Universidade do Minho
Escola de Engenharia

Paulo Jorge Lopes Maia da Silva

Algorithms and Tools for *in silico* Design of Cell Factories

January 2015

Esta investigação foi financiada pela Fundação para a Ciência e Tecnologia através da concessão de uma bolsa de doutoramento (SFRH/BD/61465/2009), co-financiada pelo POPH – QREN – Tipologia 4.1 – Formação Avançada – e comparticipado pelo Fundo Social Europeu (FSE) e por fundos nacionais do Ministério da Ciência, Tecnologia e Ensino Superior (MCTES).

QUALIFICAR É CRESCER.
QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013
Governo da República Portuguesa
UNIÃO EUROPEIA Fundo Social Europeu

Paulo Jorge Lopes Maia da Silva **Algorithms and Tools for *in silico* Design of Cell Factories**

UMinho | 2015



Universidade do Minho
Escola de Engenharia

Paulo Jorge Lopes Maia da Silva

Algorithms and Tools for *in silico* Design of Cell Factories

PhD thesis in Bioengineering

This work was realized under the supervision of:
Prof. Isabel Cristina de Almeida Pereira da Rocha
Prof. Miguel Francisco de Almeida Pereira da Rocha

January 2015

Autor: Paulo Jorge Lopes Maia da Silva

E-mail: pmaia@silicolife.com
CC: 12097790

Título da tese:
Algorithms and Tools for *in silico* Design of Cell Factories

Orientadores:
Prof. Isabel Cristina de Almeida Pereira da Rocha
Prof. Miguel Francisco de Almeida Pereira da Rocha

Ano de conclusão: 2015

Programa Doutoral em Bioengenharia - MIT Portugal

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, 30 de Janeiro de 2015

Paulo Jorge Lopes Maia da Silva

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, January 30, 2015

Paulo Jorge Lopes Maia da Silva

ACKNOWLEDGEMENTS/AGRADECIMENTOS

É difícil encontrar as palavras certas para exprimir toda a gratidão que sinto neste momento. Como não sou poeta, vou reservar este espaço para agradecer, do fundo do coração, a todas as pessoas que, de uma forma ou de outra, contribuíram para que esta tese se tenha tornado uma realidade.

Em primeiro lugar, aos meus orientadores, Doutora Isabel Rocha e Doutor Miguel Rocha. Obrigado por apostarem em mim, obrigado pela amizade, sabedoria e apoio que me deram. Obrigado por não terem desistido de mim. Sem vocês eu seria um programador anónimo a trabalhar numa qualquer empresa genérica e não um membro feliz da nossa SilicoLife.

Ao Centro de Engenharia Biológica da Universidade do Minho por ter aceitado ser a minha instituição de acolhimento, à Fundação para a Ciência e Tecnologia pela atribuição da bolsa de doutoramento que tornou este trabalho possível (SFRH/BD/61465/2009) e ao Programa MIT Portugal por me ter permitido fazer parte do seu Programa Doutoral em Bioengenharia.

A todo os colegas e amigos do BioSystems, grupo de investigação no qual estive integrado durante estes anos. Um agradecimento especial ao Doutor Eugénio Ferreira por estar sempre disponível e por toda a motivação que me deu. Um abraço ao Rui e ao Bonito, companheiros de corrida no fim da jornada.

Aos meus amigos e colegas da SilicoLife, tão belo projeto que nasceu durante esta aventura. Em particular ao Simão e ao Vilaça por serem meus amigos, meus sócios e duas das pessoas mais inteligentes que conheço. Obrigado aos meus colegas de doutoramento, Rafael e Pedro, pela amizade,

discussões interessantes e partilha de conhecimento. À Sónia pela amizade e disponibilidade para me ensinar tanta coisa em tão pouco tempo, e ao Hugo por toda a amizade, motivação e força.

Obrigado a todos os meus amigos, em particular ao Artur e ao João, que fizeram parte da minha vida ao longo destes anos, e sem os quais esta teria muito menos prazer. Obrigado ao Bruno (estás longe mas não me esqueço de ti) e ao Jorjão porque estás no meu coração. Ao Paulo Miguel e ao David porque com tantas voltas que a vida dá, acabamos sempre por nos reencontrar.

Obrigado à Dila, ao Francisco e à Lia por me terem recebido e aceitado tão bem no seio da vossa família. Obrigado por me aturarem, alimentarem e por me fazerem sentir em casa todos os dias.

À minha família, por existirem e estarem sempre cá para mim. Obrigado Mãe, Patrícia, Bárbara e obrigado aos meus sobrinhos patetas, Lara e Tiago. Obrigado Avó e Avô por terem esperado que eu terminasse isto...

Para ti Tatiana, não há palavras suficientes. Obrigado por aturares as minhas patetices, os meus cozinhados e os meus stresses. Obrigado pela paciência, dedicação e carinho, em particular nestes últimos meses difíceis. Obrigado por seres a minha confidente, a minha companheira, a minha casa...

Um obrigado final ao meu pai, onde quer que estejas.

Esta tese é dedicada à minha mãe - Ercília, às minha irmãs - Patrícia e Bárbara e à minha companheira - Tatiana.

ABSTRACT

Algorithms and Tools for *in silico* Design of Cell Factories

The progressive shift from chemical to biotechnological processes is one of the pillars of the 21^{st} century industrial biotechnology. Projections from the Organization for Economic Co-operation and Development estimate that, within the next two decades, about 35% of the production of chemicals will be guaranteed by biotechnological processes. The development of efficient cell-factories, capable of outperforming current chemical processes, is vital for this leap to happen. The development of constraint-based models of metabolism and rational computational strain optimization algorithms (CSOMs) hold the promise to accelerate these efforts.

Here, we aim to provide an in depth and critical review of the currently available constraint-based CSOMs, their strengths and limitations, as well as to discuss future trends in the field. Then, we cover in detail the main tasks in strain design and provide a taxonomy of the main CSOMs. These are presented in detail and their features and limitations are explored.

One of the identified problems is their limited offering of trade-off solutions of biotechnological objectives (e.g. overproducing desired compounds or minimizing the cost of solutions) versus cellular objectives (e.g. maximizing biomass). To tackle this problem we developed an evolutionary multi-objective (MO) framework for strain optimization capable of finding

high-quality, trade-off solutions that can be explored by metabolic engineering experts.

Also, the majority of the strain optimization algorithms rely on phenotype prediction methods based on debatable biological assumptions. We verified that, for a large percentage of solutions generated by a CSOM using one phenotype prediction method, the results would not hold when simulated with an alternative method. Leveraging on the previously developed framework and driven by the MO nature of this problem, we proposed a tandem approach capable of finding strain designs that comply with the assumptions of distinct phenotype prediction methods, validating the approach with multiple case studies.

Finally, all the algorithms developed during this work are made available in the form of an open and flexible software framework. This framework is a powerful tool for both common users, interested in exploring the available methods, and experienced programmers which are able to easily extend it to support new features.

RESUMO

Algorithms and Tools for *in silico* Design of Cell Factories

A conversão de processos químicos em processos biotecnológicos é um dos grandes objetivos da biotecnologia industrial para o século XXI. A Organização para a Cooperação e Desenvolvimento Económico estima que, nas próximas duas décadas, cerca de 35% da produção de compostos químicos sejam assegurados por processos biotecnológicos. O desenvolvimento de fábricas celulares eficientes, capazes de superar o rendimento dos atuais processos químicos, é vital para que este avanço seja possível. O desenvolvimento de modelos metabólicos e algoritmos para otimização de estirpes (AOEs), é uma das grandes esperanças para acelerar estes esforços.

Neste trabalho, pretendemos efetuar uma revisão aprofundada dos AOEs atuais baseados na modelação por restrições, analisar os seus pontes fortes e limitações, e discutir temas de interesse futuro na área. De seguida, estudamos em detalhe os tipos de estratégias comuns para o desenho de estirpes e formulamos uma taxonomia para os principais AOEs. Estes são avaliados em detalhe e as suas características principais são devidamente exploradas.

Um dos problemas identificados prende-se com a sua oferta limitada de soluções de compromisso entre objetivos industriais (como produzir em excesso um composto de interesse, ou reduzir o custo de implementar uma solução) e objetivos celulares (como a maximização do crescimento). Para

enfrentar este problema, desenvolvemos uma plataforma para otimização de estirpes baseada em computação evolucionária multiobjetivo, capaz de encontrar soluções de compromisso de elevada qualidade, que podem ser exploradas por peritos em engenharia metabólica.

Para além disso, a grande maioria dos AOEs baseia-se em métodos de previsão de fenótipos que, por sua vez, são construídos sobre assunções biológicas discutíveis. Verificamos que uma grande percentagem das soluções geradas por um AOE, usando um método de previsão de fenótipos, deixaria de ser válida quando simulada com um método alternativo. Tirando partido da plataforma desenvolvida anteriormente e motivados pela natureza multiobjetivo deste problema, propusemos uma abordagem capaz de encontrar estirpes que respeitassem as assunções de diferentes métodos de previsão de fenótipos. Esta abordagem foi validada com vários casos de estudo.

Por fim, todos os algoritmos desenvolvidos ao longo deste trabalho são disponibilizados sob a forma de uma aplicação de software aberto. Esta constitui uma ferramenta poderosa, tanto para utilizadores comuns interessados em explorar os métodos disponibilizados, como para programadores experientes que podem estendê-la facilmente com novos métodos.

TABLE OF CONTENTS

Agradecimentos/Acknowledgements **v**

Abstract **vii**

Resumo **ix**

Table of Contents **xi**

List of Figures **xv**

List of Tables **xxi**

List of Algorithms **xxiii**

1 Introduction **1**
1.1 Context and Motivation 3
1.2 Research Aims 4
1.3 Outline of the Thesis 5
References 7

2 *In silico* Constraint-based Strain Optimization Methods: The Quest for Optimal Cell Factories **9**
2.1 Introduction 11
2.2 Constraint-based Modeling: Concepts and Methods 15
2.2.1 Constraint-based Models 15
2.2.2 Constraint-based Phenotype Prediction 20
2.2.3 Unbiased Characterization of the Flux Cone: pathway analysis 24
2.3 Computational Strain Optimization Methods 27
2.3.1 Computational Strain Optimization Tasks 28
2.3.2 A Taxonomy for Computational Strain Optimization Methods. 32
2.3.3 Comparative Analysis of the different classes of CSOMs 53
2.4 Applications and Discussion 56
2.5 Concluding Remarks 64

References 67

3 MOStrain: An Evolutionary Multi-Objective Framework for Strain Optimization 91
3.1 Introduction 93
3.2 Methods 94
3.2.1 Multi-objective optimization 94
3.2.2 Multi-objective and Single-objective Algorithms . . . 96
3.2.3 Representation scheme and operators 98
3.2.4 Solution decoding and evaluation 99
3.2.5 Case studies and experimental setup 102
3.2.6 Performance metrics 104
3.2.7 Generation of reference sets 105
3.3 Results and Discussion 107
3.3.1 Results for product and biomass optimization 107
3.3.2 Minimizing the number of knock-outs 113
3.4 Conclusions 122
References 123

4 Identification of Robust Strain Designs via Multiple Phenotype Prediction 127
4.1 Introduction 129
4.2 Effects of phenotype prediction methods over strain optimization 130
4.2.1 Methods 130
4.2.2 Results 131
4.3 Robust strain optimization by means of tandem phenotype prediction 140
4.3.1 Methods 140
4.3.2 Results 141
4.3.3 LMOMA solutions act as a chaperone for the identification of pFBA-valid solutions in tandem pFBA/LMOMA optimization 147
4.3.4 Tandem optimization yields robust strain optimization designs 151
4.4 Conclusions 152
References 155

5 Development of Software Tools 157
5.1 Introduction 159
5.2 Application Programming Interface Layer 159

5.2.1 Metabolic3 . . . 160
5.2.2 Solvers . . . 169
5.2.3 Jecoli3 . . . 169
5.2.4 BioComponents . . . 171
5.2.5 Tools for batch optimization and analysis . . . 171
5.3 Graphical User Interface Layer: OptFlux . . . 178
5.4 Detailed Methods . . . 185
5.4.1 Strength Pareto Evolutionary Algorithm 2 (SPEA2) . 186
5.4.2 Evolutionary Algorithm (EA) . . . 189
5.4.3 Simulated Annealing (SA) . . . 190
5.4.4 Phenotype Evaluation . . . 191
5.5 Implementation and Availability . . . 193
5.6 Conclusions . . . 193
References . . . 195

6 Conclusions and Future Work **197**
6.1 General Conclusions . . . 199
6.2 Topics for Future Work . . . 201
References . . . 203

LIST OF FIGURES

2.1 Diagram of the various components commonly found in constraint-based metabolic (and integrated metabolic/regulatory) genome-scale models. The example includes a) - a toy network composed by 10 reactions, 6 metabolites, 8 genes and 2 transcription factors; b) - the corresponding stoichiometric matrix; and c) - the corresponding gene-protein-reaction and transcriptional regulation rules. 21
2.2 The flux cone. In the example, a) - the admissible flux space is defined by the steady-state and reversibility constraints; b) - an objective function is imposed (subjected to the previous contraints) defining an optimization problem; and c) - further constraints are imposed to redirect the flux to a desired region of the flux cone. 25
2.3 Computational strain optimization tasks. In the example, a) - 3 gene deletions force the flux through reactions producing the desired compounds; b) - the inclusion of two heterologous genes allow the production of an intermediary compound and subsequent excretion of the desired product; c) - the overexpression of two enzymes allow the excess formation of compound B which is subsequently excreted; and d) - the enzyme catalyzing the transport reaction R_3 is swapped by an heterologous enzyme using NADH. The deletion of a membrane oxidoreductase enzyme creates an excess of NADH that can be used by the new transport reaction to excrete compound F. 31
2.4 Simplified example of the coupling strategy introduced by OptKnock, allowing the transformation of a bi-level MILP into a single-level one. The top region depicts the bi-level MILP problem (left) and the dual of the inner problem (right). For each constraint in the primal, there are one or more dual equivalents. Using strong duality theory and accumulation of constraints, the single-level MILP formulation is attained (bottom). 34
2.5 Workflow of generic metaheurist CSOM. The separation of the several layers (model/phenotype prediction/strain optimization) is easily visible. 44

2.6 Properties of CSOMs and their relationships. The target plot is sectioned into 4 discrete quadrants regarding scalability and exactness. Methods in the upper region are considered more scalable while methods in the bottom region are considered less scalable. Similarly, the methods located in the right region of the figure guarantee that, if it exists, the optimal solution for the specified objective function is always found, while the methods in the left region, being inherently heuristic, do not. Other features are indicated in the legend (bottom of the figure). 54

3.1 Overview of the developed algorithms. The upper region shows the major steps of the three algorithms (See details in 5.4.1, 5.4.3 and 5.4.2). The evaluation gray box illustrates the processes of solution decoding, from Gene Knockout Sets (GKS) to Reaction Knockout Sets (RKS) (upper-left), phenotype simulation showing the added constraints (bottom) and fitness evaluation (details in 5.4.4) for both MO and SO cases (upper-right). . . . 99

3.2 Coverage plots for the bi-objective case studies in aerobic (a) and anaerobic (b) conditions, showing the PF of each method overlaying the global reference front R (in black). Areas not covered (black) show regions where the method did not reach optimal solutions (as compared to R). White dots represent solutions. . 109

3.3 Boxplots for the bi-objective case studies, representing both (a) the best BPCY ($mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$) and (b) the best CYIELD ($C{-}mmol\ product\ .\ C{-}mmol\ substrate^{-1}$) obtained in each run. 111

3.4 Monotonic plots for the bi-objective case studies, showing (a) the best BPCY ($mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$) and (b) the best CYIELD ($C-mmol\ product\ .\ C-mmol\ substrate^{-1}$) obtained by each algorithm, as a function of the number of gene deletions. 112

3.5 Monotonic plots for the tri-objective case studies, showing (a) the best BPCY ($mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$) and (b) the best CYIELD ($C{-}mmol\ product\ .\ C{-}mmol\ substrate^{-1}$) obtained by each algorithm, as a function of the number of gene deletions. 115

3.6 3D scatter plots for the aerobic succinate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident. 117
3.7 3D scatter plots for the aerobic lactate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident. 118
3.8 3D scatter plots for the aerobic pyruvate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident. 119

4.1 Convergence plots for the EA and SA algorithms applied to the aerobic production of lactate and succinate from glucose. The solid lines indicates the means of 30 runs, while the color-shaded areas indicate the standard deviation. The dashed lines represent the maximum value for each algorithm and problem. 133
4.2 Distribution of the Jaccard distances from the solutions flux distributions to the wild-type flux distributions for the aerobic production of lactate (a) and succinate (b) from glucose. Red bins and blue bins represent solutions generated by LMOMA and pFBA, respectively. Solutions were re-simulated with pFBA (top in each plot) and LMOMA (bottom in each plot). 135
4.3 Venn-like diagrams for checking the inter-method validity of the solutions. From left to right, solutions that are BPCY-valid for: only pFBA, both pFBA and LMOMA, and only LMOMA. . . . 137
4.4 BPCY ($mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$) boxplots for the Lactate (a) and Succinate (b) case studies. Solutions generated with pFBA (blue,right) and LMOMA (red,left) are re-simulated using both methods (x-axis). 138
4.5 Scatter plots showing BPCY ($mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$) values obtained by LMOMA(red) and pFBA(blue) generated solutions when simulated using pFBA (x-axis) and LMOMA (y-axis) for the lactate (a) and succinate (b) case studies. 139

4.6 Distribution of the Jaccard distances from the solutions flux distributions to the wild-type flux distributions for the aerobic production of lactate (a) and succinate (b) from glucose. All the solution were re-simulated with pFBA (top in each plot) and LMOMA (bottom in each plot). 142
4.7 Venn-like diagrams for checking the inter-method validity of the solutions. From left to right, solutions that are BPCY-valid for: only pFBA, both pFBA and LMOMA, and only LMOMA. . . . 143
4.8 BPCY Boxplots for the Lactate (a) and Succinate (b) case studies obtained by the tandem approach. Solutions are re-simulated with both methods (x-axis). 144
4.9 Scatter plots showing BPCY values obtained by the tandem approach generated solutions when simulated using pFBA (x-axis) and LMOMA (y-axis) for the lactate (a) and succinate (b) case studies. 146
4.10 Illustration of LMOMA-pFBA tandem optimization. Solution spaces with few knockouts contain no valid pFBA solutions while some LMOMA solutions can be found. In the much larger, many-knockouts, solution space, pFBA, LMOMA and pFBA/LMOMA solutions are found. 147

5.1 Overview of the developed libraries. The GUI layer is provided by the OptFlux workbench whereas the API layer is composed by a total of 5 libraries: *Metabolic3*, *BioComponents*, *MetabolicAnalysis*, *Solvers* and *Jecoli3*. 161
5.2 The API layer highlighting the Model, Simplification and Essentiality / Targets packages, as well as the interaction with the BioComponents library. 162
5.3 The API layer highlighting the *Phenotype Prediction* and *Strain Optimization* packages of the *Metabolic3* library and their interaction with the *Solvers* and *Jecoli3* libraries. 166
5.4 The *tools for batch optimization and analysis* are composed by three components. The *Configuration* and *CommandLine tools* from the *Metabolic3* library and the *MetabolicAnalysis* library. . 172
5.5 Example of a configuration file (a) where 5 variables are combined, some of which have multiple assignments. The tree of possible combinations is displayed in b) with a particular state highlighted (state 1 - dotted purple area). 175
5.6 Structure of a simple analysis workflow. Three tasks are depicted, each declaring their expected inputs and outputs. 177

5.7 A simple use case scenario for the a user interaction with the StrainOptimization plug-in. The three main MVC components and their AIBench counterparts are depicted: the user interacts with the `StrainOptimizationOperationGUI` (a view) which invokes the `StrainOptimizationOperation` (controller). The controller generates the `StrainOptimizationResultDatatype` (model) which in turn updates the views. 180
5.8 Screenshot of the strain optimization configuration dialog. . . . 181
5.9 Screenshots of the progress dialogs presented to the user during optimization. When a single objective function is selected the x-axis represents the number of function evaluations and the y-axis the objective function, resulting in a convergence plot (a). When multiple objective functions are selected both axis are used to represent objective functions (which can be interchangeable), resulting in a pareto plot (b). 183
5.10 Screenshot of the strain optimization summary view. 184
5.11 Screenshot of the strain optimization results view. 185

LIST OF TABLES

2.1 Some publicly available GSMMs. Domain: E - Eukarya, B - Bacteria, A - Archaea. Reconstruction data includes number of reconstructions (#), maximum number of functionally annotated genes (G), metabolites (M) and reactions (R). 18
2.2 Experimental validation of CSOMs found in the current literature. 59
2.3 Industrial patent application/grants of microorganisms, products or processes referring the use of at least one CSOM. 60

3.1 Size of the reference sets (synthetic) and the aggregated Pareto fronts (for each algorithm) in all the bi-objective case studies. . 107
3.2 Performance Indexes for the bi-objective case-studies. For each case, the best performance is shown in bold. 108
3.3 C-Measures for the bi-objective case studies. For each case, the best performance is shown in bold. 110
3.4 Size of the reference sets (synthetic) and the aggregated Pareto fronts (for each algorithm) in the tri-objective case studies. . . . 113
3.5 Performance Indexes results for the aerobic, tri-objective studies. 114
3.6 C-Measures for the aerobic, tri-objective case studies. 114
3.7 Best CYIELDs for the tri-objective case studies. Fitnesses considering a maximum of 6 knock-outs are also displayed. 120
3.8 Most frequently selected genes (in () the percentage of solutions where they appear). List with the 10 most frequently selected genes for deletion for the SPEA2 algorithm in the succinate case study, both for the 2 and 3 objectives approaches, when considering solutions able to reach at least 75% of the top BPCY for that case study. The full list for all algorithms and case studies can be accessed in the supplementary material. 120

4.1 Size of the merged solution sets (EAs and SAs) for the various case studies. Target products: Lac. - lactate, Suc. - succinate, Pyr. - pyruvate, Phe. - phenylalanine, Gly. - glycerol 132
4.2 Size of the solution sets for the various case studies, using the tandem optimization approach. 141

4.3 Analysis of the sub-solution space of solution s_{Φ}. For each size of sub-set of s_{Φ}, the best BPCY and percentage of valid solutions, for both pFBA and LMOMA are presented. BPCY units are *mmol product . mmol substrate*$^{-1}$. h^{-1} 149
4.4 Percentage of FBA-robust solutions in all the case studies. . . . 152

5.1 Summary of the main contributions regarding software development during this work. 160

LIST OF ALGORITHMS

1 SPEA2 main loop 186
2 EA main loop 189
3 SA main loop 190
4 Annealing Schedule 191
5 Phenotype Evaluation Function 191
6 Evaluate single solution 192
7 Archive Notification (Single Solution) 192
8 Archive Notification (Population) 193

CHAPTER 1

INTRODUCTION

In the context of Metabolic Engineering (ME), Computational Strain Optimization Methods (CSOMs) deal with the problem of finding optimal modifications to an organisms' genotype, to promote the overproduction of compounds of interest and minimize the excretion of undesired by-products. This thesis is focused on the analysis of existing CSOMs and in the design and implementation of alternatives to the current metaheuristic CSOMs. All the tools developed during this work are made available to the ME community in an accessible and extensible framework.

This chapter provides an introduction to this work, addressing its motivation, highlighting the main objectives and presenting the structure of this text.

1.1 Context and Motivation

The concept of metabolic pathway manipulation towards desirable behavior is not new. Early methods relied mostly on the use of mutagenesis and strain selection techniques, with notable examples coming from the production of aminoacids, vitamins or antibiotics [1]. However, with the increasingly demanding industrial requirements, the need to resort to more rational approaches became evident. The development of genetic engineering brought ways to more precisely modify specific genes/enzymes, thus paving the way towards the more rational introduction of direct genetic changes to create desirable strains [2].

Moreover, the recent advances in genome sequencing technologies which culminated in the development of next generation sequencing technologies [3] and semi-automated annotation techniques, made the availability of a large number of fully annotated microbial genomes a reality. This information explosion also accelerated the development of genome-scale metabolic models (GSMMs) for a large number of organisms [4]. The development of phenotype prediction methods supporting distinct genetic and environmental conditions, including the well-known method of Flux Balance Analysis [5–7], combined with GSMMs, brought powerful tools to predict the behavior of microbial strains and support rational ME efforts.

Backed by these efforts, the development of strain design methods, where bioengineering objectives could be rationally addressed, became paramount. In 2003, Burgard and co-workers released OptKnock [8], which would become the basis for a large portion of the constraint-based strain design methods for the following decade. Such *in silico* ME approaches are able to propose genetic changes based on computational simulation and mathematical optimization methods. While these approaches have provided good

results, they are still limited since they usually return a single solution to the problem or, in some cases, sets of solutions with limited variability. Among all, meta-heuristic CSOMs [9] provide the most diverse solutions, but typically those follow similar strategies to maximize the selected objective function. To overcome these limitations, information from multiple criteria is often included in a single objective function, which can introduce undesired biases in the sampling process.

An analysis of available CSOMs, reveals several shortcomings of the current methods. As an example, defining an objective function for a CSOM can be a difficult task. Because of this, and since models lack critical information to improve the quality of the predictions, the solutions proposed by most CSOMs are not only overly-optimistic, but sometimes physiologically impossible. Multi-objective (MO) approaches search for optimal trade-offs of solutions instead of a single optimal solution, thus providing a valuable tool for expert researchers, allowing them to opt for compromise solutions believed to have better chances of working *in vivo*.

1.2 Research Aims

In this context, the aim of this work will be to develop novel CSOMs, based on an MO perspective, using algorithms with given proofs from Evolutionary Computation, which can improve the current state of the art methods finding competitive and robust designs, and providing open computational tools for the ME community.

With this overall aim in mind, several scientific/technological objectives have been defined for this work:

- To conduct a comprehensive analysis on the currently available CSOMs

and related literature, highlighting their main similarities and differences, as well as their strengths and limitations, also proposing a new classification system for the different types of CSOMs;

- Aggregate and analyze successful applications of CSOMs in real-world case studies and discuss their relevance in a future bio-based economy;

- Suggest and implement new approaches for strain optimization. These will tackle some of the major problems identified in other CSOMs:

  - addressing their limited offering of trade-off solutions of biotechnological objectives (e.g. overproducing desired compounds or minimizing the cost of solutions) versus cellular objectives (e.g. maximizing biomass) through an evolutionary-based MO approach;

  - addressing their over-dependency on the biological assumptions of their inner phenotype prediction methods through tandem strain optimization techniques.

- Integrate the developed packages into the OptFlux suite [10], a software workbench for ME that includes a straightforward graphical user interface, developed by the research group where this work was integrated.

1.3 Outline of the Thesis

This manuscript has been structured in six chapters addressing all of the previously stated aims.

The thesis begins in the current chapter (**Chapter 1**) with a general introduction, together with the statement of the proposed aims and an outline of the manuscript's structure.

In **Chapter 2**, an extensive literature review, covering the main CSOMs, and evidencing both their similarities and differences is presented. Furthermore, a classification for these methods is suggested and their successes and limitations are discussed. We close with a discussion around the future challenges of ME and strain design and its relevance for a sustained bio-based economy over the coming years.

One of the main contributions of this work is introduced in **Chapter 3**, where a new evolutionary multi-objective framework for strain optimization (MOStrain) is proposed. A thorough comparison of results between this method and previous meta-heuristic CSOMs is performed, and a critical analysis of some solutions is presented.

Leveraging on the framework developed in **Chapter 3**, a new tandem strain optimization approach, capable of finding strain designs compliant with multiple phenotype prediction methods is proposed in **Chapter 4**. This new method is tested for a panoply of case studies, and its ability to find robust strain designs, in situations where previous methods displayed limitations, is evidenced.

In **Chapter 5**, all the software tools developed during this thesis are explained in detail. These tools are made available in a powerful yet accessible framework, for the ME community to use and extend.

Finally, in **Chapter 6**, the main conclusions are drawn, and topics for future work are suggested.

REFERENCES

[1] G. Stephanopoulos, A. A. Aristidou, and J. Nielsen, *Metabolic Engineering: Principles and Methodologies*. 1998.

[2] J. Nielsen, "Metabolic engineering," *Applied Microbiology and Biotechnology*, vol. 55, no. 3, pp. 263–283, 2001.

[3] S. C. Schuster, "Next-generation sequencing transforms today's biology," *Nature methods*, vol. 5, no. 1, pp. 16–18, 2008.

[4] C. S. Henry, M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens, "High-throughput generation, optimization and analysis of genome-scale metabolic models.," *Nature biotechnology*, vol. 28, pp. 977–82, Sept. 2010.

[5] J. M. Savinell and B. O. Palsson, "Optimal selection of metabolic fluxes for in vivo measurement. I. Development of mathematical methods," *Journal of Theoretical Biology*, vol. 155, pp. 201–214, Mar. 1992.

[6] J. M. Savinell and B. O. Palsson, "Optimal selection of metabolic fluxes for in vivo measurement. II. Application to Escherichia coli and hybridoma cell metabolism," *Journal of Theoretical Biology*, vol. 155, pp. 215–242, Mar. 1992.

[7] K. J. Kauffman, P. Prakash, and J. S. Edwards, "Advances in flux balance analysis," *Current Opinion in Biotechnology*, vol. 14, pp. 491–496, Oct. 2003.

[8] A. P. Burgard, P. Pharkya, and C. D. Maranas, "Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization," *Biotechnology and bioengineering*, vol. 84, no. 6, pp. 647–657, 2003.

[9] M. Rocha, P. Maia, R. Mendes, J. Pinto, E. Ferreira, J. Nielsen, K. Patil, and I. Rocha, "Natural computation meta-heuristics for the in silico optimization of microbial strains," *BMC bioinformatics*, vol. 9, no. 1, p. 499, 2008.

[10] I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S. Soares, J. P. Pinto, J. Nielsen, K. R. Patil, E. C. Ferreira, and M. Rocha, "Optflux: an open-source software platform for in silico metabolic engineering," *BMC systems biology*, vol. 4, no. 1, p. 45, 2010.

CHAPTER 2

IN SILICO CONSTRAINT-BASED STRAIN OPTIMIZATION METHODS: THE QUEST FOR OPTIMAL CELL FACTORIES

Shifting from chemical to biotechnological processes is one of the cornerstones of the 21^st century industry. The production of a great range of chemicals via biotechnological means is a key challenge towards a biobased economy. However, this shift is occurring at a pace slower than initially expected. The development of efficient cell-factories that allow for competitive production yields is of paramount importance for this leap to happen. Constraint-based models of metabolism, together with *in silico* strain design algorithms promise to reveal insights into the best genetic design strategies, a step further to achieve that goal. In this work, a thorough analysis of the main *in silico* constraint-based strain design strategies and algorithms is performed, their application in real-world case studies is analyzed and a path for the future is preconized.

2.1 Introduction

Since the early 1970s, modern Biotechnology has started to emerge as a competitor of the chemical industry towards the production of valuable chemicals, although it remains until today in a great disadvantage. However, the scenario is rapidly changing given the increasing need for sustainable manufacturing processes. This has given industrial biotechnology a new breath, boosting its use in the production of a number of valuable products, such as pharmaceuticals, fuels and food ingredients. The Organization for Economic Co-operation and Development (OECD) predicts that by 2030, 35% of chemicals and other industrial products will be largely supported by industrial biotechnology [1].

Of course, mankind has been using microbes as a way to produce food ingredients and other valuable products since the Neolithic age. However, the ability to manipulate genomes has provided the field with novel powerful tools. Indeed, the development of industrial biotechnology is deeply intertwined with the recent evolution of molecular biology and genomics technologies.

Among these advances, two must be emphasized given their relevance to the field. In the early 1970s, the development of recombinant DNA technology [2–4] fostered the efforts in genetic engineering and, eventually, gave rise to modern biotechnology. A few years later, in the mid 1970s, the development of the Sanger sequencing technique [5, 6] provided another boost, starting a real revolution on genome sequencing technologies. Indeed, the first automated sequencer was developed in the late 1980s and, in 1995, the first complete genome of a microbe, *Haemophilus influenza*, was finished and published [7], followed by many others.

The importance of these technologies for industry is due to an obvious

observation: since microbes have evolved according to their own intrinsic objectives, in order to comply with industrial purposes, their metabolism needs to be retrofitted. Indeed, a sustainable, environmental-friendly and economically viable bio-based industry requires the use of cell factories tailored to deliver near-optimal yields of substrate to product conversion, as well as high titers and productivities.

The concept of metabolic pathway manipulation towards desirable behavior is an old one, with notable examples coming from the production of aminoacids, vitamins or antibiotics [8]. These early methods relied mostly on the use of mutagenesis and creative strain selection techniques. As these traditional approaches for microbial improvement started to struggle to keep up with industrial requirements, resorting to more rational approaches became a need. Genetic engineering provides the way to more precisely modify specific genes/ enzymes, thus paving the way towards the more rational introduction of direct genetic changes to create desirable strains [9].

In 1991, Bailey coined the term "Metabolic Engineering" (ME) as the use of recombinant DNA technology for the purpose of improving cellular activities via manipulation of enzymatic, transport and regulatory functions of the cell [10]. In contrast with previous experiences in genetic engineering, he preconized a much more direct, target-oriented and mechanistic approach. ME was also more attractive to the industry and less reluctant towards parallel sciences such as systems biology. In fact, it was thought from the beginning to embrace transversal knowledge.

The remarkable advances on genome sequencing technologies have played a decisive role for the change of perception regarding the genotype-phenotype relationship, which, at the time, was still mostly treated based on qualitative analysis. With the recent developments in genome sequencing technologies,

culminated by the surge of the so-called next generation sequencing technologies [11], as well as semi-automated annotation techniques, an increasingly large number of fully annotated microbial genomes are being made available.

These full genome sequences provide comprehensive information about the genetic elements that compose an organism and, when combined with the understanding of cellular processes such as metabolism, results in structured knowledge that can be mathematically represented. This knowledge explosion accelerated the development of genome-scale metabolic models (GSMMs) for a large number of organisms [12].

While a number of attempts have been made to model the whole cell behavior within the realm of systems biology research [13], these models are typically incomplete due to the lack of kinetic and regulatory information. An alternative is provided by the field of constraint-based modeling (CBM), which has been applied to the analysis of biochemical reaction networks for over 25 years [14]. One of the major outcomes of CBM research has been the development of phenotype prediction methods supporting distinct genetic and environmental conditions including the well-known method of Flux Balance Analysis [15–17]. When combined with GSMMs, these methods provide powerful tools to predict the behavior of microbial strains and, thus, support rational ME efforts.

Based on these efforts, the time for the development of strain design methods had come, where bioengineering objectives could be rationally addressed. In 2003, Burgard and co-workers released OptKnock [18], which would become the basis for a large portion of the constraint-based strain design methods for the following decade. Such *in silico* ME approaches are able to propose genetic changes (gene deletions in the case of OptKnock)

based on computational simulation and mathematical optimization methods. In this way, they are able to speed up the process of strain design, by reducing the time in the lab, and making more effective the engineering of microbes.

However, while this last decade has witnessed a rapid proliferation of strain optimization methods, mostly based on CBM approaches, *in vivo* proofs-of-concept are lagging far behind, as well as rigorous analysis of the predicting power of both simulation and design methods. Moreover, the concomitant proliferation of available GSMMs, often of organisms poorly characterized in physiological terms, adds a new layer of uncertainty to *in silico* predictions that also needs to be considered when designing novel and improved strains.

Here, we aim to provide an in depth and critical review of the currently available CBM-based strain optimization methods, their strengths and limitations, as well as to discuss future trends in the field. The importance of these methods for ME, and their relevance to boost modern industrial biotechnology efforts will be discussed, as well as the need for large-scale *in vivo* validation of rational design-related methods.

In the remaining, we start by putting forward the main concepts and methods within CBM, which will serve as a context and support for the strain optimization methods. Then, we cover in detail the main tasks in strain design and provide a taxonomy of the main strain optimization methods. These are presented in detail, their features and limitations are explored and the connections among different methods are highlighted. The section closes with a global discussion on the merits and limitations of the methods.

The review follows with an overview of selected practical applications

of strain design, in general, and of the contribution of the reviewed optimization methods in particular, focusing on experimentally and industrially validated applications. Successes and limitations of the approaches are discussed. We close with a discussion around the future challenges of ME and strain design and its relevance for a sustained bio-based economy over the coming years.

2.2 Constraint-based Modeling: Concepts and Methods

2.2.1 Constraint-based Models

Cellular functions are dependent on a series of intertwined mechanisms, such as metabolism or transcriptional regulation, which can be affected by a multitude of factors. Understanding the relationships between these mechanisms and the environment is key to developing correct and predictive models. Based on biochemical knowledge, classical kinetic models provide detailed dynamical and quantitative descriptions of the systems. However, they depend on many, usually difficult to measure, parameters and are also computationally expensive to solve in the genome-scale context [19]. Indeed, up to this date there are no dynamic genome-scale models of metabolism that can be used effectively in ME efforts, mainly given the difficulty in obtaining the relevant kinetic data [20, 21]. For several metabolic network analysis or metabolic engineering tasks, a simpler approach might be sufficient to reach useful results. For these purposes, certain realistic assumptions can be adopted, avoiding the burden of determining kinetic rate equations and their parameters [19].

Since the metabolic transients are usually faster than both microbial growth rates and dynamic environmental changes, the internal metabolite concentrations can often be assumed to be in a quasi-steady state. This assumption is at the core of constraint-based metabolic modeling approaches and its derived consequence is that all the metabolic fluxes leading to formation or degradation of any intracellular metabolite are mass-balanced [22]. This assumption can be represented in the form:

$$S.v = 0 \tag{2.1}$$

where S is an $m \times n$ matrix of stoichiometric coefficients, for a set of m metabolites and a set of n reactions, and v is the vector of n reaction rates (fluxes) Figure (2.1,b). For each reaction, maximum and minimum flux values can also be imposed to define the thermodynamic feasibility (directionality) and flux capacity of the reactions, i.e.:

$$\begin{cases} 0 \leq v_i \leq \beta_i, \forall i \in N_{irrev} \\ \alpha_i \leq v_i \leq \beta_i, \forall i \in N_{rev} \end{cases} \tag{2.2}$$

where v_i is the flux carried over reaction i, N_{rev} and N_{irrev} are subsets of N, composed by all reversible and irreversible reactions, respectively and α_i and β_i are the lower and upper bound for the flux over reaction i. For most GSMMs, the number of reactions surpasses the number of compounds, therefore there are more variables than equations in the system defined by Equation 2.1.

Further details, such as translational/transcriptional knowledge in the form of Gene-Protein-Reaction (GPR) associations are also typically included in the models [23]. This additional knowledge enables the develop-

ment of more realistic phenotype prediction and strain design methods. The representation of GPR associations usually resorts to Boolean logic, where the relationships between reactions and their encoding genes are modeled as logical AND/OR operations, representing for instance cases of protein complexes and iso-enzymes. This allows, for instance, to determine the reactions inactivated after a set of gene deletions Figure (2.1,c).

Parallel efforts focused on the use of Boolean approaches to represent models of transcriptional regulation. By considering that each node in the network is in a binary state (active/inactive), Boolean networks try to approximate the dynamics of regulatory systems. For each node, a Boolean update rule is defined which depends on the values of other nodes. Notable applications of Boolean networks include the elucidation of regulatory interactions by Kaufmann [17, 24] and the simulation of a system behavior under various genetic/environmental conditions by Li [25]. Despite the known fact that the expression of metabolic genes is affected by a plethora of different stimuli through regulatory mechanisms, few studies have focused on the integration of these approaches into GSMMs. The main efforts have been the inclusion of transcriptional regulation Boolean constraints, first by Covert and collaborators over an *E. coli* GSMM [26] and more recently and in a more general purpose way in the TIGER framework [27], as well as the inclusion of the transcriptional and translational machinery into the GSMMs of *E. coli* by Thiele et al [28].

Reconstruction of GSMMs has skyrocketed in the last decade, with dozens of these models currently available, for organisms from all the domains of life including recent efforts to model complex eukaryotes as is the case with the human models. In Table 2.1, a collection of some publicly available reconstructions is presented aiming to portray the main efforts,

rather than to be an exhaustive recollection. It is noticeable that for the most studied organisms, as *E. coli* or *S. cerevisiae*, a large number of models have been proposed for more than a decade now, generating models that are constantly growing in the level of biological detail covered. When used in tandem with phenotype prediction and strain design methods, to be covered next, GSMMs are a powerful tool to aid in various metabolic engineering tasks [29].

Table 2.1: Some publicly available GSMMs. Domain: E - Eukarya, B - Bacteria, A - Archaea. Reconstruction data includes number of reconstructions (#), maximum number of functionally annotated genes (G), metabolites (M) and reactions (R).

Classification		**Reconstruction data**						**Ref**
Organism	*Domain*	*Dates*	*#*	*IDs*	*G*	*M*	*R*	
A. thaliana	E	2010	1	AraGEM	1419	1748	1567	[30]
A. niger	E	2008	1	iMA871	871	1045	1190	[31]
B. subtilis	B	2007 2009	2	iYO844 iBsu1103	1103	1138	1437	[32] [33]
C. reinhardtii	E	2011	1	iRC1080	1080	1068	2190	[34]
C. salexigens	B	2011	1	iOA584	584	1411	1386	[35]
C. beijerinckii	B	2011	1	iCB925	925	881	938	[36]
C. thermocellum	B	2010	1	iSR432	432	525	577	[37]
D. ethenogenes	B	2010	1	iAI549	549	549	518	[38]
D. shibae	B	2013	1	iDsh827	827	1614	1488	[39]
E. coli	B	2000 ↓ 2011	5	iJE660 iJR904 iAF1260 iJO1366 iCA1273	1366	1136	2251	[40] [23] [41] [42] [43]
G. sulfurreducens	B	2006	1	iRM588	588	541	523	[44]
H. influenza	B	1999 2000	2	iJE296 iCS400	400	451	461	[7] [45]

H. pylori	B	2002 2005	2	iCS291 iIT341	391	485	476	[46] [47]
H. sapiens	E	2007 2013	2	Recon 1 Recon 2	1789	2626	7440	[48] [49]
K. pneumoniae	B	2011	1	iYL1228	1228	1658	1970	[50]
M. acetivorans	A	2011 2012	2	iVS941 iMB745	941	715	756	[51] [52]
M. barkeri	A	2006 2013	2	iAF692 iMG746	746	719	741	[53] [54]
M. musculus	E	2005 ↓ 2010	4	iMM1415	1415	2774	3726	[55] [56] [57] [58]
M. tuberculosis	B	2007 ↓ 2010	3	iNJ661 GSMN-TB iNJ661m	726	838	1049	[59] [60] [61]
M. genitalium	B	2009	1	iPS189	189	274	262	[62]
P. pastoris	E	2010 2012	2	iPP668 iLC915	915	899	1423	[63] [64]
P. putida	B	2008 ↓ 2010	3	iNJ746 iJP815 PpuMBEL1071	900	1044	1071	[65] [66] [67]
P. aeruginosa	B	2008	1	iMO1056	1056	760	883	[68]
R. eutropha	B	2007	1	RehMBEL1391	1256	1391	1171	[69]
S. cerevisiae	E	2003 ↓ 2013	14	iFF708 iND750 iLL672 iIN800 iMH805/775 Yeast[2-7] iMM904 iAZ900 iTO977	977	2056	3334	[70] [71] [72] [73] [74] [75] [76] [77] [78]

S. typhimurium	B	2009 ↓ 2011	3	iRR1083 iMA945 STM_v1.0	1270	1119	2201	[79] [80] [81]
Z. mobilis	B	2010 2011	2	ZmobMBEL601 iZM363	363	704	747	[82] [83]

2.2.2 Constraint-based Phenotype Prediction

Phenotypic behavior can be predicted using a number of constraint-based approaches over the information kept in metabolic models. The intersection of the available biological constraints (e.g. steady-state, reversibility, flux capacity) defines the flux cone of admissible flux distributions [84] (Figure 2.2,a), representing the underdetermination of the system. Given that experimental measurements of internal fluxes are difficult to obtain, the usual approach to solve this underdetermined system is to transform it into an optimization problem (Figure 2.2,b). For this purpose, biological assumptions are usually adopted in the form of an objective function. One common approach is to rely on the rationale that organisms might have been evolutionarily shaped towards metabolic operations that favor particular objectives. Extra constraints are commonly employed by many methods, which further constraint the flux cone, redirecting the flux towards a desirable state (Figure 2.2,c).

In Flux Balance Analysis (FBA) [15–17], these assumptions are modeled using linear objective functions, usually maximizing a given reaction rate (flux), minimizing the global energy expenditures of the cell, or a panoply of other ones [85]. With a linear objective function subjected to linear constraints, the problem is conveniently translated into a readily solvable

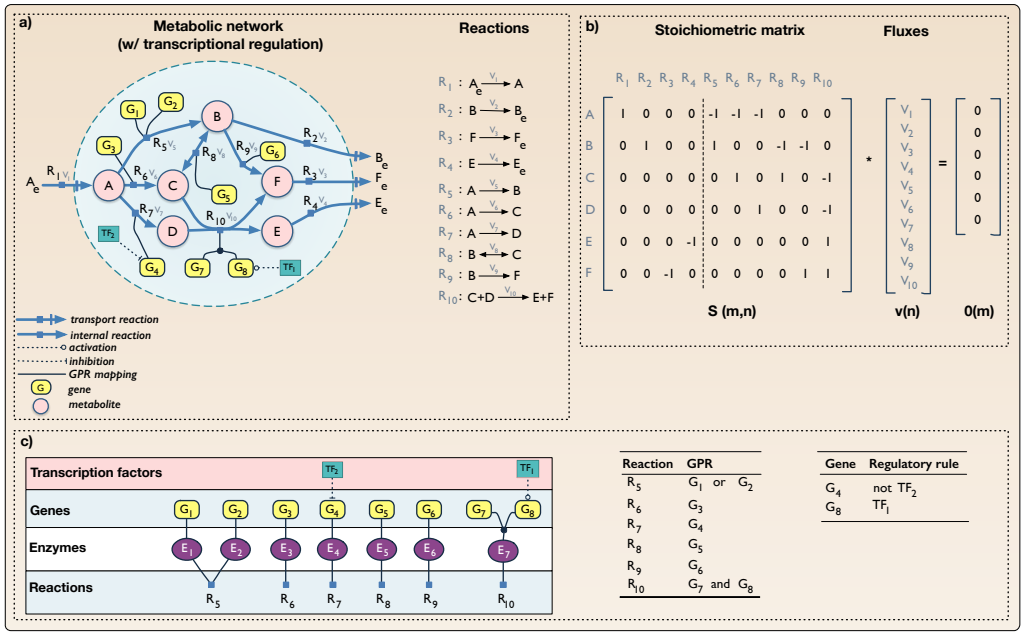


Figure 2.1: Diagram of the various components commonly found in constraint-based metabolic (and integrated metabolic/regulatory) genome-scale models. The example includes a) - a toy network composed by 10 reactions, 6 metabolites, 8 genes and 2 transcription factors; b) - the corresponding stoichiometric matrix; and c) - the corresponding gene-protein-reaction and transcriptional regulation rules.

linear programming (LP) problem. The most commonly used assumption is that microorganisms are evolutionarily adapted to maximize growth [86–88], which is modeled as a linear objective function (an artificially defined flux) that maximizes biomass formation.

Despite its utility, classical FBA is still fairly limited due to its obliviousness of several biological phenomena. As an example, the effects of regulatory constraints under certain media/environmental conditions are not accounted for. For this purpose, specialized methods such as regulatory FBA (rFBA) [89] or steady-state regulatory FBA (SR-FBA) [90] have been developed. Both methods rely on additional information, such as transcriptional regulation constraints [26], being integrated in the models.

Regulatory FBA forecasts dynamic flux profiles in changing environ-

ments by predicting a regulatory and a metabolic steady state for short successive time intervals, while ensuring consistency with the previous state in each step. Alternatively, SR-FBA simulates an ensemble metabolic-regulatory steady state under the assumption of maximal biomass production rate, which satisfies both metabolic and regulatory constraints. A mixed-integer linear programming problem (MILP) is yielded by the superimposition of the regulatory constraints and GPRs as linear functions in the model.

While the assumption of maximal growth is acceptable under natural conditions (wild-type), the same is heavily disputed when the organism is subjected to genetic perturbations, for instance when simulating the phenotype of gene deletion mutant strains. To account for the burden of shifting from one operating region to another, Segrè and co-workers introduced the Minimization of Metabolic Adjustment (MOMA) [86]. Contrarily to FBA, MOMA is not growth-coupled, meaning that the optimal flux distribution for a given set of conditions is not assumed to be dependent on the maximization of the organism's biomass production rate. Instead, it minimizes the sum of the squared differences between the wild type (typically calculated with FBA or given as a reference flux distribution) and the mutant flux distributions, thus defining a quadratic objective function, which translates into a convex quadratic optimization problem (QP) with linear constraints.

Linear MOMA (LMOMA), an alternative linear implementation of MOMA was later proposed by Becker [91], which removes the burden of the quadratic approach, by replacing the original objective function by a linear one, using the minimization of the sum of the absolute values of differences (between the a reference flux distribution and the mutant).

With a similar purpose, Shlomi and co-workers developed the regulatory

on/off minimization (ROOM) [92] algorithm that minimizes the number of significantly changed fluxes, relative to the original flux distribution, after genetic perturbations. This approach requires the introduction of binary variables in the objective function, thus converting the LP problem into a MILP one, increasing its complexity. Both MOMA and ROOM formulations rely on the assumption that, after genetic perturbations, the organism's metabolic and regulatory responses favor a new steady-state close to the original operating region, rather than maximizing cellular growth.

More recently, Brochado and co-workers developed the Minimization of Metabolites Balance (MiMBl) [93] as an alternative to MOMA, aiming at addressing some of its limitations. Instead of tackling the problem by optimizing fluxes, MiMBl resorts to metabolite turnovers (the sum of all fluxes producing or consuming it, multiplied by the corresponding stoichiometric coefficients), thus eliminating problems related with the sensitivity of the solutions to the stoichiometric representations, which can greatly affect the phenotype predictions.

The previous methods, and more notably FBA, have an important limitation, since while they provide a solution with a unique optimal value for the objective function, a large number of flux distributions may exist that lead to this value, i.e. multiple optima may exist. One way to address this issue was proposed by the Parsimonious enzyme usage FBA method which assumes that, under growth pressure, the strains that process the growth rate more efficiently (with minimum enzyme usage) will have a competitive advantage [94]. This method mimics this assumption by choosing, from these multiple optima, the flux distribution that minimizes the sum of all the fluxes in the model. A two-step approach is frequently used to approximate this formulation, where a second LP that minimizes the sum of all

fluxes while keeping the biomass flux at an optimum level, is performed [95].

Flux Variability Analysis (FVA) [96] provides a distinct approach that aims to characterize the space of possible variation of specific fluxes, given a set of constraints. It can be used to define tight bounds for the fluxes in a GSMM if no further constraints are defined, or to check the possible variation of a given flux in optimal or sub-optimal solutions, if a constraint over the objective function is defined. For instance, FVA is quite useful in checking if a flux can vary in optimal FBA solutions, by setting a constraint that requires the biomass flux to be equal to its optimal value. Among other applications, FVA is used to assess the robustness of a flux distribution, for instance in a mutant strain simulation, regarding its production capability of a certain compound. FVA is typically applied to a given reaction flux by solving a pair of LP problems that maximize and minimize the target flux, obeying the set of defined constraints.

2.2.3 Unbiased Characterization of the Flux Cone: pathway analysis

Any attempt to enumerate all the possible flux steady-state distributions lies within the realm of the intangible for typical GSMMs with large numbers of reactions and metabolites, since their complexity scales exponentially with the size of the models [97, 98]. This fact is the main driving-force behind the development of the methods described in the previous section.

Still, within the field of Pathway Analysis, a number of methods have been put forward towards this purpose, even if currently these are only applicable to small or medium scale models. The two best known approaches for the enumeration of the possible flux distributions are Elementary Flux Modes (EFMs) [99] and Extreme Pathways (ExPas) [98]. Both these meth-

ods describe minimal (non-decomposable) sub-networks of the system that operate at steady-state, defining the edges of a convex polyhedral cone (the flux cone – Figure 2.2). In turn, linear combinations of the vectors representing these minimal sub-networks yield the totality of the solution-space (all possible flux distributions).

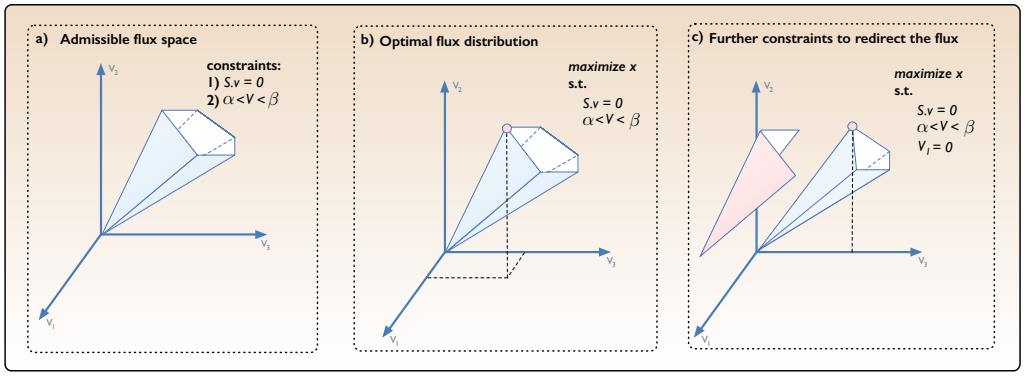


Figure 2.2: The flux cone. In the example, a) - the admissible flux space is defined by the steady-state and reversibility constraints; b) - an objective function is imposed (subjected to the previous contraints) defining an optimization problem; and c) - further constraints are imposed to redirect the flux to a desired region of the flux cone.

EFMs obey the following set of conditions [99]:

1. Steady-state: All elementary modes obey Equation 1;

2. Feasibility: All irreversible reactions proceed in the forward direction, i.e., EFMs are thermodynamically feasible, obeying Equation 2.2;

3. Non-decomposability: EFMs represent the minimal functional units in the network; therefore, no reaction can be removed from an EFM without violating either (2.1), (2.2) or both.

Moreover, these particular conditions yield some important properties:

- (P1): There is a unique set of EFMs for a given metabolic network;

- (P2): All the feasible steady-state flux distributions satisfying (1) and (2) are a non-negative superimposition of the set of EFMs in the network;

- (P3): When a reaction is removed from the network, the set of EFMs for the new network is equal to the one from the original network, but removing all the EFMs that include the removed reaction.

These properties render these approaches extremely interesting for metabolic engineering purposes (among others), since they describe the complete portfolio of steady-state phenotypes, and are conveniently presented as minimal metabolic functions.

While both methods are conceptually close to each other, the fact that ExPas resort to the decoupling of reversible reactions, into equivalent forward and backward reactions, translates into a non-complete overlap of their corresponding enumerations in many cases. In fact, ExPas are a subset of EFMs, except when the network is solely composed by irreversible reactions, in which case the two sets should completely overlap.

Given that the previously mentioned enumerations, for genome-scale or even medium-sized metabolic networks, are computationally unattainable, alternative methods based on random sampling of the flux space, such as Monte Carlo Markov Chain (MCMC) [100] or an adapted canonical basis approach [101], have been developed to circumvent this limitation. Moreover, alternative methods focused on achieving particular biological functions or searching a subset of the flux space [102, 103], are also knowledgeable ways of exploiting this class of methods.

2.3 Computational Strain Optimization Methods

Before a formal classification for computational strain optimization methods (CSOMs) is presented, our vision of what a CSOM is must be clarified. Computer-aided strain design efforts cover a broad range of applications and techniques, which leads some authors to mix together optimization approaches with others that can be considered as phenotype prediction methods or even strain design algorithms that do not explicitly use any type of optimization algorithm. To clarify our approach to the classification of these methods, and as a justification for the inclusion or absence of some well-known (and, nonetheless, very useful) approaches in this section, some rules have been drafted for whether or not to consider a method as a CSOM:

1. A CSOM must (try to) provide an answer to a specific top-level question, such as: "Which set of perturbations applied to the model (organism) favors a desired engineering goal?"

2. The exhaustive enumeration of all possible solutions (designs), albeit highly desirable, is not considered a CSOM since it is considered a trivial approach (although usually not attainable). Rather, an optimization algorithm that defines a strategy to sample the solution space must be defined to warrant the inclusion of the method in the review that follows.

3. The algorithms covered in this review are solely the ones based in constraint-based approaches, following the framework highlighted in the previous section. We will not cover here other approaches, for

instance those based in graphs or hypergraphs and their underlying algorithms, nor any approach based in the use of dynamic models.

2.3.1 Computational Strain Optimization Tasks

CSOMs can be thought of as procedures, which try to answer a practical question (or set of questions) relevant for strain design. These questions can be translated into mathematical formalisms and addressed by distinct optimization methods. Powered by phenotype prediction methods and guided by GSMMs, these methods automatically or semi-automatically search for answers to questions such as: "Which genes should be deleted from the model to couple the production of compound X to growth?" or "which foreign pathways must be added as to acquire a desirable functionality in a given host?". In fact, this last question was probably the first one to arise when molecular biologists realized the inner potential of recombinant DNA technology. The most common tasks undertaken by CSOMs are: gene deletion, gene over/under expression, heterologous insertion and, more recently, co-factor specificity swapping. Some methods also attempt combinations of these tasks to find better phenotypes.

2.3.1.1 Gene deletion

The suppression of a given metabolic function can be accomplished *in vivo* by disrupting the function of specific genes resorting to targeted modifications through homologous recombination [104] or intron introduction [105]. *In silico* CSOMs that account for gene deletion (Figure 2.3,a) usually search for combinations of metabolic function suppressions yielding desirable phenotypes. This task is commonly accomplished by imposing constraints that force the flux of the disabled reactions to zero, deterring the occurrence of

flux over those reactions, followed by the evaluation of the effect of that perturbation.

Some recent methods take advantage of the GPR information contained in the model, and search for combinations of gene deletions (instead of searching for reaction suppressions), which closer represent the *in vivo* scenario, since they inherently account for the occurrence of multifunctional and multimeric proteins, as well as isoenzymes [29].

2.3.1.2 Heterologous insertion

Analogously, the inclusion of non-native functionalities, via gene or pathway addition, might broaden the metabolic capabilities of desirable hosts, either by boosting the yields of native compounds or by allowing the production of entirely new ones (Figure 2.3,b). Typically, algorithms with this kind of capability, will sort through databases of balanced reactions for the desired functionality, and try to reconcile them with the original network. The augmented network can, afterwards, be engineered by other CSOMs to redirect the flux into the desired direction. Most algorithms specialized only in the first task, that is, the sorting of heterologous enzymes and subsequent reconciliation with a target host, are not considered in this work since they are typically not constraint-based approaches. Examples of aforesaid methods include DESHARKY [106], BioPathwayPredictor [107] or FindPath [108].

2.3.1.3 Gene over/under expression

Alternatively, the up/down regulation of gene expression have gained considerable importance in the ME community [109, 110]. Gene over/under expression concerns the fine-tuning of enzyme levels and correspond-

ing flux rates, which can be accomplished by resorting to promoter libraries [111–113] or synthetic biology tools [114]. This approach can be useful in situations where a gene deletion is lethal whereas a downregulation is not, as well as a solution to overcome flux bottlenecks in certain steps towards a desired biological function (Figure 2.3,c). These tasks are usually undertaken through the addition of extra constraints on the reaction fluxes rates, forcing them to operate closer to their maximal or minimal theoretical bounds.

2.3.1.4 Modulation of cofactor binding specificity

A distinct approach is to tackle the scarcity of some cofactors required for essential steps of some ME efforts, by modulating the cofactor binding specificities. A typical example is the modulation of NAD(H) or NADP(H) availability, due to their importance in the anabolic and catabolic processes. By playing with the co-factor binding specificities, it is possible to establish driving forces in target pathways that require, for example, the regeneration of one particular cofactor [115]. *In vivo* modulation of cofactor specificities has been reported via protein engineering [116, 117] or by replacing native enzymes with heterologous ones with different cofactor specificity [118]. This approach can be computationally simulated by swapping the cofactor specificity of some reactions in the network, followed by a phenotype prediction method to evaluate the effects of the perturbations (Figure 2.3,d).

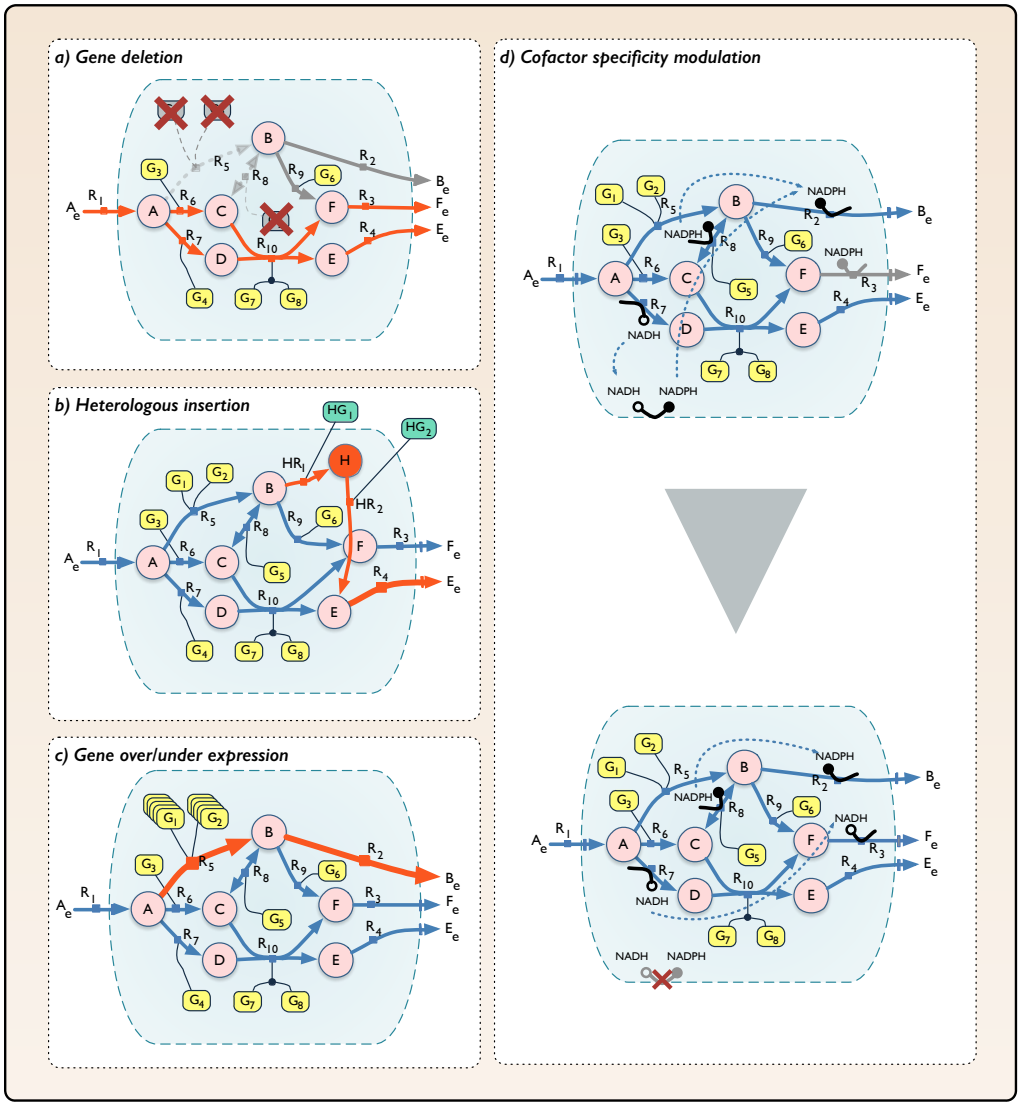


Figure 2.3: Computational strain optimization tasks. In the example, a) - 3 gene deletions force the flux through reactions producing the desired compounds; b) - the inclusion of two heterologous genes allow the production of an intermediary compound and subsequent excretion of the desired product; c) - the overexpression of two enzymes allow the excess formation of compound B which is subsequently excreted; and d) - the enzyme catalyzing the transport reaction R_3 is swapped by an heterologous enzyme using NADH. The deletion of a membrane oxidoreductase enzyme creates an excess of NADH that can be used by the new transport reaction to excrete compound F.

2.3.2 A Taxonomy for Computational Strain Optimization Methods.

The set of CSOMs reviewed in this work has been organized and classified under several aspects, including the strain design task(s) addressed, the optimization algorithm, mathematical formulation, the method's scalability, the validation case studies and the availability of the method in a software implementation. This information is provided in Supplementary Table 1, being the methods sorted chronologically.

A classification (or "taxonomy") for CSOMs is proposed in this work based on the features of each method. An obvious classification would be to group them based on the task they try to accomplish, as explained in the previous section. While keeping this also in mind, we prefer to analyze them primarily based on their optimization frameworks and mathematical formulations. Following this principle, three main branches emerged in our analysis: bilevel mixed-integer programming (MIP), metaheuristics and elementary modes analysis (EMA) based methods.

2.3.2.1 Bilevel Mixed Integer Programing methods

In 2003, Costas Maranas' group reported a specialized method to pursuit productive gene deletion strain designs coupled to cellular growth. This method, OptKnock [18], detailed a bilevel framework, where two competing objective functions were simultaneously accounted for. The inner problem concerned the biological objective of the organism, in this case maximization of cellular growth, while the outer problem focused on the engineering goal, the overproduction of a desired compound. The method suggested reaction deletions, which were imposed as constraints for the inner problem.

This elegantly formulated mathematical framework profited from the strong duality property, which stated that if the primal and the dual optimal solutions are bounded, then at optimality, the gap between both objective function values must be zero [119]. This property allows the bilevel formulation of OptKnock to be transformed into a single-level MILP, by setting the primal and dual objectives equal to one another and accumulating their respective constraints. OptKnock represented a breakthrough in the field, establishing the framework used by many of the developed CSOMs until present days. Figure 2.4 summarizes the main properties of the bilevel formulation and its conversion to a single-level MILP, introduced by OptKnock and which establishes this category of CSOMs' distinct characteristics.

One of the strong properties of OptKnock solutions is that they are mathematically guaranteed to be optimal, given the defined task and objective function. However, they can often be considered as overly optimistic in real world scenarios. The fact that the target production is coupled to the biomass formation, results in OptKnock selecting the "best" solution, the one with highest product yield given a pre-defined minimum biomass flux value and a maximum number of reaction deletions, but is not able to account for competing pathways that might redirect the flux, lowering or even zeroing the expected product yield.

To address this issue, Tepper and Shlomi suggested the RobustKnock [120] method, a reformulation of the OptKnock procedure that optimizes the worst-case scenario for the product formation coupled to cellular growth, that is, a guaranteed (robust) lower bound for the expected growth coupled product formation. The max-min formulation of RobustKnock yields a triple-level problem, an outer max-min problem that searches for a set of knockouts maximizing the minimal production rate of the target compound

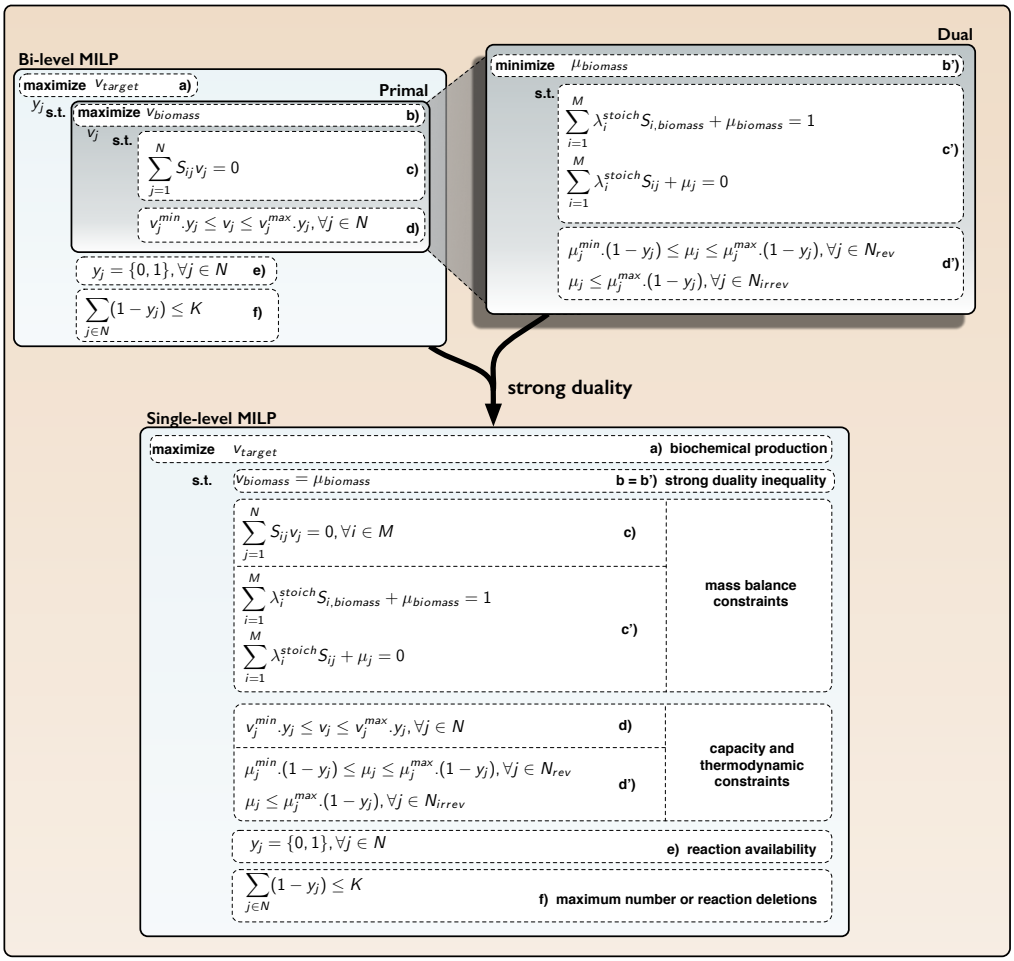


Figure 2.4: Simplified example of the coupling strategy introduced by OptKnock, allowing the transformation of a bi-level MILP into a single-level one. The top region depicts the bi-level MILP problem (left) and the dual of the inner problem (right). For each constraint in the primal, there are one or more dual equivalents. Using strong duality theory and accumulation of constraints, the single-level MILP formulation is attained (bottom).

(a bilevel problem), and an inner problem similar to that of OptKnock, searching for a feasible flux distribution maximizing biomass. The outer problem is transformed into a standard max-min problem using a procedure similar to that used in OptKnock and, subsequently, transforming it into a standard MILP problem.

Shortly after, Feist and co-workers, as an alternative approach to ad-

dress the robustness issues brought up by OptKnock, introduced objective function tilting [121]. Albeit not being a CSOM by itself, objective function tilting represents a valid, computationally lighter, alternative to RobustKnock, since it only involved small changes in the objective function of OptKnock (and also of OptGene, referred in the next section), without increasing its computational complexity. In OptKnock, this approach involved subtracting the target product multiplied by a very small weight to the inner problem's objective function. This forced the algorithm to identify the solution with the highest minimum production rate, among the ones with optimal value for the inner problem.

More recently, Kim and co-workers introduced BiMOMA as an alternative to OptKnock [122]. BiMOMA is a new bilevel CSOM for the design of gene deletion strategies. It is formulated as a mixed-integer quadratic constrained programming problem (MIQCP), which uses MOMA as the inner phenotype evaluation method (opposed to OptKnock's FBA). In their work, several techniques to reduce the scalability problems traditionally associated with MIP formulations were also suggested. These techniques include the tightening of the bounds of the dual variables using a sampling technique, the application of penalties for genetic perturbations (favoring smaller designs), reduction of the search space, and finally the use of iterative methods as a way to improve the performance of the solvers. The applicability of BiMOMA was demonstrated for the production of glutamate and pyruvate in *E. coli*.

A similar approach was suggested by Ren and co-workers [123] with MOMAKnock. The adoption of MOMA as the inner problem once again yielded a mixed integer bilevel quadratic programming (MIBQP) formulation but, contrarily to BiMOMA, it was not converted to a MIQCP. Due

to the heavy computational burden caused by the MIQCP formulation, the authors proposed the use of an adaptive piecewise linearized inner problem to approximate the quadratic objective function of MOMA. The proposed formulation was benchmarked against OptKnock results, in the search for succinate productive *E. coli* strain designs.

Very recently, researchers from the Chinese Academy of Sciences questioned the validity of the duality theory transformation employed by most of the previously described CSOMs [124]. They argue that, by not including the lower level primal variables in the dual objective, the single level MILP problem in which OptKnock relies was erroneous derived. More explicitly, they argue that if the problem had been correctly formulated, a mixed integer nonlinear programming problem (MINLP) would emerge instead of a MILP. Subsequently, they suggest a method based on the Karush-Kuhn-Tucker (KKT) [125] technique to reformulate the Mixed Integer Bilevel Linear Problems (MIBLP) problem as a single level MILP (applicable only when the inner problem is continuous), and propose ReacKnock as a more reliable alternative to OptKnock. However, more recently Chowdhury and coworkers released a new work [126] where the peculiarities of the OptKnock formulation are more thoroughly explained. In particular they elaborated on the application of the complementary slackness conditions to justify the absence of some of the lower level primal variables in the dual objective function thus restating the formulation of OptKnock as completely valid.

The optimization of reaction deletions was not the only task addressed by this class of methods. Shortly after the publication of OptKnock, Maranas and co-workers extended the developed framework to support enhancing a desired host with non-native functionalities via heterologous enzyme additions. This method was termed OptStrain [127] and was implemented as a

multi-step approach. Profiting from the myriad of biological data sources available, such as KEGG [128, 129], MetaCyc [130] or BRENDA [131], the authors created a universal reaction database which OptStrain uses as a core for finding the reactions that can be added to the host GSMM. With this knowledge base in place, the OptStrain procedure computes the maximum theoretical product yields for a given substrate, considering both native (from the GSMM) and non-native reactions (from the database). Afterwards, the minimal number of non-native functionalities yielding balanced pathways are sought and included in the original stoichiometric model. Finally, the OptKnock framework is employed to find additional knockouts that further increase the yields. The OptStrain framework was used to identify metabolic engineering strategies for the production of hydrogen and vanillin in *C. acetobutylicum* and *E. coli*, respectively.

In 2011, Kim and co-workers revisited and improved the OptStrain framework, by considering both gene deletions and heterologous insertions simultaneously [122]. This method, SimOptStrain, is aware of GPR relationships, which potentially allows for more biologically feasible designs since the target here is to find good combinations of genes to knockout. The new method was demonstrated in the design of succinate and glycerol productive strains of *Escherichia coli*.

Yet, the portfolio of possible genetic manipulations was not complete with the development of OptKnock, OptStrain and related methods. As previously stated, the tuning of gene expression and related enzyme levels is another important task in strain optimization. The OptReg [132] framework was the first CSOM to allow searching for optimal gene expression levels, together with gene-deletions as provided by OptKnock. To this end, additional binary variables referring to the up and down regulations and

knockouts were considered:

$$y_j^d = \begin{cases} 0,\ if\ reaction\ j\ is\ downregulated, \\ 1,\ otherwise \end{cases}$$

$$y_j^u = \begin{cases} 0,\ if\ reaction\ j\ is\ upregulated, \\ 1,\ otherwise \end{cases}$$

These act as switches that restrict the flux in response to the respective perturbation (over/under expression) based on the following supplementary constraints:

$$v_j^{min} \leq v_j \leq [v_{j,L}^0.(1-C) + v_j^{min}.C].(1-y_i^d) + v_j^{max}.y_j^d, \forall j \in N$$

for downregulations, and:

$$[v_{j,U}^0.(1-C) + v_j^{max}.C].(1-y_j^u) + v_j^{min}.y_j^u \leq v_j \leq v_j^{max}, \forall j \in N$$

for upregulations, were, v_j^{min} and v_j^{max} are the lower and upper limits of flux j defined in the GSMM, $v_{j,U}^0$ and $v_{j,L}^0$ the lower and upper limits of flux j determined by flux variability analysis (FVA) [133] and C is the regulation strength parameter considered in the interval $[0, 1]$. Higher values of C correspond to stronger regulations. Furthermore, constraints stating that a reaction can only be subjected to one manipulation (knockout, downregulation or upregulation), constraints limiting the maximum number of manipulations allowed and constraints forcing the mutual knockout of the two directions of reversible reactions are also taken into account. OptReg

was illustrated with the determination of engineering strategies for the production of ethanol in *E. coli*.

Almost concurrently, Ranganathan and colleagues published an alternative method named OptForce [134], which was supported on a different but rather insightful concept. The workflow is initiated by a characterization of the wild-type strain, resorting to FVA to determine the lower and upper bounds of each flux (this task can be aided by experimental data if available). By iteratively considering sets of reactions (pairs, triples, etc), OptForce generates a set of reactions that are required to change to achieve a user defined production yield (termed the MUST set). From this set, a new step proceeds to characterize the minimal set of reactions that need to be forced via genetic manipulation (termed the FORCE set). The proof of concept was performed in the production of succinate in *E. coli*. Posterior experimental validation led to interesting results for other targets in *E. coli* [135, 136] (further details in the next section).

A similar approach was recently proposed by Cotten and Reed [137], termed CosMos, that incorporates flux up/downregulations in a more flexible manner than OptForce since changes to bounds can be chosen continuously. Their results report additional solutions compared to OptForce, in a case study for succinate production with *E. coli*.

With the scalability problems of traditional MIP formulations in mind [121], Mahadevan and co-workers proposed the EMILIO (enhancing metabolism with iterative linear optimization) approach [138]. The foundational framework employed by EMILIO is very similar to that of OptReg, with the exception that the inner problem's objective function enforces the maximization of a minimal production rate, thus addressing the concerns first raised by Shlomi et al [120]. The bi-level problem is reformulated into a non-

convex single-level mathematical program with complementary constraints (MPCC), which is solved using a three-step approach. First, an iterative linear programming (ILP) is applied to establish the set of active constraints (flux bounds), followed by a recursive LP-based pruning method that identifies subsets of active constraints required to achieve a user-specified fraction of the maximum production rate. In the final step, for each subset, a MILP procedure similar to OptReg is employed to minimize the number of reaction modifications required to satisfy the user-specified fraction of the maximum production rate. EMILIO was demonstrated in the design of various deletion/ over/under expression *E. coli* mutants for the production of succinate, L-glutamate and L-serine.

A distinct method is OptORF, released in 2010 by Kim and Reed [139] that was the first to address strain optimization using metabolic-regulatory integrated models. Similarly to previous approaches, OptORF was formulated as a bi-level MILP problem, capable of suggesting metabolic engineering growth-coupled production designs. Uniquely though, OptORF designs consisted on both metabolic and regulatory gene deletions as well as metabolic gene overexpressions. These make use of Boolean rules defining the relations between reaction and metabolic genes (GPRs), as well as rules that define transcriptional regulation. Both are transformed into linear constraints. The applicability of this method was, however, limited by the scarcity of available integrated metabolic/regulatory models to profit from all its features. OptORF has been applied in the design of ethanol, isobutanol and 2-phenylethanol *E. coli* producing strains.

Complementary to the various described approaches for gene deletion, gene over/under expression and heterologous insertion, some alternative CSOMs with unique characteristics have also been proposed.

An exquisite new method, OptSwap, was very recently proposed by King and Feist [115]. OptSwap was the first and, until today, the only method to consider the cofactor binding specificity of enzymes as possible targets for computational strain optimization. More specifically, OptSwap focuses on oxidoreductase enzymes and their binding specificity for either NAD(H) or NADP(H). However, the authors state that the principles and framework are extensible to other specific sets of interest. A pool of swap candidate oxiydoreductase enzymes is selected beforehand, based on literature and *in silico* limitations. The problem is formulated as a MILP similar to RobustKnock, but including additional constraints to enforce swaps of the cofactor specificities of the previously selected reactions. This is accomplished by extending the model with a set of reactions with opposite cofactor specificity to the ones in the pool. Extra constraints are added to force the knockout of either the native or the swapped reaction and to limit the number of both swaps and deletions. The OptSwap procedure was used to identify non-intuitive designs for several end products in *E. coli*, some of which were not possible by gene deletions alone.

The most recent effort in MIP-based CSOMs combines the kinetic descriptions of metabolic steps with traditional stoichiometric models, to improve their predictive power and suggest more accurate designs [140]. By bridging the gap between stoichiometric and kinetic-based models [141], k-OptForce may represent a game changer and a new chassis for future CSOM development efforts. Alternatively to most CSOMs, k-OptForce does not rely on assumptions such as the maximization of biomass or minimization of metabolic adjustments as a fitness function, but rather makes use of available kinetic rate laws to predict flux distributions. To meet this end, the reactions in the metabolic network are split into two sets, one contain-

ing reactions for which kinetic information is available (N^{kin}), and another for the reactions with only stoichiometric information (N^{stoic}). While the reactions in N^{stoic} are constrained by mass-balances and thermodynamics, the ones in N^{kin} are subjected to the enzyme kinetics, metabolite concentrations and kinetic parameter values. Consequently, the N^{kin} part of the network is represented as a system of nonlinear ordinary differential equations (ODEs).

The k-OptForce procedure is subsequently solved in two steps: first, a characterization of the wild-type is done, by solving the system of ODEs to obtain a steady-state flux distribution for N^{kin}, and by using FVA for the N^{stoic} reactions. Similarly, the characterization of the overproducing strain is performed subject to the kinetic and concentrations constraints, where available. Finally, the computation of MUST and FORCE sets (81) is reformulated to account for the newly introduced kinetic layer. The introduction of nonlinear kinetics into the formulation translates into a single-level mixed-integer nonlinear optimization problem (MINLP), which needs to be addressed resorting to NLP solvers. The k-OptForce formulation was contrasted with the original OptForce, for the prediction of L-serine and Tal mutants, in *E. coli* and *S. cerevisiae* respectively.

Overall, the class of methods described in this section presents numerous advantages and has led to several successful applications that will be covered later. Yet, albeit the formulation of strain optimization problems on top of the OptKnock framework yields exact solutions for the defined formulations, the methods have problems with the underlying computational complexity. This constrains, for instance, OptKnock and related methods to a relatively low maximum number of allowed transformations, as the consideration of higher numbers would make the methods hard to apply in current GSMMs.

Also, the use of a MIP framework restricts these methods to the use of linear objective functions that do not necessarily express the complexity of the bioengineering objectives. Lastly, these methods rely on a tight coupling of the two levels, the phenotype simulation and the strain optimization layers, which also reduces their flexibility. This is clearly illustrated by the fact that there is a need to define a new method when an optimization approach developed for a specific simulation method is to be applied to another one, such as changing from FBA to MOMA.

2.3.2.2 Metaheuristic CSOMs

The problems associated with the previous methods stated above motivated the development of a separate class of approaches, supported on a more heuristic rationale. Heuristic methods are usually computationally less expensive approaches for a myriad of optimization problems. Although, due to their nature, they do not guarantee that the overall optimal solutions are found, they allow the definition of optimization frameworks with an enriched set of objective functions and that foster a more clearer separation of the strain optimization from the phenotype simulation methods. In Figure 2.5, a generic workflow for a typical metaheuristic CSOM is presented.

The first effort to move in this direction was OptGene [142] presented by Patil and co-workers, that appeared shortly after the publication of OptKnock and OptStrain. Inspired by the Darwinian natural evolution theory, OptGene formulates a bi-level decoupled approach, supported by the use of a genetic algorithm [143]. The idea is to encode solutions as individuals in an evolving population. Here, each solution is represented as a set of integer values encoding reaction deletions. The algorithm starts by the random generation of an initial set of candidate solutions (the initial population),

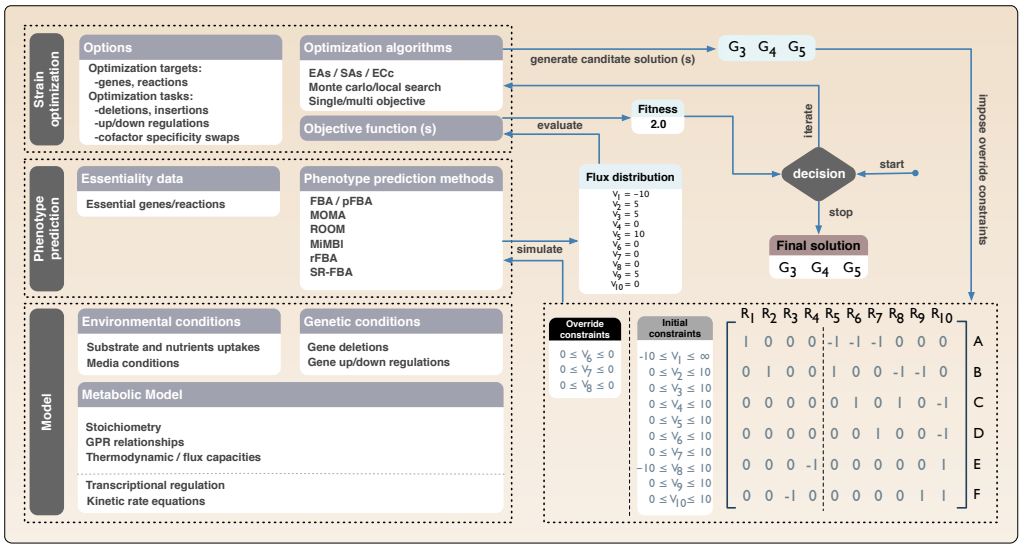


Figure 2.5: Workflow of generic metaheurist CSOM. The separation of the several layers (model/phenotype prediction/strain optimization) is easily visible.

each is decoded into a set of reaction deletions, which are translated into constraints and flux distributions are predicted using FBA. A fitness value is assigned to each solution by a user-defined objective function such as the Biomass-Product Coupled Yield (BPCY), which is an example of a non-linear objective function that resembles productivity. Subsequently, the algorithm enters an iterative stage, starting with a selection step, which chooses solutions as primary candidates for reproduction, in a stochastic way that depend on their previously assigned fitness (fitter individuals have more probability of generating offspring solutions). Finally, by combining these individuals via crossover or mutation operators, a new population is attained and re-evaluated. This cycle is repeated until a desired phenotype is achieved or another user-defined termination criterion is met.

Although these methods, as the previous ones, still follow a bi-level design, in this case the bioengineering and the biological optimization tasks are clearly decoupled and are performed independently. This decoupling of

the outer and inner optimization problems translated into some very powerful properties: first, complex designs involving a larger number of perturbations can be tested without affecting the computational tractability of the problem. Secondly, the decoupling of the outer and inner optimization problems allows for a more flexible tuning of both layers, for example, the inner phenotype evaluation method can easily be swapped amongst FBA, MOMA, ROOM, or any other phenotype simulation method, for instance including the ones using regulatory constraints (such as rFBA or SR-FBA).

Another important advantage is the flexibility in the definition of the objective function in the outer problem, which is here not bounded by linearity. Non-linear objective functions (even discontinuous) can easily be included, as is the case with the BPCY, allowing to define more meaningful and powerful functions. The flexibility gained by the decoupling of the two layers also allows the easier switch of the optimization heuristic used to search for metabolic engineering strategies and also to address the different optimization tasks with a similar framework.

In the original publication, the OptGene method was first used to suggest *S. cerevisiae* designs for the production of vanillin, glycerol and succinate. Since then, taking advantage of the flexibility provided by this approach, the framework has been thoroughly extended to support additional features and algorithms. Rocha and co-workers provided a reformulation of OptGene where set-based evolutionary algorithms (SEA) and simulated annealing (SA) algorithms were used in the outer optimization problem [144], enabling a more compact representation and the simultaneous optimization of the number of knockouts. Also, extensions were proposed where metabolic engineering strategies consider both metabolic and regulatory genes as targets [145] and gene over/under expression [146].

A very similar method, Cipher of evolutionary design (CiED) [147], was also developed and experimentally validated in *E. coli* for the increased production of malonyl-CoA. More recently, Constanza and Nicosia revisited the robustness issues of suggested genetic interventions [148]. The proposed genetic design through multi-objective optimization (GDMO) is inspired in the non-dominated sorting genetic algorithm II (NSGA-II) [149], a multi-objective evolutionary algorithm (MOEA) which uses the Pareto optimality principle to find the optimal (or near optimal) trade-off solutions in a multi-objective optimization problem. The procedure begins by the pre-processing stage, where a sensitivity analysis on the model ranks the various pathways according to their influence on the outputs. Subsequently, the MOEA searches for both gene deletions and nutrients, specified by the objective functions. Finally, a robustness analysis task selects the most locally and globally robust designs as ideal candidates for implementation. This procedure was applied in *E. coli* for the production of succinate and acetate, and its results were compared with those of GLDS, OptGene, SEAs/SAs and OptKnock.

The genetic design through local search (GDLS) method [150] embraced the problem from a different perspective. By using iterative local search steps, building on previous solutions, GDLS is capable of suggesting efficient strain designs, involving a larger number of genetic interventions (both gene deletions and over/under expressions). However, as later pointed out [138], the complexity still increased exponentially with larger scopes of each local search. GLDS proof of concept was established by suggesting acetate and succinate *E. coli* productive strains.

Later, part of the GDLS team also participated in the development of the genetic design through branch and bound (GDBB) [151] approach.

GDBB uses a truncated branch and bound technique to tackle the scalability problems associated with the bilevel MILP problem introduced by OptKnock. The formulation, similar to that of OptKnock, makes use of the Gurobi solver (Gurobi Optimization, Houston, TX, USA) particular implementation of the truncated branch and bound algorithm, and exploits its optimality and feasibility configuration options to fine-tune the truncation process. By not allowing the solver to reach optimality, but rather forcing it to stop at near-optimal solutions (considered sufficient for practical purposes), the performance of the method improved significantly when compared to previous approaches. Such technique is readily applicable to any method whose foundational framework is a bilevel MIP, for which a single-level MIP equivalent is attainable.

More recently, Rockwell and co-workers presented the Redirector approach [152]. The Redirector method is based on the iterative local search technique used by GDLS, but introducing a novel objective function reconstruction cycle in the iterative procedure. This novel cycle is composed of two steps: the first one, called "objective control", finds metabolic engineering targets and adds them to the objective function while the second one, designated "progressive target discovery", iteratively adjusts the contribution of growth to the objective function, redirecting resources to the optimization of the target compound. Important in this method is the idea to simulate over/ underexpression of enzymes, by including the fluxes in the objective function with positive and negative coefficients, respectively. This approach was applied to the discovery of novel *E. coli* designs for the production of fatty acids.

The latest noteworthy heuristic CSOM presented is FastPros [153], an efficient screening procedure based on shadow prices analysis. This algo-

rithm introduces a novel score for knockout screening (μ_{target}), which corresponds to the shadow price of the sink flux associated with the compound of interest (for an FBA problem maximizing biomass production rate). This new score represents the potential for target production, and is calculated by the following relationship:

$$\mu = \frac{\delta v_{growth}}{\delta v_{target}}$$

where δv_{growth} is the variation in the biomass production caused by an increase of δv_{target} from zero flux. A positive value of μ_{target} represents growth-coupled production of the target compound. The procedure begins with the computation of every double knockout strategy and corresponding μ_{target} values, from which a set of P parent sets are selected with regard to that score. At this point, the iterative procedure begins, consisting in the generation of knockout sets by adding every possible single knockout to every parent set in P (yielding $P \times N$ sets), recalculating the δv_{target} values and re-initializing the procedure with the new set of P parents. However, the value of δv_{target} is only related with the benefit towards growth, rendering the iterative procedure insufficient for the purpose of finding high productivity strain designs. For this purpose, the authors proposed a modification in the OptKnock procedure, where the candidates for reaction knockouts are limited to those selected by FastPros, which yielded very positive results. The FastPros procedure was used in the screening of geranyl diphosphate and L-phenylalanine *E. coli* designs.

2.3.2.3 Elementary modes analysis based methods

The final group of methods includes the CSOMs that use elementary modes analysis (EMA) as their foundational framework. Klamt and Gilles intro-

duced the concept of minimal cut sets (MCSs) in 2004 [154] representing the first draft of what an EMA-based framework for strain optimization could look like. A minimal cut set describes an irreducible group of reactions required to disrupt a determined network function (a targeted reaction - r_{obj}), and thus, in a certain way, MCSs are the opposite of EFMs, which describe the minimal functional modes. Similarly to EFMs, the set of MCSs in a network is also unique. As a limitation, the computation of the EFMs of the network is required a priori, which has limited the uses of MCSs to small to medium size networks. After the computation of the EFMs is performed, the EFMs set is divided in two sub-sets, one containing all the EFMs that involve the target reaction r_{obj} (the target modes - E^t) and another containing all the EFMs that do not involve the target reaction (the non-target modes - E^{nt}). By ensuring that all the target modes become inactive after the removal of a set C of reactions, then only non-target modes will be left hence, by the definition of EFMs, it is no longer possible to find a feasible flux distribution involving r_{obj}. An MCS "hitting" all target modes is termed a minimal hitting set and the computation of the minimal hitting sets from the set of target modes can be performed using the Berge algorithm [155]. In this work, several MCSs possible applications have been discussed, including: target identification and repression of cellular functions, network verification and mutant phenotype predictions, as well as structural fragility and robustness analyses. Later, the concept of MCSs was refined, generalized to multiple targets and its duality properties with EFMs were studied in more detail [156].

The main limitation of the MCS approach is that, by being focused on a target functionality to be disrupted, it is rendered oblivious to possible side effects over other desired functionalities. To tackle this limitation, Hädicke

and Klamt developed a generalized approach of MCSs termed constrained minimal cut sets (cMCSs), allowing for the inclusion of side constraints [157]. A cMCS C is now admissible if it hits all target modes, but also maintains a minimum number n of desired EFMs. It is not expected that this set (D) will keep all the desired modes, meaning that it is possible that some of the desired modes will also be hit by some MCSs. However, the set of desired modes not hit by any MCS (D^C) is bounded by $|D^C| \geq n$. The introduction of these constraints implicated an adaptation of the Berge algorithm. This approach has been demonstrated for the production of ethanol in *E. coli*. The inclusion of regulatory constraints into cMCSs was later done by Jungreuthmaye and Zanghellini, devising the regulatory constrained MCSs (rcMCSs) [158].

Concurrently, Srienc and collaborators presented another EMA-based approach named minimal metabolic functionality (MMF) [159]. This approach iteratively searches for all the combinations of gene deletions that will eliminate all undesired EFMs, while keeping a set of optimal or near optimal ones intact. Similarly to other approaches, coupling of biomass and product synthesis can be enforced by the selected knockout strategies. The application of MMF yields a network containing only its most efficient pathways, and its applicability was verified experimentally for the production of several secondary metabolites in *E. coli*.

The first EMA-based approach to consider other types of metabolic engineering interventions besides knockouts was FluxDesign [160]. This procedure first computes the set of EFMs resorting to the EFMTool [161], followed by a normalization step, that for a given EFM calculates the relative flux for each of its reactions, normalized to the substrate uptake. Finally, to decide whether or not a given reaction r represents a potential

intervention target, a chosen set of EFMs is searched for (statistically) significant correlations between the flux through the objective reaction and the flux through reaction r. This strategy yields targets for both amplification (up regulation) and attenuation (down regulation). FluxDesign was demonstrated for lysine and enzyme production in *C. glutamicum* and *Aspergillus niger*, respectively.

Another approach for suggesting multiple types of genetic interventions came from Hädicke and Klamt [162]. The proposed computational approach for strain optimization aimed at high productivity (CASOP) is based on reaction importance measures, where their relative contributions to yield and flux capacity are taken into account. With the purpose of finding growth-coupled high-productivity strain designs for a target product P, the procedure begins by considering an artificial external metabolite (V),which is produced from biomass and the target product by a reaction (here simplified):

$$R_v : (1-\gamma)BM + \gamma P(ext) \rightarrow V, \gamma \in [0,1]$$

with γ representing the relative production of $P(ext)$ with respect to biomass (BM) synthesis. By iteratively increasing the parameter γ, computing the set of EFMs for the new scenario, and recalculating the EM weights and reaction importances, the procedure yields a set of candidates for knockouts and overexpressions. Succinate over-productive strain designs were suggested in *E. coli* to demonstrate the applicability of this approach.

More recently, Trinh and co-workers presented the systematic multiple enzyme targeting approach (SMET) [163]. This method uses cMCS to find the set of modes maximizing a desired product yield, and ensemble metabolic modeling (EMM) to generate ensemble models representing the

steady-state phenotype of the wild-type strain. Afterwards, SMET is used to systematically identify sets of enzyme targets to engineer the wild-type strain to reach the desirable phenotype. The SMET approach is capable of suggesting designs based on deletions or over/under expressions and has been validated in the DHAP metabolic network. Until recently, the use of EMA-based methods, including application to strain optimization, was limited to small to medium scale metabolic networks, as the number of possible EFMs increases exponentially with the network size. Indeed, all previously mentioned validation examples do not use GSMMs, but rather small/ medium sized models with selected pathways.

Some approaches to circumvent this issue have been suggested in the past [164, 165], but with limited applicability in this context. A big step forward into bringing EMA-based CSOMs to the genomic scale, was taken by de Figueiredo and collaborators, with the publication of a method to generate the shortest EFMs in a genome-scale metabolic network [166]. While not being able to find all EFMs for the network, this method allows to calculate iteratively sets of EFMs of interest for a given ME task (e.g. those achieving a given, or set of, chemical transformations). As an illustration, the method has been employed within CASOP to extend it to genome-scale metabolic models via an heurist approach named CASOP-GS [167].

Even more noteworthy is the latest work of Kamp and Klamt, MCSEnumerator, published in early 2014. By exploiting the dual properties of EFMs and MCSs and combining them with a modified version of the de Figueiredo approach, MCSEnumerator is able to identify thousands of the most efficient intervention strategies in a genome-scale metabolic network, involving up to 7 knockouts. More specifically, they begin by converting the original network and intervention objective to its dual form, following

the Ballerstein approach [168] followed by an enumeration of the k shortest EFMs in the dual network using the modified version of the de Figueiredo approach. The EFMs in the dual network correspond to the MCSs in the primal, thus, the shortest EFMs in the dual will correspond to the smallest MCSs in the primal. The framework is formulated as a MILP problem, elegantly extended to represent sets. Afterwards, MCSEnumerator is demonstrated, first by enumerating all the synthetic lethals up to 5 knockouts in a GSMM of *E. coli* [41], followed by the enumeration of all the cMCSs up to 7 reactions leading to growth-coupled ethanol production in the same model.

2.3.3 Comparative Analysis of the different classes of CSOMs

After reviewing all the relevant methods, it is important to provide an overall discussion of their merits and limitations. Figure 2.6 supports this discussion by organizing the most relevant methods according to several criteria, including scalability, ability to generate exact optimal solutions and year of development. Also, other aspects related to the optimization task, experimental validation and patent applications are also represented. Importantly, in the figure, the links represent an attempt to define the methods' phylogeny, i.e. to identify methods that are extensions or, at least, are strongly based in previous ones.

One important trend that is clear from Figure 2.6, looking at the dates of development, is the growing number of published methods over the last few years. This ascertains the relevance of the problem and the growing need for efficient strain optimization methods. Another observable conclusion is the clear trade-off between scalability and the guarantee to reach the

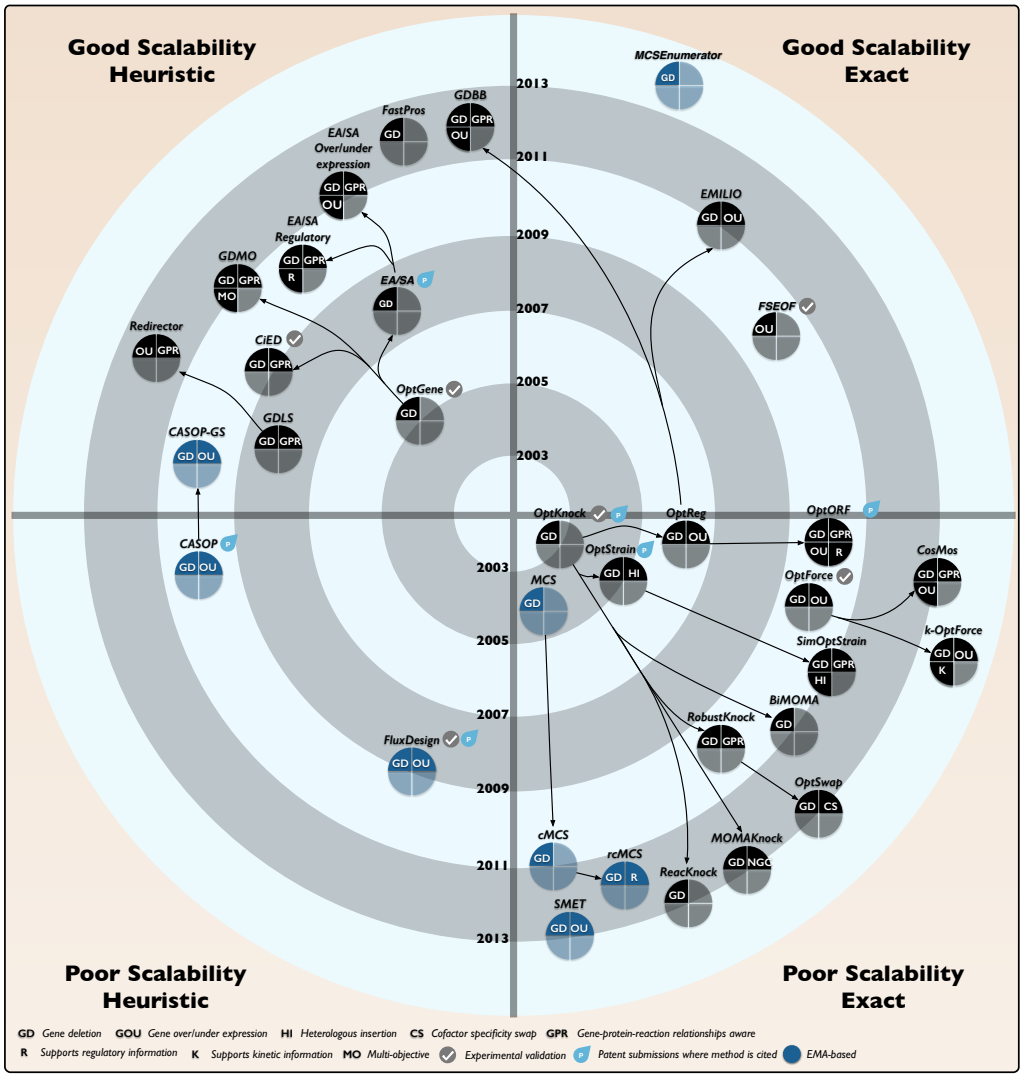


Figure 2.6: Properties of CSOMs and their relationships. The target plot is sectioned into 4 discrete quadrants regarding scalability and exactness. Methods in the upper region are considered more scalable while methods in the bottom region are considered less scalable. Similarly, the methods located in the right region of the figure guarantee that, if it exists, the optimal solution for the specified objective function is always found, while the methods in the left region, being inherently heuristic, do not. Other features are indicated in the legend (bottom of the figure).

global optimal solution. Most MIP and EMA-based methods' formulations guarantee that, for the implemented objective function, the global optimal solution is always found. To do this, the complexity of the mathematical

formulations is usually incremented in such a way that, for larger models or for a higher number of tested perturbations, they usually do not scale well. Some authors addressed these scalability problems by suggesting ways to simplify the problems in MIP-based methods [122, 138] or by computing subsets of the solution space in EMA-based approaches [166]. The recent developments from Klamt and co-workers [169] hold the promise to scale-up EMA-based strain optimization and boost their utility in real-world applications.

Another approach is the use of metaheuristic CSOMs, which usually scale well with larger models or higher number of tested perturbations. However, there is no guarantee that the global optimal will be found. When the solution space is large enough, they will usually reach optimal or near-optimal solutions faster than MIP or EMA-based methods. However, even if they often perform better than these methods in large scale problems, since further increasing the number of allowed perturbations will cause the solution space to grow exponentially, this might make it hard even for metaheuristic methods to find good solutions.

Metaheuristic methods are also more flexible when it comes to the specification of objective functions since they are not bounded by linearity. For most MIP-based methods, the introduction of a non-linear objective function will yield computationally very expensive MINLP problems.

The inclusion of support for gene based intervention strategies, through GPR information, within the strain design methods is another trend adopted by several authors along the years. Although this increases the complexity of the overall approach, most solutions reached when only reactions are accounted for, will be unfeasible once the GPR relationships are scrutinized. Common problems in this context are related with the existence of

isoenzymes and, more importantly, to the fact that the same gene might be associated with several reactions, some of which can be essential for biomass growth and others important for redirecting the metabolic flux when removed.

One important aspect for the applicability of the discussed methods is how easily they can be accessed and used. Despite descriptive formulations for most of the methods are provided in their respective publications, their implementations are generally not available. The methods from the OptGene family (EAs/SAs) and OptKnock can be readily accessed through the OptFlux workbench [170] and all the code is available under a GPL open-source license. The COBRA toolbox [171] also provides open-source implementations for some of these methods, such as OptKnock, OptGene and GDLS, however, it depends on the commercial software MATLAB (The MathWorks Inc., USA). Moreover, access to most of the methods from the Klamt group, such as CASOP and cMCS, is provided via CellNetAnalyzer [172], but the source code is not disclosed. Complete information regarding the availability of the discussed methods has been included in the supplementary table 1.

2.4 Applications and Discussion

GSMM reconstructions and related querying methods have been employed as guiding tools for the development and optimization of bioprocesses for a wide range of industrially relevant chemicals, such as lycopene [173, 174], malate and succinate [175, 176], L-threonine [177], L-valine [178] and diapolycopendioic acid [179]. Other applications include bio-fuels, such as bioethanol [180] or biohydrogen [181], and drug target discovery [59, 182].

Recent reviews discussing the various applications of GSMMs are available [14, 183–186].

Despite the positive results accomplished by employing GSMMs and related querying methods, elaborate strategies such as amplification/attenuation of gene expressions or strategies requiring multiple types of interventions, are neither attainable by simple network inspection nor by exhaustive/iterative strategies [187]. For these complex and non-intuitive strategies, the use of CSOMs is put forward. Academic researchers have concentrated some efforts into the validation of such methods. However, it is still not clear whether the application of CSOMs in guiding ME applications is already being used by the industry or if it is only limited to the academic spectrum. Since CSOMs are centerpiece of this review, in the following discussion we will mainly focus on applications where at least one of the discussed methods has been employed. To support this discussion we put together Table 2.2, which provides a summary of the main experimental validations performed for CSOMs results, and Table 2.3, where some industrial patents and patent applications referring to the use of at least one CSOM are listed.

Two years after the publication of OptKnock, Fong and co-workers implemented the predicted gene deletions for growth-coupled lactate production in *E. coli* and performed the first experimental validation of a CSOM [188]. They concluded that the constructed strains were not only overproducing lactate, but that this production was indeed coupled to growth, according to OptKnock predictions. Notwithstanding this positive confirmation, some discrepancies between the computationally predicted and experimentally observed growth rates were also reported. Since then, OptKnock has been employed in several other efforts, including increas-

ing the respiratory rate in *G. sulfurreducens* [189] and the production of 1,4-butanediol [190] and 2,3-butanediol [191] in *E. coli* and *S. cerevisiae respectively.* More recently, particularly through the hands of Genomatica (Genomatica, Inc. USA), several patent applications referring to the use of OptKnock have been filled, related with the development of microorganisms for the production of adipic acid [192], 1,4-butanediol [193, 194] and the polyester precursor cyclohexanedimethanol [195]. Another patent application mentioning OptKnock and OptStrain in the engineering of primary alcohol producing microbes has also been filed [196].

Subsequently, the usefulness of some heuristic CSOMs was also gradually asserted. Asadollahi and co-workers resorted to OptGene to find S. cerevisiae sesquiterpene-producing mutants with 85% increase in titer [197]. Here, the flexibility of the decoupled bi-level heuristics became evident, since the authors employed MOMA as the phenotype simulation method. A year later, Brochado and colleagues revisited S. cerevisiae, this time for the overproduction of vanillin, resorting to OptGene as the strain design method, which resulted in a 5-fold increase when compared to a previous report [200]. Discrepancies between the predicted results and the batch cultivations were attributed to the lack of kinetic and regulatory information, reinforcing the need to invest in the integration of such information with GSMMs and in the development of simulation and optimization methods that take this information into account. More recently, Otero and coworkers employed OptGene in the design of a S. cerevisiae strain, which improved succinate yield on biomass by 43-fold in comparison with the reference strain [205]. Moreover, in 2013, Invista (Invista North America S.A.R.L. USA) filed patent applications referring to the use of OptFlux [170], whose implemented strain optimization methods are direct de-

Table 2.2: Experimental validation of CSOMs found in the current literature.

Organism (model)	**Application**	**Year**	**Methods**	**Results**	**Ref**
E. coli (iJR904)	Lactate	2005	OptKnock	- 25% increase in pta-adhE strains; - 73% increase in pta-pfk strains; - 55% increase in pta-adhE-pfl-glk strains.	[188]
G. sulfurreducens (iRM588)	Respiratory rate	2008	OptKnock	Higher respiratory rate and increase electron transfer. Expected decrease in growh.	[189]
E. coli (iJR904)	Malonyl-CoA	2009	CiED	Specific flavone yields increased by over 600%.	[147]
S. cerevisiae (iFF708)	Sesquiterpene	2009	OptGene MOMA	85% increase in titer.	[197]
E. coli (EcoMBEL979)	Lycopene	2010	FSEOF	Maximum 3.2-fold increase in lycopene.	[198]
E. coli (iJR904)	NADPH availability	2010	CiED MOMA	- 4-fold increase in leucolyanidin; - 2-fold increase in catechin.	[199]
S. cerevisiae (iFF708)	Vanillin	2010	OptGene MOMA FVA MMT[a]	5-fold increase compared to [200].	[201]
C. glutamicum (ATCC 13032)	L-lysine	2011	FluxDesign EMA	Nearly 2-fold yield increase compared. to [203]	[202]
E. coli (iJR904)	1,4-butanediol	2011	OptKnock	Over 3-fold increase in titer.	[190]
E. coli (iJR904)	Malonyl-CoA	2011	OptForce	- 4-fold increase in intracellular levels of malonyl-CoA; - Highest yield of naringenin in a lab-scale fermentation ever achieved.	[135]
E. coli (iAF1260)	Fatty acids	2012	OptForce	Over 20% yield increase among all the tested strains.	[136]
B. subtilis (iYO844)	Isobutanol	2012	FluxDesign EMA	61% of maximum theoretical yield, a 2.3-fold increase.	[204]
S. cerevisiae (iMM904)	2,3-butanediol	2012	OptKnock	2,3-butanediol titer (2.29 g/L) and yield (0.113 g/g) under anaerobic conditions.	[191]
S. cerevisiae (iMM904)	Succinate	2013	OptGene	30-fold improvement in succinate titer and 43-fold improvement in succinate yield compared to the reference strain.	[205]

[a] *MMT - minimization of metabolite turnover - described in the supplementary material of [201].*

Table 2.3: Industrial patent application/grants of microorganisms, products or processes referring the use of at least one CSOM.

Description	Patent number	Year	Assignee	CSOM	Ref
Microorganisms for the production of adipic acid and other compounds.	US7799545 B2	2010	Genomatica, Inc.	OptKnock	[192]
Microorganisms for the production of 1,4-butanediol, 4-hydroxybutanal, 4-hydroxybutyryl-coa, putrescine and related compounds, and methods related thereto.	US20110229946 A1	2011	Genomatica, Inc.	OptKnock	[193]
Primary alcohol producing organisms.	US7977084 B2	2011	Genomatica, Inc.	OptKnock OptStrain	[196]
Microorganisms and methods for the production of 1,4-cyclohexanedimethanol.	US20120156740 A1	2012	Genomatica, Inc.	OptKnock	[195]
Cytosolic isobutanol pathway localization for the production of isobutanol.	US8232089 B2	2012	Gevo, Inc.	OptORF	[206]
Microorganisms for the production of 1,4-butanediol and related methods.	US8129169 B2	2012	Genomatica, Inc.	OptKnock	[194]
Methods of producing 6-carbon chemicals via CoA-dependent carbon chain elongation associated with carbon storage.	US20130183728 A1	2013	Invista North America S.A R.L.	OptFlux (EA/SA)	[207]
Method for the preparation of 2,4-dihydroxybutyrate.	WO2014009435 A1	2014	Adisseo France S.A.S.	CASOP	[208]
A microorganism modified for the production of 1,3-propanediol.	WO2014009432 A2	2014	Institut National Des Sciences Appliquées	CASOP	[209]

scendants of OptGene, describing a method for producing 6-carbon chemicals [207].

Other heuristic approaches were also experimentally validated by Fowler and co-workers for the overproduction of malonyl-CoA [147] and by Chemler and colleagues [199] for the increased availability of NADPH, both in *E.*

coli. Fowler's work culminated in flavone yields increased by over 600% while Chemler's translated into a 4-fold increase in leucocyanidin and a 2-fold increase in catechin.

Naturally, validations of gene deletions suggested by CSOMs were the first to be performed and they were indeed successful up to a certain degree. However, other more elaborate strategies are not so easy to attain. An obvious example is the identification of gene amplification/attenuation targets, which are not necessarily reflected by an increase in the metabolic flux, due to the complexity of the regulatory machinery. Despite the efforts directed to this subject, quantitative predictions of the flux distribution following this type of interventions are still in their infancy. OptReg was specifically designed for this purpose but has never been experimentally validated.

In fact, the first validation of a method suggesting over/under expression of genes was performed by Choi for FSEOF [198], revisiting the lycopene overproduction in *E. coli*. The FSEOF procedure consists in changing the objective function of the classic biomass maximization FBA problem, by considering an additional constraint enforcing the production of the desired target. The iterative procedure progressively increases the enforced minimum value of the product, while scanning the remaining flux distribution for relevant flux changes (relatively to the wild-type). The most commonly changing fluxes are considered preferential targets for manipulation. The best strain achieved in this work resulted in a 3.2 fold lycopene increase in comparison with the control strain, and a titer slightly in excess of that reported by Alper and co-workers [174].

Another of such methods is OptForce, which was promptly validated in *E. coli* for the production of malonyl-CoA [135]. OptForce application was translated into a 4-fold increase in intracellular levels of malonyl-CoA

and the highest yield of naringenin in a lab-scale fermentation reported. OptForce was also successfully applied in the *E. coli* overproduction of fatty-acids [136], with an overall 20% increase among all the strains.

The last class of methods discussed in this review, EMA-based CSOMs, has only one member for which experimental validation was performed. FluxDesign was first validated for the production of L-lysine in *C. glutamicum* where a 2-fold increase in yield was achieved in comparison with an existing strain [202], and later for the production of isobutanol, where a strain operating at 61% of the maximum theoretical yield was engineered.

Despite no academic validation has been performed for CASOP, it has recently been cited in a patent application of Adisseo (Adisseo France S.A.S. France), describing a method for the preparation of 2,4-dihydroxubutyrate [208], and another by the Institut National Des Sciences Appliquées (France) for the engineering of a 1,3-propanediol producing microorganism [209].

Up until recently, EMA-based approaches were limited to small to medium networks, which forced its application to be subjected to a biased selection process of the most promising pathway(s), which ultimately hinders the holistic vision characteristic to systems biology applications.

Overall, however, and despite these successful in vivo applications, the obtained improvements in yields, productivities and titers are still far away from the ones obtained with non-rational traditional approaches such as random mutagenesis. In fact, it has been reported, as an illustrative example, over 1000-fold increase on the amount of penicillin produced in Fleming's original culture of *P. chrysogenum* [210] by the use of X-Rays, UV rays or other mutagens since the 1950s. Although this time-span is not compatible with the needs of modern biotechnology, this scale of improvement is clearly currently unattainable with rational methods.

One of the reasons for this is clearly the lack of relevant information in the used models, but it is also difficult to assess where additional developments should be focused: if in the development of better models, in simulation methods or in optimization tools, since most validation efforts are not exploited in sufficient detail. For example, there is a clear lack of studies that include more than one round of *in silico* design to *in vivo* implementation, and where advanced –omics tools are used to characterize the developed strains, feeding and improving the in silico predictions. In the future, studies focusing on characterizing in detail the strains constructed from rational approaches, reporting failed efforts and several cycles of intervention are necessary to assess where the main bottlenecks are. In fact, since most of the few validation efforts have been reported associated with specific CSOMs, which would be the final layer of an in silico approach, it is often impossible to decouple the effects of model predictions from the CSOM results themselves. More studies are thus needed that would allow the separate the validation of GSMMs, simulation tools and CSOMs. Only to mention an extreme example, the validation of simulation methods such as FBA, MOMA or ROOM in a large scale has never been performed apart from the examples used when the methods have been developed or improved, a gap attributable in part to the scarcity of flux distribution experimental data.

Other studies that are lacking include the systematic utilization of the vast amount of information on strains developed using non-rational approaches to understand if the existing models/simulation/optimization tools allow to reproduce the successful approaches and what can be learnt from those strains.

In summary, although improvements are clearly needed in the develop-

ment of novel CSOM methods that were very scarce until recently, it is probably also important to focus research efforts on the experimental validation of *in silico* approaches to foster the adoption of rational tools in industrial biotechnology.

2.5 Concluding Remarks

The shift from traditional chemical synthesis processes to biotechnological ones holds the promise to reshape the industrial landscape in the 21^st century. Essentially driven by energy security and climate changes, the road towards a bio-based economy still faces several barriers. The investment in high-risk R&D, infrastructure to support it and guarantees of a steady and controlled supply of raw materials (mostly agricultural products or by-products) are among the most sensitive issues to be addressed. Another barrier is the public perception towards some biotechnologies, such as the use of genetically modified microorganisms (GMOs) in agriculture and food processing. Moreover, the use of farmable land for non-food crops in a growing world population raises ethical concerns that must be properly addressed [211]. Current worldwide revenue estimates of biotechnology-derived goods reach around EUR 60 billion while some predictions for 2030 place this value in nearly EUR 300 billion [211]. The adoption of these knowledge-based approaches is dependent on global policies supporting the improvement, validation and scaling-up of these technologies, reducing their risk and making them more attractive to the industry. In fact, both the USA with the National Bioeconomy Blueprint [212] and Europe with the Knowledge-based Bio-Economy (KBBE) program [213] are setting clear objectives and allocating public resources into these matters.

While the development and validation of rational-approaches focused on the development of microorganisms for the production of biofuels and other bio-based chemicals is evolving steadily, the same cannot be said regarding agricultural efforts. A clear indicator of this gap is the relative scarcity of GSMMs in the plant kingdom with reconstructions for only four higher plants available [214]. The rational engineering of plants for both food crops and biomass (for other purposes, such as biofuels) will require a stronger investment both at the economic and at the policy-making levels.

In the various research fronts towards a biobased economy, the development of reliable GSMMs, robust phenotype prediction methods and efficient strain optimization algorithms will increasingly become more relevant. GSMMs and phenotype prediction methods are already used to some extent in order to guide and evaluate rational engineering efforts [215], although, as mentioned above, further validation efforts are necessary in both fronts.

Specific limitations regarding the scalability of exact strain optimization methods, such as the ones supported by MIP-based or EMA-based implementations are gradually being dealt with or circumvented by various authors. More recently these limitations are becoming more tractable and a convergence between EMA-based and MIP-based approaches is to be expected. In fact, Hädicke and Klamt, describe how both the OptKnock and RobustKnock methods can be reformulated as corresponding cMCS problems [157].

Another limitation affecting the precision and feasibility of the metabolic engineering strategies is intimately connected with the lack of kinetic and regulatory information available and considered in the discussed approaches. The recently proposed k-OptForce from Chowdhury and co-workers [140] looks like a step in the right direction. While acquiring kinetic data will re-

main a difficult problem, already well-know phenomena and parameters can be exploited by this approach in tandem with the more common stoichiometric representations of metabolism, improving the predictive capabilities of previous methods.

Also within the last decade, a large panoply of methods proposing the integration of transcriptional data within constraint-based metabolic models in order to improve the predictive capabilities of flux distributions have also been proposed [216]. However, the number of CSOMs exploring these capabilities is still scarce, leaving room for the emergence of new branches from previously proposed CSOMs or even entirely new ones.

In this review, we presented a highlight of the main computational strain optimization methods proposed until the today. These methods were segmented into three distinct categories and their functionalities and main applications were analyzed. As the number of genome-scale reconstructions increase each year and the complexity of these models is able of capturing more and more information [217], new methods capable of exploring this information to generate new knowledge are expected to rise, holding the promise of finally bridging the gap between the academic-grade efforts and the industry-grade standards required for a full adoption of CSOMs as a standard tool for guiding metabolic engineering efforts. Nevertheless, this adoption is only possible if underlying tools such as modelling and simulation are also further developed and validated.

REFERENCES

[1] OECD, "The Bioeconomy to 2030: Designing a policy agenda," tech. rep., July 2009.

[2] D. A. Jackson, R. H. Symons, and P. Berg, "Biochemical method for inserting new genetic information into DNA of Simian Virus 40: circular SV40 DNA molecules containing lambda phage genes and the galactose operon of Escherichia coli.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 69, pp. 2904–9, Oct. 1972.

[3] P. E. Lobban and A. Kaiser, "Enzymatic end-to-end joining of DNA molecules," *Journal of Molecular Biology*, vol. 78, pp. 453–471, Aug. 1973.

[4] S. N. Cohen, A. C. Y. Chang, H. W. Boyer, and R. B. Helling, "Construction of Biologically Functional Bacterial Plasmids In Vitro," *Proceedings of the National Academy of Sciences*, vol. 70, pp. 3240–3244, Nov. 1973.

[5] F. Sanger and A. Coulson, "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase," *Journal of Molecular Biology*, vol. 94, pp. 441–448, May 1975.

[6] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 74, pp. 5463–7, Dec. 1977.

[7] R. Fleischmann, M. Adams, O. White, R. Clayton, E. Kirkness, A. Kerlavage, C. Bult, J. Tomb, B. Dougherty, J. Merrick, and e. Al., "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd," *Science*, vol. 269, pp. 496–512, July 1995.

[8] G. Stephanopoulos, A. A. Aristidou, and J. Nielsen, *Metabolic Engineering: Principles and Methodologies*. 1998.

[9] J. Nielsen, "Metabolic engineering," *Applied Microbiology and Biotechnology*, vol. 55, pp. 263–283, Apr. 2001.

[10] J. E. Bailey, "Toward a science of metabolic engineering," *Science*, vol. 252, pp. 1668–1675, June 1991.

[11] S. C. Schuster, "Next-generation sequencing transforms today's biology.," *Nature methods*, vol. 5, pp. 16–8, Jan. 2008.

[12] C. S. Henry, M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens, "High-throughput generation, optimization and analysis of genome-scale metabolic models.," *Nature biotechnology*, vol. 28, pp. 977–82, Sept. 2010.

[13] M. Tomita, "Whole-cell simulation: a grand challenge of the 21st century," *Trends in Biotechnology*, vol. 19, pp. 205–210, June 2001.

[14] A. Bordbar, J. M. Monk, Z. A. King, and B. O. Palsson, "Constraint-based models predict metabolic and associated cellular functions.," *Nature reviews. Genetics*, vol. 15, pp. 107–20, Feb. 2014.

[15] J. M. Savinell and B. O. Palsson, "Optimal selection of metabolic fluxes for in vivo measurement. I. Development of mathematical methods," *Journal of Theoretical Biology*, vol. 155, pp. 201–214, Mar. 1992.

[16] J. M. Savinell and B. O. Palsson, "Optimal selection of metabolic fluxes for in vivo measurement. II. Application to Escherichia coli and hybridoma cell metabolism," *Journal of Theoretical Biology*, vol. 155, pp. 215–242, Mar. 1992.

[17] K. J. Kauffman, P. Prakash, and J. S. Edwards, "Advances in flux balance analysis," *Current Opinion in Biotechnology*, vol. 14, pp. 491–496, Oct. 2003.

[18] A. P. Burgard, P. Pharkya, and C. D. Maranas, "Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization," *Biotechnology and bioengineering*, vol. 84, no. 6, pp. 647–657, 2003.

[19] M. Ederer, R. Schlatter, J. Witt, R. Feuer, J. Bona-Lovasz, S. Henkel, and O. Sawodny, "An introduction to kinetic, constraint-based and Boolean modeling in systems biology," in *2010 IEEE International Conference on Control Applications*, pp. 129–134, IEEE, Sept. 2010.

[20] K. Smallbone, E. Simeonidis, N. Swainston, and P. Mendes, "Towards a genome-scale kinetic model of cellular metabolism.," *BMC systems biology*, vol. 4, p. 6, Jan. 2010.

[21] K. Smallbone, H. L. Messiha, K. M. Carroll, C. L. Winder, N. Malys, W. B. Dunn, E. Murabito, N. Swainston, J. O. Dada, F. Khan, P. Pir, E. Simeonidis, I. Spasić, J. Wishart, D. Weichart, N. W. Hayes, D. Jameson, D. S. Broomhead, S. G. Oliver, S. J. Gaskell, J. E. G. McCarthy, N. W. Paton, H. V. Westerhoff, D. B. Kell, and P. Mendes, "A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes.," *FEBS letters*, vol. 587, pp. 2832–41, Sept. 2013.

[22] A. Varma and B. O. Palsson, "Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use," *Bio/Technology*, vol. 12, pp. 994–998, Oct. 1994.

[23] J. L. Reed, T. D. Vo, C. H. Schilling, and B. O. Palsson, "An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR).," *Genome biology*, vol. 4, p. R54, Jan. 2003.

[24] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Genetic networks with canalyzing Boolean rules are always stable.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 17102–7, Dec. 2004.

[25] S. Li, S. M. Assmann, and R. Albert, "Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling.," *PLoS biology*, vol. 4, p. e312, Oct. 2006.

[26] M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson, "Integrating high-throughput and computational data elucidates bacterial networks.," *Nature*, vol. 429, pp. 92–6, May 2004.

[27] P. A. Jensen, K. A. Lutz, and J. A. Papin, "TIGER: Toolbox for integrating genome-scale metabolic models, expression data, and transcriptional regulatory networks.," *BMC systems biology*, vol. 5, p. 147, Jan. 2011.

[28] I. Thiele, N. Jamshidi, R. M. T. Fleming, and B. O. Palsson, "Genome-scale reconstruction of Escherichia coli's transcriptional and translational machinery: a knowledge base, its mathematical formulation, and its functional characterization.," *PLoS computational biology*, vol. 5, p. e1000312, Mar. 2009.

[29] M. A. Garcia-Albornoz and J. Nielsen, "Application of Genome-Scale Metabolic Models in Metabolic Engineering," *Industrial Biotechnology*, vol. 9, pp. 203–214, Aug. 2013.

[30] C. G. de Oliveira Dal'Molin, L.-E. Quek, R. W. Palfreyman, S. M. Brumbley, and L. K. Nielsen, "AraGEM, a genome-scale reconstruction of the primary metabolic network in Arabidopsis.," *Plant physiology*, vol. 152, pp. 579–89, Feb. 2010.

[31] M. R. r. Andersen, M. L. Nielsen, and J. Nielsen, "Metabolic model integration of the bibliome, genome, metabolome and reactome of Aspergillus niger.," *Molecular systems biology*, vol. 4, p. 178, Jan. 2008.

[32] Y.-K. Oh, B. O. Palsson, S. M. Park, C. H. Schilling, and R. Mahadevan, "Genome-scale reconstruction of metabolic network in Bacillus subtilis based on high-throughput phenotyping and gene essentiality data.," *The Journal of biological chemistry*, vol. 282, pp. 28791–9, Sept. 2007.

[33] C. S. Henry, J. F. Zinner, M. P. Cohoon, and R. L. Stevens, "iBsu1103: a new genome-scale metabolic model of Bacillus subtilis based on SEED annotations.," *Genome biology*, vol. 10, p. R69, Jan. 2009.

[34] R. L. Chang, L. Ghamsari, A. Manichaikul, E. F. Y. Hom, S. Balaji, W. Fu, Y. Shen, T. Hao, B. O. Palsson, K. Salehi-Ashtiani, and J. A. Papin, "Metabolic network reconstruction of Chlamydomonas offers insight into light-driven algal metabolism.," *Molecular systems biology*, vol. 7, p. 518, Jan. 2011.

[35] O. Ates, E. T. Oner, and K. Y. Arga, "Genome-scale reconstruction of metabolic network for a halophilic extremophile, Chromohalobacter salexigens DSM 3043.," *BMC systems biology*, vol. 5, p. 12, Jan. 2011.

[36] C. B. Milne, J. A. Eddy, R. Raju, S. Ardekani, P.-J. Kim, R. S. Senger, Y.-S. Jin, H. P. Blaschek, and N. D. Price, "Metabolic network reconstruction and genome-scale model of butanol-producing strain Clostridium beijerinckii NCIMB 8052.," *BMC systems biology*, vol. 5, p. 130, Jan. 2011.

[37] S. B. Roberts, C. M. Gowen, J. P. Brooks, and S. S. Fong, "Genome-scale metabolic analysis of Clostridium thermocellum for bioethanol production.," *BMC systems biology*, vol. 4, p. 31, Jan. 2010.

[38] M. Ahsanul Islam, E. A. Edwards, and R. Mahadevan, "Characterizing the metabolism of Dehalococcoides with a constraint-based model.," *PLoS computational biology*, vol. 6, Jan. 2010.

[39] R. Rex, N. Bill, K. Schmidt-Hohagen, and D. Schomburg, "Swimming in light: a large-scale computational analysis of the metabolism of Dinoroseobacter shibae.," *PLoS computational biology*, vol. 9, p. e1003224, Jan. 2013.

[40] J. S. Edwards and B. O. Palsson, "The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities," *Proceedings of the National Academy of Sciences*, vol. 97, pp. 5528–5533, May 2000.

[41] A. M. Feist, C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis, and B. O. Palsson, "A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information.," *Molecular systems biology*, vol. 3, p. 121, Jan. 2007.

[42] J. D. Orth, T. M. Conrad, J. Na, J. A. Lerman, H. Nam, A. M. Feist, and B. O. Palsson, "A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011.," *Molecular systems biology*, vol. 7, p. 535, Jan. 2011.

[43] C. T. Archer, J. F. Kim, H. Jeong, J. H. Park, C. E. Vickers, S. Y. Lee, and L. K. Nielsen, "The genome sequence of E. coli W (ATCC 9637): comparative genome analysis and an improved genome-scale reconstruction of E. coli.," *BMC genomics*, vol. 12, p. 9, Jan. 2011.

[44] R. Mahadevan, D. R. Bond, J. E. Butler, A. Esteve-Nuñez, M. V. Coppi, B. O. Palsson, C. H. Schilling, and D. R. Lovley, "Characterization of metabolism in the Fe(III)-reducing organism Geobacter sulfurreducens by constraint-based modeling.," *Applied and environmental microbiology*, vol. 72, pp. 1558–68, Mar. 2006.

[45] C. H. Schilling, D. Letscher, and B. O. Palsson, "Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective.," *Journal of theoretical biology*, vol. 203, pp. 229–48, Apr. 2000.

[46] C. H. Schilling, M. W. Covert, I. Famili, G. M. Church, J. S. Edwards, and B. O. Palsson, "Genome-Scale Metabolic Model of Helicobacter pylori 26695," *Journal of Bacteriology*, vol. 184, pp. 4582–4593, Aug. 2002.

[47] I. Thiele, T. D. Vo, N. D. Price, and B. O. Palsson, "Expanded metabolic reconstruction of Helicobacter pylori (iIT341 GSM/GPR):

an in silico genome-scale characterization of single- and double-deletion mutants.," *Journal of bacteriology*, vol. 187, pp. 5818–30, Aug. 2005.

[48] N. C. Duarte, S. A. Becker, N. Jamshidi, I. Thiele, M. L. Mo, T. D. Vo, R. Srivas, and B. O. Palsson, "Global reconstruction of the human metabolic network based on genomic and bibliomic data.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 1777–82, Feb. 2007.

[49] I. Thiele, N. Swainston, R. M. T. Fleming, A. Hoppe, S. Sahoo, M. K. Aurich, H. Haraldsdottir, M. L. Mo, O. Rolfsson, M. D. Stobbe, S. G. Thorleifsson, R. Agren, C. Bölling, S. Bordel, A. K. Chavali, P. Dobson, W. B. Dunn, L. Endler, D. Hala, M. Hucka, D. Hull, D. Jameson, N. Jamshidi, J. J. Jonsson, N. Juty, S. Keating, I. Nookaew, N. Le Novère, N. Malys, A. Mazein, J. A. Papin, N. D. Price, E. Selkov, M. I. Sigurdsson, E. Simeonidis, N. Sonnenschein, K. Smallbone, A. Sorokin, J. H. G. M. van Beek, D. Weichart, I. Goryanin, J. Nielsen, H. V. Westerhoff, D. B. Kell, P. Mendes, and B. O. Palsson, "A community-driven global reconstruction of human metabolism.," *Nature biotechnology*, vol. 31, pp. 419–25, May 2013.

[50] Y.-C. Liao, T.-W. Huang, F.-C. Chen, P. Charusanti, J. S. J. Hong, H.-Y. Chang, S.-F. Tsai, B. O. Palsson, and C. A. Hsiung, "An experimentally validated genome-scale metabolic reconstruction of Klebsiella pneumoniae MGH 78578, iYL1228.," *Journal of bacteriology*, vol. 193, pp. 1710–7, Apr. 2011.

[51] V. Satish Kumar, J. G. Ferry, and C. D. Maranas, "Metabolic reconstruction of the archaeon methanogen Methanosarcina Acetivorans.," *BMC systems biology*, vol. 5, p. 28, Jan. 2011.

[52] M. N. Benedict, M. C. Gonnerman, W. W. Metcalf, and N. D. Price, "Genome-scale metabolic reconstruction and hypothesis testing in the methanogenic archaeon Methanosarcina acetivorans C2A.," *Journal of bacteriology*, vol. 194, pp. 855–65, Feb. 2012.

[53] A. M. Feist, J. C. M. Scholten, B. O. Palsson, F. J. Brockman, and T. Ideker, "Modeling methanogenesis with a genome-scale metabolic reconstruction of Methanosarcina barkeri.," *Molecular systems biology*, vol. 2, p. 2006.0004, Jan. 2006.

[54] M. C. Gonnerman, M. N. Benedict, A. M. Feist, W. W. Metcalf, and N. D. Price, "Genomically and biochemically accurate metabolic

reconstruction of Methanosarcina barkeri Fusaro, iMG746.," *Biotechnology journal*, vol. 8, pp. 1070–9, Sept. 2013.

[55] K. Sheikh, J. Förster, and L. K. Nielsen, "Modeling hybridoma cell metabolism using a generic genome-scale metabolic model of Mus musculus.," *Biotechnology progress*, vol. 21, no. 1, pp. 112–21, 2005.

[56] L.-E. Quek and L. K. Nielsen, "On the reconstruction of the Mus musculus genome-scale metabolic network model.," *Genome informatics. International Conference on Genome Informatics*, vol. 21, pp. 89–100, Jan. 2008.

[57] S. Selvarasu, I. A. Karimi, G.-H. Ghim, and D.-Y. Lee, "Genome-scale modeling and in silico analysis of mouse cell metabolic network.," *Molecular bioSystems*, vol. 6, pp. 152–61, Jan. 2010.

[58] M. I. Sigurdsson, N. Jamshidi, E. Steingrimsson, I. Thiele, and B. O. Palsson, "A detailed genome-wide reconstruction of mouse metabolism based on human Recon 1.," *BMC systems biology*, vol. 4, p. 140, Jan. 2010.

[59] N. Jamshidi and B. O. Palsson, "Investigating the metabolic capabilities of Mycobacterium tuberculosis H37Rv using the in silico strain iNJ661 and proposing alternative drug targets.," *BMC systems biology*, vol. 1, p. 26, Jan. 2007.

[60] D. J. V. Beste, T. Hooper, G. Stewart, B. Bonde, C. Avignone-Rossa, M. E. Bushell, P. Wheeler, S. Klamt, A. M. Kierzek, and J. McFadden, "GSMN-TB: a web-based genome-scale network model of Mycobacterium tuberculosis metabolism.," *Genome biology*, vol. 8, p. R89, Jan. 2007.

[61] X. Fang, A. Wallqvist, and J. Reifman, "Development and analysis of an in vivo-compatible metabolic network of Mycobacterium tuberculosis.," *BMC systems biology*, vol. 4, p. 160, Jan. 2010.

[62] P. F. Suthers, M. S. Dasika, V. S. Kumar, G. Denisov, J. I. Glass, and C. D. Maranas, "A genome-scale metabolic reconstruction of Mycoplasma genitalium, iPS189.," *PLoS computational biology*, vol. 5, p. e1000285, Feb. 2009.

[63] B. K. Chung, S. Selvarasu, C. Andrea, J. Ryu, H. Lee, J. Ahn, H. Lee, and D.-Y. Lee, "Genome-scale metabolic reconstruction and in silico analysis of methylotrophic yeast Pichia pastoris for strain improvement.," *Microbial cell factories*, vol. 9, p. 50, Jan. 2010.

[64] L. Caspeta, S. Shoaie, R. Agren, I. Nookaew, and J. Nielsen, "Genome-scale metabolic reconstructions of Pichia stipitis and Pichia pastoris and in silico evaluation of their potentials.," *BMC systems biology*, vol. 6, p. 24, Jan. 2012.

[65] J. Nogales, B. O. Palsson, and I. Thiele, "A genome-scale metabolic reconstruction of Pseudomonas putida KT2440: iJN746 as a cell factory.," *BMC systems biology*, vol. 2, p. 79, Jan. 2008.

[66] J. Puchałka, M. A. Oberhardt, M. Godinho, A. Bielecka, D. Regenhardt, K. N. Timmis, J. A. Papin, and V. A. P. Martins dos Santos, "Genome-scale reconstruction and analysis of the Pseudomonas putida KT2440 metabolic network facilitates applications in biotechnology.," *PLoS computational biology*, vol. 4, p. e1000210, Oct. 2008.

[67] S. B. Sohn, T. Y. Kim, J. M. Park, and S. Y. Lee, "In silico genome-scale metabolic analysis of Pseudomonas putida KT2440 for polyhydroxyalkanoate synthesis, degradation of aromatics and anaerobic survival.," *Biotechnology journal*, vol. 5, pp. 739–50, July 2010.

[68] M. A. Oberhardt, J. Puchałka, K. E. Fryer, V. A. P. Martins dos Santos, and J. A. Papin, "Genome-scale metabolic network analysis of the opportunistic pathogen Pseudomonas aeruginosa PAO1.," *Journal of bacteriology*, vol. 190, pp. 2790–803, Apr. 2008.

[69] J. M. Park, T. Y. Kim, and S. Y. Lee, "Genome-scale reconstruction and in silico analysis of the Ralstonia eutropha H16 for polyhydroxyalkanoate synthesis, lithoautotrophic growth, and 2-methyl citric acid production.," *BMC systems biology*, vol. 5, p. 101, Jan. 2011.

[70] J. Förster, I. Famili, P. Fu, B. O. Palsson, and J. Nielsen, "Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network.," *Genome research*, vol. 13, pp. 244–53, Feb. 2003.

[71] N. C. Duarte, M. J. Herrgå rd, and B. O. Palsson, "Reconstruction and validation of Saccharomyces cerevisiae iND750, a fully compartmentalized genome-scale metabolic model.," *Genome research*, vol. 14, pp. 1298–309, July 2004.

[72] L. Kuepfer, U. Sauer, and L. M. Blank, "Metabolic functions of duplicate genes in Saccharomyces cerevisiae.," *Genome research*, vol. 15, pp. 1421–30, Oct. 2005.

[73] I. Nookaew, M. C. Jewett, A. Meechai, C. Thammarongtham, K. Laoteng, S. Cheevadhanarak, J. Nielsen, and S. Bhumiratana, "The genome-scale metabolic model iIN800 of Saccharomyces cerevisiae and its validation: a scaffold to query lipid metabolism.," *BMC systems biology*, vol. 2, p. 71, Jan. 2008.

[74] M. J. Herrgå rd, N. Swainston, P. Dobson, W. B. Dunn, K. Y. Arga, M. Arvas, N. Blüthgen, S. Borger, R. Costenoble, M. Heinemann, M. Hucka, N. Le Novère, P. Li, W. Liebermeister, M. L. Mo, A. P. Oliveira, D. Petranovic, S. Pettifer, E. Simeonidis, K. Smallbone, I. Spasić, D. Weichart, R. Brent, D. S. Broomhead, H. V. Westerhoff, B. Kirdar, M. Penttilä, E. Klipp, B. O. Palsson, U. Sauer, S. G. Oliver, P. Mendes, J. Nielsen, and D. B. Kell, "A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology.," *Nature biotechnology*, vol. 26, pp. 1155–60, Oct. 2008.

[75] M. L. Mo, B. O. Palsson, and M. J. Herrgå rd, "Connecting extracellular metabolomic measurements to intracellular flux states in yeast.," *BMC systems biology*, vol. 3, p. 37, Jan. 2009.

[76] A. R. Zomorrodi and C. D. Maranas, "Improving the iMM904 S. cerevisiae metabolic model using essentiality and synthetic lethality data.," *BMC systems biology*, vol. 4, p. 178, Jan. 2010.

[77] T. Österlund, I. Nookaew, S. Bordel, and J. Nielsen, "Mapping condition-dependent regulation of metabolism in yeast through genome-scale modeling.," *BMC systems biology*, vol. 7, p. 36, Jan. 2013.

[78] H. W. Aung, S. A. Henry, and L. P. Walker, "Revising the Representation of Fatty Acid, Glycerolipid, and Glycerophospholipid Metabolism in the Consensus Model of Yeast Metabolism," *Industrial Biotechnology*, vol. 9, pp. 215–228, Aug. 2013.

[79] A. Raghunathan, J. Reed, S. Shin, B. Palsson, and S. Daefler, "Constraint-based analysis of metabolic capacity of Salmonella typhimurium during host-pathogen interaction.," *BMC systems biology*, vol. 3, p. 38, Jan. 2009.

[80] M. AbuOun, P. F. Suthers, G. I. Jones, B. R. Carter, M. P. Saunders, C. D. Maranas, M. J. Woodward, and M. F. Anjum, "Genome scale reconstruction of a Salmonella metabolic model: comparison of similarity and differences with a commensal Escherichia coli strain.," *The Journal of biological chemistry*, vol. 284, pp. 29480–8, Oct. 2009.

[81] I. Thiele, D. R. Hyduke, B. Steeb, G. Fankam, D. K. Allen, S. Bazzani, P. Charusanti, F.-C. Chen, R. M. T. Fleming, C. A. Hsiung, S. C. J. De Keersmaecker, Y.-C. Liao, K. Marchal, M. L. Mo, E. Özdemir, A. Raghunathan, J. L. Reed, S.-i. Shin, S. Sigurbjörnsdóttir, J. Steinmann, S. Sudarsan, N. Swainston, I. M. Thijs, K. Zengler, B. O. Palsson, J. N. Adkins, and D. Bumann, "A community effort towards a knowledge-base and mathematical model of the human pathogen Salmonella Typhimurium LT2.," *BMC systems biology*, vol. 5, p. 8, Jan. 2011.

[82] K. Y. Lee, J. M. Park, T. Y. Kim, H. Yun, and S. Y. Lee, "The genome-scale metabolic network analysis of Zymomonas mobilis ZM4 explains physiological features and suggests ethanol and succinic acid production strategies.," *Microbial cell factories*, vol. 9, p. 94, Jan. 2010.

[83] H. Widiastuti, J. Y. Kim, S. Selvarasu, I. A. Karimi, H. Kim, J.-S. Seo, and D.-Y. Lee, "Genome-scale modeling and in silico analysis of ethanologenic bacteria Zymomonas mobilis.," *Biotechnology and bioengineering*, vol. 108, pp. 655–65, Mar. 2011.

[84] C. Wagner and R. Urbanczik, "The geometry of the flux cone of a metabolic network.," *Biophysical journal*, vol. 89, pp. 3837–45, Dec. 2005.

[85] R. Schuetz, L. Kuepfer, and U. Sauer, "Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli.," *Molecular systems biology*, vol. 3, p. 119, Jan. 2007.

[86] D. Segrè, D. Vitkup, and G. M. Church, "Analysis of optimality in natural and perturbed metabolic networks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 15112–7, Nov. 2002.

[87] R. U. Ibarra, J. S. Edwards, and B. O. Palsson, "Escherichia coli K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth.," *Nature*, vol. 420, pp. 186–9, Nov. 2002.

[88] J. S. Edwards, R. U. Ibarra, and B. O. Palsson, "In silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data.," *Nature biotechnology*, vol. 19, pp. 125–30, Feb. 2001.

[89] M. W. Covert, C. H. Schilling, and B. Palsson, "Regulation of gene expression in flux balance models of metabolism.," *Journal of theoretical biology*, vol. 213, pp. 73–88, Nov. 2001.

[90] T. Shlomi, Y. Eisenberg, R. Sharan, and E. Ruppin, "A genome-scale computational study of the interplay between transcriptional regulation and metabolism.," *Molecular systems biology*, vol. 3, p. 101, Jan. 2007.

[91] S. A. Becker, A. M. Feist, M. L. Mo, G. Hannum, B. Ø. Palsson, and M. J. Herrgard, "Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox," *Nature protocols*, vol. 2, no. 3, pp. 727–738, 2007.

[92] T. Shlomi, O. Berkman, and E. Ruppin, "Regulatory on/off minimization of metabolic flux changes after genetic perturbations.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 7695–700, May 2005.

[93] A. R. Brochado, S. Andrejev, C. D. Maranas, and K. R. Patil, "Impact of stoichiometry representation on simulation of genotype-phenotype relationships in metabolic networks.," *PLoS computational biology*, vol. 8, p. e1002758, Jan. 2012.

[94] N. E. Lewis, K. K. Hixson, T. M. Conrad, J. A. Lerman, P. Charusanti, A. D. Polpitiya, J. N. Adkins, G. Schramm, S. O. Purvine, D. Lopez-Ferrer, K. K. Weitz, R. Eils, R. König, R. D. Smith, and B. O. Palsson, "Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models.," *Molecular systems biology*, vol. 6, p. 390, July 2010.

[95] R. Carreira, P. Evangelista, P. Maia, P. Vilaça, M. Pont, J.-F. Tomb, I. Rocha, and M. Rocha, "Cbfa: phenotype prediction integrating metabolic models with constraints derived from experimental data," *BMC systems biology*, vol. 8, no. 1, p. 123, 2014.

[96] R. Mahadevan and C. Schilling, "The effects of alternate optimal solutions in constraint-based genome-scale metabolic models," *Metabolic Engineering*, vol. 5, pp. 264–276, Oct. 2003.

[97] S. Klamt and J. Stelling, "Combinatorial Complexity of Pathway Analysis in Metabolic Networks," *Molecular Biology Reports*, vol. 29, pp. 233–236, Mar. 2002.

[98] M. Yeung, I. Thiele, and B. O. Palsson, "Estimation of the number of extreme pathways for metabolic networks.," *BMC bioinformatics*, vol. 8, p. 363, Jan. 2007.

[99] S. Schuster and C. Hilgetag, "On Elementary Flux Modes in Biochemical Reaction Systems at Steady State," *Journal of Biological Systems*, vol. 02, pp. 165–182, June 1994.

[100] J. Schellenberger and B. O. Palsson, "Use of randomized sampling for analysis of metabolic networks.," *The Journal of biological chemistry*, vol. 284, pp. 5457–61, Feb. 2009.

[101] D. Machado, Z. Soons, K. R. Patil, E. C. Ferreira, and I. Rocha, "Random sampling of elementary flux modes in large-scale metabolic networks.," *Bioinformatics (Oxford, England)*, vol. 28, pp. i515–i521, Sept. 2012.

[102] C. Kaleta, L. F. de Figueiredo, and S. Schuster, "Can the whole be less than the sum of its parts? Pathway analysis in genome-scale metabolic networks using elementary flux patterns.," *Genome research*, vol. 19, pp. 1872–83, Oct. 2009.

[103] K. Ip, C. Colijn, and D. S. Lun, "Analysis of complex metabolic behavior through pathway decomposition.," *BMC systems biology*, vol. 5, p. 91, Jan. 2011.

[104] K. A. Datsenko and B. L. Wanner, "One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, pp. 6640–5, June 2000.

[105] C. L. Frazier, J. San Filippo, A. M. Lambowitz, and D. A. Mills, "Genetic Manipulation of Lactococcus lactis by Using Targeted Group II Introns: Generation of Stable Insertions without Selection," *Applied and Environmental Microbiology*, vol. 69, pp. 1121–1128, Feb. 2003.

[106] G. Rodrigo, J. Carrera, K. J. Prather, and A. Jaramillo, "DESHARKY: automatic design of metabolic pathways for optimal cell growth.," *Bioinformatics (Oxford, England)*, vol. 24, pp. 2554–6, Nov. 2008.

[107] A. Cho, H. Yun, J. H. Park, S. Y. Lee, and S. Park, "Prediction of novel synthetic pathways for the production of desired chemicals.," *BMC systems biology*, vol. 4, p. 35, Jan. 2010.

[108] P. Carbonell, D. Fichera, S. B. Pandit, and J.-L. Faulon, "Enumerating metabolic pathways for the production of heterologous target chemicals in chassis organisms.," *BMC systems biology*, vol. 6, p. 10, Jan. 2012.

[109] P. R. Jensen and K. Hammer, "The Sequence of Spacers between the Consensus Sequences Modulates the Strength of Prokaryotic Promoters," *Appl. Envir. Microbiol.*, vol. 64, pp. 82–87, Jan. 1998.

[110] M. A. G. Koffas, G. Y. Jung, and G. Stephanopoulos, "Engineering metabolism and product formation in Corynebacterium glutamicum by coordinated gene overexpression.," *Metabolic engineering*, vol. 5, pp. 32–41, Jan. 2003.

[111] K. Hammer, I. Mijakovic, and P. R. Jensen, "Synthetic promoter libraries–tuning of gene expression.," *Trends in biotechnology*, vol. 24, pp. 53–5, Feb. 2006.

[112] T. Siegl, B. Tokovenko, M. Myronovskyi, and A. Luzhetskyy, "Design, construction and characterisation of a synthetic promoter library for fine-tuned gene expression in actinomycetes.," *Metabolic engineering*, vol. 19, pp. 98–106, Sept. 2013.

[113] A. S. Rajkumar, N. Dénervaud, and S. J. Maerkl, "Mapping the fine structure of a eukaryotic promoter input-output function.," *Nature genetics*, vol. 45, pp. 1207–15, Oct. 2013.

[114] H. M. Salis, E. A. Mirsky, and C. A. Voigt, "Automated design of synthetic ribosome binding sites to control protein expression.," *Nature biotechnology*, vol. 27, pp. 946–50, Oct. 2009.

[115] Z. A. King and A. M. Feist, "Optimizing Cofactor Specificity of Oxidoreductase Enzymes for the Generation of Microbial Production Strains—OptSwap," *Industrial Biotechnology*, vol. 9, pp. 236–246, Aug. 2013.

[116] J. H. Hurley, R. Chen, and A. M. Dean, "Determinants of cofactor specificity in isocitrate dehydrogenase: structure of an engineered NADP+ –¿ NAD+ specificity-reversal mutant.," *Biochemistry*, vol. 35, pp. 5670–8, May 1996.

[117] J. L. Harris and C. S. Craik, "Engineering enzyme specificity," *Current Opinion in Chemical Biology*, vol. 2, pp. 127–132, Feb. 1998.

[118] R. Verho, J. Londesborough, M. Penttila, and P. Richard, "Engineering Redox Cofactor Regeneration for Improved Pentose Fermentation in Saccharomyces cerevisiae," *Applied and Environmental Microbiology*, vol. 69, pp. 5892–5897, Oct. 2003.

[119] J. P. Ignizio and T. M. Cavalier, *Linear programming*. Prentice-Hall international series in industrial and systems engineering, Englewood Cliffs, N.J: Prentice Hall, 1994.

[120] N. Tepper and T. Shlomi, "Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways," *Bioinformatics (Oxford, England)*, vol. 26, no. 4, pp. 536–543, 2010.

[121] A. M. Feist, D. C. Zielinski, J. D. Orth, J. Schellenberger, M. J. Herrgard, and B. O. Palsson, "Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli," *Metabolic Engineering*, vol. 12, no. 3, pp. 173–186, 2010.

[122] J. Kim, J. L. Reed, and C. T. Maravelias, "Large-scale bi-level strain design approaches and mixed-integer programming solution techniques.," *PloS one*, vol. 6, p. e24162, Jan. 2011.

[123] S. Ren, B. Zeng, and X. Qian, "Adaptive bi-level programming for optimal gene knockouts for targeted overproduction under phenotypic constraints.," *BMC bioinformatics*, vol. 14 Suppl 2, p. S17, Jan. 2013.

[124] Z. Xu, P. Zheng, J. Sun, and Y. Ma, "ReacKnock: identifying reaction deletion strategies for microbial strain optimization based on genome-scale metabolic network.," *PloS one*, vol. 8, p. e72150, Jan. 2013.

[125] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, The Regents of the University of California, 1951.

[126] A. Chowdhury, A. R. Zomorrodi, and C. D. Maranas, "Bilevel optimization techniques in computational strain design," *Computers & Chemical Engineering*, July 2014.

[127] P. Pharkya, A. P. Burgard, and C. D. Maranas, "OptStrain: a computational framework for redesign of microbial production systems," *Genome research*, vol. 14, no. 11, pp. 2367–2376, 2004.

[128] M. Kanehisa, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, pp. 27–30, Jan. 2000.

[129] M. Kanehisa, S. Goto, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "Data, information, knowledge and principle: back to metabolism in KEGG.," *Nucleic acids research*, vol. 42, pp. D199–205, Jan. 2014.

[130] R. Caspi, H. Foerster, C. A. Fulcher, R. Hopkinson, J. Ingraham, P. Kaipa, M. Krummenacker, S. Paley, J. Pick, S. Y. Rhee, C. Tissier, P. Zhang, and P. D. Karp, "MetaCyc: a multiorganism database of metabolic pathways and enzymes.," *Nucleic acids research*, vol. 34, pp. D511–6, Jan. 2006.

[131] M. Scheer, A. Grote, A. Chang, I. Schomburg, C. Munaretto, M. Rother, C. Söhngen, M. Stelzer, J. Thiele, and D. Schomburg, "BRENDA, the enzyme information system in 2011.," *Nucleic acids research*, vol. 39, pp. D670–6, Jan. 2011.

[132] P. Pharkya and C. D. Maranas, "An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems," *Metabolic engineering*, vol. 8, no. 1, pp. 1–13, 2006.

[133] A. P. Burgard, S. Vaidyaraman, and C. D. Maranas, "Minimal reaction sets for escherichia coli metabolism under different growth requirements and uptake environments," *Biotechnology progress*, vol. 17, no. 5, pp. 791–797, 2001.

[134] S. Ranganathan, P. F. Suthers, and C. D. Maranas, "OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions," *PLoS computational biology*, vol. 6, no. 4, p. e1000744, 2010.

[135] P. Xu, S. Ranganathan, Z. L. Fowler, C. D. Maranas, and M. A. G. Koffas, "Genome-scale metabolic network modeling results in minimal interventions that cooperatively force carbon flux towards malonyl-CoA.," *Metabolic engineering*, vol. 13, pp. 578–87, Sept. 2011.

[136] S. Ranganathan, T. W. Tee, A. Chowdhury, A. R. Zomorrodi, J. M. Yoon, Y. Fu, J. V. Shanks, and C. D. Maranas, "An integrated computational and experimental study for overproducing fatty acids in Escherichia coli.," *Metabolic engineering*, vol. 14, pp. 687–704, Nov. 2012.

[137] C. Cotten and J. L. Reed, "Constraint-based strain design using continuous modifications (CosMos) of flux bounds finds new strategies for

metabolic engineering.," *Biotechnology journal*, vol. 8, pp. 595–604, May 2013.

[138] L. Yang, W. R. Cluett, and R. Mahadevan, "EMILiO: a fast algorithm for genome-scale strain design.," *Metabolic Engineering*, vol. 13, no. 3, pp. 272–281, 2011.

[139] J. Kim and J. L. Reed, "OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains," *BMC systems biology*, vol. 4, p. 53, Jan. 2010.

[140] A. Chowdhury, A. R. Zomorrodi, and C. D. Maranas, "k-OptForce: Integrating Kinetics with Flux Balance Analysis for Strain Design," *PLoS Computational Biology*, vol. 10, p. e1003487, Feb. 2014.

[141] D. Machado, R. S. Costa, E. C. Ferreira, I. Rocha, and B. Tidor, "Exploring the gap between dynamic and constraint-based models of metabolism.," *Metabolic engineering*, vol. 14, pp. 112–9, Mar. 2012.

[142] K. R. Patil, I. Rocha, J. Förster, and J. Nielsen, "Evolutionary programming as a platform for in silico metabolic engineering," *BMC Bioinformatics*, vol. 6, no. 1, p. 308, 2005.

[143] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[144] M. Rocha, P. Maia, R. Mendes, J. P. Pinto, E. C. Ferreira, J. Nielsen, K. R. Patil, and I. Rocha, "Natural computation meta-heuristics for the in silico optimization of . . . ," *BMC Bioinformatics*, vol. 9, no. 499, 2008.

[145] P. Vilaça, I. Rocha, and M. Rocha, "A computational tool for the simulation and optimization of microbial strains accounting integrated metabolic/regulatory information.," *Bio Systems*, vol. 103, pp. 435–41, Mar. 2011.

[146] E. Gonçalves, R. Pereira, I. Rocha, and M. Rocha, "Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression.," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 19, pp. 102–14, Mar. 2012.

[147] Z. L. Fowler, W. W. Gikandi, and M. A. G. Koffas, "Increased malonyl coenzyme A biosynthesis by tuning the Escherichia coli metabolic

network and its application to flavanone production.," *Applied and environmental microbiology*, vol. 75, pp. 5831–9, Sept. 2009.

[148] J. Costanza, G. Carapezza, C. Angione, P. Lió, and G. Nicosia, "Robust Design of Microbial Strains.," *Bioinformatics (Oxford, England)*, vol. 28, pp. 3097–3104, Oct. 2012.

[149] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, Apr. 2002.

[150] D. S. Lun, G. Rockwell, N. J. Guido, M. Baym, J. A. Kelner, B. Berger, J. E. Galagan, and G. M. Church, "Large-scale identification of genetic design strategies using local search," *Molecular systems biology*, vol. 5, p. 296, 2009.

[151] D. Egen and D. S. Lun, "Truncated branch and bound achieves efficient constraint-based genetic design.," *Bioinformatics (Oxford, England)*, vol. 28, pp. 1619–23, June 2012.

[152] G. Rockwell, N. J. Guido, and G. M. Church, "Redirector: designing cell factories by reconstructing the metabolic objective.," *PLoS computational biology*, vol. 9, p. e1002882, Jan. 2013.

[153] S. Ohno, H. Shimizu, and C. Furusawa, "FastPros: screening of reaction knockout strategies for metabolic engineering.," *Bioinformatics (Oxford, England)*, vol. 30, pp. 981–987, Nov. 2013.

[154] S. Klamt and E. D. Gilles, "Minimal cut sets in biochemical reaction networks," *Bioinformatics*, vol. 20, pp. 226–234, Jan. 2004.

[155] J. Behre, T. Wilhelm, A. von Kamp, E. Ruppin, and S. Schuster, "Structural robustness of metabolic networks with respect to multiple knockouts.," *Journal of theoretical biology*, vol. 252, pp. 433–41, June 2008.

[156] S. Klamt, "Generalized concept of minimal cut sets in biochemical networks.," *Bio Systems*, vol. 83, pp. 233–47, Jan. 2006.

[157] O. Hädicke and S. Klamt, "Computing complex metabolic intervention strategies using constrained minimal cut sets.," *Metabolic engineering*, vol. 13, pp. 204–13, Mar. 2011.

[158] C. Jungreuthmayer and J. Zanghellini, "Designing optimal cell factories: integer programming couples elementary mode analysis with regulation.," *BMC systems biology*, vol. 6, p. 103, Jan. 2012.

[159] C. T. Trinh, R. Carlson, A. Wlaschin, and F. Srienc, "Design, construction and performance of the most efficient biomass producing E. coli bacterium.," *Metabolic engineering*, vol. 8, pp. 628–38, Nov. 2006.

[160] G. Melzer, M. E. Esfandabadi, E. Franco-Lara, and C. Wittmann, "Flux Design: In silico design of cell factories based on correlation of pathway fluxes to desired properties," *BMC systems biology*, vol. 3, p. 120, 2009.

[161] M. Terzer and J. Stelling, "Large-scale computation of elementary flux modes with bit pattern trees.," *Bioinformatics (Oxford, England)*, vol. 24, pp. 2229–35, Oct. 2008.

[162] O. Hädicke and S. Klamt, "CASOP: a computational approach for strain optimization aiming at high productivity," *Journal of biotechnology*, vol. 147, no. 2, pp. 88–101, 2010.

[163] D. Flowers, R. A. Thompson, D. Birdwell, T. Wang, and C. T. Trinh, "SMET: systematic multiple enzyme targeting - a method to rationally design optimal strains for target chemical overproduction.," *Biotechnology journal*, vol. 8, pp. 605–18, May 2013.

[164] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar, "Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae," *Bioinformatics*, vol. 18, pp. 351–361, Feb. 2002.

[165] M. Imielinski and C. Belta, "Exploiting the pathway structure of metabolism to reveal high-order epistasis.," *BMC systems biology*, vol. 2, p. 40, Jan. 2008.

[166] L. F. de Figueiredo, A. Podhorski, A. Rubio, C. Kaleta, J. E. Beasley, S. Schuster, and F. J. Planes, "Computing the shortest elementary flux modes in genome-scale metabolic networks.," *Bioinformatics (Oxford, England)*, vol. 25, pp. 3158–65, Dec. 2009.

[167] K. Bohl, L. F. de Figueiredo, O. Hädicke, S. Klamt, C. Kost, S. Schuster, and C. Kaleta, "CASOP GS: Computing Intervention Strategies Targeted at Production Improvement in Genome-scale Metabolic Networks.," in *GCB*, pp. 71–80, 2010.

[168] K. Ballerstein, A. von Kamp, S. Klamt, and U.-U. Haus, "Minimal cut sets in a metabolic network are elementary modes in a dual network.," *Bioinformatics (Oxford, England)*, vol. 28, pp. 381–7, Mar. 2012.

[169] A. von Kamp and S. Klamt, "Enumeration of smallest intervention strategies in genome-scale metabolic networks.," *PLoS computational biology*, vol. 10, p. e1003378, Jan. 2014.

[170] I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S. a. Soares, J. P. Pinto, J. Nielsen, K. R. Patil, E. C. Ferreira, and M. Rocha, "OptFlux: an open-source software platform for in silico metabolic engineering," *BMC systems biology*, vol. 4, no. 1, p. 45, 2010.

[171] N. E. Lewis, H. Nagarajan, and B. O. Palsson, "Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods.," *Nature reviews. Microbiology*, vol. 10, pp. 291–305, Apr. 2012.

[172] S. Klamt, J. Saez-Rodriguez, and E. D. Gilles, "Structural and functional analysis of cellular networks with CellNetAnalyzer.," *BMC systems biology*, vol. 1, p. 2, Jan. 2007.

[173] H. Alper, Y.-S. Jin, J. F. Moxley, and G. Stephanopoulos, "Identifying gene targets for the metabolic engineering of lycopene biosynthesis in Escherichia coli.," *Metabolic engineering*, vol. 7, pp. 155–64, May 2005.

[174] H. Alper, K. Miyaoku, and G. Stephanopoulos, "Characterization of lycopene-overproducing E. coli strains in high cell density fermentations.," *Applied microbiology and biotechnology*, vol. 72, pp. 968–74, Oct. 2006.

[175] K. Jantama, M. J. Haupt, S. A. Svoronos, X. Zhang, J. C. Moore, K. T. Shanmugam, and L. O. Ingram, "Combining metabolic engineering and metabolic evolution to develop nonrecombinant strains of Escherichia coli C that produce succinate and malate.," *Biotechnology and bioengineering*, vol. 99, pp. 1140–53, Apr. 2008.

[176] S. H. Hong and S. Y. Lee, "Metabolic flux analysis for succinic acid production by recombinant Escherichia coli with amplified malic enzyme activity.," *Biotechnology and bioengineering*, vol. 74, pp. 89–95, July 2001.

[177] K. H. Lee, J. H. Park, T. Y. Kim, H. U. Kim, and S. Y. Lee, "Systems metabolic engineering of Escherichia coli for L-threonine production.," *Molecular systems biology*, vol. 3, p. 149, Jan. 2007.

[178] J. H. Park, K. H. Lee, T. Y. Kim, and S. Y. Lee, "Metabolic engineering of Escherichia coli for the production of L-valine based on transcriptome analysis and in silico gene knockout simulation.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 7797–802, May 2007.

[179] P. Unrean, C. T. Trinh, and F. Srienc, "Rational design and construction of an efficient E. coli for production of diapolycopendioic acid.," *Metabolic engineering*, vol. 12, pp. 112–22, Mar. 2010.

[180] C. T. Trinh and F. Srienc, "Metabolic engineering of Escherichia coli for efficient conversion of glycerol to ethanol.," *Applied and environmental microbiology*, vol. 75, pp. 6696–705, Nov. 2009.

[181] J. Nogales, S. Gudmundsson, and I. Thiele, "An in silico re-design of the metabolism in Thermotoga maritima for increased biohydrogen production," *International Journal of Hydrogen Energy*, vol. 37, pp. 12205–12218, Sept. 2012.

[182] D.-Y. Lee, L. T. Fan, S. Park, S. Y. Lee, S. Shafie, B. Bertók, and F. Friedler, "Complementary identification of multiple flux distributions and multiple metabolic pathways.," *Metabolic engineering*, vol. 7, pp. 182–200, May 2005.

[183] M. A. Oberhardt, K. Yizhak, and E. Ruppin, "Metabolically remodeling the drug pipeline.," *Current opinion in pharmacology*, vol. 13, pp. 778–85, Oct. 2013.

[184] T. Y. Kim, S. B. Sohn, Y. B. Kim, W. J. Kim, and S. Y. Lee, "Recent advances in reconstruction and applications of genome-scale metabolic models.," *Current opinion in biotechnology*, vol. 23, pp. 617–23, Aug. 2012.

[185] A. M. Feist and B. O. Palsson, "The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli.," *Nature biotechnology*, vol. 26, pp. 659–67, June 2008.

[186] M. Durot, P.-Y. Bourguignon, and V. Schachter, "Genome-scale models of bacterial metabolism: reconstruction and applications.," *FEMS microbiology reviews*, vol. 33, pp. 164–90, Jan. 2009.

[187] S. Ohno, C. Furusawa, and H. Shimizu, "In silico screening of triple reaction knockout Escherichia coli strains for overproduction of useful metabolites.," *Journal of bioscience and bioengineering*, vol. 115, pp. 221–8, Feb. 2013.

[188] S. S. Fong, A. P. Burgard, C. D. Herring, E. M. Knight, F. R. Blattner, C. D. Maranas, and B. O. Palsson, "In silico design and adaptive evolution of Escherichia coli for production of lactic acid.," *Biotechnology and bioengineering*, vol. 91, pp. 643–8, Oct. 2005.

[189] M. Izallalen, R. Mahadevan, A. Burgard, B. Postier, R. Didonato, J. Sun, C. H. Schilling, and D. R. Lovley, "Geobacter sulfurreducens strain engineered for increased rates of respiration.," *Metabolic engineering*, vol. 10, pp. 267–75, Sept. 2008.

[190] H. Yim, R. Haselbeck, W. Niu, C. Pujol-Baxley, A. Burgard, J. Boldt, J. Khandurina, J. D. Trawick, R. E. Osterhout, R. Stephen, J. Estadilla, S. Teisan, H. B. Schreyer, S. Andrae, T. H. Yang, S. Y. Lee, M. J. Burk, and S. Van Dien, "Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol.," *Nature chemical biology*, vol. 7, pp. 445–52, July 2011.

[191] C. Y. Ng, M.-Y. Jung, J. Lee, and M.-K. Oh, "Production of 2,3-butanediol in Saccharomyces cerevisiae by in silico aided metabolic engineering.," *Microbial cell factories*, vol. 11, p. 68, Jan. 2012.

[192] A. P. Burgard, P. Pharkya, and R. E. Osterhout, "Microorganisms for the production of adipic acid and other compounds," 2010.

[193] R. Haselbeck, J. D. Trawick, W. Niu, and A. P. Burgard, "Microorganisms for the production of 1,4-butanediol, 4-hydroxybutanal, 4-hydroxybutyryl-coa, putrescine and related compounds, and methods related thereto," 2011.

[194] S. J. Van Dien, A. P. Burgard, R. Haselbeck, C. J. Pujol-Baxley, W. Niu, J. D. Trawick, H. Yim, M. J. Burk, R. E. Osterhout, and J. Sun, "Microorganisms for the production of 1,4-butanediol and related methods," 2012.

[195] P. Pharkya and M. J. Burk, "Microorganisms and methods for the production of 1,4-cyclohexanedimethanol," 2012.

[196] J. Sun, P. Pharkya, and A. P. Burgard, "Primary alcohol producing organisms," 2011.

[197] M. A. Asadollahi, J. Maury, K. R. Patil, M. Schalk, A. Clark, and J. Nielsen, "Enhancing sesquiterpene production in Saccharomyces cerevisiae through in silico driven metabolic engineering.," *Metabolic engineering*, vol. 11, pp. 328–34, Dec. 2009.

[198] H. S. Choi, S. Y. Lee, T. Y. Kim, and H. M. Woo, "In silico identification of gene amplification targets for improvement of lycopene production.," *Applied and environmental microbiology*, vol. 76, pp. 3097–105, May 2010.

[199] J. A. Chemler, Z. L. Fowler, K. P. McHugh, and M. A. G. Koffas, "Improving NADPH availability for natural product biosynthesis in Escherichia coli by metabolic engineering.," *Metabolic engineering*, vol. 12, pp. 96–104, Mar. 2010.

[200] E. H. Hansen, B. L. Mø ller, G. R. Kock, C. M. Bünner, C. Kristensen, O. R. Jensen, F. T. Okkels, C. E. Olsen, M. S. Motawia, and J. r. Hansen, "De novo biosynthesis of vanillin in fission yeast (Schizosaccharomyces pombe) and baker's yeast (Saccharomyces cerevisiae).," *Applied and environmental microbiology*, vol. 75, pp. 2765–74, May 2009.

[201] A. R. Brochado, C. Matos, B. L. Mø ller, J. r. Hansen, U. H. Mortensen, and K. R. Patil, "Improved vanillin production in baker's yeast through in silico design.," *Microbial cell factories*, vol. 9, p. 84, Jan. 2010.

[202] J. Becker, O. Zelder, S. Häfner, H. Schröder, and C. Wittmann, "From zero to hero–design-based systems metabolic engineering of Corynebacterium glutamicum for L-lysine production.," *Metabolic engineering*, vol. 13, pp. 159–68, Mar. 2011.

[203] T. Hirao, T. Nakano, T. Azuma, M. Sugimoto, and T. Nakanishi, "l-Lysine production in continuous culture of an l-lysine hyperproducing mutant of Corynebacterium glutamicum," *Applied Microbiology and Biotechnology*, vol. 32, Dec. 1989.

[204] S. Li, D. Huang, Y. Li, J. Wen, and X. Jia, "Rational improvement of the engineered isobutanol-producing Bacillus subtilis by elementary mode analysis.," *Microbial cell factories*, vol. 11, p. 101, Jan. 2012.

[205] J. M. Otero, D. Cimini, K. R. Patil, S. G. Poulsen, L. Olsson, and J. Nielsen, "Industrial systems biology of Saccharomyces cerevisiae

enables novel succinic acid cell factory.," *PloS one*, vol. 8, p. e54144, Jan. 2013.

[206] J. URANO and C. A. Dundon, "Cytosolic isobutanol pathway localization for the production of isobutanol," 2012.

[207] A. L. Botes and A. Van Eck Conradie, "Methods of producing 6-carbon chemicals via CoA-dependent carbon chain elongation associated with carbon storage," 2013.

[208] T. Walther, H. CORDIER, C. DRESSAIRE, J. M. Francois, and R. Huet, "Method for the preparation of 2,4-dihydroxybutyrate," 2014.

[209] T. Walther and J. M. Francois, "A microorganism modified for the production of 1,3-propanediol," 2014.

[210] R. Elander, "Industrial production of β-lactam antibiotics," *Applied microbiology and biotechnology*, vol. 61, no. 5-6, pp. 385–392, 2003.

[211] OECD, "Future Prospects for Industrial Biotechnology," tech. rep., 2011.

[212] The White House, "National Bioeconomy Blueprint," tech. rep., The White House, Washington, USA, 2012.

[213] European Comission, "Bio-based economy for Europe: state of play and future potential," tech. rep., European Commission, Brussels, 2011.

[214] C. G. de Oliveira Dal'Molin and L. K. Nielsen, "Plant genome-scale metabolic reconstruction and modelling.," *Current opinion in biotechnology*, vol. 24, pp. 271–7, Apr. 2013.

[215] C. Xu, L. Liu, Z. Zhang, D. Jin, J. Qiu, and M. Chen, "Genome-scale metabolic model in guiding metabolic engineering of microbial improvement.," *Applied microbiology and biotechnology*, vol. 97, pp. 519–39, Jan. 2013.

[216] D. Machado and M. Herrgå rd, "Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism.," *PLoS computational biology*, vol. 10, p. e1003580, Apr. 2014.

[217] J. Monk, J. Nogales, and B. O. Palsson, "Optimizing genome-scale network reconstructions," *Nature Biotechnology*, vol. 32, pp. 447–452, May 2014.

CHAPTER 3

MOSTRAIN: AN EVOLUTIONARY MULTI-OBJECTIVE FRAMEWORK FOR STRAIN OPTIMIZATION

As abundantly described in Chapter 2, the Metabolic Engineering (ME) arena has been providing advances in strain optimization, through the development of methods able to select genetic modifications capable of achieving an improved production of desired products. As shown, these strain optimization tasks can be formulated as bi-level problems, adding, over phenotype prediction, a layer that searches for the best mutant that can be obtained by performing genetic alterations to the wild type. The most studied problem consists in discovering a subset of genes to delete to achieve strains that maximize an objective function (e.g. yield) related with the production of a target compound, while keeping the host viable. Different approaches were proposed, namely OptKnock that uses MILP [1] and meta-heuristics such as Evolutionary Algorithms (EAs) [2] and Simulated Annealing (SA) [3].

Although these approaches have provided good results, they are still limited since they usually return a single solution to the problem or, in some cases, sets of solutions with limited variability.

3.1 Introduction

Among all the previously discussed methods, EAs provide the most diverse solutions, but typically those follow similar strategies to maximize the selected objective function. In order to overcome these limitations, information from multiple criteria is often included in a single objective function. However, these aggregation techniques can introduce undesired biases in the sampling process.

In this context, we here propose the use of multi-objective approaches to provide strains with different trade-offs between the distinct factors involved. For ME applications, it is relevant to find, for example, different trade-offs between the production capability regarding the desired compound, the strain viability (measured by the biomass flux) or the predicted cost of implementing the genetic modifications.

In this work, an approach based on Multi-Objective Evolutionary Algorithms (MOEAs) is thus proposed. Since the mid-1980s, MOEAs have been used to solve multiple criteria problems in distinct scenarios and the nature of the ME problem suggests they can be good candidates here. The MOEA chosen for this task is one of the best performers in the field, the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [4]. Other algorithms, such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5], were tested in preliminary experiments with not so satisfactory results (not shown).

The approaches are validated with three case studies considering the production of succinate, lactate and pyruvate using *E. coli* as the host in aerobic and anaerobic conditions. Two scenarios were tested: in the former, two objectives are set including the maximization of the production of the compound of interest and maximization of biomass production; in

the latter, a third objective is added: the minimization of the number of gene knockouts.

The overall results show that MOEAs were able to find a set of trade-offs between the two or three optimization aims more effectively than single-objective algorithms, discovering more diverse solutions. Also, we show that the quality of the solutions is comparable or superior even when using an aggregation function in single objective EAs. All the proposed methods are implemented in a plug-in for the OptFlux platform [6], an open-source, user-friendly software for ME, and thus can be freely used by the community.

3.2 Methods

3.2.1 Multi-objective optimization

The complexity of many optimization problems is often difficult to represent using a single objective. Therefore, multi-objective (MO) optimization applications have been increasing and the development of suitable algorithms has been an active area of research. In MO problems, there is no longer a single optimal solution, but rather a set of solutions of non-comparable quality. In this context, some definitions need to be put forward:

The classical definition of a multi-objective optimization problem can be formally written in the form: $minimize[f_1(x), f_2(x), \ldots, f_K(x)]$ where $f_i : \Omega \rightarrow \mathcal{R}^K$ are K objective functions. The task is to determine from the set of feasible solutions, the particular subset yielding optimum values for all objective functions [7]. Yet, it is very rare to find a solution that simultaneously optimizes all objective functions. Therefore, we usually seek the optimal set of trade-offs (Pareto Optimal Front), rather than a single solu-

tion. This clearly changes the concept of "optimality" to the one proposed by Pareto [8], working as described in the following definitions.

Definition 1 *Parameter Space and Fitness Space*: The set of all solutions to the optimization problem is known as the Parameter Space or Solution Space. The Feasible Region, Ω, is the set of all admissible solutions to the problem where all the constraints are satisfied. The solution space is projected onto a K-dimensional Fitness Space, by the K objective functions.

Definition 2 *Dominance*: A solution $a \in \Omega$ is said to dominate another solution $b \in \Omega$, when $\forall i : f_i(a) \leq f_i(b)$ and $\exists i : f_i(a) < f_i(b)$, where f_i is the i-th objective (assuming a minimization problem without losing generality). This relation is notated as $a \prec b$. In a less formal manner, a solution a dominates another solution b, if a is not worse than b in any of the objectives and it is at least better in one.

Definition 3 *Pareto Optimality*: A solution $x \in \Omega$ is Pareto Optimal, if $\nexists\ w \in \Omega$ such that $w \prec x$. The set of all the Pareto Optimal solutions is called Pareto Optimal Front (notated as P_{true} in the solution space and as PF_{true} in the fitness space). Since it is usually impossible to find P_{true} or PF_{true}, an approximation is taken. Thus, to evaluate a given solution set S, there is the need to specify an artificial or desired solution set termed Reference Set, notated as R.

3.2.2 Multi-objective and Single-objective Algorithms

Evolutionary Algorithms (EAs) [9] are a popular family of metaheuristic optimization methods, inspired in biological evolution through natural selection. They work by evolving a set of individuals encoding solutions to a given problem. Each individual is evaluated via a fitness function that assigns a numerical value to it, depending on the quality of the solution. New solutions are created using reproduction operators over selected parents taken probabilistically from the current population.

On the other hand, Simulated Annealing (SA) [10] works in a distinct fashion, keeping just one solution. This is used to create new solutions through the application of a mutation operator. Each time a new solution is created, it replaces the original one, if it is better or, in some cases, even when it is worse. The probability of accepting worse solutions depends on the difference between the fitness values and the current temperature, a variable that is reduced as the SA iterates.

In previous work by the authors, both algorithms have been proposed to address strain optimization problems, selecting (near-)optimal sets of genes/ reactions to delete from a model, to overproduce a given compound. Both used the same representation, the set-based representation presented in the next section. The full configuration of these algorithms and the results obtained can be found in [3].

The populational and stochastic nature of EAs enables them to find several possible members of the Pareto-front in a single run. It must be noted, however, that to properly work in multi-objective scenarios, a new class of MOEAs has been proposed which, while keeping the main principles, introduce some changes into the selection and re-insertion steps. Typically,

MOEAs keep sets of solutions that are Pareto optimal. The first MOEAs were developed in [11], soon followed by several approaches developing more specialized EAs. One advantage of using MOEAs over other MO optimization methods lies in the few constraints imposed over the objective function (e.g. they do not depend on the continuity of the Pareto-front). Also, they are relatively easy to implement, very robust and can be easily adapted to parallel computing. A general idea of MOEAs and the recent history of the field can be found in [12].

In this work, the SPEA2 [4] was tested, an evolution of the previous SPEA [13] algorithm. SPEA uses an external archive that contains non-dominated solutions (called the external non-dominated set). At each generation, non-dominated individuals are copied from the population to this external non-dominated set. For each individual in the archive, a strength value, proportional to the number of solutions in the archive it dominates, is computed. The fitness of each individual in the current population is computed according to the strengths of all external individuals that dominate it. This strategy is used to promote the convergence of the algorithm. The fact that the external non-dominated set is used in the selection process brings the problem that, if that set grows too much, the selection pressure might be reduced, thus slowing down the global search process. To prevent this, a clustering technique called "average linkage method" was adopted to prune the external non-dominated set, thus maintaining diversity.

To allow the fairer comparison of the algorithms, some changes were applied to the EA and SA algorithms, regarding the information that is kept during each run, without compromising their procedure nor interfering with the way they search for solutions. The major change was the implementation of an archive that keeps the best solutions found by the EA and SA

during the run. This is not used by the algorithms to select or create new solutions, but allows to build a Pareto-front with the solutions found during their execution. The archive has a maximum size, being managed using the same methods provided in SPEA2, including methods for selecting solutions for removal when the maximum size is reached.

The structure of the 3 algorithms, EA, SA and SPEA2, used in this work, is depicted in Figure 3.1, in an attempt to highlight their major features. Extended details about the implementation can be found in Section 5.4.

3.2.3 Representation scheme and operators

The optimization problem addressed here consists in selecting, from a set of genes of an organism (represented in the respective metabolic model), a subset to be deleted, in order to optimize a set of objective functions.

The encoding of a solution is achieved by a set-based representation [3], where only gene deletions are represented. The representation scheme uses variable-sized solutions and hence sets with distinct cardinalities can compete within the search process.

Two types of reproduction operators were used: crossover and mutation. The first is inspired on *uniform crossover* and works as follows: the genes that are present in both parent solutions are kept in both offspring solutions; the genes that are present in only one of the parent solutions are sent to one of the offspring solutions, selected randomly with equal probabilities (used in EA and SPEA2).

Regarding mutation, three operators were used: *random mutation* replaces a gene in the solution by another gene in the allowed range; *grow mutation* introduces a new gene into the solution, whose value (index) is randomly generated (avoiding duplicates in the set) and *shrink mutation*:

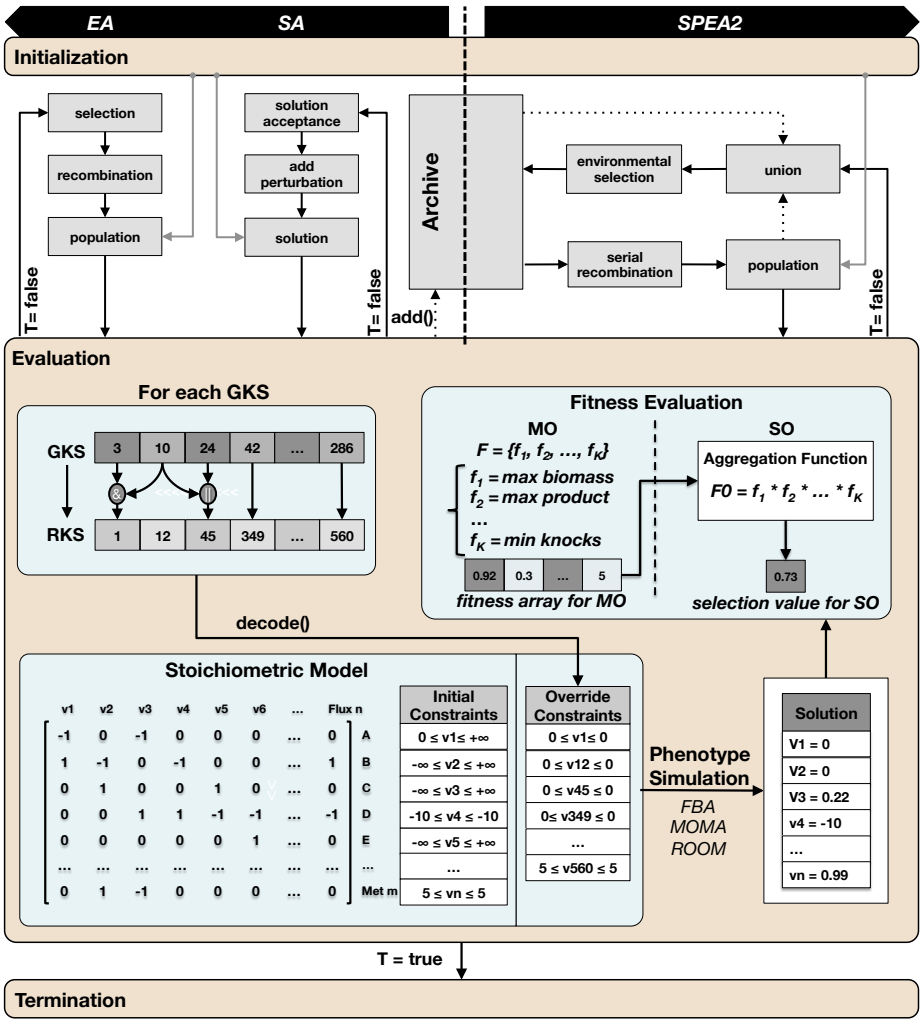


Figure 3.1: Overview of the developed algorithms. The upper region shows the major steps of the three algorithms (See details in 5.4.1, 5.4.3 and 5.4.2). The evaluation gray box illustrates the processes of solution decoding, from Gene Knockout Sets (GKS) to Reaction Knockout Sets (RKS) (upper-left), phenotype simulation showing the added constraints (bottom) and fitness evaluation (details in 5.4.4) for both MO and SO cases (upper-right).

removes a randomly selected gene (these are used in EA, SA and SPEA2).

3.2.4 Solution decoding and evaluation

Each solution is represented as a gene knockout set $GKS_{i,t}$ (i-th solution of the population at generation t) containing a list of indexes for genes

in the model, a subset of the full set of genes that can be deleted (essential genes are not allowed). Transcriptional/ translational information is encoded in the model as a set of Gene-Reaction (GR) associations, represented by Boolean rules, including AND and OR operations. Each GKS is converted through those rules into a reaction knockout set ($RKS_{i,t}$). Each RKS consists of a set of integer values between 1 and N, where N is the number of reactions in the model. When simulating this solution, for each reaction in this set, the flux is forced to be 0, therefore disabling it from the model.

The process proceeds with the phenotype simulation of the mutant using one of the available methods. In the experiments performed within this work, the FBA method was used, but any of the methods available within the OptFlux framework can be used with this MO framework (e.g. MOMA or ROOM).

FBA is a constraint-based approach where feasible flux distributions are mainly defined by three types of constraints (Equation 1b): i) stoichiometric constraints, defining the mass balance equations over internal metabolites assuming steady-state (in the formulation v_j corresponds to the flux of reaction j and S_{ij}, stands for the stoichiometric coefficient of metabolite i in reaction j); ii) thermodynamic or capacity constraints, mainly defining reaction reversibility ($v_{j,min}$ for the lower limit and $v_{j,max}$ for the upper limit of the reaction v_j); and iii) those imposed by the knockouts defined in the respective RKS. FBA uses linear programming to determine the optimal flux distributions using a specified objective function, usually corresponding to the maximization of a flux representing biomass production.

The overall optimization problem addressed in this work can be formulated as:

$$
\begin{array}{ll}
\text{a)} & \left\{ \begin{array}{l} max\ f_i, \forall i \in k \\ subject\ to: \end{array} \right. \\
\text{b)} & \left\{ \begin{array}{l} max\ f_{biomass} \\ subject\ to: \\ \quad i)\ Stoichiometric\ constraints \\ \quad\quad \sum_{j=1}^{N} S_{ij} v_j = 0, \forall i \in \{1, .., M\} \\ \quad ii)\ Thermodynamic\ or\ capacity\ constraints \\ \quad\quad v_{j,min} \leq v_j \leq v_{j,max}, \forall j \in \{1, .., N\} \\ \quad iii)\ Knockout\ constraints \\ \quad\quad v_j = 0, \forall j \in RKS \end{array} \right.
\end{array}
\tag{1}
$$

where N corresponds to the set of reactions and M to the set of metabolites in the model. The layer (a) addresses the objective functions at the bioengineering level to be addressed by the MO algorithm, while the layer (b) depicts the objective function and constraints of the inner phenotype simulation (cellular level). In this bi-level formulation, the strain optimization and the phenotype simulation methods can be chosen independently from all available options (e.g. SPEA2, EA or SA in a) and FBA, pFBA, MOMA, LMOMA, ROOM or MiMBl in b)).

The output of the phenotype simulation is the set of flux values for all reactions in the model. These are used by the optimization algorithms to compute the fitness value of the solution, using an appropriate objective function. One possible objective function is the Biomass-Product Coupled Yield (BPCY) [2], given by $\frac{PB}{S}$, where P stands for the flux of the desired product; B for the biomass flux and S for the substrate uptake flux. Besides optimizing for the production of the desired compound, it also allows to select mutants that exhibit high growth rates, i.e., that are likely to exhibit a higher productivity. An overview of the full framework for solution evaluation is depicted in the respective box of Figure 1.

In this work, when dealing with multi-objective problems, we seek for

solutions optimizing three different objective functions - (OF1) the value of the flux of the compound of interest (P), (OF2) the biomass flux value (B) and (OF3) a function representing the cost of implementing the solution (aims at minimizing the number of knockouts) given by $\frac{1}{s+1}$, where s is the number of gene knockouts (size of the solution).

3.2.5 Case studies and experimental setup

Three case studies were used to test the aforementioned algorithms considering the bacterium *Escherichia coli* in the production of succinate, lactate and pyruvate in both aerobic and anaerobic conditions.

Succinate is one of the key intermediates in cellular metabolism and therefore an important case study for ME [14]. Succinate and its derivatives have been used to synthesize polymers, as additives and flavouring agents in foods, supplements for pharmaceuticals, or surfactants. Currently, succinate is produced through petrochemical processes that can be expensive and have significant environmental impacts. The knockout solutions that lead to an improved production are not straightforward to identify since they often involve a large number of interacting reactions [15].

On the other hand, lactate and its derivatives have been used in a wide range of food processing and industrial applications like meat preservation, cosmetics, oral and health care products and baked goods. Additionally, and because it can be easily converted to readily biodegradable polyesters, it is emerging as a potential material for producing plastics from sugars.

The commercial demand for pyruvate has been increasing due to the variety of fields where it is applicable. It is currently used as a dietary and weight control supplement [16], nutraceutical [17] and antioxidant [18]. Moreover, because it possesses both reactive ketonic and carboxyl groups,

its applications as a starting material are widespread to the chemical, agrochemical and pharmaceutical industries [19–21].

E. coli has many advantageous characteristics as a production host, such as rapid growth and simple nutritional requirements. The iAF1260 *E. coli* genome-scale metabolic model used in this work was developed by [22]. This model includes a total of N=2078 reactions, M=1369 metabolites and 1260 genes. To reduce the algorithm's search space, a pipeline of operations to identify possible knockouts targets was applied. This includes steps to identify fluxes mathematically constrained to zero, as well as equivalent fluxes as described in [3]. Further restrictions to the number of genes available as possible knockout targets were also applied following the pipeline described in [23]. This process resulted in a list of 250 and 223 variables (genes) to be considered by the optimization algorithms, for the aerobic and anaerobic case studies, respectively. For all the case studies, the glucose uptake flux was limited to a maximum of 20 $mmol.gDW^{-1}.h^{-1}$ and the oxygen uptake flux, in the aerobic case studies, was limited to a maximum of 20 $mmol.gDW^{-1}.h^{-1}$.

For all algorithms, the termination criterion for each run was defined based on a maximum of 50000 solution evaluations. The initial solutions are sets with randomly generated elements. The size of the individuals is randomly created in the range [1,20]. For each experimental setup, the process was repeated for 30 runs. The archives' maximum size was set to 100 solutions. A typical run of these algorithms, using this model after the pre-processing stages is executed in approximately 30 minutes in common `Intel` i7 CPU.

In the implementation of the phenotype prediction methods in this work, IBM ILOG CPLEX was used to run the simplex algorithm for linear pro-

gramming.

3.2.6 Performance metrics

Although the comparison between Pareto fronts is not a straightforward task, some metrics have been developed enabling to better evaluate the quality of the solutions and compare the performance of the algorithms. The definition of quality in MO optimization involves a number of optimization goals: the distance of the resulting set to the reference set should be minimized; a good distribution of the solutions along the front should be obtained; and, the spread of the front should be maximized, i.e., for each objective, a wide range of values should be covered. In this work, three Performance Indexes (PIs) were used. In the following definitions, PF_{known} refers to the experimental Pareto-front being tested in each instant and R to the reference set that is used to approximate the true optimal Pareto front (calculated in this work as explained in the next section).

- **Generational Distance (GD)** [24]: This metric calculates how far, on average, the experimental pareto front (PF_{known}) is from R. It is mathematically defined by:

$$GD \triangleq \frac{(\sum_{i=1}^{n} d_i^K)^{\frac{1}{K}}}{n} \tag{3.2}$$

  where n is the number of solutions in PF_{known}, K is the number of objectives and d_i is the distance in the fitness space (Euclidean Distance when K=2) between each solution and the nearest member in the reference set R.

- **Hypervolume (HV)** [13]: This indicator is defined as the area of

coverage of PF_{known} with respect to the objective space (in a bi-objective problem). Its mathematical definition can be interpreted as the sum of all the rectangular areas (volumes, hypervolumes,...) bounded by the solutions and ($f_1(x)$,$f_2(x)$,...,$f_K(x)$) for a K-objectives problem:

$$HV \triangleq \left\{ \bigcup_i vol_i | sol_i \in PF_{known} \right\} \tag{3.3}$$

where vol_i is the calculated volume (area, hypervolume, ...) correspondent to each solution sol_i contained in PF_{known}.

- **C-Measure (C)** [25]: It is based on the concept of solution dominance. Given two PFs (PF1, PF2) the measure C(PF1, PF2) returns the fraction of solutions in PF2 that are dominated by at least one solution in PF1. Values near 1 clearly favor the method that generated PF1; values near 0 show that few solutions in PF2 are dominated by solutions in PF1.

3.2.7 Generation of reference sets

One difficulty that arises in the comparison of multi-objective algorithms is the fact that the optimal Pareto set is not known *a priori* and is typically impossible to calculate. Thus, the alternative is to build a reference set (R) that can work as an approximation.

The strategy followed was to synthetically generate these reference sets by defining maximum theoretical values. Note that a large part of this space can be unreachable in practice. The process of reference set generation followed these steps:

- Compute the biomass flux value for the wild-type (via FBA simulation

without further constraints);

- Constrain the minimum biomass allowed to a percentage of the previously computed wild-type biomass. These percentages were varied in the range $[0, 100]\%$ in 0.05% steps. For each, an FBA maximizing the respective product was performed. This allowed to generate a reference set with 2001 points for each of the bi-objective case studies;

- To generate the reference set for the tri-objective case studies, and since an approach similar to the previous one would not be reasonable, we simply extended the previous reference set along the third objective axis (repeating R for each of the number of knockouts in the range $[0, 20]$). This resulted in reference sets composed of 42021 points for each of the tri-objective case studies.

Furthermore, aggregated PFs for each algorithm in each case study were also created. This was accomplished by assembling the PFs for each algorithm in each case study, encompassing all non-dominated solutions reached by that algorithm in all the runs. This process implies taking the final archives from each run, simplifying its solutions (removing genes not contributing to the objective functions), creating its union (which removes repetitions) and removing all dominated solutions from this set. An extra step was also used to discard all non-robust solutions from these sets. A solution is considered robust, if, by enforcing the biomass to be at least 99% that of the solution (via an additional constraint) followed by an FBA minimizing the target product, the production of that compound is still predicted.

3.3 Results and Discussion

3.3.1 Results for product and biomass optimization

The first set of experiments considered the simplest scenario: two objective functions, OF1 and OF2, as defined above. Thus, the objective was to find trade-offs between the capacity of the strain to produce the target compound and its predicted viability. SPEA2 will be compared with the single objective EA and SA [3], including the adaptations described above. In both, a single objective function has to be employed, an aggregation function combining both objectives. The option was to consider the product of OF1 and OF2 (equivalent to the BPCY defined above when the substrate uptake is fixed).

The first analysis concerns the size of the PFs generated by each algorithm, for each case study. The sizes of the synthetically generated reference sets, as well as aggregated PFs for each algorithm can be found in Table 3.1.

Table 3.1: Size of the reference sets (synthetic) and the aggregated Pareto fronts (for each algorithm) in all the bi-objective case studies.

		Reference	**SPEA2**	**EA**	**SA**
Aerobic	*Succinate*	2001	60	5	58
	Lactate	2001	910	181	265
	Pyruvate	2001	746	360	542
Anaerobic	*Succinate*	2001	162	95	94
	Lactate	2001	924	119	205
	Pyruvate	2001	326	68	128

This provides a first result, showing that SPEA2 fronts are larger and thus this algorithm is able to obtain more non-dominated solutions. However, this does not guarantee their diversity. The visualization of the PFs

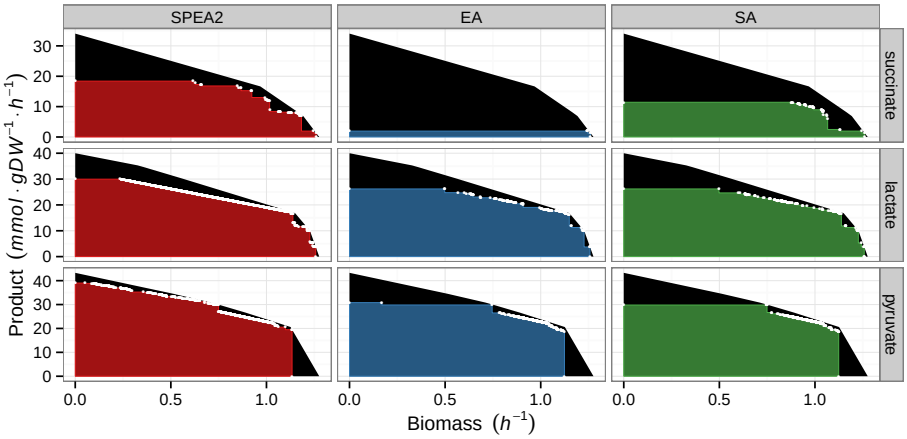
for each algorithm provides an important analysis tool. These are displayed in Figures 3.2 that show the ability of SPEA2 to find a good set of non-dominated solutions that are near the reference set, and well distributed along the Pareto front.

These visual conclusions are further reinforced looking at the results from the application of the defined metrics to the results (compiled in Tables 3.2 and 3.3). In this analysis, note that for GD smaller values are better, while for HV, higher values are preferable. For the C-measures, lower values in the columns (dominated) and higher values in the rows (dominating) are preferable. For all the case studies the advantage of SPEA2 is clearer, obtaining the best results in nearly all the PIs (Table 3.2) and also having significantly lower counts of dominated solutions (Table 3.3). The only exceptions to this trend occur for the aerobic production of pyruvate case study with the SA algorithm and the aerobic production of succinate with the EA algorithm. However, since the GD metric calculates the average of the distance of the solutions to the reference, if a given case study has few solutions but those solutions are good, the metric can be deceptive.

Table 3.2: Performance Indexes for the bi-objective case-studies. For each case, the best performance is shown in bold.

		Aerobic		**Anaerobic**	
		GD	*HV*	*GD*	*HV*
Succinate	*SPEA2*	0.00690	**0.43994**	**0.00495**	**0.50041**
	EA	**0.00237**	0.05625	0.01111	0.38252
	SA	0.01296	0.28224	0.00565	0.47497
Lactate	*SPEA2*	**0.00269**	**0.57577**	**0.00006**	**0.67455**
	EA	0.00375	0.54422	0.00016	0.65317
	SA	0.00336	0.54524	0.00010	0.65331
Pyruvate	*SPEA2*	0.00130	**0.63765**	**0.00672**	**0.44947**
	EA	0.00150	0.56707	0.01405	0.44044
	SA	**0.00111**	0.56397	0.00675	0.44903

(a) Aerobic



(b) Anaerobic

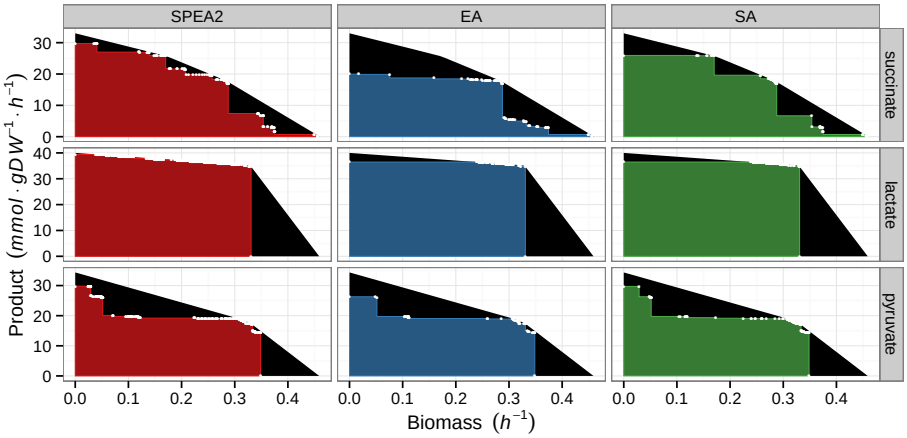


Figure 3.2: Coverage plots for the bi-objective case studies in aerobic (a) and anaerobic (b) conditions, showing the PF of each method overlaying the global reference front R (in black). Areas not covered (black) show regions where the method did not reach optimal solutions (as compared to R). White dots represent solutions.

An alternative view is provided by Figure 3.3, where all methods are evaluated measuring their capacity to generate good solutions both in the

Table 3.3: C-Measures for the bi-objective case studies. For each case, the best performance is shown in bold.

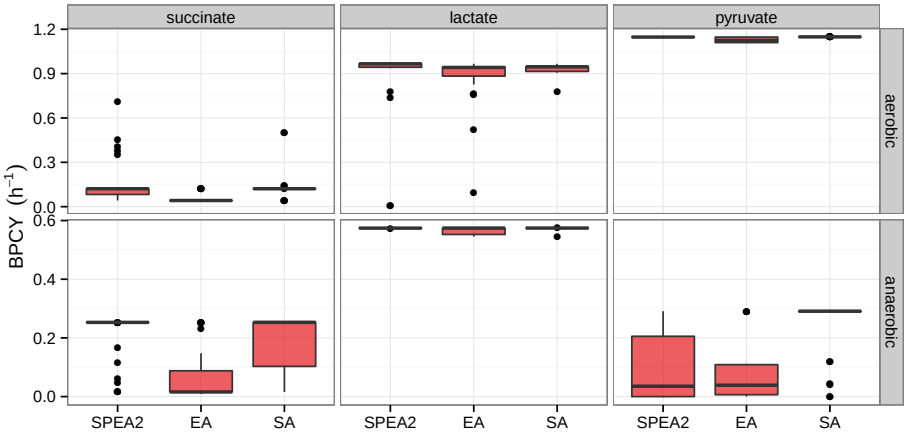
		Aerobic			**Anaerobic**		
		SPEA2	*EA*	*SA*	*SPEA2*	*EA*	*SA*
Succinate	*SPEA2*		40.00%	74.14%		92.63%	39.36%
	EA	1.67%		**0.00%**	**3.70%**		5.32%
	SA	**0.00%**	40.00%		**20.99%**	89.47%	
Lactate	*SPEA2*		49.72%	34.34%		19.33%	20.98%
	EA	**3.30%**		15.09%	**3.35%**		22.93%
	SA	**3.96%**	53.04%		**5.19%**	44.54%	
Pyruvate	*SPEA2*		30.28%	20.30%		54.41%	25.78%
	EA	**7.91%**		18.45%	3.99%		**3.91%**
	SA	**8.58%**	36.11%		**12.58%**	58.82%	

perspective of an aggregation function, in this case the BPCY, and the substrate to product carbon yield (CYIELD). For each run of each algorithm, the solution in the final archive that maximizes the BPCY and the solution that maximizes the CYIELD are taken. Boxplots showing these values for each algorithm and case study are presented. The results show that the SPEA2 is able to perform at a comparable, if not better, level in this regard. This shows its ability to find good overall solutions in the perspective of a single objective evaluation function, even if it is not directly maximizing it, as it is the case with EA/SA.

Another way of measuring the performance of the various algorithms is presented in Figure 3.4, depicting the best solutions (highest BPCY and highest CYIELD respectively - y-axis) found by the algorithms, given a maximum number of allowed gene deletions (x-axis).

Despite not being configured to optimize neither of the objective functions, the SPEA2 results are on par with the ones reached by the EA and SA algorithms for the BPCY objective function (superior in the aerobic production of succinate). For the CYIELD objective function, the SPEA2

(a) BPCY



(b) CYIELD

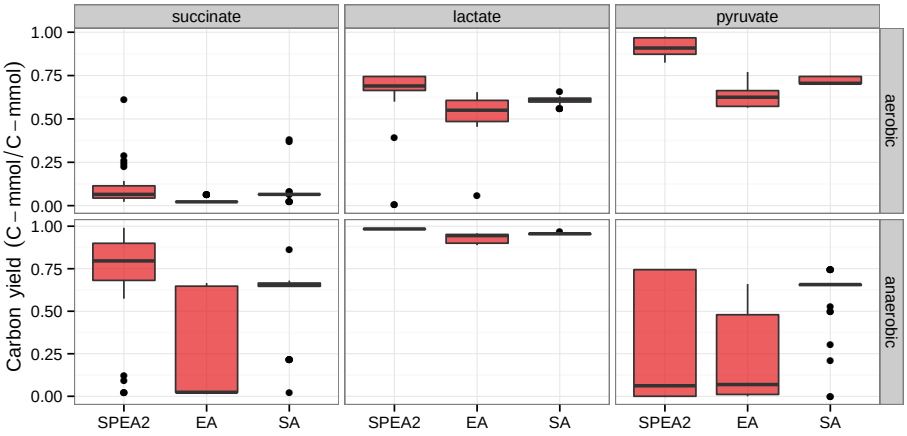
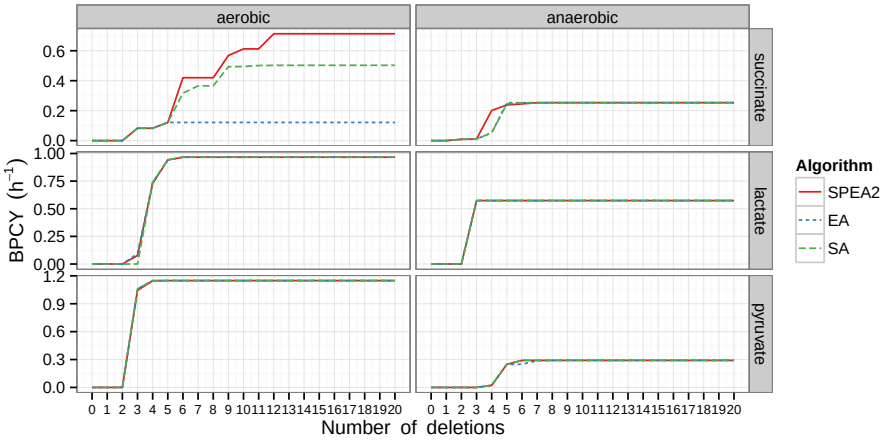


Figure 3.3: Boxplots for the bi-objective case studies, representing both (a) the best BPCY (mmol product . mmol substrate^{-1} . h^{-1}) and (b) the best CYIELD (C − mmol product . C − mmol substrate^{-1}) obtained in each run.

results are visibly superior. It is interesting to note that overall BPCY best results are achieved by all algorithms, except for the succinate case study, even though only EA and SA specifically search for the best BPCY. When looking at the CYIELD result, for which none of the approaches directly

(a) BPCY



(b) CYIELD

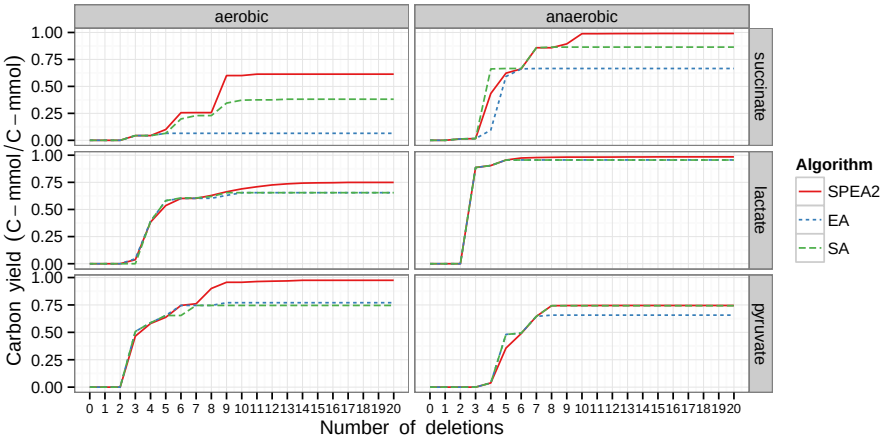


Figure 3.4: Monotonic plots for the bi-objective case studies, showing (a) the best BPCY (mmol product . mmol substrate^{-1} . h^{-1}) and (b) the best CYIELD (C − mmol product . C − mmol substrate^{-1}) obtained by each algorithm, as a function of the number of gene deletions.

optimizes for, only SPEA2 seem to be able to find the best overall solutions in the 3 case studies, meaning that SPEA2 results are sufficiently diverse to be able to include good solutions independently on the evaluation criteria.

It is worth remembering that all the solutions (for all the algorithms) have been previously simplified, thus removing all the genes not contributing directly to the original objective functions (OF1 and OF2), reducing their effective sizes.

3.3.2 Minimizing the number of knock-outs

Minimizing the number of gene knock-outs is an important objective when considering strain optimization approaches, given that the practical implementation of a gene deletion is both time and economically demanding. Also, wet-lab implementations of mutant strains with many knock-outs may be very difficult tasks. So, we took this aim as the third optimization objective, adding OF3 to the list of functions to maximize. The three algorithms were again executed, considering only the aerobic case studies. The aggregation function used in the case of EA and SA (single objective algorithms) is still the product of the three values (in this case OF1, OF2 and OF3). The size of the reference sets and aggregated PF is presented in Table 3.4.

Table 3.4: Size of the reference sets (synthetic) and the aggregated Pareto fronts (for each algorithm) in the tri-objective case studies.

		Reference	**SPEA2**	**EA**	**SA**
Aerobic	*Succinate*	42021	79	9	9
	Lactate	42021	908	60	153
	Pyruvate	42021	1231	294	244

Tables 3.5 and 3.6 show the results for the calculation of the same set of metrics used in the previous case. It is clear that, in this case, the advantage of the SPEA2 is even more pronounced than before. In this case, the only exception happens in the succinate case study, where both the EA and SA

algorithms have a better GD value, which is easily explained by the very low number of solutions that they found.

Table 3.5: Performance Indexes results for the aerobic, tri-objective studies.

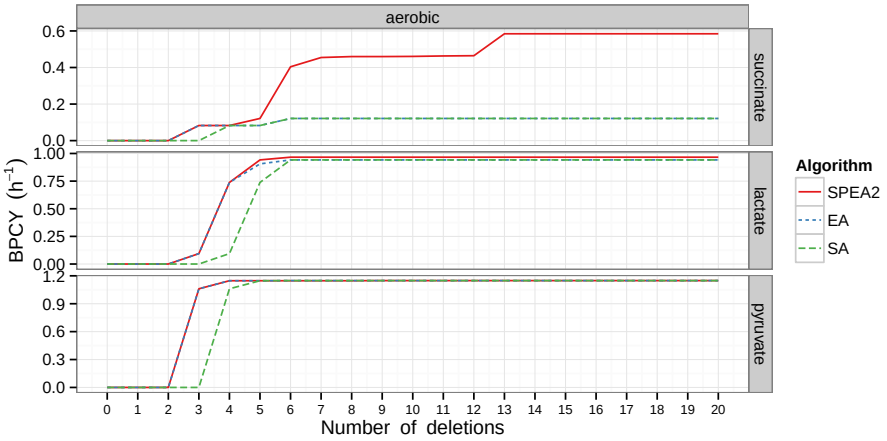
		Aerobic	
		GD	*HV*
Succinate	*SPEA2*	0.01238	**0.03072**
	EA	**0.00066**	0.00989
	SA	0.00114	0.00790
Lactate	*SPEA2*	**0.00261**	**0.07852**
	EA	0.01238	0.07187
	SA	0.00610	0.05772
Pyruvate	*SPEA2*	**0.00101**	**0.11064**
	EA	0.00152	0.09928
	SA	0.00168	0.07734

Table 3.6: C-Measures for the aerobic, tri-objective case studies.

		Aerobic		
		SPEA2	*EA*	*SA*
Succinate	*SPEA2*		33.33%	44.44%
	EA	**0.00%**		0.00%
	SA	**3.80%**	22.22%	
Lactate	*SPEA2*		41.67%	26.14%
	EA	**2.09%**		18.30%
	SA	**2.64%**	11.67%	
Pyruvate	*SPEA2*		27.89%	38.52%
	EA	**5.77%**		46.31%
	SA	**3.41%**	14.97%	

The monotonic plots for the tri-objective case studies are presented in Figure 3.5. Here, the quality of the SPEA2 results can further be confirmed, since it is the algorithm with the best overall behavior, able to obtain the solutions with both the highest BPCY and highest CYIELD, for each number of gene deletions.

(a) BPCY



(b) CYIELD

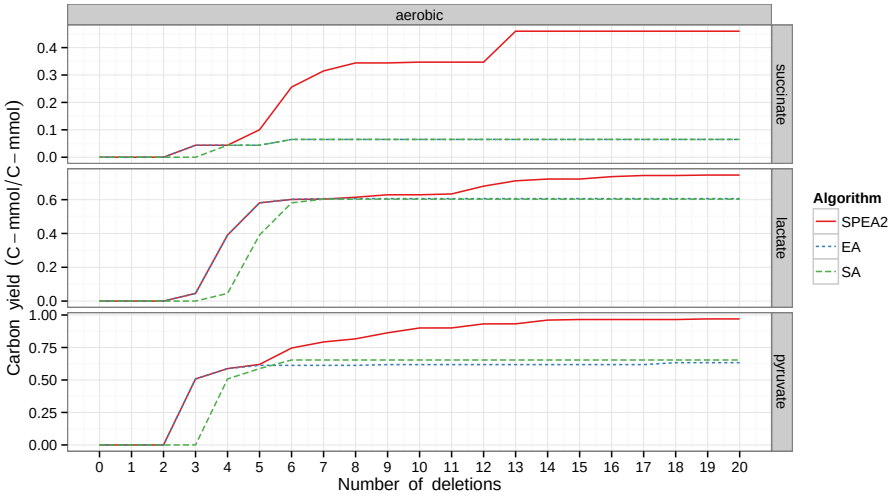


Figure 3.5: Monotonic plots for the tri-objective case studies, showing (a) the best BPCY (mmol product . mmol substrate^{-1} . h^{-1}) and (b) the best CYIELD (C − mmol product . C − mmol substrate^{-1}) obtained by each algorithm, as a function of the number of gene deletions.

Visualizing the non-dominated fronts when working with more than 2 objectives becomes a harder task. To try to address this issue, in Figures 3.6

(succinate production), 3.7 (lactate production) and 3.8 (pyruvate production), 3D plots of the PFs are displayed showing only comparisons of pairs of algorithms to foster interpretability. For each case study, we show the comparison of SPEA2 with the other algorithms. For each pair, both the 3D plot and the projection to the biomass and product (either succinate or lactate) axes are provided. One conclusion that arises almost immediately is the higher number of non-dominated solutions available for the SPEA2 algorithm.

From the projections, one can also assert that the coverage of the solution space for biomass and product is, once again, superior in the SPEA2 case. Duly note that the extreme solutions are always covered by the SPEA2 algorithm even though an extra objective has been added. From the 3D plots it is also possible to observe that SPEA2 is capable of reaching similar solutions in terms of biomass and product fluxes, with a smaller number of knock-outs, when compared to the SO alternatives. These conclusions are reinforced by the solutions displayed in Table 3.7 that shows the best overall and 6 knock-out solutions in terms of carbon yield.

In the table, three solutions are presented in more detail ($s1$, $s2$ and $s3$) corresponding to the best solutions found by SPEA2 for a maximum of 6 knock-outs. We can easily observe that the fitnesses of these solutions are either equal or better than the best overall solutions found by the other algorithms. Moreover, the best overall fitnesses for the SPEA2 algorithm are always better than the ones obtained by the other algorithms, even if it requires more deletions, demonstrating the superior sampling prowess of the MO approach. Another interesting observation is that all the genes that compose solution $s1$ (succinate) are among the top 10 genes for the same problem presented in Table 3.8.

It is interesting to observe that, from the top 10 genes present in solutions that reach at least 75% of the maximum BPCY, a very small subset is kept when we added the third objective function. This suggests that the core mechanisms underlying the top solutions has shifted significantly.

The full analysis of all (or a significant part) of the solutions obtained is impossible in this scope (even because of its diversity). Thus, we pro-

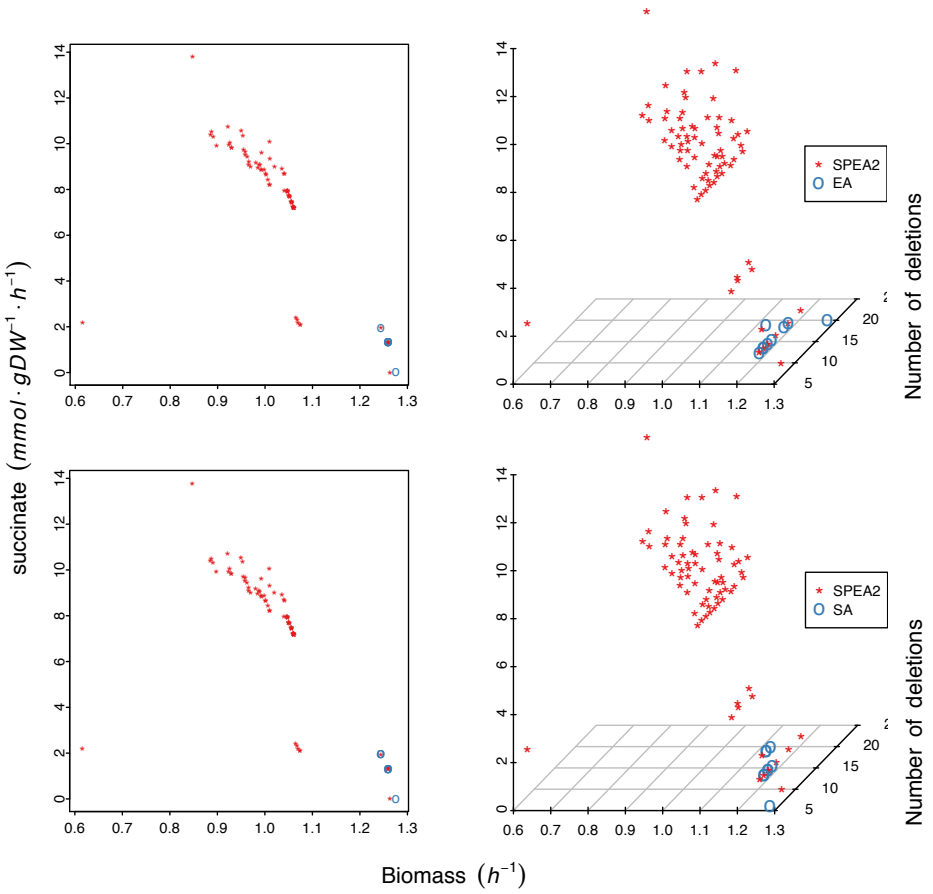


Figure 3.6: 3D scatter plots for the aerobic succinate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident.

vide in supplementary material a list of all non-dominated unique solutions obtained by each algorithm in both studies and considering the 2 and 3 objective scenarios. This list is given in a spreadsheet, including for each solution the number and list of gene deletions, as well as the product and biomass flux values obtained in the FBA simulation (BPCY and some other

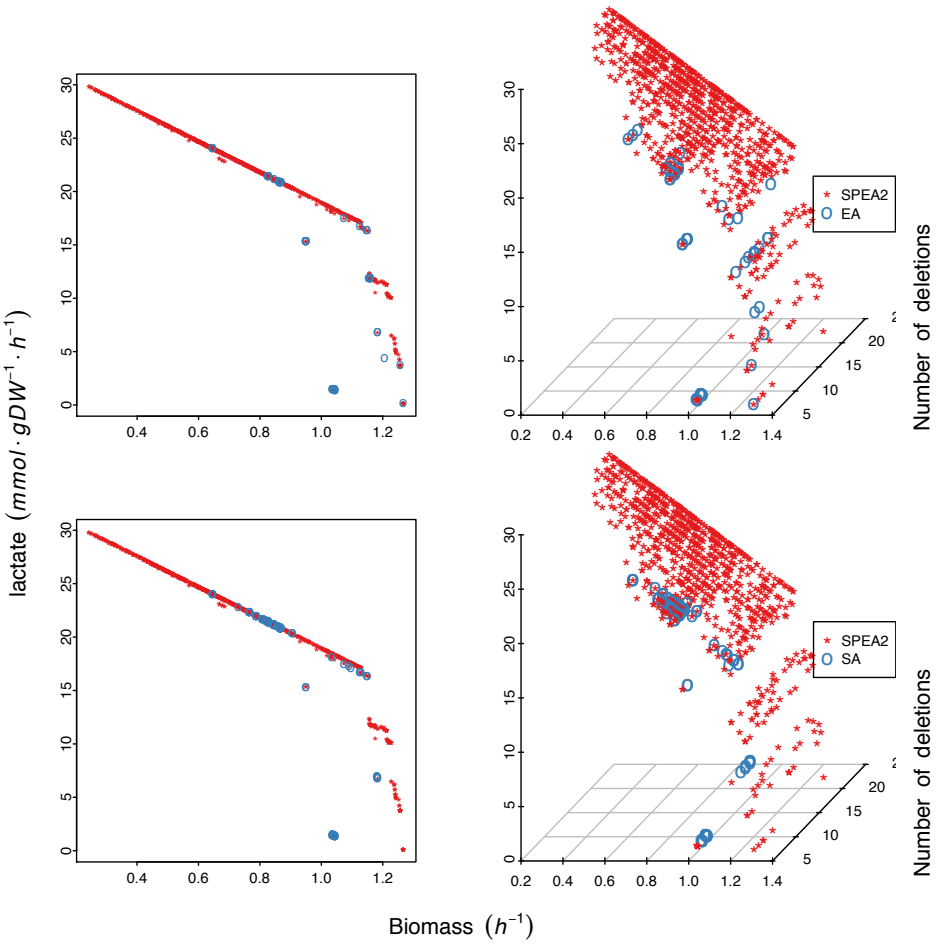


Figure 3.7: 3D scatter plots for the aerobic lactate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident.

auxiliary objective functions' values are also given). Taking advantage of the sorting features of the spreadsheet it is possible to quickly find solutions using desired criteria.

While trying to understand the biological validity of the solutions proposed by the algorithms, we have analysed knockout solutions provided in literature that were implemented *in vivo*, which have been achieved with-

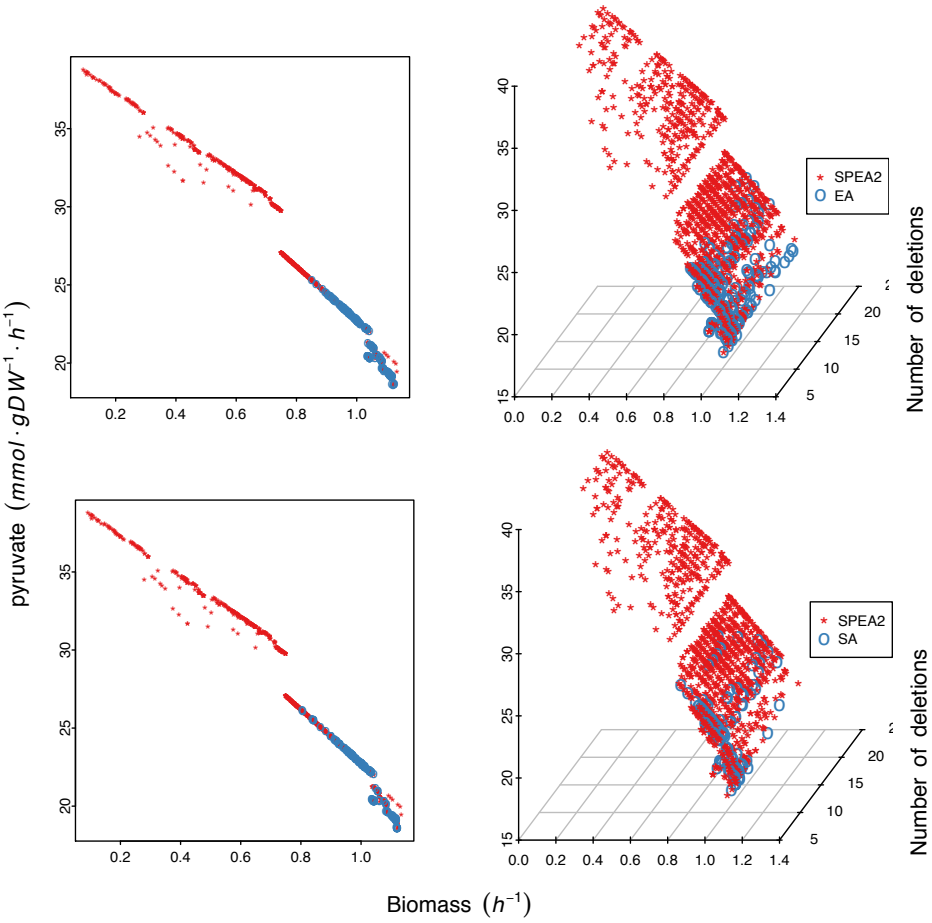


Figure 3.8: 3D scatter plots for the aerobic pyruvate production case study. In each, SPEA2 solutions (red asterisks) are plotted against either EA (top) or SA (bottom) solutions (blue circles). The projections for each problem are always presented on the left side of the full Pareto-front. From these projections, the wider distribution and spread of SPEA2 fronts becomes evident.

Table 3.7: Best CYIELDs for the tri-objective case studies. Fitnesses considering a maximum of 6 knock-outs are also displayed.

		Best (K)	**Best 6K**
Succinate	*SPEA2*	**0.45988 (13)**	**0.25596** $^{(s1)}$
	EA	0.06494 (6)	0.06494
	SA	0.06494 (6)	0.06494
Lactate	*SPEA2*	**0.74562 (20)**	**0.60148** $^{(s2)}$
	EA	0.60616 (7)	0.60148
	SA	0.60298 (7)	0.58122
Pyruvate	*SPEA2*	**0.96989 (19)**	**0.74569** $^{(s3)}$
	EA	0.63323 (18)	0.61239
	SA	0.65368 (6)	0.65368

$^{(s1)}$ *mae*B - *pnt*B - *sdh*A - *ald*B - *gnd* - *edd*
$^{(s2)}$ *gld*A - *sdh*D - *ppc* - *pfl*A - *lpd* - *pfl*C
$^{(s3)}$ *sdh*C - *pox*B - *pfl*A - *lpd* - *pfl*D - *gnd*

Table 3.8: Most frequently selected genes (in () the percentage of solutions where they appear). List with the 10 most frequently selected genes for deletion for the SPEA2 algorithm in the succinate case study, both for the 2 and 3 objectives approaches, when considering solutions able to reach at least 75% of the top BPCY for that case study. The full list for all algorithms and case studies can be accessed in the supplementary material.

	Aerobic	
	2 objectives	*3 objectives*
Succinate	pntA (100.00%)	pntB (100.00%)
	*mae*B (100.00%)	*gnd* (100.00%)
	*pyk*A (100.00%)	*ald*B (100.00%)
	*pyk*F (100.00%)	*sdh*A (91.89%)
	zwf (100.00%)	*mae*B (91.89%)
	*mae*A (96.30%)	*eda* (62.16%)
	eda (96.30%)	*ser*A (59.46%)
	*fsa*B (74.07%)	*suc*C (45.95%)
	*fsa*A (74.07%)	*edd* (43.24%)
	*gcv*H (48.15%)	*scp*C (40.54%)

out the help of this type of algorithms, rather by the knowledge of experts. The first problem we had to deal with is related with the few number of solutions that have been implemented *in vivo* in any of the case studies. Nevertheless, we have chosen the succinate case study to analyse, since

it is the most difficult one for all the algorithms. The results from this analysis can be found in the supplementary material (Excel spreadsheet *chapter3_literature_mutants.xlsx*).

We also realized that often the conditions of *in vivo* studies were not the ones chosen in our optimization (e.g. many of the studies available are in anaerobiosis), and also that several of the targets implemented *in vivo* are not accounted for in our model (e.g. regulatory proteins) or are considered essential in the model due to incompleteness of information (*icd* or *pts*G). These factors, allied to the fact that, as observed *in vivo*, there seems to be a great diversity of genetic mutations that originate high performing mutants (as shown in the supplementary material), explains why there is not a perfect match between the best solutions we found and the ones implemented so far. Nevertheless, it is still possible to conclude that many of the common *in vivo* targets were also found by our algorithm in several solutions. Examples of those genes are the obvious *sdh* (codifying for succinate dehydrogenase), *pta* (phosphate acetyltransferase) or *pox*B (pyruvate oxidase) that, besides interrupting the TCA cycle (*sdh*), avoid the formation of the main competing carbon sink (acetate, produced either by *pox*B or *pta*/*ack*A).

On the other hand, the variety of solutions found by our algorithms is much broader than what can be found implemented *in vivo*, which is natural given the disproportion between the number of *in vivo* and *in silico* solutions. An interesting outcome is the realization that these algorithms provide a great diversity of solutions expected to have different compromises between different objectives that have not yet been exploited commercially. Examples of such non-intuitive targets are related with the replacement of the common serine and glycine biosynthesis (knockout of genes *gly*A - $\sim$

20% of the solutions - and *ser*A/*ser*B - $\sim$ 60% of the solutions) by synthesis via threonine, promoting a higher flux through the glycolysis and the TCA cycle. Although this approach still needs validation, similar approaches involving the aminoacids metabolism have been highlighted for *S. cerevisiae* [2].

3.4 Conclusions

In the Metabolic Engineering arena, the possibility of selecting the appropriate genetic modifications to a given microorganism offers a lot of promises. However, either due to lack of information or incorrect assumptions, the uncertainty of the results provided by the current approaches renders some of the solutions unfeasible. Here, we propose the MOStrain framework that provides a very attractive approach for strain optimization tasks, given its ability to find high quality solutions, as well as providing in a single run a set of alternatives that can be explored by the ME experts.

Future improvements, currently in development, include the ability to use local optimizers over a selected region of the Pareto-front computed by the framework, as well as the inclusion of over/under expression of genes as optimization targets [26]. Another extension will allow to account for both metabolic and regulatory information during the optimization process [27] Also, user-friendly tools are being developed to foster the visualization and interpretation of the solutions before their wet-lab implementation, aiming at the understanding of the strategies found by mutant strains to achieve the overproduction of the compounds of interest.

REFERENCES

[1] A. Burgard, P. Pharkya, and C. Maranas, "Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization," *Biotechnology and bioengineering*, vol. 84, no. 6, pp. 647–657, 2003.

[2] K. Patil, I. Rocha, J. Förster, and J. Nielsen, "Evolutionary programming as a platform for in silico metabolic engineering," *BMC bioinformatics*, vol. 6, no. 1, p. 308, 2005.

[3] M. Rocha, P. Maia, R. Mendes, J. Pinto, E. Ferreira, J. Nielsen, K. Patil, and I. Rocha, "Natural computation meta-heuristics for the in silico optimization of microbial strains," *BMC bioinformatics*, vol. 9, no. 1, p. 499, 2008.

[4] E. Zitzler, M. Laumanns, L. Thiele, *et al.*, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *EUROGEN*, pp. 95–100, 2001.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[6] I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S. Soares, J. Pinto, J. Nielsen, K. Patil, E. Ferreira, and M. Rocha, "Optflux: an open-source software platform for in silico metabolic engineering," *BMC systems biology*, vol. 4, no. 1, p. 45, 2010.

[7] C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information systems*, vol. 1, no. 3, pp. 129–156, 1999.

[8] V. Pareto, "Cours D'Economie Politique, volume I and II," *F. Rouge, Lausanne*, vol. 250, 1896.

[9] Z. Michalewicz, *Genetic algorithms+ data structures*. Springer, 1996.

[10] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[11] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st international Conference on Genetic Algorithms*, pp. 93–100, L. Erlbaum Associates Inc., 1985.

[12] K. Deb, *Multi-objective optimization using evolutionary algorithms*, vol. 16. Wiley, 2001.

[13] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, 1999.

[14] S. Lee, S. Hong, and S. Moon, "In silico metabolic pathway analysis and design: succinic acid production by metabolically engineered *Escherichia coli* as an example," *Genome informatics series*, pp. 214–223, 2002.

[15] H. Lin, G. Bennett, and K. San, "Genetic reconstruction of the aerobic central metabolism in *Escherichia coli* for the absolute aerobic production of succinate," *Biotechnology and bioengineering*, vol. 89, no. 2, pp. 148–156, 2005.

[16] P. K. Koh-Banerjee, M. P. Ferreira, M. Greenwood, R. G. Bowden, P. N. Cowan, A. Almada, and R. B. Kreider, "Effects of calcium pyruvate supplementation during training on body composition, exercise capacity, and metabolic responses to exercise," *Nutrition*, vol. 21, no. 3, pp. 312–319, 2005.

[17] M. McCarty, "Toward a wholly nutritional therapy for type 2 diabetes," *Medical hypotheses*, vol. 54, no. 3, pp. 483–487, 2000.

[18] X. Wang, E. Perez, R. Liu, L.-J. Yan, R. T. Mallet, and S.-H. Yang, "Pyruvate protects mitochondria from oxidative stress in human neuroblastoma sk-n-sh cells," *Brain research*, vol. 1132, pp. 1–9, 2007.

[19] T. Lütke-Eversloh, C. N. S. Santos, and G. Stephanopoulos, "Perspectives of biotechnological production of l-tyrosine and its applications," *Applied microbiology and biotechnology*, vol. 77, no. 4, pp. 751–762, 2007.

[20] Y.-C. Lee, H.-C. R. Chien, and W.-H. Hsu, "Production of¡ i¿ n¡/i¿-acetyl-d-neuraminic acid by recombinant whole cells expressing¡ i¿ anabaena¡/i¿ sp. ch1¡ i¿ n¡/i¿-acetyl-d-glucosamine 2-epimerase and¡ i¿ escherichia coli n¡/i¿-acetyl-d-neuraminic acid lyase," *Journal of biotechnology*, vol. 129, no. 3, pp. 453–460, 2007.

[21] C. Gunawan, G. Satianegara, A. K. Chen, M. Breuer, B. Hauer, P. L. Rogers, and B. Rosche, "Yeast pyruvate decarboxylases: variation in biocatalytic characteristics for (r)-phenylacetylcarbinol production," *FEMS yeast research*, vol. 7, no. 1, pp. 33–39, 2007.

[22] A. Feist, C. Henry, J. Reed, M. Krummenacker, A. Joyce, P. Karp, L. Broadbelt, V. Hatzimanikatis, and B. Palsson, "A genome-scale metabolic reconstruction for *Escherichia coli* k-12 mg1655 that accounts for 1260 orfs and thermodynamic information," *Molecular systems biology*, vol. 3, no. 1, 2007.

[23] A. Feist, D. Zielinski, J. Orth, J. Schellenberger, M. Herrgard, and B. Palsson, "Model-driven evaluation of the production potential for growth-coupled products of *Escherichia coli*," *Metabolic engineering*, vol. 12, no. 3, pp. 173–186, 2010.

[24] D. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," tech. rep., DTIC Document, 1999.

[25] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[26] E. Gonçalves, R. Pereira, I. Rocha, and M. Rocha, "Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression," *Journal of Computational Biology*, vol. 19, no. 2, pp. 102–114, 2012.

[27] P. Vilaça, I. Rocha, and M. Rocha, "A computational tool for the simulation and optimization of microbial strains accounting integrated metabolic/regulatory information," *Biosystems*, vol. 103, no. 3, pp. 435 – 441, 2011.

CHAPTER 4

IDENTIFICATION OF ROBUST STRAIN DESIGNS VIA MULTIPLE PHENOTYPE PREDICTION

As it has been described in Chapter 2, the past two decades have witnessed great advances in the computational modeling and systems biology fields. Soon after the first models of metabolism were developed, several methods for the prediction of phenotypes were also put forward. With the ever-growing information provided by such methods, new questions arose. Metabolic Engineering in particular posed some interesting questions. Recently, Schuetz and co-workers proposed that the metabolism of bacteria operates close to the Pareto-optimal surface of a three-dimensional space defined by competing objectives and demonstrated the validity of their claims for various environmental perturbations [1]. Albeit multi-objective approaches focused on the bio-engineering objectives have been proposed (Chaper 3), none tackles the multi-objective nature of the cellular objectives.

4.1 Introduction

Assumptions regarding the cellular objectives of an organism when subjected to distinct conditions (environmental, genetic, etc.) are still the object of active discussion. The most common approach is to consider the cell to be in a pseudo steady-state and, since the solution space for the metabolic fluxes of the cell is usually very large, constraint based optimization approaches are often applied for simulating metabolic fluxes. Given this assumption, it is therefore plausible to predict cellular behaviour by solving optimization problems, as long as biologically realistic objective functions are put forward. Several methods have been developed to solve these problems. Among these, Flux Balance Analysis (FBA) [2] is a widely used phenotype prediction tool that uses a linear programming (LP) formulation often for the maximization of growth (synthesis of biomass constituents) as the objective function, considering the biological assumption that organisms tend to maximize their growth [3].

However, to predict the cellular behaviour of mutant organisms, such assumption is not widely accepted and, for that specific purpose, other methods have been proposed such as Minimization of Metabolic Adjustment (MOMA) [4] and Regulatory On/Off Minimization of metabolic flux changes (ROOM) [5]. MOMA's formulation is based on Quadratic Programming (QP) and the objective function is the minimization of flux variations relatively to the wild-type.

As for ROOM, the principle is similar, only the formulation is based on Mixed Integer-Linear Programming (MILP) and the objective function is the minimization of the number of perturbed fluxes from the mutant relatively to the wild-type. The hypothesis underlying both MOMA and ROOM is that fluxes in a perturbed cell (e.g. a mutant) will be redistributed

in order to be as similar as possible to the wild-type [6]. These methods are described with greater detail in Chapter 2.

In this work, we focused on variations of two of the most widely used phenotype prediction methods, the parsimonious enzyme usage FBA (pFBA) (a variation of FBA that minimizes the overall sum of enzyme-associated fluxes [7]) and LMOMA (a linear implementation of MOMA [8]). We analyze the influence of the simulation methods on the results of strain optimization metaheuristic algorithms and suggest a multi-objective approach capable of finding designs compliant with the cellular objectives assumed by the various phenotype prediction methods.

We have confirmed that the results of strain optimization meta-heuristics are highly dependent on the phenotype prediction methods, and specifically, the use of FBA/pFBA leads to sub-optimal results in more challenging tasks. The proposed multi-objective approach reveals itself as a valid method to overcome these limitations, being able to reach solutions that are robust, both taking into account the phenotype simulation method and the minimum granted compound production.

4.2 Effects of phenotype prediction methods over strain optimization

4.2.1 Methods

Using a recent model of *Escherichia coli* K12 (iAF1260) [9], and as a way of avoiding being mislead by any particular set of experimental conditions, the experiments were setup for multiple environmental conditions (aerobiose, anaerobiose), multiple carbon sources (glucose, xylose and glycerol) and

multiple target products (lactate, succinate, pyruvate, phenilalanine and glycerol).

Initially, the two metaheuristics described in Chapter 3 (EAs and SAs) were executed using both pFBA and LMOMA as the phenotype prediction method, using the BPCY objective function. For each, the execution was halted after 50000 function evaluations and the process was repeated 30 times. The remaining configuration parameters for these algorithms are the same that were used in the previous chapter. For analysis purposes, the resulting solution sets for the EA and SA algorithms were merged for each set of conditions. However, the convergence analysis is done separately (algorithm-dependent).

4.2.2 Results

A summary of the number of solutions generated by each of the algorithms in each conditions is presented in Table 4.1. In this table, only solutions that are BPCY-valid are considered, i.e. where $BPCY \geq 1 \times 10^{-5}$. Note that this was a criterion used to take into account solutions where both the biomass and target compound fluxes are larger than zero.

In an initial analysis of this table, the larger number of solutions reachable when LMOMA is the used phenotype prediction method is easily observable. This seems to imply that the space of BPCY-valid solutions is larger when LMOMA is used in the simulation, leading the algorithms to more rapidly finding interesting solutions. This fact is one of the insights that lead to the development of our multiobjective approach, as will be later discussed in more detail.

While the conclusions are extensible for all the case studies, in the following the results and discussion will be pointed only for the aerobic produc-

Table 4.1: Size of the merged solution sets (EAs and SAs) for the various case studies. Target products: Lac. - lactate, Suc. - succinate, Pyr. - pyruvate, Phe. - phenylalanine, Gly. - glycerol

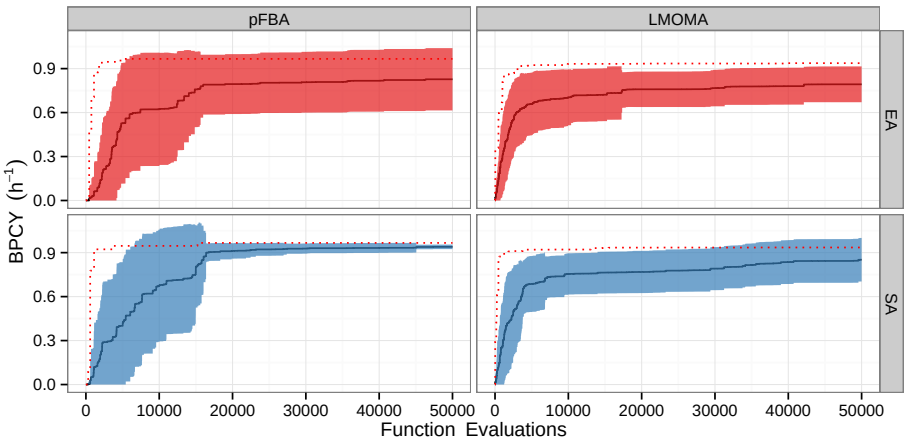
Condition	**Substrate**	**Method**	**Target Product**				
			Lac.	*Suc.*	*Pyr.*	*Phe.*	*Gly.*
Aerobic	*Glucose*	pFBA	292	143	20	2	1
		LMOMA	661	1187	1317	489	433
	Glycerol	pFBA	343	354	309	12	–
		LMOMA	506	730	839	526	–
	Xylose	pFBA	350	211	69	0	3
		LMOMA	757	654	407	351	350
Anaerobic	*Glucose*	pFBA	15	9	95	73	0
		LMOMA	1186	123	597	259	258
	Xylose	pFBA	15	7	151	0	0
		LMOMA	587	133	946	449	314

tion of lactate and succinate using glucose as a carbon source. The results for the remaining case studies are available in supplementary material in http://darwin.di.uminho.pt/pmaia/phdthesis.

In order to understand how different phenotype prediction methods affects the solutions reached by the algorithms, the convergences of the EA and SA, when using pFBA and LMOMA were analyzed separately. Figure 4.1 depicts the convergences in the Lactate and Succinate case studies.

In a first observation of the convergence plots, a smoother convergence when the LMOMA phenotype prediction method is being used becomes evident. When pFBA is the selected method, the convergence evolves in a stepped pattern, with no observable change in fitness for several evaluation functions and larger fitness jumps in some steps. The LMOMA pattern is a smoother one, represented by slight but constant increases in fitness until convergence is attained. In the easier case study, lactate, these differences are not so easily observable, while in the more difficult one, succinate, this trend becomes evident. These trends are extensible to the additional case

(a) Lactate



(b) Succinate

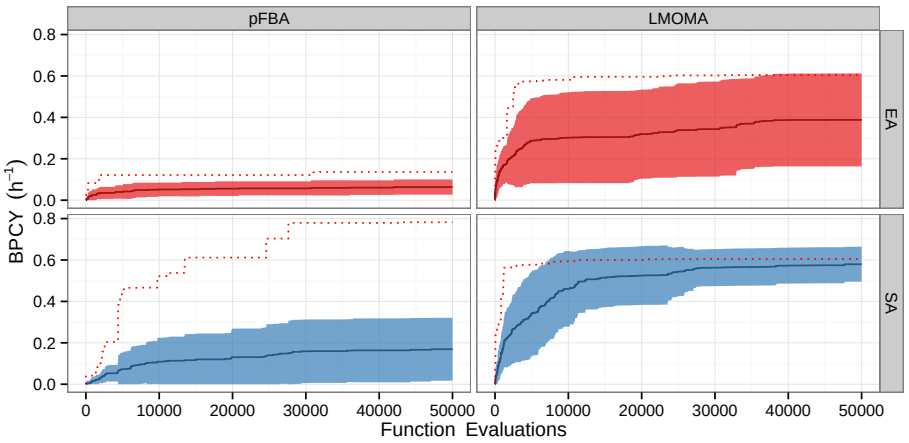


Figure 4.1: Convergence plots for the EA and SA algorithms applied to the aerobic production of lactate and succinate from glucose. The solid lines indicates the means of 30 runs, while the color-shaded areas indicate the standard deviation. The dashed lines represent the maximum value for each algorithm and problem.

studies (in supplementary material) where, in some cases, these patterns are even more declared.

To evaluate the effect that these differences had in the phenotype (flux

values) of the solutions reached by the algorithms with each of the phenotype prediction methods, an analysis on the flux distribution of such solutions was devised. A wild-type flux distribution was predicted using pFBA and taken as the reference flux distribution. The distribution of flux distances from the mutant phenotypes to this reference was then computed (Figure 4.2). To meet this end, the Jaccard distance for asymmetric binary attributes (d_J) was employed:

$$d_J = \frac{M_{01} + M_{10}}{M_{01} + M_{10} + M_{11}} \tag{4.1}$$

where M_{01} represents the total number of fluxes active in the mutant, but inactive in the wild-type; M_{10} is the total number of fluxes active in the wild-type but not in the mutant; and, M_{11} is the number of fluxes that are active in both the mutant and the wild type flux distributions. This metric only considers the flux differences as a binary array (on or off), thus ignoring the effective flux levels.

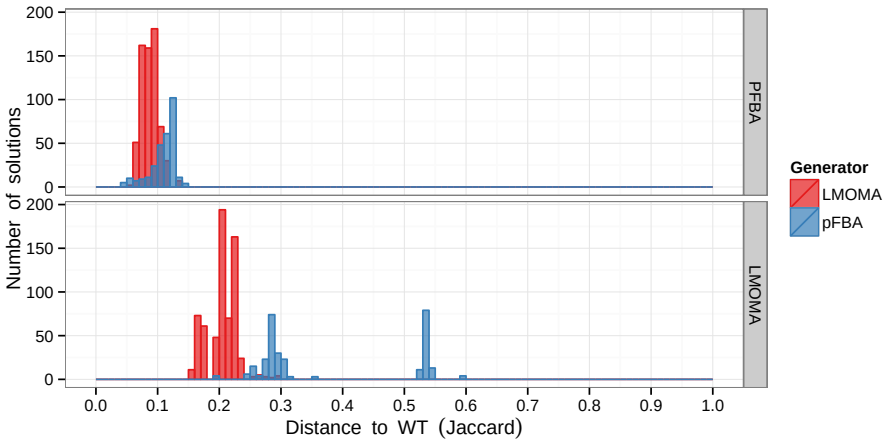
Every solution generated by the EA and SA, while using LMOMA (LMOMA-generated) was re-simulated using pFBA (top histogram in each chart) while every solution generated using pFBA (pFBA-generated) was re-simulated using LMOMA (bottom histogram in each chart). By visually inspecting the histograms, some observations are possible:

1. Overall, the solutions simulated by LMOMA are usually farther from the wild-type than the ones simulated by pFBA;

2. When re-simulated with pFBA, the LMOMA-generated solutions, are generally closer to the wild-type than the pFBA ones;

3. When re-simulated with LMOMA, the pFBA-generated solutions are generally much farther from the wild-type than the LMOMA ones.

(a) Lactate



(b) Succinate

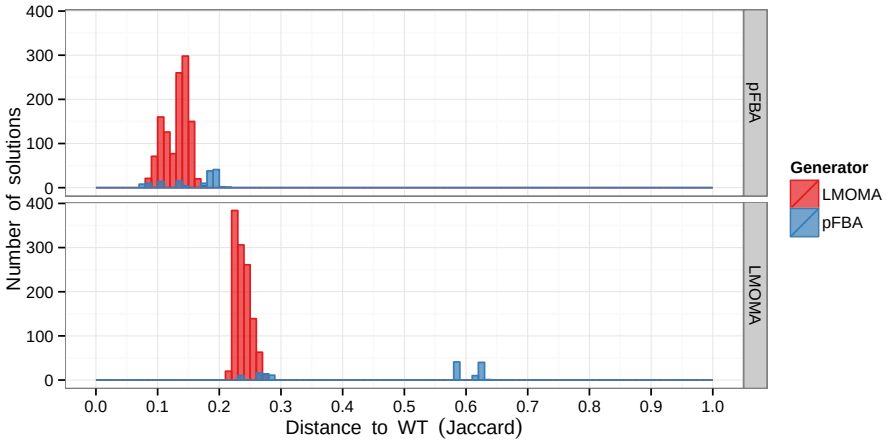


Figure 4.2: Distribution of the Jaccard distances from the solutions flux distributions to the wild-type flux distributions for the aerobic production of lactate (a) and succinate (b) from glucose. Red bins and blue bins represent solutions generated by LMOMA and pFBA, respectively. Solutions were re-simulated with pFBA (top in each plot) and LMOMA (bottom in each plot).

Some of them even have a $d_J > 0.5$.

These facts can be dissected and analyzed in more detail. The formula-

tion of the pFBA procedure helps explaining the observation 1 given that, for a given flux space that maximizes biomass, it will return the flux distribution that minimizes the overall sum of fluxes. On the other hand, the fact that LMOMA solutions are closer to the wild-type than pFBA ones, even when simulated with pFBA (observation 2), can be attributed to the fact that the LMOMA objective function tries to minimize the distance between the wild-type and the mutant flux distributions (i.e., there is a bias in the LMOMA optimization towards this objective). This is important, because we assume that solutions that are closer to the wild-type are more likely to work in reality [4]. This is observable for simulations with pFBA and LMOMA. On the other hand, the solutions simulated with pFBA are generally closer to the wild-type, which can be attributed to the pFBA objective function that minimizes the overall sum of fluxes. This means that pFBA simulations are probably closer to ROOM than LMOMA. Finally, in observation 3, the pFBA-generated solutions are clearly modifying a higher number of fluxes when they are simulated with LMOMA. This can be explained by the fact that the pFBA procedure does not have any bias towards flux distributions that are closer to the wild-type.

Another question that quickly arises is how many solutions, generated by each of the methods, are actually valid when simulated with the other. Venn-like diagrams are presented in Figure 4.3 to provide a first answer to this question.

It is easily observable that a large number of solutions (about half) are not BPCY-valid for both methods. From these, the majority are BPCY-valid for LMOMA only. This trend can also be observed in the other case studies available in the supplementary material, although with some variations. This confirms that the phenotype prediction method is a determining

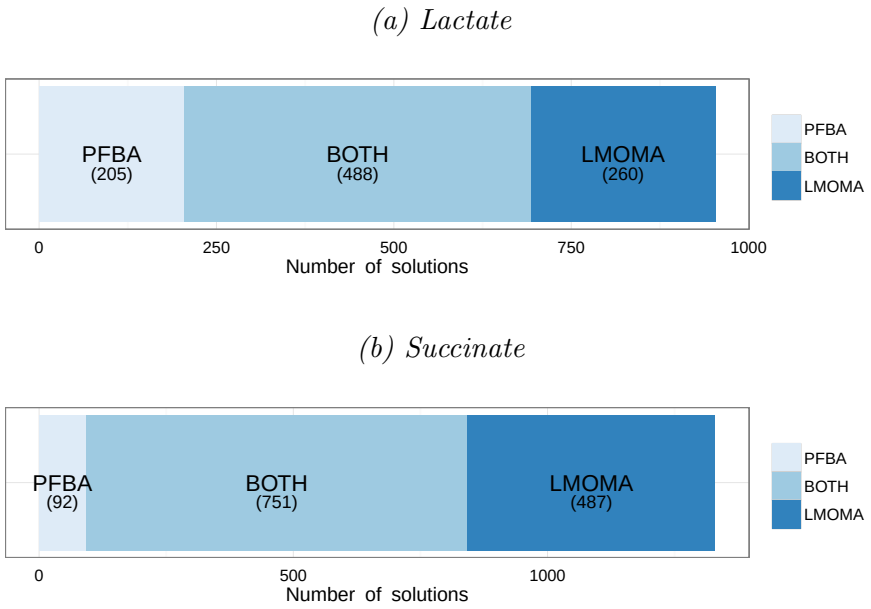


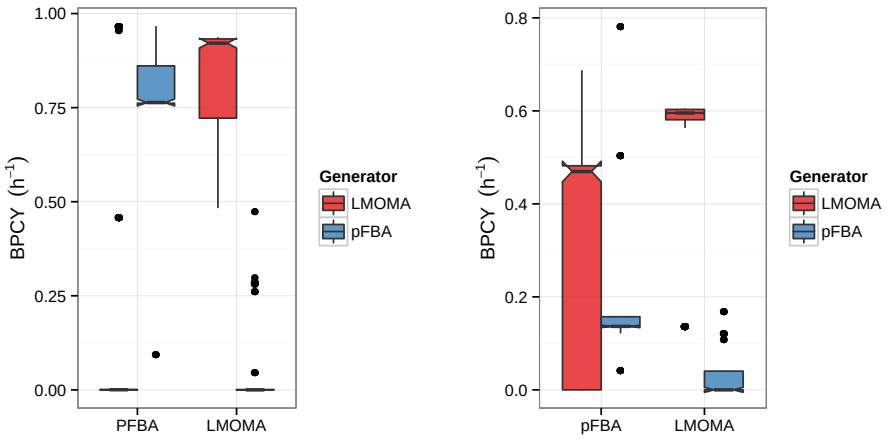
Figure 4.3: Venn-like diagrams for checking the inter-method validity of the solutions. From left to right, solutions that are BPCY-valid for: only pFBA, both pFBA and LMOMA, and only LMOMA.

factor not only on the performance of the strain optimization algorithms, but also in the sets of solutions they yield. If we assume that the likelihood of these solutions working in reality increases if they are valid using different phenotype prediction methods methods, then most of the solutions found are not robust.

While it is also clear that, in these case studies, there is a good set of solutions that are BPCY-valid for both methods, the quality of these solutions is not addressed, since this analysis includes solutions whose BPCY values are close to zero ($\geq 1 \times 10^{-5}$).

To better understand how the fitnesses vary as a function of the phenotype prediction methods, Figures 4.4 and 4.5 are put forward, where the BPCY-values are taken into account. In Figure 4.4, solutions generated by

the strain optimization algorithms when one of the phenotype prediction methods was used were re-simulated with the other, their BPCY values were calculated for both cases and represented in the form of boxplots.



(a) Lactate *(b) Succinate*

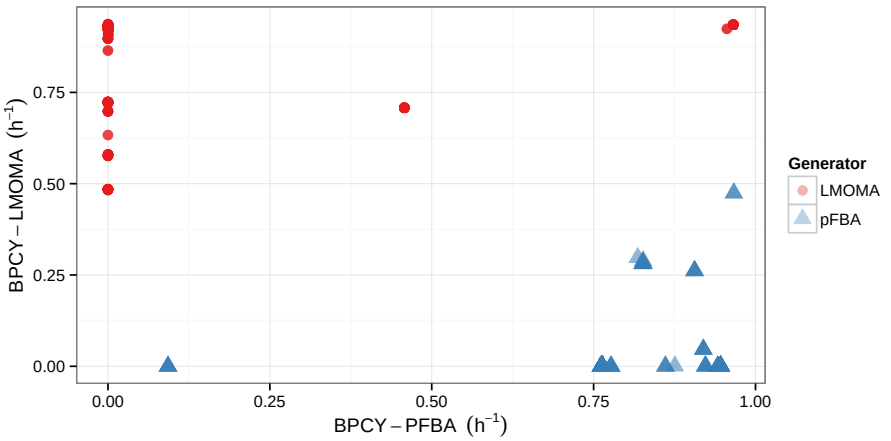
Figure 4.4: BPCY (mmol product . mmol substrate^{-1} . h^{-1}) boxplots for the Lactate (a) and Succinate (b) case studies. Solutions generated with pFBA (blue,right) and LMOMA (red,left) are re-simulated using both methods (x-axis).

From the boxplots it is clear that the distribution of the BPCY values of the solutions generated when using one of the phenotype prediction methods changes dramatically when using the other. Remarkably, in the succinate case study, the average BPCY of the LMOMA-generated solutions when simulated with pFBA is superior to the average of the pFBA-generated solutions.

This fact is further supported by the scatter plots presented in Figure 4.5, which allow the visualization of the BPCY obtained using the two different methods for individual solutions.

In these plots, particular attention should be paid to the LMOMA solutions in the top right region of the plots. In the perspective of this work,

(a) Lactate



(b) Succinate

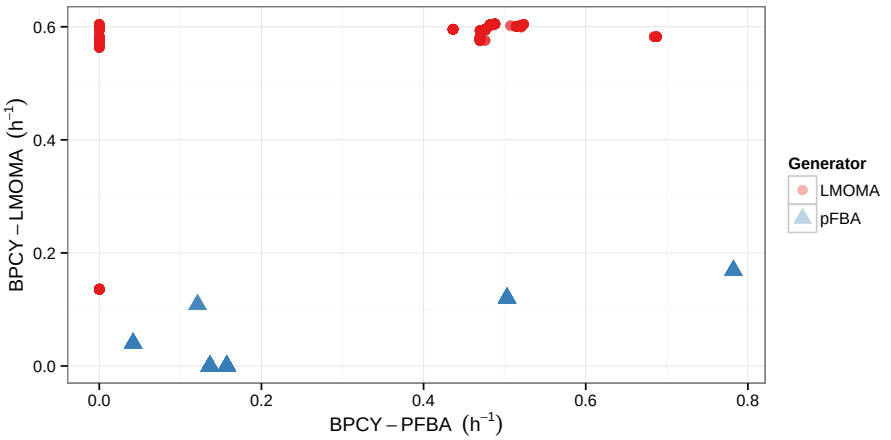


Figure 4.5: Scatter plots showing BPCY (mmol product . mmol substrate^{-1} . h^{-1}) values obtained by LMOMA(red) and pFBA(blue) generated solutions when simulated using pFBA (x-axis) and LMOMA (y-axis) for the lactate (a) and succinate (b) case studies.

these will be the desired solutions since they provide good results using both prediction methods, being considered more reliable.

Albeit being curious, this can be partially attributed to the larger/less

constrained LMOMA solution space. That is, in cases where few valid FBA solutions exist, and there is the necessity for a specific (and restricted) combination of knockouts to guarantee the production of a desired compound, that specific combination might be hard to reach using FBA, thus rendering the optimization process close to a random sampling while no valid solution is found. Alternatively, LMOMA solutions can spread the flux by multiple reactions reaching a multitude of valid solutions from the early stages of the optimization, i.e, with few knockouts (this effect can be observed in the convergence plots). Some of these solutions or areas of the LMOMA solution space are BPCY-valid for FBA as well, as shown by Figures 4.4 and 4.5. This supports the rationale that LMOMA-based optimization can be used to guide the FBA-based optimization, which was used as one of the pivotal reasonings behind the development of the tandem optimization approach detailed in the next section.

4.3 Robust strain optimization by means of tandem phenotype prediction

4.3.1 Methods

Leveraging on the developments presented in Chapter 3, a multi-objective mechanism, capable of searching for genetic designs compliant with two or more phenotype prediction methods, was devised. The SPEA2 algorithm is used with the same configuration as the one used in the previous chapter (details of the algorithm are available in Section 5.4). The main difference concerns the evaluation of the solutions. In this case, each solution is decoded as before and simulated independently using the selected pheno-

type prediction methods, which in the experiments will be two: pFBA and LMOMA. These originate two distinct flux distributions which will be evaluated using BPCY. These two values, BPCY-pFBA and BPCY-LMOMA, make the two objective functions used by SPEA2.

4.3.2 Results

We applied our approach to all previously presented case studies and compared the results. However, as before, the analysis will be focused on the aerobic production of succinate and lactate using glucose as the carbon source, while full results are provided in supplementary material. Table 4.2 summarizes the number of solutions found by our method in every case study.

Table 4.2: Size of the solution sets for the various case studies, using the tandem optimization approach.

		Target Product				
		Lactate	*Succinate*	*Pyruvate*	*Phenylalanine*	*Glycerol*
Aerobic	*Glucose*	1184	709	671	1393	610
	Glycerol	741	537	494	459	–
	Xylose	568	745	789	256	60
Anaerobic	*Glucose*	981	1376	669	347	49
	Xylose	469	1646	594	234	62

With a quick inspection of Table 4.2, the generally larger number of BPCY-valid solutions found across the various case studies when compared with Table 4.1 is easily perceptible.

Figure 4.6 represents the Jaccard distance of the mutant flux distributions (solutions found by the tandem approach) to the wild-type flux distribution. From the histograms, it is possible to conclude that, while the average distance of the LMOMA-based flux distributions when simulated

with both pFBA and LMOMA has not decreased significantly in comparison with the EA/SA approaches, the outliers found in the pFBA-based flux distributions when simulated with LMOMA are not present anymore.

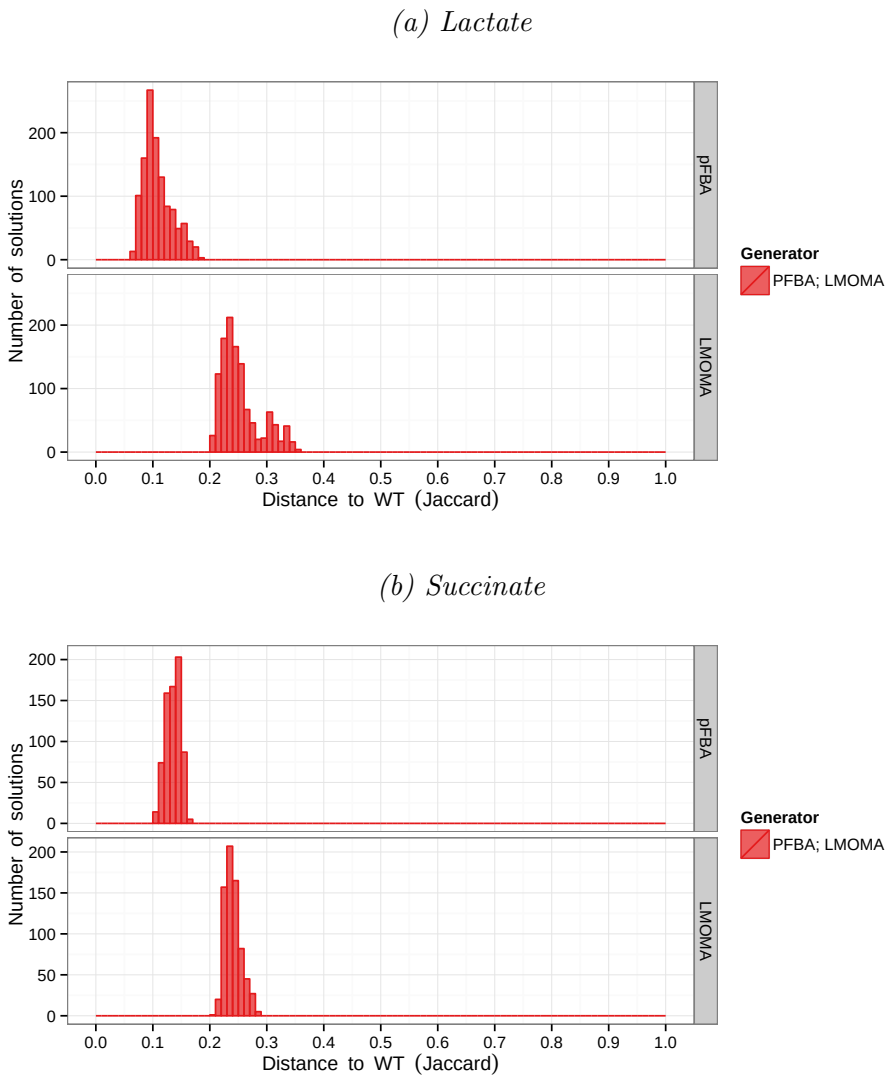


Figure 4.6: Distribution of the Jaccard distances from the solutions flux distributions to the wild-type flux distributions for the aerobic production of lactate (a) and succinate (b) from glucose. All the solution were re-simulated with pFBA (top in each plot) and LMOMA (bottom in each plot).

This observation suggests that the current solutions are closer to each other in terms of flux distributions. Notwithstanding, as stated in the previous section, if we assume that the likelihood of the solutions working in reality increases, if they are valid for different phenotype prediction methods, no conclusions can be derived about the inter-method validity of these solutions. To access the validity in both pFBA and LMOMA phenotype prediction methods, Figure 4.7 is introduced. The results presented in the Venn-like diagrams are self-explanatory, with all solutions but 2 in the succinate case study and 1 in the lactate case study being BPCY-valid for both methods. This result is extremely positive by itself, however, the precise performance of these solutions remains to be evaluated.

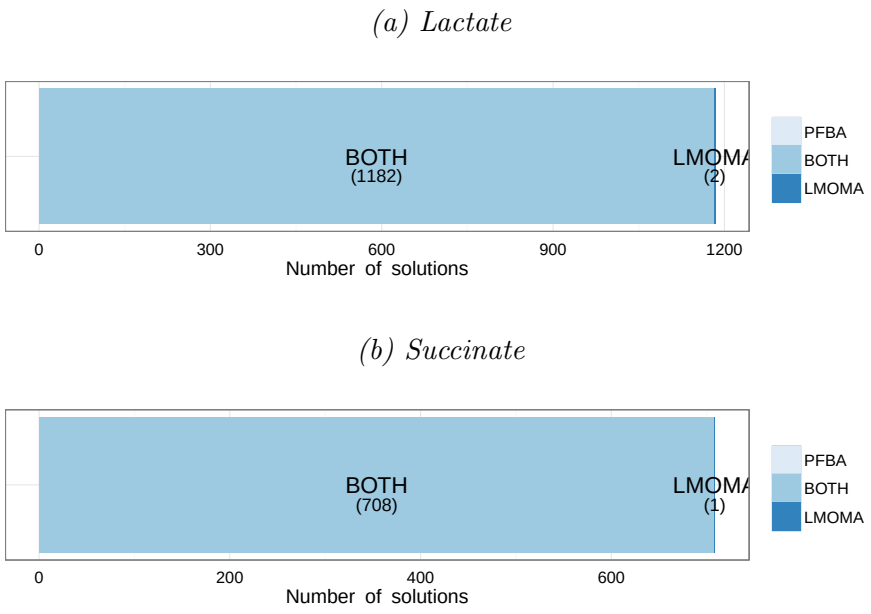
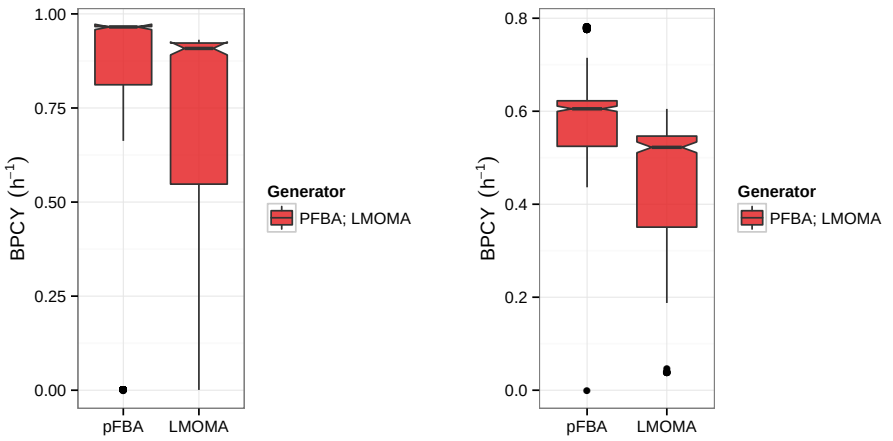


Figure 4.7: Venn-like diagrams for checking the inter-method validity of the solutions. From left to right, solutions that are BPCY-valid for: only pFBA, both pFBA and LMOMA, and only LMOMA.

The hypothesis raised in the previous section, that the LMOMA-based

optimization could be used to guide the pFBA-based optimization, is now revisited here. The corresponding boxplots were generated and are presented in Figure 4.8.



(a) Lactate *(b) Succinate*

Figure 4.8: BPCY Boxplots for the Lactate (a) and Succinate (b) case studies obtained by the tandem approach. Solutions are re-simulated with both methods (x-axis).

By analyzing the boxplots it is now evident that the results have improved greatly in comparison with the previous approach. In the Lactate case study, while in the former approach the pFBA-generated solutions were generally not valid with LMOMA and the LMOMA-generated solutions were not valid with pFBA, here, the solutions are not only valid, but the average of their BPCY values is better, in particular for the pFBA method.

Even more interesting are the results of the Succinate case study. In the previous section, we pointed out the curious results found for this example, where the LMOMA-generated solutions achieved better BPCY values when simulated with pFBA than the pFBA-generated solutions themselves. It is

clear that the tandem approach is able to find still better solutions that are valid with pFBA, than in the previous approach, with an average BPCY of 0.6 $mmol\ product\ .\ mmol\ substrate^{-1}\ .\ h^{-1}$. This improvement in the results can also be witnessed in the scatter plots presented in Figure 4.9 where a large portion of the solutions are located on the top-right region of the plots.

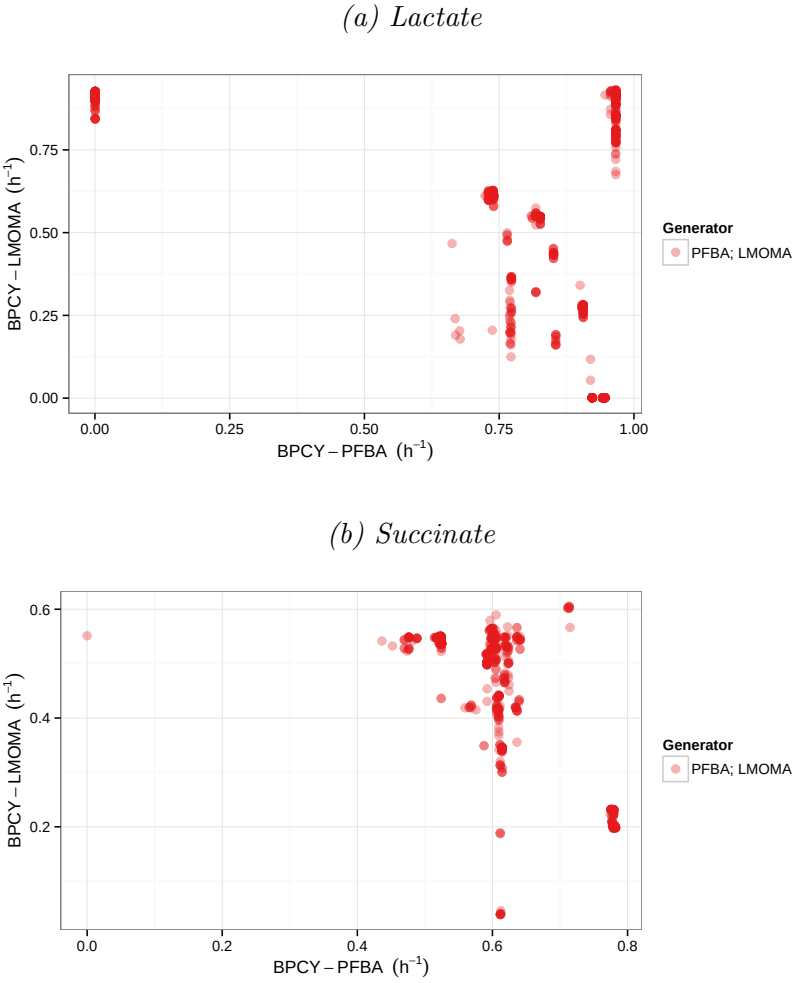


Figure 4.9: Scatter plots showing BPCY values obtained by the tandem approach generated solutions when simulated using pFBA (x-axis) and LMOMA (y-axis) for the lactate (a) and succinate (b) case studies.

4.3.3 LMOMA solutions act as a chaperone for the identification of pFBA-valid solutions in tandem pFBA/LMOMA optimization

One of our claims is that, in very constrained FBA solution spaces where few BPCY-valid solutions exist, i.e., where to reach solutions than can couple biomass growth and target overproduction a large number of specific deletions is required, the LMOMA-based optimization process can act as chaperone for the FBA-based optimization. To help illustrate this process, Figure 4.10 is put forward.

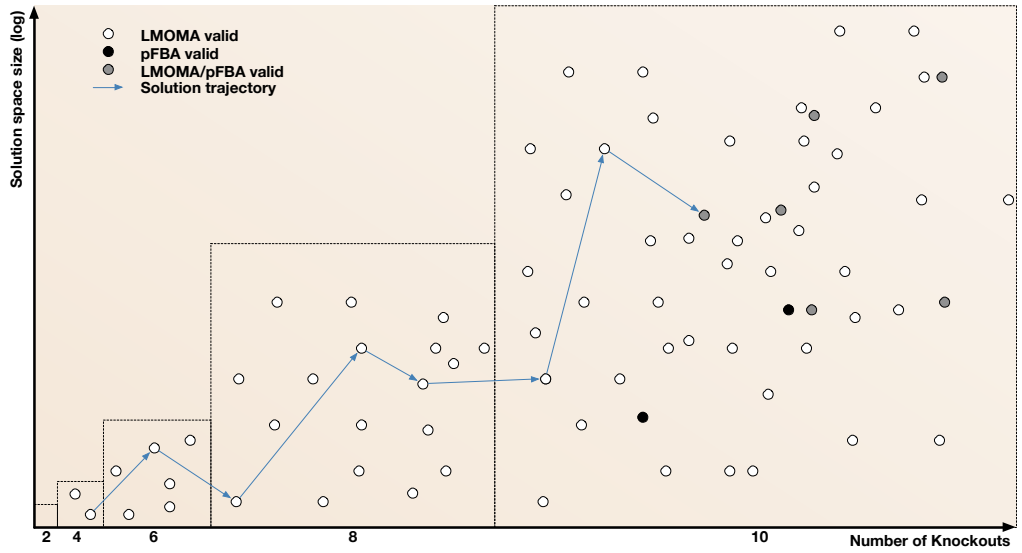


Figure 4.10: Illustration of LMOMA-pFBA tandem optimization. Solution spaces with few knockouts contain no valid pFBA solutions while some LMOMA solutions can be found. In the much larger, many-knockouts, solution space, pFBA, LMOMA and pFBA/LMOMA solutions are found.

The early stage LMOMA solutions allow the algorithm to initialize convergence towards interesting regions of the solution space, by using solely LMOMA solutions' fitnesses. For higher numbers of deletions, despite being scarce, valid pFBA solutions exist. However, the probability of a valid

combination of deletions being found by an evolutionary heuristic using only pFBA as the phenotype prediction method is low. When (and if) the LMOMA and pFBA feasible solution spaces intersect, the tandem optimization approach starts to attribute more value to solutions that are valid for both methods. This is a natural outcome of the dominance property of the underlying MO approach.

To analyze the validity of these claims, we selected the top pFBA / LMOMA solution (the solution that attained the best mean BPCY-pFBA/BPCY-LMOMA) from the succinate case study, and generated all its sub-solutions.

The selected solution (s_Φ) comprised the following 19 gene deletions: s_Φ = ΔprpE, ΔmhpF, ΔsucD, ΔpflA, Δndh, ΔdhaK, ΔadhE, ΔldhA, ΔpntA, Δedd, Δgnd, Δdld, ΔmaeB, ΔgcvT, ΔaldB, ΔtnaA, ΔgldA, ΔpflD, ΔidnO. For each size smaller than the one of s_Φ,all possible combinations of sub-sets of s_Φ were tested, and the best BPCY and percentage of valid solutions for both pFBA and LMOMA were compiled (Table 4.3). The total number of combinations amounted to over 500.000 and the respective phenotype simulations (pFBA/LMOMA) to over 1 million.

While we only accounted for 19 possible genes, the total number of genes available for the algorithms to combine is, in this case study, 240. The possible number of combinations of these 240 genes in order to generate 10-deletion solutions is 1.44×10^{17}. If the pFBA-feasible sub-solution space of these solutions is proportional to that of this solution, it will be very hard for any heuristic algorithm to find a valid combination. It is also worth mentioning that deterministic methods, such as OptKnock [10], do not to scale to these sizes.

By analyzing Table 4.3, it is possible to observe that until reaching

Table 4.3: Analysis of the sub-solution space of solution s_Φ. For each size of sub-set of s_Φ, the best BPCY and percentage of valid solutions, for both pFBA and LMOMA are presented. BPCY units are mmol product . mmol substrate^{-1} . h^{-1}

Deletions	**# Solutions**	**Best BPCY**		**Valid Solutions**	
		pFBA	*LMOMA*	*pFBA*	*LMOMA*
1	19	0.00000	0.00001	0.000%	5.263%
2	171	0.00000	0.03992	0.000%	10.526%
3	969	0.00000	0.19762	0.000%	15.996%
4	3876	0.00000	0.32428	0.000%	21.827%
5	11628	0.00000	0.35743	0.000%	28.113%
6	27132	0.00058	0.41588	0.004%	34.896%
7	50388	0.41190	0.41774	0.026%	42.153%
8	75582	0.45917	0.46551	0.106%	49.780%
9	92378	0.47226	0.49940	0.336%	57.587%
10	92378	0.70233	0.51212	0.913%	65.310%
11	75582	0.70283	0.52409	2.231%	72.652%
12	50388	0.71425	0.53329	5.001%	79.342%
13	27132	0.71468	0.56377	10.353%	85.176%
14	11628	0.71468	0.59254	19.788%	90.033%
15	3876	0.71491	0.59458	34.778%	93.885%
16	969	0.71491	0.59984	55.728%	96.801%
17	171	0.71491	0.60317	80.117%	98.830%
18	19	0.71491	0.60502	100.000%	100.000%
19	1	0.71348	0.60504	100.000%	100.000%

6 gene deletions (gray region), no sub-solutions attained BPCY-valid flux distributions for pFBA and, before 10 gene deletions (light gray region), the number of valid sub-combinations was bellow 1%. Alternatively, the LMOMA solution sub-space allowed for valid combinations with as low as 1 deletion (1 deletion out of 19 predicted both biomass and succinate). The percentage of valid combinations for LMOMA increases steadily with the number of gene deletions. This allows the algorithm to use these solutions to navigate the solution space, ranking them on top of the current population and allowing them to recombine more often in order to generate offspring.

When a valid pFBA/LMOMA solution is found (a point in the intersection of the pFBA and LMOMA feasible solution spaces), the dominance property of the algorithm attributes a higher ranking to this solution which,

in turn, increases its chance to generate derivative solutions. Solutions that are both pFBA and LMOMA valid are always given more reproductive priority, which introduces a bias for the algorithm to explore that region of the flux cone.

To give better support to this analysis, we argue that the smallest pFBA-valid solution found: (s_ψ = Δ*mhp*F, Δ*pflA*, Δ*adh*E, Δ*ldh*A, Δ*dld*, Δ*pfl*D) is either an increment of a valid LMOMA 5-deletion solution or a combination of two LMOMA valid solutions. To investigate this, we scanned all the sub-solutions with 5 deletions or less, and found out that:

- there are two LMOMA valid 5-deletion solutions that could have been incremented with an extra deletion to generate s_ψ, namely:

  1. Δ*mhp*F, Δ*pfl*A, Δ*ldh*A, Δ*dld*, Δ*pfl*D;
  2. Δ*pfl*A, Δ*adh*E, Δ*ldh*A, Δ*dld*, Δ*pfl*D

- there are 1746 valid LMOMA sub-solutions (sizes 1 to 5) that could be combined to generate s_ψ;

Also, there are 36 LMOMA valid 6-deletion solutions that could have been mutated to generate s_ψ, by replacing one of their elements.

Naturally, a more thorough analysis of these solutions and their subspaces is required to give a stronger statistical significance to this claim. Nonetheless, the results and preliminary analysis, strongly support our claim.

4.3.4 Tandem optimization yields robust strain optimization designs

In this context, our interpretation of robustness is two-fold. First, we introduce a new concept of robustness in which solutions that are predicted by more than one phenotype prediction method are more robust, since they comply with more than one assumption regarding the behavior of the organism when subjected to perturbations. While we will not provide any further tests supporting this claim, this robustness is a natural consequence of the objective functions of our tandem approach, which are sufficiently detailed in our previous analyses.

Secondly, we argue that the tandem optimization process is able to attain solutions that are also robust in the LP (FBA) solution cone (FBA-robustness). The problems associated with competing pathways not being accounted for by strain optimization algorithms were first brought to light by Tepper and Schlomi in [11] where they introduced the concept of robust solutions.

The FBA-robustness is tested in a 2-step approach, first a regular FBA phenotype prediction is performed, maximizing the biomass (bio_{step1}), while subjected to the genetic and environmental conditions of the solutions. Next, FBA is performed with the objective of minimizing the production of the target compound, but an extra constraint - $bio_{step2} \geq bio_{step1} * (1 - \alpha)$ where $\alpha = 0.00001$- is added to the problem. If FBA is still able to predict the production of the target compound in these conditions, we consider the solution to be FBA-robust.

To analyze the validity of this claim, we observed the FBA-robustness of the solutions reached by the tandem algorithm in every case study. The results of this analysis are summarized in Table 4.4.

Table 4.4: Percentage of FBA-robust solutions in all the case studies.

		Target Product				
		Lactate	*Succinate*	*Pyruvate*	*Phenylalanine*	*Glycerol*
Aerobic	*Glucose*	75.9%	99.9%	100%	68.7%	35.5%
	Glycerol	86.0%	100%	100%	72.6%	[redacted]
	Xylose	75.4%	93.2%	100%	0.0%	8.3%
Anaerobic	*Glucose*	99.5%	100%	96.1%	79.8%	2.0%
	Xylose	99.2%	100%	99.8%	0.0%	0.0%

A quick analysis of the table allows us to conclude that, for the majority of the case studies, most of the solutions found by the tandem algorithm are FBA-robust. The percentage of FBA-robust solutions found by our method is in the same range of the previous methods (consult supplementary material), however, our method is able to find a much larger number of solutions. Thus, as a consequence, a higher number of FBA-robust solutions is made available to the researchers. For the most difficult case studies, such as the anerobic production of phenylalanine and glycerol from xylose, few or no robust solutions could be found, suggesting that either they do not exist (for up to 20 gene deletions) or they are extremely scarce and difficult to find.

4.4 Conclusions

In this work, a new tandem optimization approach capable of finding robust strain designs compliant with multiple phenotype prediction methods is proposed. Several advantages emerge from using this tandem approach. First, the algorithm helps uncovering pFBA solutions that would otherwise be difficult to find by traditional approaches. Secondly, the majority of these solutions are both FBA-robust and multi-phenotype robust.

Arguably, a valid alternative would be to ignore FBA/pFBA as a phenotype prediction method for perturbed/mutant organisms and use MOMA/LMOMA. However, LMOMA designs suffer from some limitations. Given that the objective function in MOMA/LMOMA is to minimize the distance to the wild-type flux distribution and since it is not bound to the maximization of biomass constraint, MOMA can artificially activate/deactivate a large number of reactions in order to reach this minimum value. This results in flux distributions with a large number of minimally activated fluxes, which is unlikely to be biologically sound. Furthermore, because of this, the analysis of the flux distribution of MOMA/LMOMA solutions is a challenging task, whereas analyzing pFBA flux distributions is an amenable one. The solutions attained by our tandem algorithm provide the advantages of both approaches with none of the shortcomings.

It is also worth mentioning that the implementation of a similar approach via a bi-level design algorithm, such as RobustKnock [11] is not a trivial (if at all possible) task. Moreover, the unique perspective granted by the MO nature of the tandem algorithm would not be possible.

Further work includes testing alternative phenotype prediction methods such as ROOM and MiMBl, which are already supported by the implemented framework (see Chapter 5).

REFERENCES

[1] R. Schuetz, N. Zamboni, M. Zampieri, M. Heinemann, and U. Sauer, "Multidimensional optimality of microbial metabolism," *Science*, vol. 336, no. 6081, pp. 601–604, 2012.

[2] K. J. Kauffman, P. Prakash, and J. S. Edwards, "Advances in flux balance analysis," *Current opinion in biotechnology*, vol. 14, no. 5, pp. 491–496, 2003.

[3] R. U. Ibarra, J. S. Edwards, and B. O. Palsson, "Escherichia coli k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth," *Nature*, vol. 420, no. 6912, pp. 186–189, 2002.

[4] D. Segre, D. Vitkup, and G. M. Church, "Analysis of optimality in natural and perturbed metabolic networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 23, pp. 15112–15117, 2002.

[5] T. Shlomi, O. Berkman, and E. Ruppin, "Regulatory on/off minimization of metabolic flux changes after genetic perturbations," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 21, pp. 7695–7700, 2005.

[6] A. R. Brochado, S. Andrejev, C. D. Maranas, and K. R. Patil, "Impact of stoichiometry representation on simulation of genotype-phenotype relationships in metabolic networks," *PLoS computational biology*, vol. 8, no. 11, p. e1002758, 2012.

[7] N. E. Lewis, K. K. Hixson, T. M. Conrad, J. A. Lerman, P. Charusanti, A. D. Polpitiya, J. N. Adkins, G. Schramm, S. O. Purvine, D. Lopez-Ferrer, *et al.*, "Omic data from evolved e. coli are consistent with computed optimal growth from genome-scale models," *Molecular systems biology*, vol. 6, no. 1, 2010.

[8] S. A. Becker, A. M. Feist, M. L. Mo, G. Hannum, B. Ø. Palsson, and M. J. Herrgard, "Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox," *Nature protocols*, vol. 2, no. 3, pp. 727–738, 2007.

[9] A. M. Feist, C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis, and B. Ø. Palsson, "A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information," *Molecular systems biology*, vol. 3, no. 1, 2007.

[10] A. P. Burgard, P. Pharkya, and C. D. Maranas, "Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization," *Biotechnology and bioengineering*, vol. 84, no. 6, pp. 647–657, 2003.

[11] N. Tepper and T. Shlomi, "Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways," *Bioinformatics*, vol. 26, no. 4, pp. 536–543, 2010.

CHAPTER 5

DEVELOPMENT OF SOFTWARE TOOLS

The need to develop the software tools required for the optimizations and analysis discussed in both Chapters 3 and 4, and the conscience acquired in the previous chapters that no single method is optimal for all particular strain optimization problem, led us to design and implement a well structured software framework that could be made available to the ME community.

5.1 Introduction

The developed framework is structured into multiple layers and provides different levels of user experience, thus allowing both experienced programmers and expert researchers to configure and fine tune strain optimization strategies tailored to their own needs. This is accomplished by a modular software design, allowing both a graphical user interface (GUI) which encapsulates powerful tools for users without programming experience, and a versatile application programming interface (API) for developers that are able to use the provided libraries or contribute to their extension with new methods. Note that some of these libraries are a joint effort of several members of the institute where this thesis was developed. The libraries and packages exclusively developed or significantly extended during this work will be specially detailed. A summary of the main contributions regarding software development are presented in Table 5.1.

5.2 Application Programming Interface Layer

The API and GUI constitute the two top level software layers and their structure is represented in Figure 5.1. The API layer comprises 5 distinct libraries: *Metabolic3*, *BioComponents*, *Solvers*, *Jecoli3* and *MetabolicAnalysis*. The GUI layer is provided by the *OptFlux3* Metabolic Engineering Workbench [1] via the optimization plug-in and will be discussed in the next section.

Table 5.1: Summary of the main contributions regarding software development during this work.

Library	Main contributions
Metabolic3	The configuration package has been fully developed during this work. A completely new, more powerful version of the commandline tools has been implemented to support the new configuration package. The simplification package has been fully updated to support the new persistent mode developed for the solvers library. The essentiality/targets package has been fully redesigned. Only 2 of the 9 currently available methods were developed developed prior to this work. A control center was also developed to centralize and offer API access to these procedures. All the phenotype prediction methods have been re-formulated and re-implemented from scratch to support the new persistent mode developed for the solvers library. A new version of the strain optimization package was developed during this work. New methods and new strategies were implemented on top of a completely new optimization framework."
Solvers	The solvers library has been extended to support a persistent mode. This persistent mode is implemented as a wrapper to the CPLEX optimizer that allows keeping the inner model of the optimizer constant, changing only specific variables, constraints or objective functions as needed. This extension has allowed a 10 to 20 fold speed-up relatively to the previous version."
Jecoli3	"The main additions to the Jecoli library were made in the evolutionary multi-objective domain. Two new MOEAs (SPEA2 and NSGA-II) were added to list of available methods. Moreover, an archive manager, common to all the algorithms has also been developed. Smaller improvements were made throughout this thesis.
MetabolicAnalysis	The MetabolicAnalysis package was fully developed during this work.
OptFlux3	A new Strain Optimization plug-in was fully developed during this work.

5.2.1 Metabolic3

The *Metabolic3* library is the core of the framework. It is responsible for the interaction with the models in their various formats, formulation of all the previously discussed phenotype prediction methods, the computation of the essential information and the strain optimization procedures. The *Metabolic3* library includes an API that allows programmers to take advantage of all the provided facilities. The library is segmented in several packages, from which the most relevant ones are detailed next.

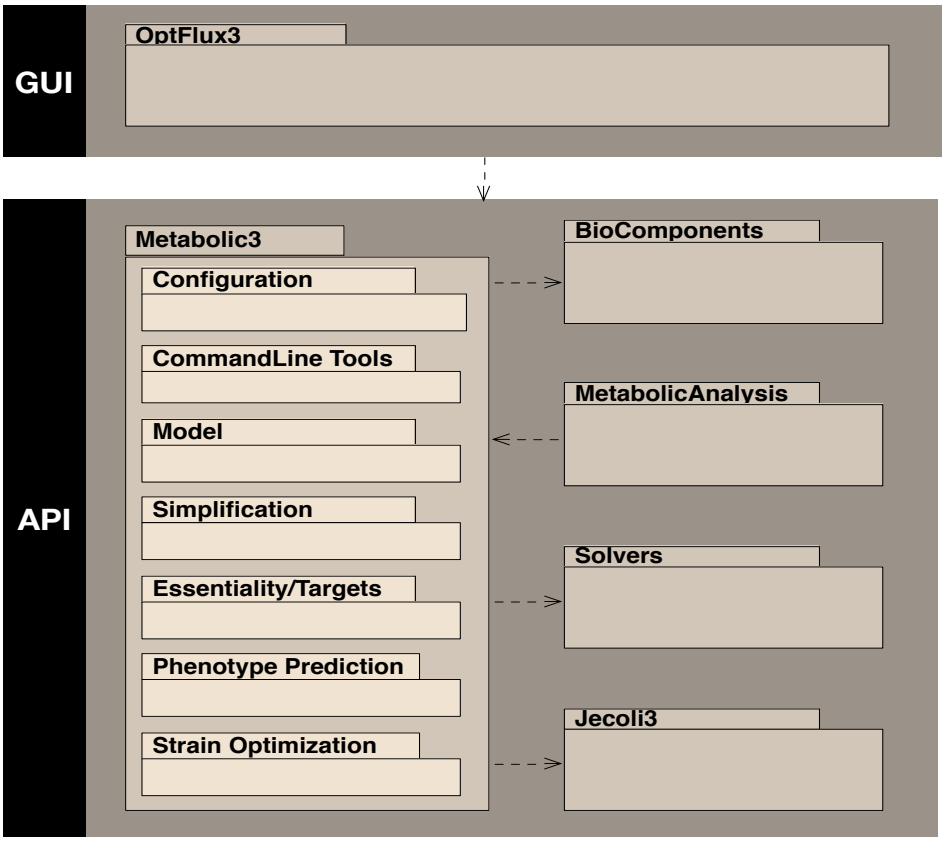


Figure 5.1: Overview of the developed libraries. The GUI layer is provided by the OptFlux workbench whereas the API layer is composed by a total of 5 libraries: Metabolic3, BioComponents, MetabolicAnalysis, Solvers and Jecoli3.

Model

The *Model* package provides data structures to support all the information pertinent to stoichiometric metabolic models. A stoichiometric model is composed by the list of reactions and their constraints (flux capacity, reversibility), the list of metabolites and the stoichiometric coefficients in a matrix form. If available, the genes and respective gene-protein-reaction (GPRs) rules, as well as the pathways, are also integrated in the model information. The input/output facilities are provided via the *BioComponents*

library as displayed in Figure 5.2.

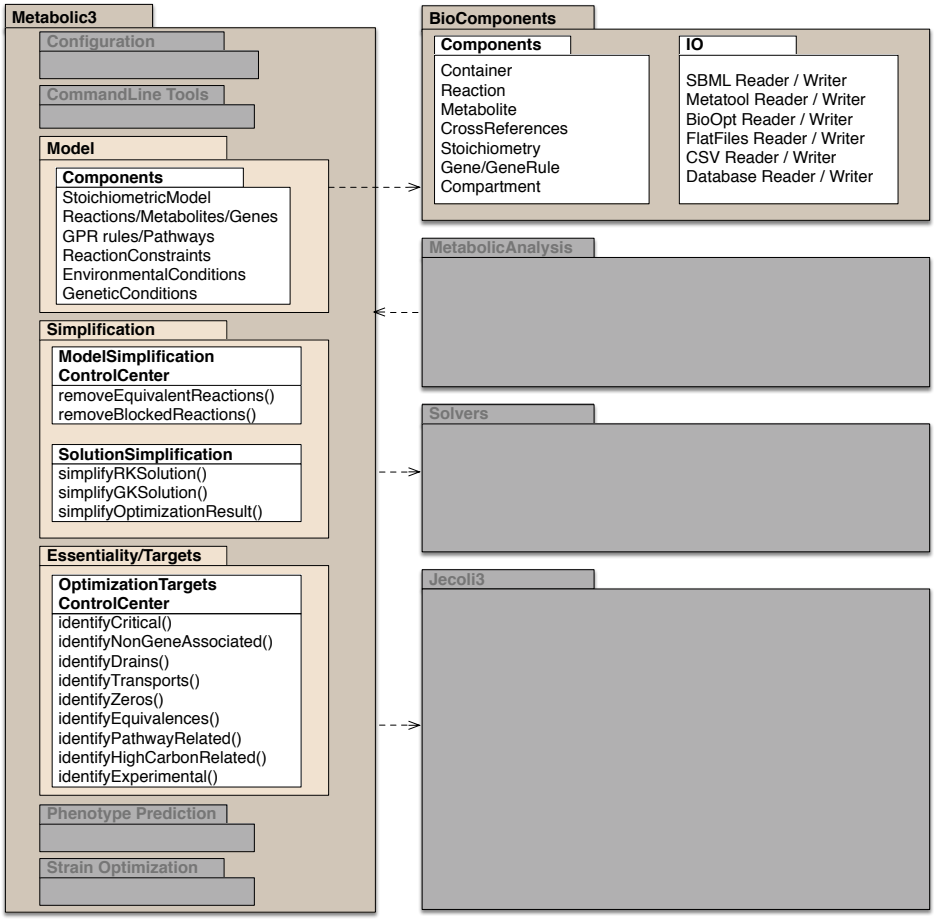


Figure 5.2: The API layer highlighting the Model, Simplification and Essentiality / Targets packages, as well as the interaction with the BioComponents library.

Simplification

The *Simplification* package includes procedures for both model reduction and solution simplification. Model simplification is a common step in the workflow of computational strain optimization procedures, allowing savings of memory and improved performance. Both model reduction procedures

are detailed in [2]. The solution simplification procedures allow the removal of modifications that are not contributing to a set of specified objective functions.

Essentiality/Targets

The *Essentiality/Targets* package provides methods to identify critical information that should be withheld from the list of optimization targets. Most of these strategies were implemented according to the steps reported by Feist and co-workers [3]. These include:

- **Essential and nearly essential reactions** - all the reactions that, when no flux is allowed to be carried over them, render the biomass growth below a given percentage of the wild-type;
- **Non gene-associated reactions** - reactions (metabolic) that either occur spontaneously or, due to deficient functional annotation, do not possess any gene association;
- **Drains** - identify the sinks and sources, pseudo-reactions added to allow interaction of the model with external medium;
- **Transports** - identify membrane transport protein associated reactions;
- **Blocked or zero flux reactions** - reactions that, for the current environmental conditions (reaction constraints) are blocked, thus not allowing flux to be carried through them;
- **Co-sets - equivalent reactions** - reactions that are topologically equivalent (e.g. linear pathways) are grouped and only one of them is considered in the analysis;

- **Reactions in selected pathways** - reactions occurring exclusively in user-defined pathways (e.g. peripherical) can be marked as non-targets;

- **Reactions acting on high-carbon metabolites** - all the reactions that act on high-carbon containing molecules will be removed from the targets list. The number of carbons is defined by the user (the default value is 7 carbons). This strategy requires the definition of a list of exceptions, i.e., a list of metabolites (cofactors and other exceptions) that, despite having high-carbon content, must not be included in this analysis;

- **Reactions related to experimentally validated lethal genes** - By using databases of essential genes, the associated reactions are calculated and marked as non-targets. Two databases were included:

  - **Keio collection** [4] - a list of experimentally validated single-gene knockouts for *E. coli*;
  - **OGEE** - [5] - an online gene essentiality database for multiple organisms.

Phenotype Prediction

The *Phenotype Prediction* package is responsible for the formulation of all the phenotype prediction problems, offering different methods for their solution. This package includes classes that allow the specification of a phenotype prediction method and its use to simulate the behaviour of both wild-type and mutant strains. It allows the prediction of the flux distributions of wild-type strains using the environmental conditions defined in the *Model* package. Moreover, it allows the prediction of flux distributions

for mutant strains, using both the environmental and genetic conditions defined in the model layer. It currently includes implementations of the following formulations: FBA, pFBA, MOMA LMOMA, ROOM and MiMBl as well as FVA. All of these methods were previously described in Chapter 2.

Each formulation is the realization of an abstract problem class provided by the Solvers library. In Figure 5.3, this interaction with the *Solvers* library (described in Section 5.2.2) is exemplified. Finally, the `SteadyState-SimulationControlCenter` API class serves as a control center that aggregates and provides easy access to all the features and functions of the package.

Strain Optimization

The *Strain Optimization* package is the core subject of this work and the functionalities developed in this context present one of thesis major contributions. This layer allows for the definition, tuning and solving of the strain optimization problem that better suits the user requests. In the context of strain optimization, a problem can be defined as: for a given optimization strategy, find the optimal set of modifications that optimize a set of objective functions.

The optimization strategies that are allowed in this framework include gene knockouts (GK), reaction knockouts (RK), gene over/under expression (GOU) and reaction over/under expression (ROU). A reaction knockout is accomplished by blocking that reactions' flux by setting its upper and lower constraints to zero. Since the models are reaction-oriented (i.e. the genes are not directly represent in the metabolic network), a gene knockout is implemented by analyzing that gene's rule (the GPR) and blocking the flux

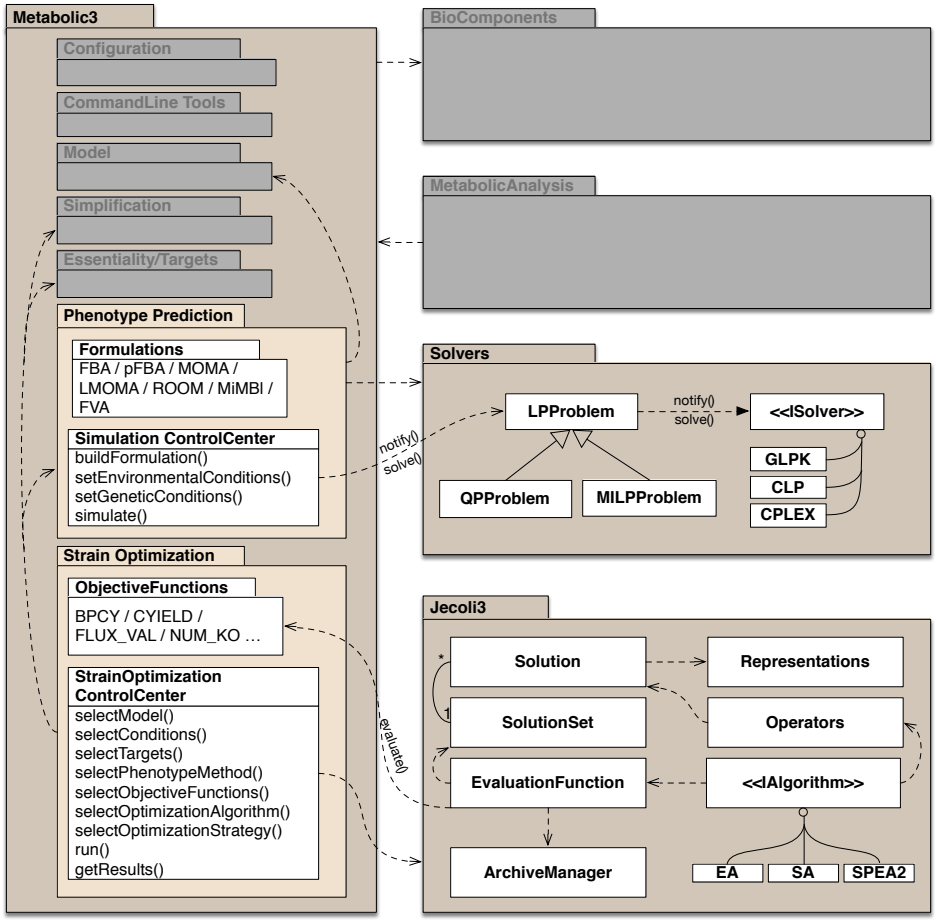


Figure 5.3: The API layer highlighting the Phenotype Prediction and Strain Optimization packages of the Metabolic3 library and their interaction with the Solvers and Jecoli3 libraries.

of all the reactions that are exclusively promoted by that gene. Alternatively, an over/under expression is accomplished by forcing the flux through a given reaction to be increased or decreased relatively to the original value in a reference flux distribution (typically the wild-type flux distribution). More details about the over/under expression approach can be explored in [6].

The objective functions are the most important user input, in the sense

that they identify what are the optimization targets/objectives. An example is the maximization of the flux of a given reaction (over-production of a compound of interest). Several objective functions are readily available, including:

- **Biomass-Product Coupled Yield (BPCY) [7]**: This objective function couples (via multiplication) the biomass flux (B) with the selected product flux (P) relative to the substrate (S) consumption: $BPCY(solution) = \frac{B \times P}{S}$;
- **Carbon Yield (CYIELD)**: Computes the yield of the desired product flux (P) relative to the substrate consumption flux (SS) normalized to the carbon content of both compounds (C_P and C_S, respectively): $CYIELD(solution) = \frac{P.C_P}{S.C_S}$.. This objective function requires that the formulas of the compounds are available;
- **Product Yield with minimum biomass (YIELD)**: Computes the yield of the desired product flux (P) relative to the substrate consumption (S) ensuring that a minimum percentage of biomass (relative to the wild-type) is produced: $YIELD(solution) = \frac{P}{S}, B_{mut} \geq \alpha B_{wt}$, where B_{mut} is the biomass of the mutant, B_{wt} is the biomass of the wild-type strain and α is a user defined percentage;
- **Flux value**: Maximizes or minimizes the value of a specified reaction flux;
- **Number of knockouts (or over/under expressions)**: Minimize the number of modifications performed to the mutant strain;
- **Sum of flux measures**: Maximize or minimize the sum of all the fluxes in the flux distribution (metabolic activity).

The package supports the selection of multiple objective functions to be optimized in parallel. In fact, even if a single objective (SO) algorithm is selected and multiple objective functions are defined, the framework is still able to follow the multi-objective principles. This feature is accomplished through the use of aggregation functions (that can be specified by the user) and the work of an archive manager which manages the solutions reached by the algorithms according to user-defined criteria (refer to Section 5.2.3 for more details).

The optimization package also allows the selection of environmental conditions to use in the optimization problems, as well as taking into account the essentiality information and targets to be considered by the optimization procedure. Regarding the optimization algorithms, 3 alternatives are provided:

- *Evolutionary Algorithms (EAs)* - for single objective;
- *Simulated Annealing (SA)* - for single objective;
- *Strength Pareto Evolutionary Algorithm 2 (SPEA2)* - for multiobjective.

Details of the algorithms' implementations are available in Section 5.4 while their overall description can be found in Chapter 3 and in [2]. An API class named `StrainOptimizationControlCenter` allows easy programmatic access and fine tuning of every feature and parameter previously described. This class can also be fully initialized using configuration files (described in section 5.2.5).

5.2.2 Solvers

The *Solvers* library implements an abstraction layer between generic problems, such as LP, MILP and QP problems (discussed in Chapter 2) and specific formulations, such as FBA, MOMA, ROOM, etc. It provides a complete portfolio of generic components (variables, constraints, objectives, etc.) that can be combined to formulate any of the previously mentioned methods. Furthermore, it also provides connectors that provide transparent communication with various open source and commercial solvers/optimizers, including GLPK[1], CLP[2] and CPLEX[3]. When a phenotype prediction for a given solution's genotype is required from the *Phenotype Prediction* package, the *Solvers* library is notified of changes in the formulations' constraints (via environmental conditions, genetic conditions, etc) and reflects those changes in its inner representation of the problem. Afterwards, the solver optimizes the problem and returns a result (an array of values for each of the problem's variables). The *Solvers* library interprets this result, and its variables' values are translated into a flux distribution that represents the final phenotype of the solution.

5.2.3 Jecoli3

The Java Evolutionary Computation Library (JECoLi 3) [8] is a generic multi-purpose evolutionary computation library. It is an adaptable, flexible, extensible and reliable software framework implementing metaheuristic optimization algorithms, using the Java programming language. In its current version, JECoLi aims to offer a solution suited for the integration of

[1]http://www.gnu.org/software/glpk/glpk.html
[2]https://projects.coin-or.org/Clp
[3]http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

EC-based approaches in larger applications, as well as a way to the rapid and efficient benchmarking of EC algorithms in specific problems taking full advantage of the available hardware. Moreover, several metaheuristic algorithms are provided: general purpose Evolutionary Algorithms, including Genetic Algorithms (GAs) and Evolutionary Programming; Differential Evolution; Genetic Programming (GP) and Linear GP; Simulated Annealing; Cellular Automata GAs [1] and multi-objective optimization Evolutionary Algorithms (NSGA II and SPEA2). JECoLi 3 allows to encode solutions using different types of representations such as: binary, integer, real, permutations, sets and trees. JECoLi also provides a panoply of almost 40 reproduction operators (crossover and mutation) as well as several distinct selection operators implemented for EAs. A number of pre-processing schemes applicable to the selection operators are also made available (e.g. scaling, ranking). The library also includes an API that allows all its components to be easily extended.

One the main features provided by *Jecoli* is the Archive Manager, a facility capable of managing the solutions that are found by the algorithms and providing a common layer between them. The Archive Manager runs in parallel with the optimization algorithms and it is notified by the evaluation functions. Each time a new solution is evaluated, the archive is notified and decides whether or not to keep that solution. Several operations are performed by the Archive Manager to maximize the number of interesting solutions that are kept. The user is allowed to specify the maximum number of solutions to keep, as well as the strategies to keep or remove solutions. When multiple objective functions are defined, the archive manager uses the Zitzler truncation method [9] to prune the archive to the maximum allowed size.

5.2.4 BioComponents

The `BioComponents` library includes API classes for reading and writing metabolic models in several formats. An information abstraction layer (`Container`) works as the pivot between the inputs and outputs, and it is the connecting point for other software packages, such as the `Metabolic3` library (Figure 5.2). All the available information is loaded into this Container and can be either written to other standard formats or converted to other datastructure. As an example, the `Metabolic3` library uses a `ContainerConverter` to extract a Model (which can include GPRs, Pathways, etc., depending on the information available in the original input). Currently, the `BioComponents` library includes readers and writers for several standard formats including the Systems Biology Markup Language (SBML) [10], Metatool [11], Flat-files [1], BioOpt/BioMet [12] and a generic table format (csv, tsv, etc.).

5.2.5 Tools for batch optimization and analysis

The requirement for large workflows of optimization and analysis procedures, where the same methodologies would be repeated for various metabolic models, several different carbon sources and alternative target compounds, using multiple phenotype prediction methods and strain optimization algorithms, led us to create a flexible batch optimization/analysis sub-framework seamlessly integrated in the API. While this problem could have been solved by hard-coding some scripts for each particular problem, we believed that the ME community would lose a valuable research tool. These tools are composed by three different components, the *Configuration* and *CommandLine Tools* packages from the *Metabolic3* library and the complementary

MetabolicAnalysis library, whose dependency relationships are depicted in Figure 5.4.

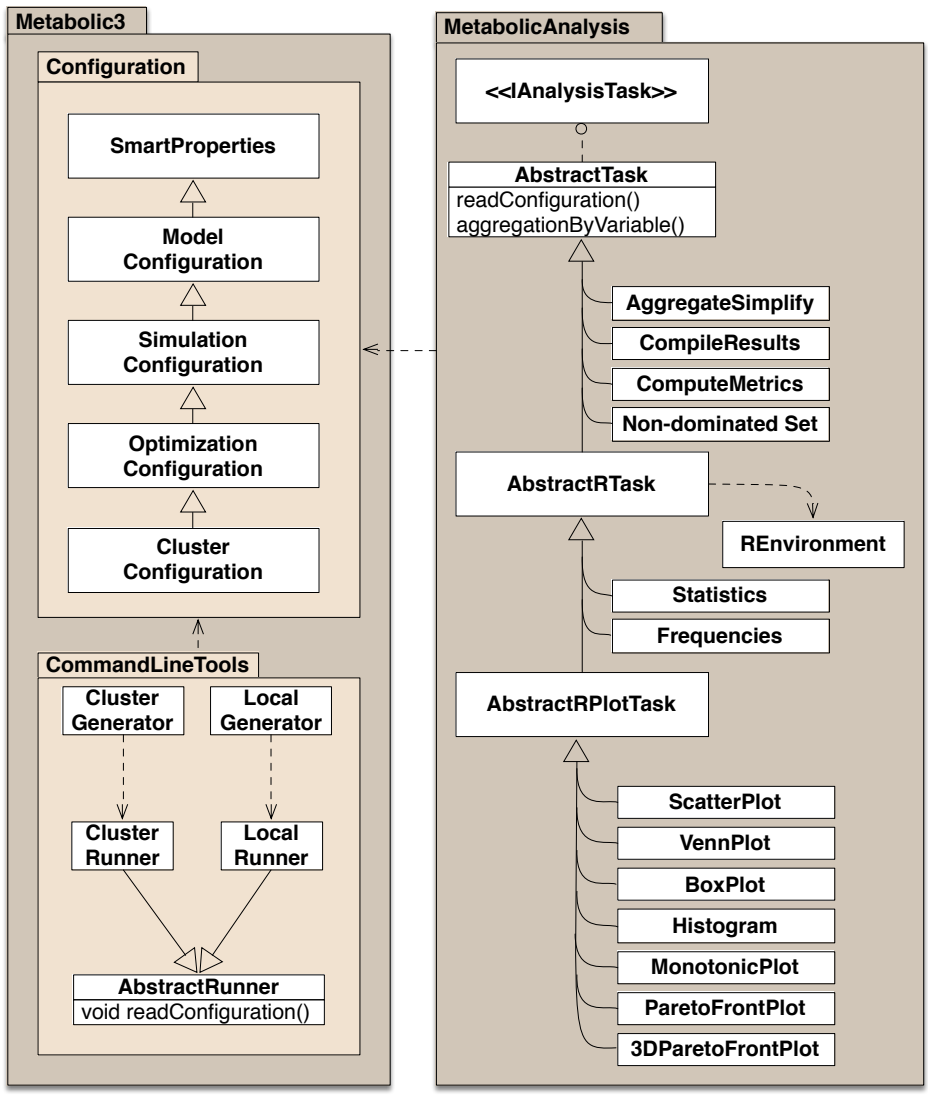


Figure 5.4: The tools for batch optimization and analysis are composed by three components. The Configuration and CommandLine tools from the Metabolic3 library and the MetabolicAnalysis library.

Configuration

The *Configuration* package was developed from scratch in this work and integrated with the core API components of the *Metabolic3* library and it acts as the mediator between the *CommandLine Tools* package and the *MetabolicAnalysis* library. This package allows configurations for single or multiple optimization procedures to be stored, executed and shared effortlessly. The package is composed of several hierarchically related classes as depicted in Figure 5.4.

At the top of the hierarchy is the `SmartProperties` utility class. Any configuration sub-class, whether it is a `ModelConfiguration`, a `Simulation-Configuration`, an `OptimizationConfiguration` or a `ClusterConfiguration`, is an extension of this class, thus inheriting all of its properties and features, which include:

- **Order awareness:** - The class is fully aware of the ordering of the properties therein contained, either by their in-memory declaration time or by their position in the configuration file;
- **Self awareness:** - A configuration can refer to itself or to its current directory using the reserved special variables `${THIS}` and `${CDIR}`, respectively.
- **Properties inheritance:** - A configuration instance can inherit from another configuration instance, by using the special modifier `@DEPENDS`;
- **Variable declaration:** - User defined variables can be declared by using the special modifier `$` followed by the desired variable name. Declared variables can be later accessed by using the modifier `${<VAR>}`.

All the system variables are also made available via the same modifier. Multiple values can be assigned to a variable by separating them with the delimiter `|`.

- **Aliases declaration:** - User defined aliases can be attributed to variables' assignments via the special modifier `~`. This is useful for referring to long or unorthodox variables' assignments.

- **Multiple states:** - Using the special modifier `@COMBINE`, variables can be combined.

A configuration can be declared to combine some or all of its variables, thus possibly generating multiple states. This happens when a variable possesses multiple assignments and is combined with other variables. A configuration will have as many states as the number of possible unique combinations generated by the multiply-assigned variables combined with the `@COMBINE` modifier. An example configuration file and respective multi-state tree, representing one of the optimization workflows used in Chapter 4, is provided in Figure 5.5, where 5 variables are combined. In this example, the `ENV_COND` (environmental conditions) variable has 2 assignments - aerobic and anaerobic; the `SUB` (substrate) variable has 3 assignments - glucose, xylose and glycerol; the `PROD` (target product) variable has 5 assignments - succinate, lactate, pyruvate, glycerol and phenylalanine; the `SIM` (simulation method) variable has 2 assignments - PFBA and LMOMA; and the `OPT_ALG` (optimization algorithm) variable has 3 assignments - EA, SA and SPEA2. When we combine these 5 variables, a state tree is generated where each state is represented by a path from the root to each of the leafs. In this case, a total of 180 states/paths ($2 \times 3 \times 5 \times 2 \times 3$) are generated, each representing a unique combination of each of the 5 variables. This facility

provides a powerful and flexible tool to generate complex workflows where multiple alternatives can be tested and summarized in a single file.

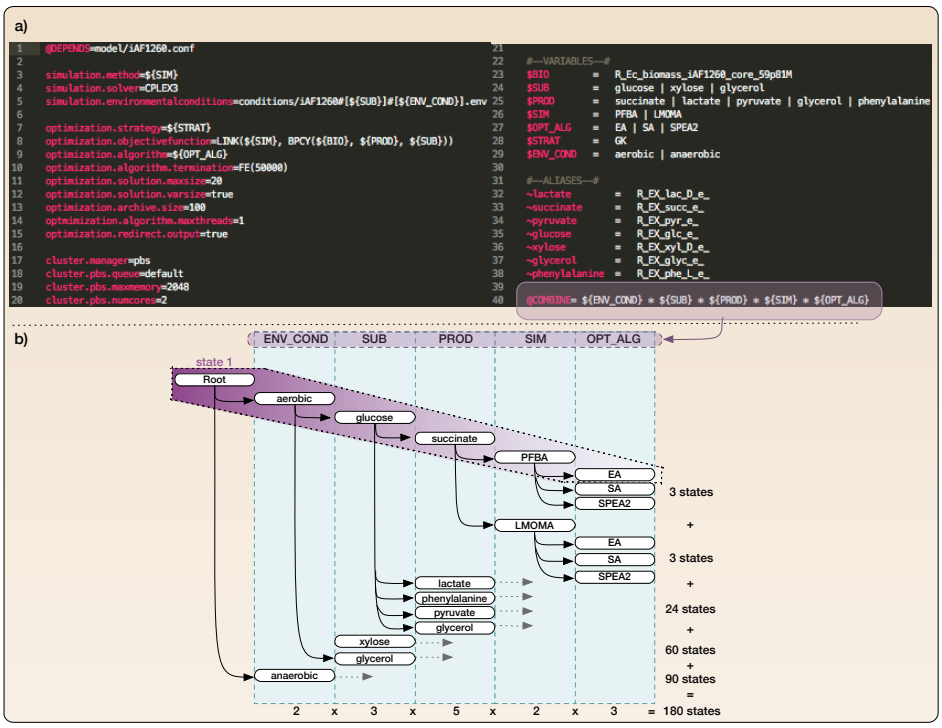


Figure 5.5: Example of a configuration file (a) where 5 variables are combined, some of which have multiple assignments. The tree of possible combinations is displayed in b) with a particular state highlighted (state 1 - dotted purple area).

The `ModelConfiguration`, `SimulationConfiguration`, `Optimization-Configuration` and `ClusterConfiguration` classes hierarchically extend each other adding particular properties and API methods for their specific purposes. The `ModelConfiguration` is an extension of the basic configuration class (`SmartProperties`) providing methods for declaring and loading metabolic models. The `SimulationConfiguration` extends the `ModelConfiguration` adding methods to declare and extract information pertaining to the phenotype prediction methods, such as the method and solver to use or the environmental conditions. Similarly, `Optimization-`

`Configuration` extends the `SimulationConfiguration`, adding methods to declare and configure the optimization algorithms and objective functions of the problem. Finally, the `ClusterConfiguration` extends the `OptimizationConfiguration` and adds methods to configure a computer cluster or grid system, such as the job-scheduling system, submission queue or memory and CPU requirements.

CommandLine Tools

The *CommandLine Tools* package contains API classes to work via the command line (5.4). In particular, the `ClusterGenerator` class can interpret a `ClusterConfiguration` file, which contains all the model, simulation, optimization and cluster properties, and generate the appropriate scripts to submit and execute in a cluster environment (the example in Figure 5.5 reflects the configuration of the search cluster[4] used during this work intensive optimization runs). An extra parameter can be provided defining the number of repetitions that each state must be executed (e.g. each experiment can be repeated 30 times for statistical significance). The `LocalGenerator` works in a similar way for environments without queue managers or job scheduling systems (e.g. local systems).

This tool was used throughout this thesis, replacing traditional scripting strategies and allowing a simpler organization of all the configurations and results, as well as making its computationally intensive nature more amenable.

[4] http://search.di.uminho.pt

MetabolicAnalysis

The *MetabolicAnalysis* library provides a set of analysis tasks that can be executed over the results returned by the strain optimization algorithms. This library was fully developed during the period of this thesis and it was extensively useful for the processing and analysis of the results presented so far. All the tasks extend the `AbstractTask` class that provides seamless interaction with the *Metabolic3* `Configuration` package (Figure 5.4).

Upon interpreting an optimization configuration, every task becomes aware of all the possible optimization states (the various alternatives that were executed), the number of repetitions for each state and the created directory structure. Each task has a well defined set of inputs and outputs, which can be combined in a black-box oriented fashion to design an analysis workflow (Figure 5.6).

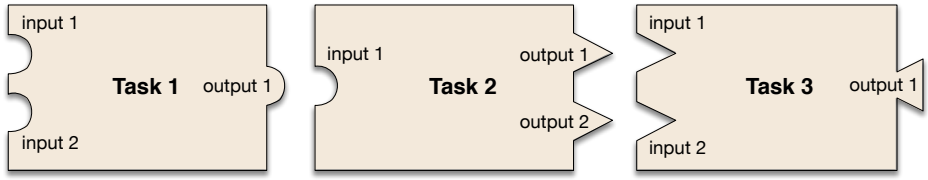


Figure 5.6: Structure of a simple analysis workflow. Three tasks are depicted, each declaring their expected inputs and outputs.

Each task verifies if its prerequisites are met and executes its procedure for each declared configuration state, otherwise the user is informed of missing inputs. Furthermore, some tasks can use up to 3 variables aggregation, to generate aggregated results for several states (e.g., a plotting task can aggregate results for several algorithms and return a grid of plots).

Two distinct sets of tasks are provided, the first set is fully implemented in `Java` (directly extends the `AbstractTask`) and include tasks such as: ag-

gregation and simplification of optimization results (`AggregateSimplify`), compilation of results (`CompileResults`, compilation of various metrics, such as dominance metrics (`ComputeMetrics`) and extraction of non-dominated sets (`Non-dominated Set`). The second one relies on a developed `REnvironment` which provides a communication layer with the `R` language (`AbstractRTask`). This environment is responsible for loading all the `R` dependencies and handling bi-directional communication between `R` and `Java`. Such `R`-dependent tasks include computation of statistics such as averages, means, standard-deviations, etc (`Statistics`), frequencies (`Frequencies`) and multiple plots, such as the venn-like plots presented in Chapter 4 (`Venn`), box-plots used in Chapters 3 and 4 or the Pareto-front plots (`ParetoFrontPlot` and `3DPareto-FrontPlot`) exhibited in Chapter 3 (Figure 5.4).

5.3 Graphical User Interface Layer: OptFlux

The GUI layer is provided by the OptFlux metabolic engineering workbench [1]. The discussed optimization methods were seamlessly integrated in the form of the optimization plug-in. OptFlux aims at being the reference software platform for ME, providing an extensive set of tools for ME experts with user-friendly interfaces.

It was the first tool to incorporate strain optimization tasks, i.e., the identification of ME targets, using Evolutionary Algorithms/Simulated Annealing metaheuristics. It also allows the use of stoichiometric metabolic models for prediction of wild-type and mutant organisms' phenotypes, using FBA, pFBA, MOMA, LMOMA, ROOM and MiMBl. Furthermore, it also incorporates Metabolic Flux Analysis (MFA), computing the admis-

sible flux space given a set of measured fluxes, and support for pathway analysis through the calculation of Elementary Flux Modes [13, 14].

OptFlux also provides access to several of the previously presented methods for model simplification, as well as pre-processing operations aimed at reducing the search space for optimization algorithms. The workbench includes a visualization module that allows the analysis of the model structure that is compatible with the multiple layout formats, allowing the superimposition of simulation results with the model graph [15].

One of the major features of OptFlux is its modular architecture that allows the easy integration of novel components. This is achieved thanks to the framework on top of which OptFlux was build - AIBench [16]. AIBench is a software development framework that was born as a collaborative project between researchers from the University of Minho and researchers from the University of Vigo, Spain. Building applications over AIBench brings important advantages to both the developers and the users, given its design principles and architecture. All the AIBench applications incorporate three types of well defined objects: operations, datatypes and views, following the MVC (model-view-controller) design pattern as exemplified in Figure 5.7.

Developing on top of AIBench forces the programmers to design highly coherent applications, whose components can be easily combined and reused. Additionally, AIBench includes a powerful plug-in engine: applications are designed by plugging components (plug-ins), each of which contains a set of AIBench objects. During the development of this work, the optimization plug-in that ships in the base release of OptFlux was refactored to support the features described in the previous chapters. Following the MVC design pattern, several new components were added, from which the most relevant

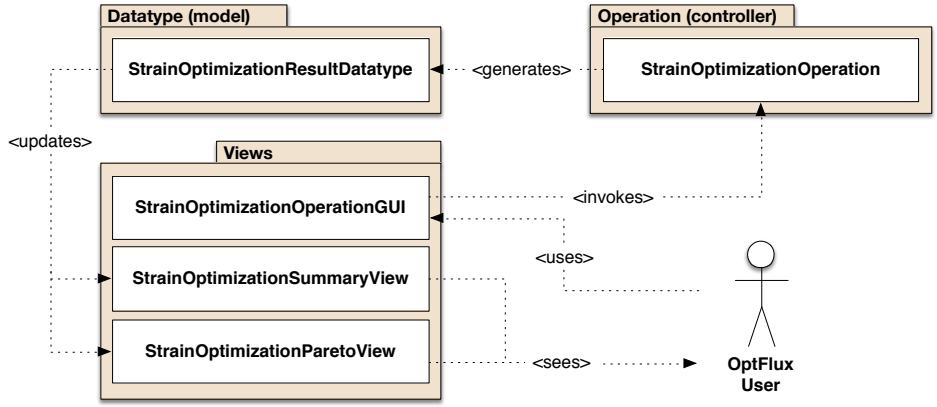


Figure 5.7: A simple use case scenario for the a user interaction with the StrainOptimization plug-in. The three main MVC components and their AIBench counterparts are depicted: the user interacts with the `StrainOptimizationOperationGUI` (a view) which invokes the `StrainOptimizationOperation` (controller). The controller generates the `StrainOptimizationResultDatatype` (model) which in turn updates the views.

will be detailed.

The `StrainOptimizationOperationGUI` provides a dialog for the `StrainOptimizationOperation`, which allows to the user to configure and launch a strain optimization procedure. This dialog is depicted in Figure 5.8

This GUI allows setting up and configuring several optimization parameters:

- **Select Simulation method:** The method used to perform the phenotype simulation: FBA, pFBA, MOMA, LMOMA, ROOM and MiMBl are available;
- **Select Environmental Conditions:** The list of available environmental conditions for this project.
- **Select the objective functions:** OptFlux supports multiple-criteria optimisation; this means that the user can select several objective

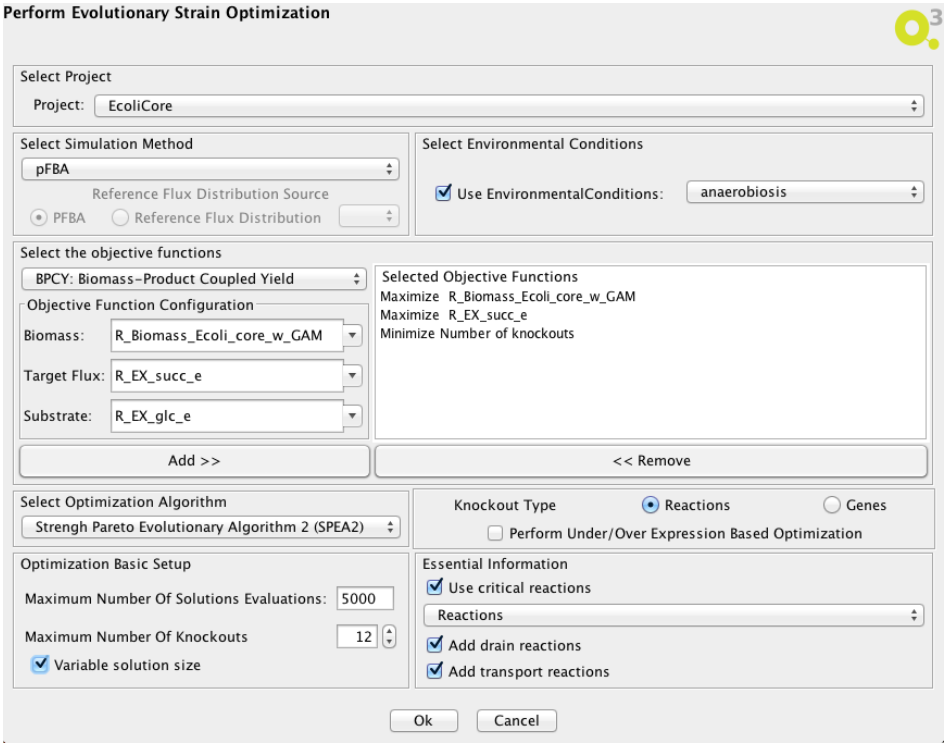


Figure 5.8: Screenshot of the strain optimization configuration dialog.

functions to optimize at the same time, among the following: BPCY, YIELD, Max/Min of reaction flux value, Max/Min of the number of knockouts and Max/Min of the sum of flux measures. The user can configure each objective function individually and add them to the list (on the right);

- **Select Optimization Algorithm:** The optimization method to use, selected from the available ones: EA, SA and SPEA2. SPEA2 is natively multi-objective, thus prone to provide better solutions when multiple objective functions are selected;

- **Knockout Type:** The user can select the strain optimization strategy, by using reactions (R) or genes (G) as targets for deletion/knock-

out (K) or over/under expression (OU);

- **Optimization Basic Setup:**
  - *Maximum Number of Solution Evaluations* - The number of simulations allowed for this optimization. Increasing this parameter will also increase the chances of finding optimal solutions, but the optimization process will take more time to complete;
  - *Maximum Number of Knockouts* - The maximum number of deletions/regulations allowed;
  - *Variable size solution* - When selected, the size of the solutions is not fixed and can vary up to the maximum previously configured.
- **Essential Information:** Depending on whether the optimization strategy is targeting reactions or genes, the user can select a set of reactions/genes that will not be considered for deletion. This set can be computed automatically by OptFlux or loaded from file.

During the optimization process, the user is presented with one of the progress dialogs represented in Figure 5.9. These dialogs allow the user to stop the optimization process when a desired result has been reached, returning the results found thus far.

After the optimization process completes, the results (instances of the `StrainOptimizationResultsDatatype`) are placed in the OptFlux results clipboard. If the user selects this object, the views are launched. The two available views for an optimization result are instances of the `StrainOptimizationSummaryView` and `StrainOptimizationParetoView` classes and they fulfill two distinct objectives. The *Strain Optimization Summary* view, represented in Figure 5.10, provides a summarized overlook of the optimization



Figure 5.9: Screenshots of the progress dialogs presented to the user during optimization. When a single objective function is selected the x-axis represents the number of function evaluations and the y-axis the objective function, resulting in a convergence plot (a). When multiple objective functions are selected both axis are used to represent objective functions (which can be interchangeable), resulting in a pareto plot (b).

execution and configuration, including its time of creation, execution time, number of solutions found during optimization and after post-processing (simplification). It also keeps information about the phenotype prediction method used, environmental conditions, optimization algorithm and respective configurations, as well as the objective functions and the set of critical information (essentiality). This allows the users to know exactly how and when those results were generated since multiple optimizations can be stored in the clipboard.

The second view, the *Strain Optimization Results* view, depicted in Figure 5.11 is where the solutions found during the optimization process are displayed. The view is segmented into three areas. At the top, a list with all the solutions is presented where, for each solution, a solution id and the values for each of the used objective functions are displayed. This list has search, sort (by each column) and export capabilities. Directly below the solutions list, a button (*"Add to simulation results"*) allows the user

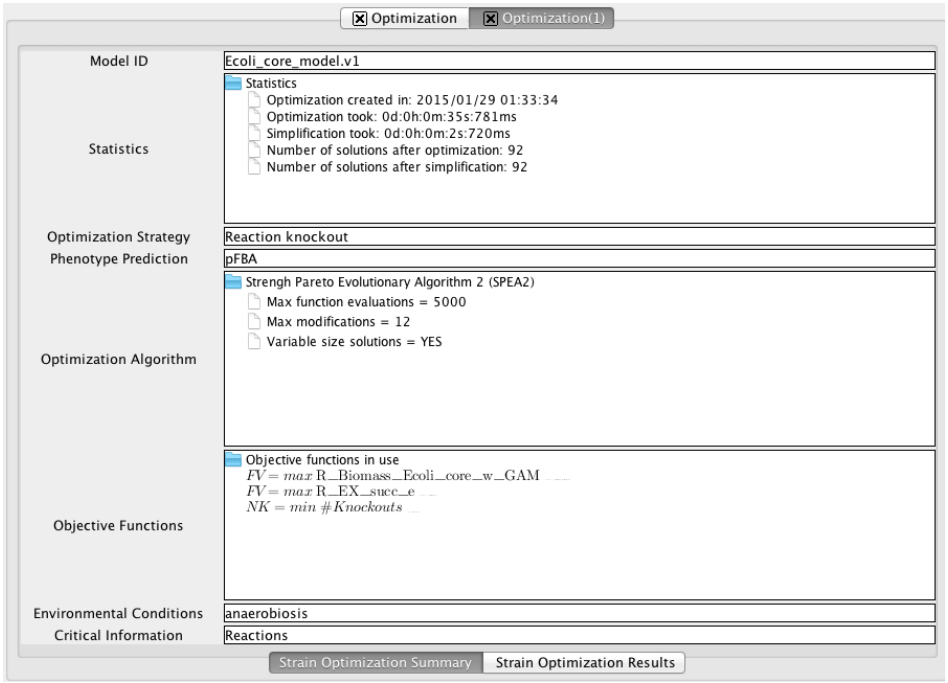


Figure 5.10: Screenshot of the strain optimization summary view.

to send one or multiple selected solutions to the clipboard as independent simulation results to be further analyzed or visualized in detail. When a solution is selected the genetic modifications that compose that solution are displayed in the bottom-left area. Knockouts are represent by $< id >= 0.0$ where $< id >$ can be either a gene or a reaction identifier, and over/under expressions are represented by $< id >=< value >$, where $< value >$ is the number of fold-increases or decreases of that gene expression/reaction value relatively to their reference value (usually from a wild-type simulation).

The bottom-right area is a plot where each solution is represented by their objective function values. This plot allows the user to have a general overview of the success of the optimization procedure, as well as to discriminate between multiple solutions. As an example, a user can opt for

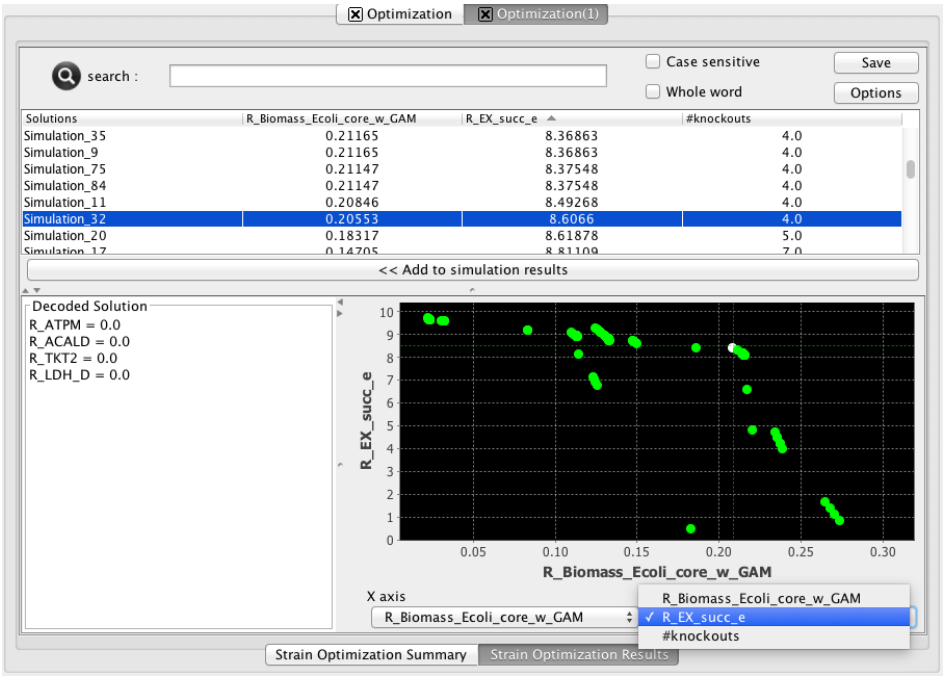


Figure 5.11: Screenshot of the strain optimization results view.

trading-off a solution with a higher production of the desired compound for another with more biomass or one which has less knockouts. Furthermore, the plot is fully interactive, allowing the users to select the solutions directly from it, switching the objective functions represented in each axis, zooming in/out and even exporting the plot to a portable image format.

5.4 Detailed Methods

During this thesis and in particular in Chapter 3, several algorithms were introduced but not explored extensively. In this section, the algorithmic details and particular features of some of those algorithms will be presented and discussed.

5.4.1 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

Main Loop

The overall algorithm main loop is defined in Algorithm 1.

Algorithm 1 The SPEA2 main loop

1: **Input:**
   N {*population size*}
   $\bar{N}$ {*archive size*}
   E {*maximum number of evaluations*}
2: **Output:**
   A {*nondominated set*}
3: **Step 1 - Initialization:**
4: P_0 {*randomly generate an initial population*}
5: $\bar{P}_0 = \emptyset$ {*create the empty archive (external set)*}
6: Set $e = 0$ {*initialize evaluation counter*}
7: **Step 2:**
8: `evaluatePhenotypes`(P_e) {*Phenotype Evaluation*}
9: `fit`(P_e) and `fit`$(\bar{P}_e)$ {*Calculate fitness values of solutions in* P_e. *and* $\bar{P}_e$. *See 5.4.1*}
10: $U_e \leftarrow$ `union`$(P_e,\bar{P}_e)$ {*Calculate multi-set union between population and archive*}
11: **Step 3 - Environmental Selection:** Copy all nondominated individuals in U_e to $\bar{P}_{e+1}$. If size of $\bar{P}_{e+1}$ exceeds $\bar{N}$ then reduce $\bar{P}_{e+1}$ by means of the truncation operator, otherwise if size of $\bar{P}_{e+1}$ is less than $\bar{N}$ then fill $\bar{P}_{e+1}$ with dominated individuals from P_e and $\bar{P}_e$.
12: **Step 4 - Termination:** if $e \geq E$ or another stopping criterion is satisfied, then set A to the set of decision variables represented by the nondominated in $\bar{P}_{e+1}$ and stop. Otherwise, proceed to step 5.
13: **Step 5 - Mating Selection:** Perform binary tournament selection with replacement on $\bar{P}_{e+1}$ to fill the mating pool
14: **Step 6 - Variation:** Apply recombination and mutation operators to the pool and set $\bar{P}_{e+1}$ to the resulting population. Increment evaluation counter ($e = e + |P_e|$). Go to Step 2.

In SPEA2, individuals that are dominated by the same archive members have identical fitness values. This means that in the case when the archive contains only a single individual, all population members have the same rank independently of whether they dominate each other or not. As a consequence, the selection pressure is decreased substantially and, in this particular case, the SPEA behaves like a random search algorithm. To avoid this, in SPEA2, for each individual both dominating and dominated solutions are taken into account.

Fitness assignment

Each individual i is assigned a strength value S_i representing the number of solutions it dominates:

$$S(i) = |j : j \in P_t \cup \bar{P}_t \wedge i \prec j| \tag{5.1}$$

where $|.|$ is the cardinality of the set, P_t is the current population, $\bar{P}_t$ the current archive, $\bigcup$ stands for multiset union and $\prec$ corresponds to the Pareto dominance relation. The raw fitness $R(i)$ of an individual i is calculated:

$$R(i) = \sum_{j \in P_t \cup \bar{P}_t, j \prec i} S(j) \tag{5.2}$$

The raw fitness alone is not robust when most individuals do not dominate each other. Additional density information is then added to discriminate between individuals with identical raw fitness values. Density is defined as:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{5.3}$$

where σ_i^k is the distance of the individual i to k, and $k= \sqrt{N + \bar{N}}$, with N and $\bar{N}$ being the size of the population and the archive, respectively. Finally, the final fitness of the individual i is calculated as:

$$F(i) = R(i) + D(i) \tag{5.4}$$

Density Estimation and Archive Truncation

If many individuals of the current generation are indifferent, i.e., do not dominate each other, none or very little information can be obtained on the basis of the ordering defined by the dominance relation. In this situation, which is very likely to occur in the presence of more than two objectives, density information has to be used in order to guide the search more effectively. Clustering makes use of this information, but only with regard to the archive and not to the population.

The SPEA2 algorithm is able to reduce the nondominated set without losing its characterists via the clustering technique previously mentioned, however, to avoid losing outer solutions an archive truncation method that preserves these, is also contemplated. More details about these two procedures can be found in [9].

Environmental Selection

There are two main differences between the archive update operation between SPEA and SPEA2: the first is that the number of individuals in the archive is constant over time; the second regards the truncation method that prevents boundary solutions from being removed. The first step is to copy all nondominated individuals from the archive and the population to the next generation:

$$\bar{P}_{t+1} = \{i : (i \in P_t \cup \bar{P}_t \wedge F(i) < 1)\} \tag{5.5}$$

If the nondominated front fits the archive ($|\bar{P}_{t+1}| = \bar{N}$), the environmental selection step is completed. Otherwise, either the archive is too small or too large. If the archive is too small, the best $\bar{N} - |\bar{P}_{t+1}|$ dominated individuals in the previous archive and population are copied to the new archive. In the second case, the archive truncation procedure is applied until the archive size is equal to $\bar{N}$.

5.4.2 Evolutionary Algorithm (EA)

The EA main loop is defined in Algorithm 2. Further details can be found in [2].

Algorithm 2 The EA main loop

1: **Input:**
   N {*population size*}
   Nr {*number of replacements per generation*}
   E {*maximum number of evaluations*}
2: **Output:**
   P_E {*final population, i.e., when e=E*}
3: **Initialization:**
   Randomly generate an initial population P_0;
   $e \leftarrow 0$ {*set evaluation counter to 0*}
   `evaluatePhenotypes`(P_0)
4: **while** $(e \leq E)$ **do**
5: $Par = [Par_0, \ldots, Par_n] \leftarrow P_e$ {*select parents*}
6: $P_e \leftarrow$ `recombine`(Par) {*apply recombination to generate offspring*}
7: $P_e \leftarrow$ `mutate`(P_e) {*apply mutation*}
8: `evaluatePhenotypes`(P_e)
9: $P_{e+1} \leftarrow$ `selection`(P_e) {*select individuals for the next generation*}
10: $e \leftarrow e + Nr$. {*increase counter by the number of new solutions*}
11: **end while**

5.4.3 Simulated Annealing (SA)

The SA main loop is defined in algorithm 3. Further details can be found in [2].

Algorithm 3 The SA main loop (assuming maximization)

1: **Input:**
   E {*maximum number of evaluations*}
   TpT {*number of trials per temperature*}
   Δ_i {*allowed initial delta*}
   Δ_f {*allowed final delta*}
2: **Output:**
   S_E {*final solution, i.e., when e=E*}
3: **Initialization:**
   $e \leftarrow 0$ {*set evaluation counter to 0*}
   $\{\alpha, T_e\} \leftarrow$ `initializeAnnealingSchedule`(Δ_i, Δ_f)
   S_0 {*generate an initial solution*}
   $F_{S_0} \leftarrow$ `evaluateSolution`(S_0) {*evaluate initial solution fitness*}
4: **while** $(e \leq E)$ **do**
5: **for** $(trial = 0 \rightarrow TpT)$ **do**
6: $S_{trial} \leftarrow$ `mutate`(S_e) {*create trial solution via mutation*}
7: $F_{S_{trial}} \leftarrow$ `evaluateSolution`(S_{trial}) {*evaluate trial solution*}
8: **if** $(F_{S_{trial}} \geq F_{S_e})$ **then**
9: $S_e \leftarrow S_{trial}$ {*move to new solution*}
10: **else**
11: $rnd \leftarrow [0..1]$ {*create random number in range*}
12: $boltz \leftarrow e^{\frac{fit(S_{trial}) - fit(S_e)}{T_e}}$ {*accept. prob. via Boltzmann factor*}
13: **if** $(rnd < boltz)$ **then**
14: $S_e \leftarrow S_{trial}$ {*move to new solution*}
15: **end if**
16: **end if**
17: $trial \leftarrow trial + 1$
18: **end for**
19: $T_e \leftarrow T_e * \alpha$ {*update current temperature*}
20: $e \leftarrow e + TpT$ {*increase evaluation counter*}
21: **end while**

The Annealing Schedule initialization (4) is used to compute the initial temperature as well as the cooling schedule. This is done using the Δ_i and

Δ_f, user variables that are usually based on the fitness landscape of the problem.

Algorithm 4 initializeAnnealingSchedule(Δ_i,Δ_f)

1: $T_0 \leftarrow \frac{-\Delta_i}{\log(0.5)}$ {*initial temperature*}
2: $T_{min} \leftarrow \frac{-\Delta_f}{\log(0.5)}$ {*minimum temperature*}
3: $\#T_{var} \leftarrow \frac{E}{TpT}$ {*total number of temperature variations*}
4: $\alpha \leftarrow e^{\frac{log(\frac{T_{min}}{T_0})}{\#T_{var}}}$ {*alpha factor*}
5: e: $T_e \leftarrow T_0$ {*current temperature*}
6: **return** $\{\alpha, T_e\}$

5.4.4 Phenotype Evaluation

The phenotype evaluation modules are responsible to decode the candidate solutions into valid constraints of the model, apply the phenotype simulation methods (FBA, MOMA or ROOM) and, from the resulting flux distribution, compute the fitness or fitnesses for the selected objective functions. The evaluation procedure also notifies the Archive Manager that a solution or a population of solutions has been evaluated, allowing this to either keep that solution or execute a series of maintenance procedures (truncation, sorting, storing). This is detailed in algorithms 5 and 6.

Algorithm 5 evaluatePhenotypes(P)

1: **for all** $S \in P$ **do**
2: $F_S \leftarrow$ `evaluateSolution`(S) {*Evaluate the solution S and set its fitness array F_S.*}
3: **end for**
4: `notifyArchive`(P) {*notify the archive manager that a population has been evaluated*}

It should be noted that the archive manager is created separately, regardless of the optimization algorithm (including SPEA2 that uses its own

Algorithm 6 evaluateSolution(S)

Inputs:
  M {*stoichiometric model*}
  OF {*array of objective functions*}
$C \leftarrow S$ {*compute override constraints from solution*}
$M_c \leftarrow apply(C, M)$ {*stoichiometric model w/override constraints*}
$FD \leftarrow$ fluxDistribution(M_c) {*flux distribution computation using one of the available methods: FBA, MOMA, ROOM*}
$F \leftarrow \emptyset$ {*create empty fitness array of size* $|OF|$}
for ($i = 0$; $i < |OF|$; $i \leftarrow i + 1$) **do**
  $objFunc \leftarrow OF_i$
  $F_i \leftarrow$ `objFunc`(FD) {*apply objective function to flux distribution*}
end for
`notifyArchive`(S) {*notify the archive manager that a solution has been evaluated*}
return: F

Algorithm 7 notifyArchive(S)

Inputs:
  A {*the global archive*}
$sub \leftarrow$ existsSubSB(A, S) {*verifies if there exists a subset, better solution than* S *in* A}
if ($\neg sub$) **then**
  $SnB \leftarrow$ existsSuperSNB(A, S)
  remove(SnB, A) {*remove all superset non better solutions from the archive*}
  add(S, A) {*if not, adds the solution to the global archive*}
end if

internal archive). This allows to keep solutions using the dominance properties natural to MO algorithms, even when an SO algorithm is selected. Details are presented in algorithms 7 and 8.

Algorithm 8 notifyArchive(P)

1: **Inputs:**
2: A {*the global archive*}
3: `processArchive`(A)
4: applyTruncation(A) {*Apply truncation based on dominance. See* **??**}
5: sort(A)
6: store(A)

5.5 Implementation and Availability

The framework was fully developed in the `Java` programming language with the exception of the `MetabolicAnalysis` library that was developed in both `Java` and `R` (installation of `R` v3.0.2 or superior is mandatory). OptFlux can be downloaded from www.optflux.org. All the code is readily available in the project sourceforge page at sourceforge.net/projects/optflux/.

5.6 Conclusions

This work proposes an integrated framework that allows its users to configure and fine tune strain optimization strategies tailored to their specific needs. The framework is segmented in two layers, allowing both experienced programmers and regular users to take advantage of the methods developed during this thesis, as well as easily extend them to support new features. The integration of this framework in the open-source OptFlux platform, makes it an attractive resource to an ever increasing ME community.

Future work contemplates the generalization of the optimization layer to support some of the other CSOMs discussed in this thesis (Chapter 2), as well as the improvement of the programming interfaces making it even easier to extend the framework with third-party software.

REFERENCES

[1] I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S. Soares, J. P. Pinto, J. Nielsen, K. R. Patil, E. C. Ferreira, and M. Rocha, "Optflux: an open-source software platform for in silico metabolic engineering," *BMC systems biology*, vol. 4, no. 1, p. 45, 2010.

[2] M. Rocha, P. Maia, R. Mendes, J. P. Pinto, E. C. Ferreira, J. Nielsen, K. R. Patil, and I. Rocha, "Natural computation meta-heuristics for the in silico optimization of microbial strains," *BMC bioinformatics*, vol. 9, no. 1, p. 499, 2008.

[3] A. M. Feist, D. C. Zielinski, J. D. Orth, J. Schellenberger, M. J. Herrgard, and B. Ø. Palsson, "Model-driven evaluation of the production potential for growth-coupled products of¡ i¿ escherichia coli¡/i¿," *Metabolic engineering*, vol. 12, no. 3, pp. 173–186, 2010.

[4] T. Baba, T. Ara, M. Hasegawa, Y. Takai, Y. Okumura, M. Baba, K. A. Datsenko, M. Tomita, B. L. Wanner, and H. Mori, "Construction of escherichia coli k-12 in-frame, single-gene knockout mutants: the keio collection," *Molecular systems biology*, vol. 2, no. 1, 2006.

[5] W.-H. Chen, P. Minguez, M. J. Lercher, and P. Bork, "Ogee: an online gene essentiality database," *Nucleic acids research*, vol. 40, no. D1, pp. D901–D906, 2012.

[6] E. Gonçalves, R. Pereira, I. Rocha, and M. Rocha, "Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression.," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 19, pp. 102–14, Mar. 2012.

[7] K. R. Patil, I. Rocha, J. Förster, and J. Nielsen, "Evolutionary programming as a platform for in silico metabolic engineering," *BMC bioinformatics*, vol. 6, no. 1, p. 308, 2005.

[8] P. Evangelista, P. Maia, and M. Rocha, "Implementing metaheuristic optimization algorithms with jecoli," in *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, pp. 505–510, IEEE, 2009.

[9] E. Zitzler, M. Laumanns, L. Thiele, *et al.*, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *EUROGEN*, pp. 95–100, 2001.

[10] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, *et al.*, "The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[11] T. Pfeiffer, J. Nu, F. Montero, S. Schuster, *et al.*, "Metatool: for studying metabolic networks.," *Bioinformatics*, vol. 15, no. 3, pp. 251–257, 1999.

[12] M. Cvijovic, R. Olivares-Hernández, R. Agren, N. Dahr, W. Vongsangnak, I. Nookaew, K. R. Patil, and J. Nielsen, "Biomet toolbox: genome-wide analysis of metabolism," *Nucleic acids research*, vol. 38, no. suppl 2, pp. W144–W149, 2010.

[13] R. Carreira, P. Evangelista, P. Maia, P. Vilaça, M. Pont, J.-F. Tomb, I. Rocha, and M. Rocha, "Cbfa: phenotype prediction integrating metabolic models with constraints derived from experimental data," *BMC systems biology*, vol. 8, no. 1, p. 123, 2014.

[14] P. Maia, P. VilaÁ? a, I. Rocha, M. Pont, J.-F. o. Tomb, and M. Rocha, "An integrated computational environment for elementary modes analysis of biochemical networks," *International journal of data mining and bioinformatics*, vol. 6, no. 4, pp. 382–395, 2012.

[15] A. Noronha, P. Vilaça, and M. Rocha, "An integrated network visualization framework towards metabolic engineering applications," *BMC bioinformatics*, vol. 15, no. 1, p. 6594, 2014.

[16] D. Glez-Peña, M. Reboiro-Jato, P. Maia, M. Rocha, F. Díaz, and F. Fdez-Riverola, "Aibench: a rapid application development framework for translational research in biomedicine," *Computer methods and programs in biomedicine*, vol. 98, no. 2, pp. 191–203, 2010.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, a summary of the main findings and contributions of this thesis is presented. Some general conclusions are drawn and topics for future work are suggested.

6.1 General Conclusions

The main objective of this thesis was the development of new algorithms and tools for *in silico* optimization of mutant strains for industrial biotechnology applications. The work began by a thorough assessment of the state-of-the-art of the field, performing a comprehensive analysis of the major CSOMs developed in recent years. These methods were classified into three groups, according to their most prominent characteristics, an analysis of their successful applications *in vivo* was devised and their level of adoption by the industry was scrutinized.

We concluded that, despite successful *in vivo* applications, the obtained improvements in yields, productivities and titers are still far away from the ones obtained with non-rational traditional approaches such as random mutagenesis. While we clearly attribute some of these shortcomings to the lack of relevant information in the models, a lack of validation efforts of the current methods was also detected. In fact, since most of the few validation efforts have been reported associated with specific CSOMs, which would be the final layer of an *in silico* approach, it is often impossible to decouple the effects of model predictions from the CSOM results themselves. More studies are thus needed that would allow to the separate the validation of GSMMs, simulation tools and CSOMs. Only to mention an extreme example, the validation of simulation methods such as FBA, MOMA or ROOM in a large scale has never been performed apart from the examples used when the methods have been developed or improved, a gap attributable in part to the scarcity of flux distribution experimental data.

In summary, although improvements are clearly needed in the development of novel CSOMs, that were very scarce until recently, it is probably also important to focus research efforts on the experimental validation of

in silico approaches to foster the adoption of rational tools in industrial biotechnology.

Another limitation that we identified with current CSOMs was that they usually return a single solution to the problem or, in some cases, sets of solutions with limited variability. We began addressing that problem by proposing an evolutionary multi-objective framework for strain optimization (MOStrain). The developed framework was tested for the *in silico* production of succinate, lactate and pyruvate using a recent GSMM of *Escherichia coli* with promising results. One of the main advantages of the proposed method is the ability of specifying multiple objective functions, providing, in a single run, a set of alternative, high-quality, trade-off solutions that can be explored by ME experts.

Leveraging on the developed multi-objective framework, we addressed another common limitation of the current methods, their dependence on the used phenotype prediction method. The majority of the CSOMs, excluding the ones based in pathway analysis, encode some form of phenotype prediction method in their formulations, which in turn were built on top of a given biological assumption. We verified that, for a large percentage of solutions generated by a CSOM using one phenotype prediction method, the results would not hold when simulated with an alternative method. There is still a heated debate concerning the behaviour of a microorganism when subjected to some form of perturbation. Supported by the recent work of Schuetz and co-workers [1], where they claimed that the metabolism of bacteria operates close to the Pareto-optimal surface of a three-dimensional space defined by competing objectives, demonstrating the validity of their claims for various environmental perturbations, we argued that the same principle could be applied to genetic perturbations. We proposed a tandem approach capable

of finding strain designs that comply with the assumptions of multiple phenotype prediction methods, and validated the approach with an array of case studies, using pFBA and LMOMA as phenotype prediction methods.

Finally, we provided all the software and tools developed throughout this work in the form of a well-designed, and flexible framework. This framework reaches both common users with no programming background, able to access a large range of the features via the developed OptFlux plug-in, and experienced programmers which are able to use all the included features, or extend it with new ones.

6.2 Topics for Future Work

The methods developed during this thesis provide a valuable contribution to the state of the art of the area. Notwithstanding, some interesting follow-up alternatives can be explored:

- During the thesis, only strategies based on gene deletions were explored. The application of the developed framework to other types of strain optimization tasks, such as, heterologous gene insertion [2], co-factor specificity modulation [3] and gene over/under expression [4] could be valuable alternatives.

- One of the under-explored aspects of this work concerns the limited range of objective functions and phenotype prediction methods that were effectively explored. In particular the application of the tandem approaches using alternative methods such as ROOM and MiMBl is of particular interest.

- Another area where big developments could be achieved, is in combining the scalability advantages of meta-heuristics with the unbiased characterization of the metabolic space provided by pathway analysis approaches.

- From a software development perspective, an ongoing objective is the generalization of the developed framework to allow the inclusion other types of CSOMS not based on meta-heuristics.

REFERENCES

[1] R. Schuetz, N. Zamboni, M. Zampieri, M. Heinemann, and U. Sauer, "Multidimensional optimality of microbial metabolism," *Science*, vol. 336, no. 6081, pp. 601–604, 2012.

[2] P. Pharkya, A. P. Burgard, and C. D. Maranas, "OptStrain: a computational framework for redesign of microbial production systems," *Genome research*, vol. 14, no. 11, pp. 2367–2376, 2004.

[3] Z. A. King and A. M. Feist, "Optimizing Cofactor Specificity of Oxidoreductase Enzymes for the Generation of Microbial Production Strains—OptSwap," *Industrial Biotechnology*, vol. 9, pp. 236–246, Aug. 2013.

[4] E. Gonçalves, R. Pereira, I. Rocha, and M. Rocha, "Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression.," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 19, pp. 102–14, Mar. 2012.