tr>
<tr>
<td>L-Lactate</td>
<td>EX_lac-L(e)</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Leucine</td>
<td>EX_leu-L(e)</td>
<td>GND</td>
</tr>
<tr>
<td>KDO(2)-lipid (A)</td>
<td>EX_lipa(e)</td>
<td>CS</td>
</tr>
<tr>
<td>Cold adapted KDO(2)-lipid (A)</td>
<td>EX_lipa_cold(e)</td>
<td>—</td>
</tr>
<tr>
<td>L-Lysine</td>
<td>EX_lys-L(e)</td>
<td>CS</td>
</tr>
<tr>
<td>L-Malate</td>
<td>EX_mal-L(e)</td>
<td>FRD2</td>
</tr>
<tr>
<td>Methanol</td>
<td>EX_mehl(e)</td>
<td>—</td>
</tr>
<tr>
<td>Ornithine</td>
<td>EX_orn(e)</td>
<td>—</td>
</tr>
</tbody>
</table>
Table B.1 (continued): Targeted products in *E. coli* genome-scale model, associated reactions, and identified valves for both full and partial decoupling. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Full decoupling</th>
<th>Partial decoupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-Phenylalanine</td>
<td>EX_phe-L(e)</td>
<td>PPM</td>
<td>PDH</td>
</tr>
<tr>
<td>Protoheme</td>
<td>EX_pheme(e)</td>
<td>RPI</td>
<td>AKGDH</td>
</tr>
<tr>
<td>L-Proline</td>
<td>EX_pro-L(e)</td>
<td>MTHFC</td>
<td>—</td>
</tr>
<tr>
<td>Putrescine</td>
<td>EX_ptrc(e)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Pyruvate</td>
<td>EX_pyr(e)</td>
<td>—</td>
<td>PDH</td>
</tr>
<tr>
<td>Quinate</td>
<td>EX_quin(e)</td>
<td>PDH</td>
<td>DHAPT</td>
</tr>
<tr>
<td>L-Serine</td>
<td>EX_ser-L(e)</td>
<td>PGM</td>
<td>AKGDH</td>
</tr>
<tr>
<td>Succinate</td>
<td>EX_succ(e)</td>
<td>PPKr</td>
<td>EX_o2_e</td>
</tr>
<tr>
<td>L-Threonine</td>
<td>EX_thr-L(e)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Thymine</td>
<td>EX_thym(e)</td>
<td>TKT2</td>
<td>—</td>
</tr>
<tr>
<td>Thymidine</td>
<td>EX_thymd(e)</td>
<td>PDH</td>
<td>PGM</td>
</tr>
<tr>
<td>L-Tryptophan</td>
<td>EX_trp-L(e)</td>
<td>PDH</td>
<td>EDA</td>
</tr>
<tr>
<td>L-Tyrosine</td>
<td>EX_tyr-L(e)</td>
<td>CS</td>
<td>PGI</td>
</tr>
<tr>
<td>Uracil</td>
<td>EX_ura(e)</td>
<td>DRPA</td>
<td>—</td>
</tr>
<tr>
<td>Urea</td>
<td>EX_urea(e)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Uridine</td>
<td>EX_uri(e)</td>
<td>—</td>
<td>HEX7</td>
</tr>
<tr>
<td>L-Valine</td>
<td>EX_val-L(e)</td>
<td>EX_o2_e</td>
<td>—</td>
</tr>
<tr>
<td>Xanthine</td>
<td>EX_xan(e)</td>
<td>—</td>
<td>PGM</td>
</tr>
<tr>
<td>Xanthosine</td>
<td>EX_xtsn(e)</td>
<td>DRPA</td>
<td>TKT1</td>
</tr>
</tbody>
</table>
Table B.2: Targeted products in *S. cerevisiae* genome-scale model, associated reactions, and identified valves. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Hydroxybutyrate C4H7O3</td>
<td>EX_2hb_e</td>
<td>NADDPp</td>
</tr>
<tr>
<td>2 methylbutyl acetate C7H14O2</td>
<td>EX_2mbac_e</td>
<td>–</td>
</tr>
<tr>
<td>2 methylbutyraldehyde C5H10O</td>
<td>EX_2mbald_e</td>
<td>–</td>
</tr>
<tr>
<td>2 methyl 1 butanol C5H12O</td>
<td>EX_2mbtoh_e</td>
<td>MDH</td>
</tr>
<tr>
<td>2 methylpropanal C4H8O</td>
<td>EX_2mppal_e</td>
<td>PDHm</td>
</tr>
<tr>
<td>2 phenylethanol C8H10O</td>
<td>EX_2phethoh_e</td>
<td>MDH</td>
</tr>
<tr>
<td>3-Carboxy-3-hydroxy-4-methylpentanoate</td>
<td>EX_3c3hmp_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>3 Methylbutanal C5H10O</td>
<td>EX_3mbald_e</td>
<td>SUCD3_u6m</td>
</tr>
<tr>
<td>(S)-3-Methyl-2-oxopentanoate</td>
<td>EX_3mop_e</td>
<td>–</td>
</tr>
<tr>
<td>4-Aminobutanoate</td>
<td>EX_4abut_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>4-Aminobenzoate</td>
<td>EX_4abz_e</td>
<td>PYK</td>
</tr>
<tr>
<td>N N bisformyl dityrosine C20H22N2O8</td>
<td>EX_Nbfortyr_e</td>
<td>–</td>
</tr>
<tr>
<td>Acetate</td>
<td>EX_ac_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Acetaldehyde</td>
<td>EX_acald_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Acetic ester C4H8O2</td>
<td>EX_aces_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>2-Oxoglutarate</td>
<td>EX_akg_e</td>
<td>FUMm</td>
</tr>
<tr>
<td>L-Alanine</td>
<td>EX_ala_L_e</td>
<td>CYOR_u6m</td>
</tr>
<tr>
<td>L-Arginine</td>
<td>EX_arg_L_e</td>
<td>–</td>
</tr>
<tr>
<td>L-Asparagine</td>
<td>EX_asn_L_e</td>
<td>–</td>
</tr>
<tr>
<td>L-Aspartate</td>
<td>EX_asp_L_e</td>
<td>–</td>
</tr>
<tr>
<td>R R 2 3 Butanediol C4H10O2</td>
<td>EX_btd_RR_e</td>
<td>CYOOm</td>
</tr>
<tr>
<td>Citrate</td>
<td>EXCit_e</td>
<td>SUCD3_u6m</td>
</tr>
<tr>
<td>CO2 CO2</td>
<td>EX_co2_e</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Cysteine</td>
<td>EX_cys_L_e</td>
<td>MDH</td>
</tr>
<tr>
<td>DTTP C10H13N2O14P3</td>
<td>EX_dttp_e</td>
<td>–</td>
</tr>
<tr>
<td>Episterol C28H46O</td>
<td>EX_epist_e</td>
<td>–</td>
</tr>
<tr>
<td>Ergosterol C28H44O</td>
<td>EX_ergst_e</td>
<td>–</td>
</tr>
</tbody>
</table>
Table B.2 (continued): Targeted products in *S. cerevisiae* genome-scale model, associated reactions, and identified valves. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethanol</td>
<td>EX_ etoh_e</td>
<td>CYOOm</td>
</tr>
<tr>
<td>Fecosterol C28H46O</td>
<td>EX_ fecost_e</td>
<td>–</td>
</tr>
<tr>
<td>Formate</td>
<td>EX_ for_e</td>
<td>FBA3</td>
</tr>
<tr>
<td>Fumarate</td>
<td>EX_ fum_e</td>
<td>PDHm</td>
</tr>
<tr>
<td>Sn-Glycero-3-phosphocholine</td>
<td>EX_ g3pc_e</td>
<td>GLPT</td>
</tr>
<tr>
<td>D-Glucosamine 6-phosphate</td>
<td>EX_ gam6p_e</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Glutamine</td>
<td>EX_ gln_L_e</td>
<td>FUMm</td>
</tr>
<tr>
<td>L-Glutamate</td>
<td>EX_ glu_L_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Glyoxylate</td>
<td>EX_ glx_e</td>
<td>–</td>
</tr>
<tr>
<td>Glycine</td>
<td>EX_ gly_e</td>
<td>–</td>
</tr>
<tr>
<td>Glycerol</td>
<td>EX_ glyc_e</td>
<td>NADDPp</td>
</tr>
<tr>
<td>Guanine</td>
<td>EX_ gua_e</td>
<td>–</td>
</tr>
<tr>
<td>H2O H2O</td>
<td>EX_h2o_e</td>
<td>PYK</td>
</tr>
<tr>
<td>H+</td>
<td>EX_h_e</td>
<td>–</td>
</tr>
<tr>
<td>L-Histidine</td>
<td>EX_his_L_e</td>
<td>HSDxi</td>
</tr>
<tr>
<td>Hypoxanthine</td>
<td>EX_hxan_e</td>
<td>–</td>
</tr>
<tr>
<td>Isoamyl acetate C7H14O2</td>
<td>EX_iamac_e</td>
<td>SUCD1m</td>
</tr>
<tr>
<td>Isoamyl alcohol C5H12O</td>
<td>EX_iamoh_e</td>
<td>SUCD3_u6m</td>
</tr>
<tr>
<td>Isobutyl acetate C6H12O2</td>
<td>EX_ibutac_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Isobutyl alcohol C4H10O</td>
<td>EX_ibutoh_e</td>
<td>CYOOm</td>
</tr>
<tr>
<td>Indole 3 acetaldehyde C10H9NO</td>
<td>EX_id3acald_e</td>
<td>SUCD3_u6m</td>
</tr>
<tr>
<td>L-Isoleucine</td>
<td>EX_ ile_L_e</td>
<td>–</td>
</tr>
<tr>
<td>Indole 3 ethanol C10H11NO</td>
<td>EX_ind3eth_e</td>
<td>ALCD2x_copy2</td>
</tr>
<tr>
<td>D-Lactate</td>
<td>EX_lac_D_e</td>
<td>GAPD</td>
</tr>
<tr>
<td>L-Lactate</td>
<td>EX_lac_L_e</td>
<td>GAPD</td>
</tr>
<tr>
<td>Lanosterol C30H50O</td>
<td>EX_lanost_e</td>
<td>SUCD1m</td>
</tr>
<tr>
<td>L-Leucine</td>
<td>EX_leu_L_e</td>
<td>SUCD3_u6m</td>
</tr>
<tr>
<td>L-Lysine</td>
<td>EX_ lys_L_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>L-Malate</td>
<td>EX_ mal_L_e</td>
<td>PDHm</td>
</tr>
</tbody>
</table>
Table B.2 (continued): Targeted products in *S. cerevisiae* genome-scale model, associated reactions, and identified valves. Dash indicates no solution was found.

<table>
<thead>
<tr>
<th>Metabolite</th>
<th>Reaction</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-Methionine</td>
<td>EX_met_L_e</td>
<td>GLUDxi</td>
</tr>
<tr>
<td>Nicotinamide adenine dinucleotide phosphate</td>
<td>EX_nadp_e</td>
<td>AKGDbm</td>
</tr>
<tr>
<td>Ammonium</td>
<td>EX_nh4_e</td>
<td>–</td>
</tr>
<tr>
<td>Oxaloacetate</td>
<td>EX_oaa_e</td>
<td>PDHm</td>
</tr>
<tr>
<td>Ornithine</td>
<td>EX_orn_e</td>
<td>NADDPp</td>
</tr>
<tr>
<td>Phenylacetaldehyde</td>
<td>EX_pacald_e</td>
<td>PYK</td>
</tr>
<tr>
<td>Adenosine 3',5'-bisphosphate</td>
<td>EX_pap_e</td>
<td>TKT1</td>
</tr>
<tr>
<td>L-Phenylalanine</td>
<td>EX_phe_L_e</td>
<td>–</td>
</tr>
<tr>
<td>Phenethyl acetate C10H12O2</td>
<td>EX_pheac_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Phosphate</td>
<td>EX_pi_e</td>
<td>–</td>
</tr>
<tr>
<td>(R)-Pantothenate</td>
<td>EX_pnto_R_e</td>
<td>ABTt</td>
</tr>
<tr>
<td>L-Proline</td>
<td>EX_pro_L_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Putrescine</td>
<td>EX_ptrc_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Pyruvate</td>
<td>EX_pyr_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>D-Sorbitol</td>
<td>EX_sbt_D_e</td>
<td>CYOR_u6m</td>
</tr>
<tr>
<td>L-Serine</td>
<td>EX_ser_L_e</td>
<td>PDHm</td>
</tr>
<tr>
<td>Sulfite</td>
<td>EX_so3_e</td>
<td>FBA3</td>
</tr>
<tr>
<td>Spermidine</td>
<td>EX_spmd_e</td>
<td>MDH</td>
</tr>
<tr>
<td>Succinate</td>
<td>EX_succe_e</td>
<td>CYOR_u6m</td>
</tr>
<tr>
<td>L-Threonine</td>
<td>EX_thr_L_e</td>
<td>HEX4</td>
</tr>
<tr>
<td>Thymine C5H6N2O2</td>
<td>EX_thym_e</td>
<td>–</td>
</tr>
<tr>
<td>L-Tryptophan</td>
<td>EX_trp_L_e</td>
<td>PYK</td>
</tr>
<tr>
<td>Tetradecanoate (n-C14:0)</td>
<td>EX_ttdca_e</td>
<td>MDH</td>
</tr>
<tr>
<td>L-Tyrosine</td>
<td>EX_tyr_L_e</td>
<td>SUCD2_u6m</td>
</tr>
<tr>
<td>Urea CH4N2O</td>
<td>EX_urea_e</td>
<td>–</td>
</tr>
<tr>
<td>L-Valine</td>
<td>EX_val_L_e</td>
<td>CYOOm</td>
</tr>
<tr>
<td>Xanthine</td>
<td>EX_xan_e</td>
<td>–</td>
</tr>
<tr>
<td>Zymosterol C27H44O</td>
<td>EX_zymst_e</td>
<td>SUCD2_u6m</td>
</tr>
</tbody>
</table>
Appendix C

Impact: a python framework for data analysis workflows to interpret microbial physiology

C.1 Supplementary information

<table>
<thead>
<tr>
<th>Table C.1: Default titers data format, typical of HPLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>glucose</td>
</tr>
<tr>
<td>substrate</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,0</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,3</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,6</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,9</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,12</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,15</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,24</td>
</tr>
<tr>
<td>strain1,plasmid1,,1,24.1</td>
</tr>
</tbody>
</table>
### Table C.2: Spectromax OD data format

<table>
<thead>
<tr>
<th>#BLOCKS= 1</th>
<th>Plate#1</th>
<th>1.3</th>
<th>PlateFormat</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time(hh:mm:ss)</td>
<td>Temperature(C)</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0:00</td>
<td>37.00</td>
<td>0.1467</td>
<td>0.1375</td>
<td>0.148</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1269</td>
<td>0.1336</td>
<td>0.137</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1369</td>
<td>0.122</td>
<td>0.1536</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1509</td>
<td>0.1637</td>
<td>0.1592</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.157</td>
<td>0.1484</td>
<td>0.1638</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1814</td>
<td>0.1928</td>
<td>0.1779</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0669</td>
<td>0.0623</td>
<td>0.0691</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0791</td>
<td>0.0998</td>
<td>0.0598</td>
</tr>
<tr>
<td>15:00</td>
<td>37.00</td>
<td>0.164</td>
<td>0.1594</td>
<td>0.1652</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1377</td>
<td>0.1462</td>
<td>0.1514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1508</td>
<td>0.1255</td>
<td>0.1858</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1701</td>
<td>0.1831</td>
<td>0.1789</td>
</tr>
</tbody>
</table>

### Table C.3: Default kinetic data format, typical of plate readers.

<table>
<thead>
<tr>
<th>A</th>
<th>Identifier</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>strain:MG</td>
<td>ko:pgi</td>
<td>media_base:LIMS</td>
<td>rep:1</td>
<td>0.1743</td>
</tr>
<tr>
<td>C</td>
<td>strain:MG</td>
<td>ko:pgi</td>
<td>media_base:LIMS</td>
<td>rep:2</td>
<td>0.1649</td>
</tr>
<tr>
<td>D</td>
<td>strain:MG</td>
<td>ko:pgi</td>
<td>media_base:LIMS</td>
<td>rep:3</td>
<td>0.1649</td>
</tr>
<tr>
<td>E</td>
<td>strain:MG</td>
<td>ko:pgi</td>
<td>media_base:LIMS</td>
<td>rep:4</td>
<td>0.1251</td>
</tr>
</tbody>
</table>
Figure C.1: Schematic of a growth curve with death phase highlighted in blue, and exponential highlighted in orange.