Winter 12-15-2018

Decoding Complexity in Metabolic Networks using Integrated Mechanistic and Machine Learning Approaches

Tolutola Timothy Oyetunde
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Energy, Environmental and Chemical Engineering

Dissertation Examination Committee:
Yinjie Tang, Chair
Pratim Biswas
Michael Brent
Roman Garnett
Tae Seok Moon

Decoding Complexity in Metabolic Networks using Integrated Mechanistic and Machine Learning Approaches

by

Tolutola Oyetunde

A dissertation presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

December, 2018
St. Louis, Missouri

Table of Contents

List of Figures vi

List of Tables ix

Acknowledgments x

Abstract xiii

Chapter 1: Introduction 1

1.1 Introduction to computational biology and machine learning 1

1.1.1 Computational modeling in biology 1

1.1.2 What is machine learning? 3

1.1.3 Machine learning in computational biology 4

1.2 Computational strain design 5

1.2.1 Basics of computational strain design 5

1.2.2 Challenges of computational strain design 7

1.3 Machine learning for computational strain design 9

1.3.1 Databases for metabolic engineering design 10

1.3.2 Practical applications of machine learning in metabolic engineering 12

1.4 Perspectives on applying machine learning in computational strain design 16

1.5 Hindrances and possible solutions to successful application of machine learning 20

Chapter 2: Refining genome-scale metabolic network reconstructions 23

2.1 Introduction 23

2.2 Methods 24

2.2.1 Step A: Conversion of incomplete stoichiometric matrix to metabolite adjacency matrix 26

2.2.2. Step B: Completion of metabolite adjacency matrix using matrix factorization 26

2.2.3 Step C: Prediction of new reactions from a universal reaction set 26

2.2.4 Modes of running BoostGAPFILL 27

2.3 Results and discussion 28

Chapter 3: Data-driven computational strain design 35

3.1 Introduction 35

3.2 Methodology 38

3.2.1 Database curation 38

3.2.2 Constraint-based simulations 40

3.2.3 Data pre-processing and augmentation 43

3.2.4 Ensemble learning and hyperparameter tuning 43

3.3 Results and discussion 44

3.3.1 Description of curated database 44

3.3.2 Identification of critical metabolic engineering factors 46

3.3.3 Model performance validation 50

3.3.4 Model improvement 53

Chapter 4: Thermodynamic framework for mutant phenotype prediction 56

4.1 Introduction 56

4.2 Methodology 61
4.2.1 Mathematical formulation of REMEP 61
4.2.2 Description of computational experiments 65
4.3 Results 65
4.3.1 E. coli mutants 65
4.3.3 S. cerevisiae mutants 68
4.4 Discussion 69
Chapter 5: Conclusions 73
5.1 Gap filling of metabolic networks 73
5.2 Data-driven computational strain design 74
5.3 Thermodynamic framework for mutant phenotype predictions 75
5.4 Monsanto co-op experience 76
5.5 Recommendations for future work 77
5.4.1 Automatic knowledge extraction from metabolic engineering literature 77
5.4.2 Multi-omics data integration in a thermodynamic framework 77
5.4.3 Machine learning techniques for 'small' data 77
5.6 Publications and conference presentations 78
5.6.1 Publications 78
5.6.2 Conference presentations 78
References 80
Appendix 100

Appendix A: Mathematical formulation of the metabolic network refinement problem in BoostGAPFILL 100

Appendix B: Technical implementation details and limitations of BoostGapFill 103

B.1 Stochasticity of algorithm 103

B.2 Prediction of reactions with new metabolites 103

B.3 Options available 104

B.4 Timing 104

B. 5 Notes on the computational methodology 104

B.6 Running BoostGapFill 105

CV 106

List of Figures

Figure 1.1 Basic classification of machine learning algorithms 3

Figure 1.2 Applications of machine learning in molecular systems biology 5

Figure 1.3 Pathway-level strain design strategies: Lycopene production case study 6

Figure 1.4 Basic schematic of microbial metabolism and strain design showing the interplay between carbon and energy processes subject to regulation/influential factors (highlighted in yellow boxes). 10

Figure 1.5 Rapid increase of metabolic engineering data 11

Figure 1.6 Paradigms of data-driven techniques in systems metabolic engineering 18

Figure 1.7 Common biosynthesis pathways from the central metabolic network 22

Figure 2.1 Basic concepts of the pattern-based module of BoostGAPFILL 25

Figure 2.2 Modes of using BoostGapFill 28

Figure 2.3 Comparison of the performance of gap-filling algorithms on E.coli model iAF1260 30

Figure 2.4 Simulation of artificial gaps 31

Figure 2.5 Effect of added reactions on number of gaps 31

Figure 2.6 Comparison of BoostGapFill and FastGapFill across different metabolic network reconstructions 33

Figure 2.7 BoostGapFill with newMet option set to ‘true’ 34

Figure 3.1 Database curation and feature extraction methodology 37

Figure 3.2 Feature additions via genome scale model simulations and data augmentation based on case studies described in the literatures 38

Figure 3.3 Machine learning pipeline. Ensemble learning using stacked regressors. 41

Figure 3.4 Summary of curated database showing distribution of titers (units in g/L) for 25 different products from the bacterium E. coli. 45

Figure 3.5 Comparison of production metrics (titer, rate, and yield) 46

Figure 3.6 Inferring possible influential factors on metabolic engineering design performance . 49

Figure 3.7 Prediction of production metrics TRY 51

Figure 3.8 Prediction of production metrics (titer, yield and rate) 52

Figure 3.9 Model performance analyses 53

Figure 3.10 Titer learning curve as the function of size of training data set 54

Figure 3.11 Rate learning curve 54

Figure 3.12 Yield learning curve 55

Figure 4.1 Comparison of flux-based to metabolite based mutant prediction algorithms 58

Figure 4.2 Comparing the effects of genetic and environmental perturbations on gene expression, metabolite concentrations and intracellular flux 60

Figure 4.3 The REMEP algorithm workflow 63

Figure 4.4 REMEP's two-step iterative solution procedure 64

Figure 4.5 Predicted changes in *E. coli*'s central metabolism upon knockout of pgi gene 66

Figure 4.6 Comparison of different phenotype prediction algorithms on *E. coli* mutant strains . 68

Figure 4.7 Predicting the effect of changing carbon sources 68

Figure 4.8 Beanplot comparison of RELATCH and REMEP on S. cerevisiae mutant strains 69

Figure 4.9 Basic physicochemical principles constraining key players in cellular metabolism... 71

Figure 4.10 Heat Map showing percentage change in selected reactions of central metabolism. 72

List of Tables

Table 1.1 Application of data-driven techniques in metabolic engineering 13

Acknowledgments

I would like to acknowledge the invaluable contributions of the following people to the success of my studies:

My research mentor, Dr. Yinjie Tang, - for his kind support and guidance over these past four years. Dr. Tang invests considerable time, money and effort in the personal and professional development of his students. I have truly enjoyed working with him and look forward to future collaborations.

My thesis committee members, Dr. Pratim Biswas, Dr. Michael Brent, Dr. Tae Seok Moon, and Dr. Roman Garnett, - for taking out time from their extremely busy schedules to serve on my committee.

Tang lab members, Gang Wu, Lian He, Le You, Whitney Hollinshead, Ni Wan, Mary Abernathy, Jeffrey Czajka, and Garrett Roell, - past and present for creating a friendly and cooperative atmosphere in the lab.

My research collaborators, Dr. Cynthia Lo, Dr. Hector Garcia, Dr. Yixin Chen and Dr. Forrest Bao - for all the technical support and guidance.

To the awesome cohort of 2014, Apoorva, Yeunook, Philip, Claire, Deanna, Nathan, Merima, Cheyenne, Jiaman, Zhichao, Jiayu, John, and Wei. You guys are the best!

The members of the CAML lab, Sungyoon, Alireza, Ahmed, Wei, and Shane – friends indeed!

The African community in St Louis, Wale, Funmi, Zion, Daniella, Omotunde, Ileri, Tatenda and Onochie, - Thanks for making St Louis a home away from home!

Pastors of Crosspoint Church and members of my bible study group, Shawn Craig, Josiah Serra, David Verret, Wyatt Trafton, Steve Shorette, Gary Tilley, Jennifer Powers, Sir Terry Carson,

Rene Parker, Bob and Cathleen Westwood, Elizabeth Jones, and Tana Murphy. I wish I could move all of you to Minnesota with me!

My lovely wife, Toyo and gracious daughter, JoGrace for putting up with me. You guys rock my world! My parents, Timothy and Victoria Oyetunde and siblings, Tomi, Tobi and Fajulo for all the love, prayers and support. Most importantly, I would like to thank God for seeing me through the program and giving me a reason to live.

I would also like to acknowledge research funding from the National Science Foundation (NSF MCB 1616619 Productivity Prediction of Microbial Cell Factories using Machine learning and Knowledge Engineering) and the Department of Energy (DOE DESC 0018324 Systems Engineering of Rhodococcus opacus to Enable Production of Drop-in Fuels from Lignocellulose)

Tolutola Oyetunde

Washington University in St. Louis

December 2018

Dedicated to my two favorite ladies: Toyo and JoGrace

Abstract

Decoding complexity in metabolic networks using integrated mechanistic and machine learning approaches

by

Tolutola Oyetunde

Doctor of Philosophy in Energy, Environmental and Chemical Engineering

Washington University in St. Louis, 2018

Dr. Yinjie Tang, Chair

How can we get living cells to do what we want? What do they actually 'want'? What 'rules' do they observe? How can we better understand and manipulate them? Answers to fundamental research questions like these are critical to overcoming bottlenecks in metabolic engineering and optimizing heterologous pathways for synthetic biology applications. Unfortunately, biological systems are too complex to be completely described by physicochemical modeling alone.

In this research, I developed and applied integrated mechanistic and data-driven frameworks to help uncover the mysteries of cellular regulation and control. These tools provide a computational framework for seeking answers to pertinent biological questions. Four major tasks were accomplished.

First, I developed innovative tools for key areas in the genome-to-phenome mapping pipeline. An efficient gap filling algorithm (called BoostGAPFILL) that integrates mechanistic and machine learning techniques was developed for the refinement of genome-scale metabolic network reconstructions. Genome-scale metabolic network reconstructions are finding ever

increasing applications in metabolic engineering for industrial, medical and environmental purposes.

Second, I designed a thermodynamics-based framework (called REMEP) for mutant phenotype prediction (integrating metabolomics, fluxomics and thermodynamics data). These tools will go a long way in improving the fidelity of model predictions of microbial cell factories.

Third, I designed a data-driven framework for characterizing and predicting the effectiveness of metabolic engineering strategies. This involved building a knowledgebase of historical microbial cell factory performance from published literature. Advanced machine learning concepts, such as ensemble learning and data augmentation, were employed in combination with standard mechanistic models to develop a predictive platform for important industrial biotechnology metrics such as yield, titer, and productivity.

Fourth, my modeling tools and skills have been used for case studies on fungal lipid metabolism analyses, *E. coli* resource allocation balances, reconstruction of the genome scale metabolic network for a non-model species, *R. opacus*, as well as the rapid prediction of bacterial heterotrophic fluxomics.

In the long run, this integrated modeling approach will significantly shorten the “design-build-test-learn” cycle of metabolic engineering, as well as provide a platform for biological discovery.

Chapter 1:Introduction[1]

In this chapter, I present a bird's eye view of biological modeling and machine learning. I also discuss computational strain design as one key area for integrating machine learning and mechanistic biological modeling. All of the subsequent chapters discuss my contributions to critical aspects of computational strain design.

1.1 Introduction to computational biology and machine learning

1.1.1 Computational modeling in biology

Computational modeling has become more and more important in recent years as molecular biology transitioned from reductionist to a systems approach[1]. Moreover, the breakthroughs in high throughput technologies has provided huge datasets that in principle allow the behavior of integrated cellular systems to be observed in detail. This provides an incentive to develop models and modeling techniques to better understand and predict these systems. It also potentially enables the realization of the promise of molecular systems biology – the ability to understand cells and their functions from the knowledge of the individual molecules that make up the cells.

For practical applications in environmental remediation, industrial biotechnology, and medicine, a lot of efforts have been focused on understanding metabolic networks – the network of

[1] This chapter is adapted from my publication: Oyetunde, T., Bao, F. S., Chen, J. W., Martin, H. G., & Tang, Y. J. (2018). Leveraging knowledge engineering and machine learning for microbial bio-manufacturing. **Biotechnology advances.**

reactions by which cells utilize substrates for growth. This has resulted in a suite of phenomenological modeling techniques largely based on physicochemical principles. Models 'document' biological information and allow for the generation of testable predictions. Modeling also can provide a platform for rational redesigning of cellular metabolism towards desired ends and help overcome problems in scale-up of metabolic engineering designs.

Unfortunately, the use of mathematical models in systems biology is not without its challenges[2]. Metabolic models become increasingly complicated as we try to account for more observed phenomena from other biological processes. Other issues include standardization of modeling techniques to ensure reusability and sharing as well as integration of heterogeneous, spare and often noisy datasets. There is also a growing interest to exploit the wealth of experimental biological information using machine learning techniques[3].

1.1.2 What is machine learning?

ML is a branch of artificial intelligence that train computers to perform tasks by gaining the capability from 'experience' (data) rather than being specifically programmed to do so. ML studies are broadly classified into supervised, unsupervised and reinforced learning. In supervised learning, the computer develops an input-output model from sets of inputs and 'correct' (i.e., labeled) outputs. In unsupervised learning (e.g., cluster analysis), hidden patterns and structures can be uncovered from the data. ML has many varied real-life applications such as finance, personalized medicine, computer vision, and energy forecasting [4], [5]. Figure 1 provides the basic classification of machine learning algorithms and their applications.

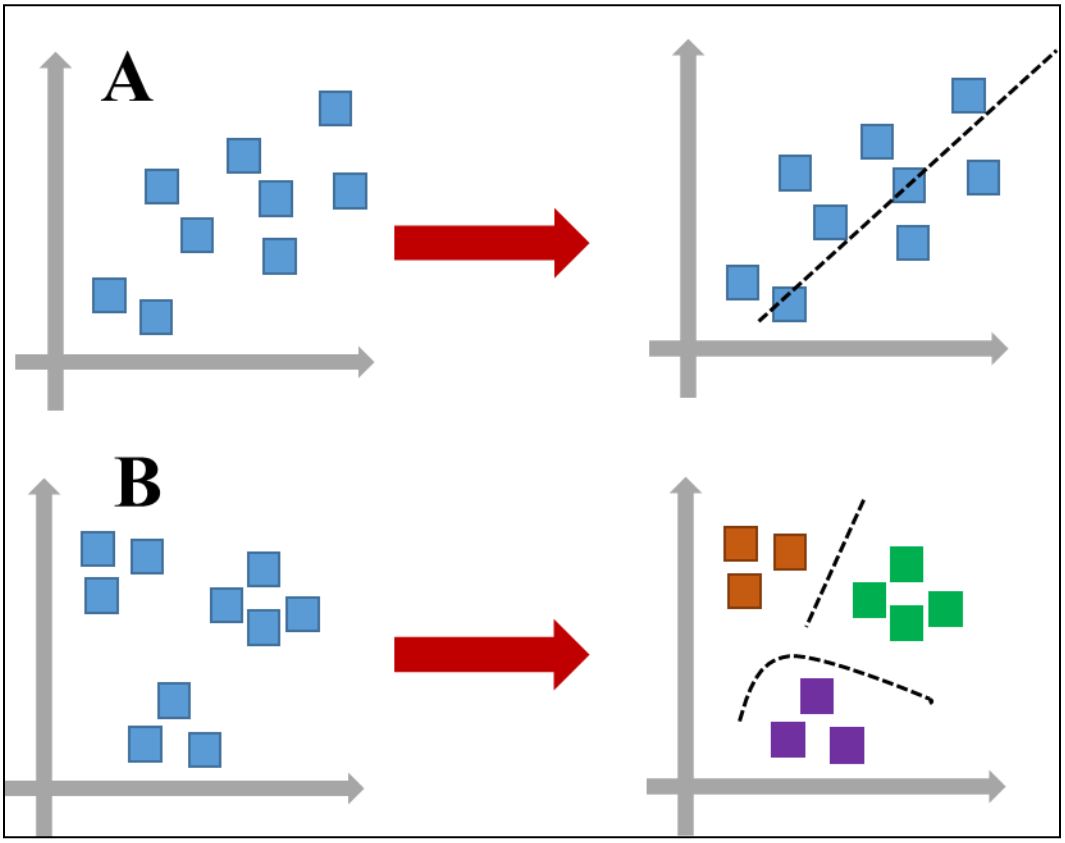


Figure 1.1 Basic classification of machine learning algorithms **A. Supervised learning**. The program 'learns' from a set of training examples. The output of supervised learning is a quantitative description of the relationship between variables in the data. Supervised learning algorithms can be grouped into two (1) Classification algorithms that

predict discrete responses e.g. support vector machines, naive Bayes, Nearest Neighbor and discriminant analysis. Applications include medical imaging and speech recognition (2) Regression algorithms predict continuous responses e.g. linear regression, ensemble methods, decision tress and neural networks. Typical applications include electrical load forecasting and computational finance. **B. Unsupervised learning** aims to find intrinsic structures in data without labeled responses. The major unsupervised learning method is clustering. Example algorithms include K-means, Fuzzy C-means, hierarchical clustering, Gaussian mixture, neural networks and hidden markov models. Typical applications include object recognition and genomic analysis. (adapted from Andrew Ng's machine learning course)

1.1.3 Machine learning in computational biology

Machine learning techniques have gained widespread use in computational biology [3], [6], [7]. Traditional applications include discovery and analysis of gene and protein networks and the identification of functionally important sites in proteins and protein function prediction, to name a few. Recent applications include cancer diagnosis, personalized medicine, cell imaging analysis, and pharmacogenomics. Deep learning is one of the fastest growing fields of machine learning and is finding increasing applications in image analysis and regulatory genomics [8]. Figure 1.2 gives an overview of typical applications of machine learning in computational biology.

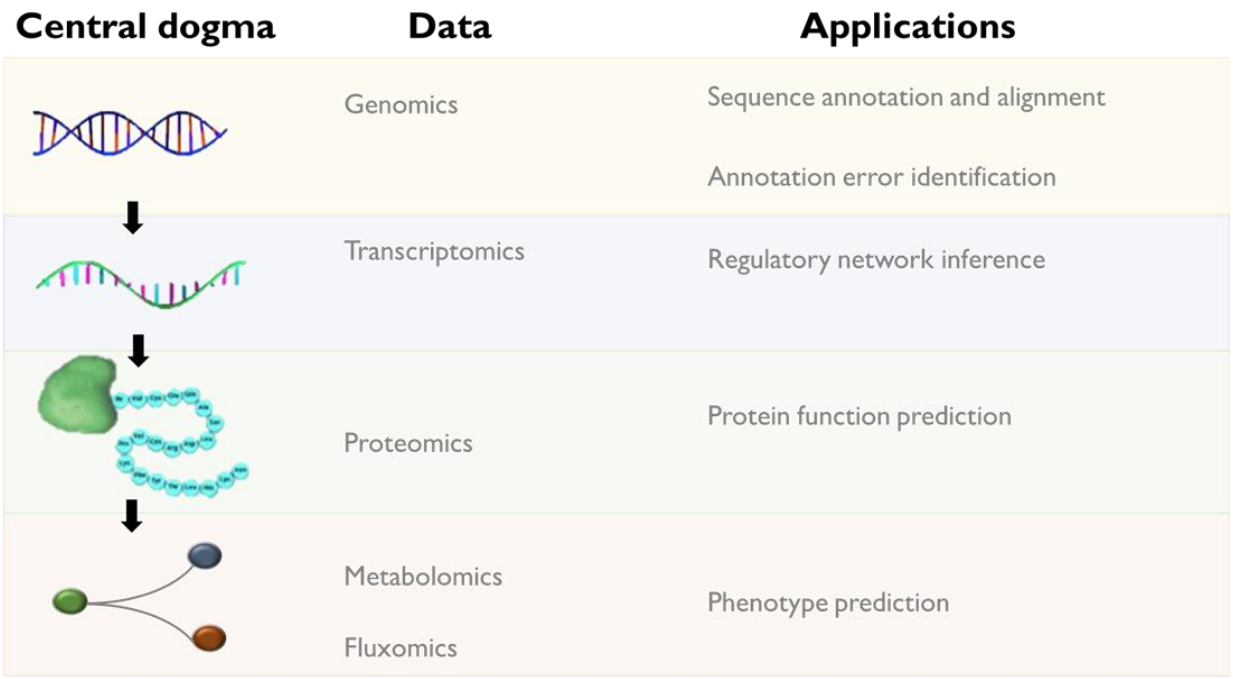


Figure 1.2 Applications of machine learning in molecular systems biology

1.2 Computational strain design

1.2.1 Basics of computational strain design

Strain design requires identifications of genetic strategies to hijack cell metabolism for useful ends. In the past, strain improvement was achieved via random mutation strategies or overexpression of a single biosynthesis gene. With the advance of genome sequencing and synthetic biology technologies, targeted modifications of multiple genes or pathways have become commonplace in order to redirect carbon and energy flows to desired products [9], [10]. 3PB principles (PULL, POWER, PUSH AND BLOCK) have been widely used to manipulate cell performance (Fig. 1.3). For example, common strategies to optimize the yeast strain for the *de novo* production of lycopene include PUSH (increase the supply of the precursor cytoplasmic acetyl-CoA), PULL (improve enzyme activities for lycopene synthesis), POWER (enhance ATP generation and NAD(P)H balances), and BLOCK (inhibit competing pathways) steps.

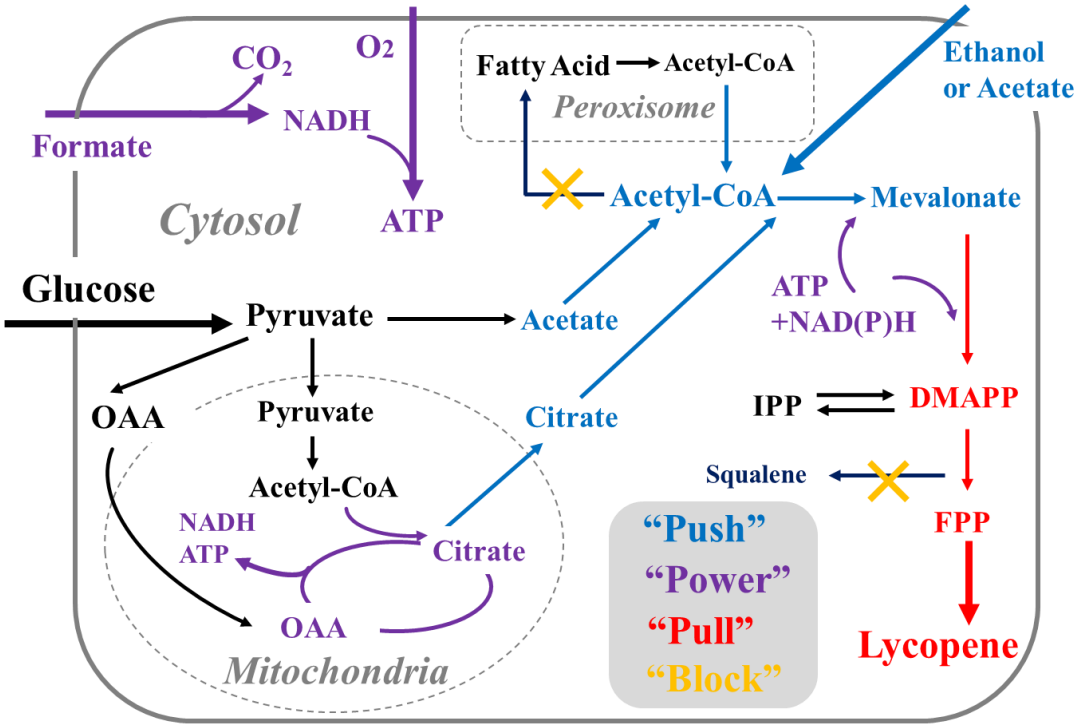


Figure 1.3 Pathway-level strain design strategies: Lycopene production case study To improve production in yeast, several modifications are required including: (1) "Push" carbon flows towards the acetyl-CoA precursor, in which several acetyl-CoA routes (including acetyl-CoA synthase and citrate lyase reactions). (2) "Pull" carbon flow towards lycopene (i.e., overexpress mevalonate pathways). (3) "Block" fluxes competing for mevalonate pathways (e.g., lipid synthesis); (4) "Power" cell metabolism by engineering redox cofactor balances and promoting ATP production (i.e., increase oxidative phosphorylation).

3PB strategies are not always effective because fluxes re-organization may induce new bottlenecks in upstream pathways. To achieve commercial yields/titers/rates, genome wide pathway modifications must be performed after creation of proof-of-concept laboratory strains. In this context, GSMs become commonly used tools to predict mutant physiologies and search possible gene targets through entire metabolic network. GSMs estimate cell growth and product secretion rates using constraint-based reconstruction and analyses (COBRA), in which complex biological processes are inherently constrained by steady-state mass balances and physical/chemical laws (e.g., thermodynamic constraints)[11]. Such underdetermined systems

are solved by objective functions [12]. For example, biomass growth objective has shown decent accuracy to describe cultures in carbon limited conditions [13]. New COBRA tools leverage omics, kinetic and thermodynamic information to improve metabolic insights [14]. Particularly, COBRA combined with transcriptomics data has shown successes to predict strain performance based on the relationship between gene profiles and the fluxome (e.g. TFBA[15], GIMME [16], iMAT [17], ME-Models [18], and E-FLUX [19]). Using gene data from high throughput sequencing technique, GSMs can not only narrow flux intervals and reduce bias/uncertainty, but also identify genes that likely regulate microbial fluxes [20]. In general, computer strain design via GSMs has one or more of the following layers (1) an algorithm for predicting intracellular reaction rates (fluxes), (2) an algorithm for selecting appropriate target reaction(s) and pathway(s), and (3) an algorithm for determining the type of modification to be performed on the selected target reactions(s). For example, k-OPTFORCE [21] determines the minimum number of interventions required to increase a specified flux through desired reaction(s).

1.2.2 Challenges of computational strain design

Informed by high-throughput technologies, the behavior of integrated cellular systems can be observed, and a better understanding of inner workings of cellular regulation has been obtained. Despite this progress, the practical utility of CSD tools has been demonstrated only in specific cases. In practical terms, increasing flux through a reaction is much more complicated to achieve than decreasing or eliminating it. This is due to several reasons. First**,** microbial catabolism/anabolism typically displays innate regulations that limit the effectiveness of metabolic re-programming by synthetic biology. Moreover, the cell needs consumption of energy molecules (e.g., NAD(P)H and ATP) and building blocks (e.g., amino acids) to construct engineered components (e.g., enzymes and plasmids), and it is difficult to estimate the

carbon/energy burdens from each synthetic biology components. Besides, we do not know the ATP maintenance cost in producer strains under stressed cultivations conditions [22]. Second, the performance of engineered hosts is often unstable in bioreactor conditions due to genetic mutations and non-genetic cell-to-cell variations. Cell behavior or genetic stability is closely related to nutrient supplies, growth conditions, and fermentation duration. Because of complex influential factors, metabolic engineering strains have poor reproducibility from study to study which is difficult to capture in a modeling framework. Third, there are many unknown mechanisms (e.g., transcriptional or allosteric regulations) that control cell flux organizations. Even the order of genes in a pathway may change productivity of a heterologous pathway due to unknown expression balance of cascade enzymes [23]. Besides, innate enzymes may employ channeling (i.e., co-localize cascade enzymes to shuttle metabolites more effectively than if enzymes were randomly distributed) to overcome diffusion barriers and protect intermediate from competing pathways [24]. However, the enzyme proximity effects on intracellular metabolic fluxes are highly controversial.

Although there have been attempts to use transcriptional or proteomic data for improving GSM, omics data are still considered insufficient to fully determine metabolic outcomes [25]. For example, while it is recognized that transcript levels affect fluxes in combination with metabolite concentrations [26], [27], a mechanistic prediction of fluxes based on metabolite concentrations or enzyme abundance is still inaccessible for the majority of metabolic reactions [27], [28]. In general, strain development requires overexpression or modification of numerous genetic targets. Modeling the effect of these intervention genetic inputs and their nontrivial interactions/tradeoffs on cellular metabolism as a whole presents a formidable challenge.

1.3 Machine learning for computational strain design

Unlike typical models encoding fundamental laws (such as mass and energy balances), data driven algorithms (machine learning, ML) make predictions by deriving patterns from training sets comprising large amounts of experimental data. Since these models are black boxes deriving predictive capabilities purely from experimental data, simulations do not require a complete mechanistic understanding of cell physiologies. Data mining and ML techniques can leverage complex fermentation data and omics results for highlighting scenarios (such as different promoter strengths and induction characteristics) that may maximally yield metabolic outputs [29]–[31]. Moreover, with rapid increase of published metabolic engineering studies and recent advances in artificial intelligence research, the use of data driven approaches may facilitate the understanding of cellular processes and assist mechanistic modeling for quality CSD.

Currently, genomics data at different cellular levels are still insufficient to determine holistic metabolic regulations [25]. While transcript levels affect fluxes in combination with metabolite concentrations, prediction of fluxes based on metabolite concentrations or enzyme abundance is still inaccessible for the majority of metabolic reactions [27], [28]. Due to these limitations, data driven approaches may be used in conjunction to mechanistic models to simulate complex cellular behavior by transforming both accountable and unaccountable influential variables (Figure 1.4).

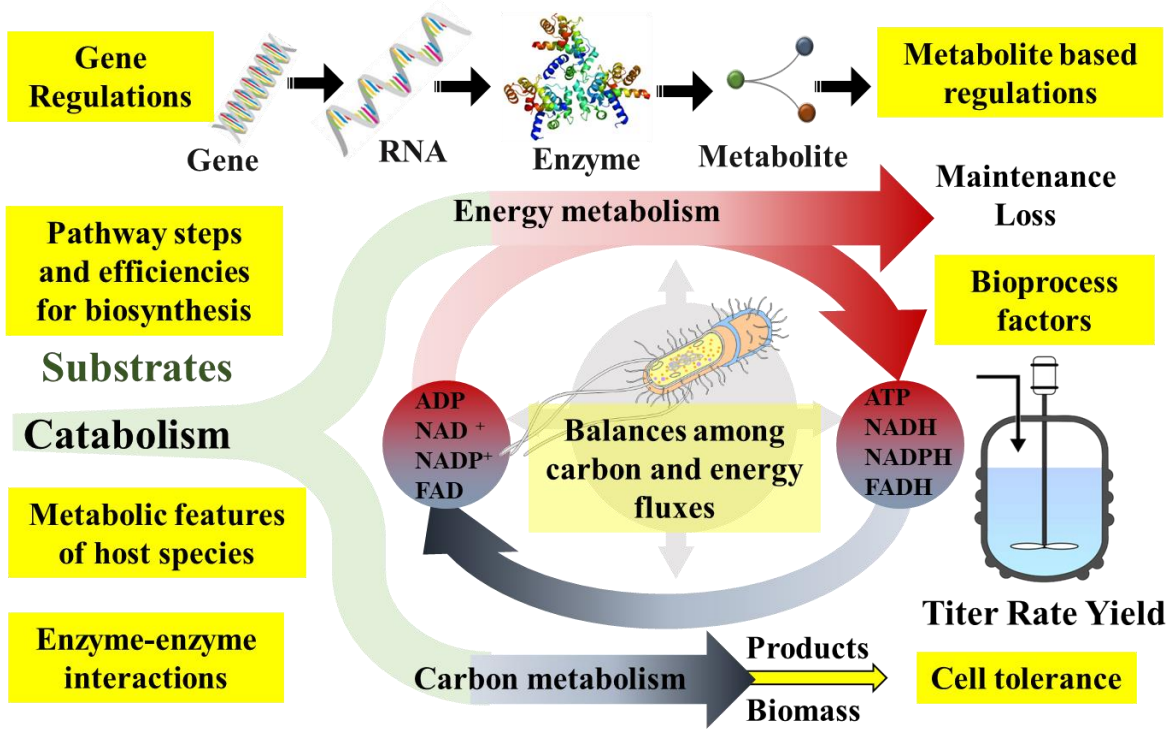
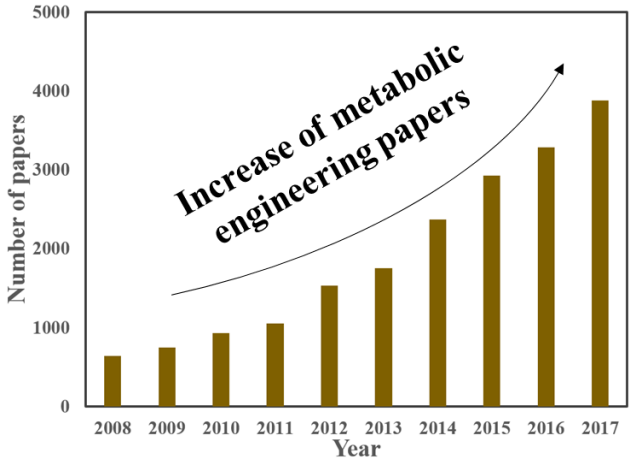


Figure 1.4 Basic schematic of microbial metabolism and strain design showing the interplay between carbon and energy processes subject to regulation/influential factors (highlighted in yellow boxes).

1.3.1 Databases for metabolic engineering design

Rapid growth of synthetic biology in the past decade has generated a large amount of literature and experimental databases (Fig. 1.5). However, every case study uses different conditions and the number of variables is very large.



Number of papers on hosts*		Number of papers on tools**	
E. coli	~3500	Genome engineering	~40300
Saccharomyces	~1750	Protein scaffold	~27000
Bacillus	~570	Directed evolution	~18000
Pseudomonas	~550	Plasmid engineering	~11200
Clostridium	~500	Promoter engineering	~9500
Corynebacterium	~420	Crispr	~8100
Synechocystis or Synechococcus	~300	Gene knockout	~7200
Pichia	~230	Efflux pump	~6900
Chinese hamster ovary	~170	Codon optimization	~1000
Yarrowia	~130	Dynamic control & flux	~450

Figure 1.5 Rapid increase of metabolic engineering data Information was based on PubMed search on Jan 25, 2018.

In the ML field, better organized data always trump better algorithms. Thus, it is necessary to standardize the datasets and build databases by extracting and clustering published data (i.e., Knowledge Engineering) [32], [33]. To date, there are many databases that focus on documenting known knowledge about cellular networks (genomic, transcriptomic, metabolic, and regulatory networks) and the interactions between them [34]. These include KEGG [35], [36], BiGG [37], Rhea [38], CecaFDB [39], MetaCyc [40], and BioCyc [41]. While such databases can potentially provide considerable insight into cellular metabolism and its regulation, they have limitations since they do not contain information about performance of engineered strains (yield, titer, and production rate) nor parameters related to bioprocess conditions (such as reactor configurations and growth medium). Recently, a number of efforts have focused on curating experimental metabolic information from published literature (Winkler et al., 2015; Wu et al., 2016). Frameworks like Experimental Data Depot (EDD) [44] , LASER[42], and OMERO [45] have been developed to standardize documentation and integration of biological experimental information. Frameworks for specific microorganisms have also been developed [46]. These frameworks also enable basic data visualization as well as a suite of tools for data

manipulation/analyses. Other frameworks like KBase [47] focus on integrating not only data but computational methods for enhanced predictive fidelity of biological functions.

Knowledge databases will benefit data standardization and pave the way for artificial intelligence to boost CSD and automation of strain development. Detailed information (including fermentation process variables, omics data, genetic tools or components) is valuable for ML to make predictions. Frameworks like LASER and EDD provide templates for such information to be gathered and standardized. Typical mechanistic models need to simplify complex biological systems, while ML can estimate strain physiological responses under diverse bioprocess (such as nutrients and bio-reactor modes) and genetic factors (e.g., metabolic burdens from gene overexpression or other synthetic biology parts) without understanding cellular processes. Particularly, the deep learning (DL), a recent powerful class of ML techniques, capable of handling massive datasets and mining complicated patterns hidden in data, will prove useful towards this end [8]. Nonetheless, DL algorithms require much larger amounts of quality data than traditional ML approaches, which can be practical only after significant progresses in knowledge engineering.

1.3.2 Practical applications of machine learning in metabolic engineering

Both bioprocessing and systems biology have widely employed ML, which can play an important role in design-build-test-learn cycle for strain improvement and fermentation optimizations. Table 1 gives published ML applications to predict metabolic outcomes. Most of these applications follow a similar workflow: (1) identification of output variables (like yield, titer, or rate); (2) iterative feature selection to identify input factors that are most influential on performance metrics; (3) model selection depending on data availability; and (4) model training

and validation. Data driven model provide complementary information to GSM. The later focuses on predicting biosynthesis yields, while production rates and titers are determined by the synergistic impact of product yields, bioprocesses, strain tolerance, and biomass growth. ML could take into account the genetic design of the microbial host system and the "suboptimal" conditions under which the fermentation process occurs. The hybrid of ML-GSM may identify effective metabolic strategies or targets and qualitatively benchmark various performances of engineered production platforms.

Table 1.1 Application of data-driven techniques in metabolic engineering

ML technique	**Application**	**Comment**	**Ref**
Neural networks	Improve the yield of target protein	Used NN technique to build predictive model from experimental results and stochastic sampling. Discovered experimental conditions that give ~350% improvement of yield	[48]
Naïve Bayes, kNN, decision trees, logistic regression	Metabolic pathway prediction	The ML methods performed as well as the well-designed and refined algorithm (PathoLogic). Besides, ML methods have the advantage of easily adding new features to test and further optimize the performance.	[49]
Multiple kernel learning, transfer learning	Predicting protein interactions in fungal secretion pathways	They predicted the protein-protein interaction in the cross-species *T. reesei* by the learning features obtained from from *S. cerevisiae*.	[50]

SVM + transfer learning	Predict the matrix metalloprotease(MMPs) substrate cleavage sites	They learn the knowledge from the source domain (MMP-9 and MMP-12) to improve the prediction of cleavage sites of other MMPs (MMP-2, -3, -7, and -8) in the target domain.	[51]
Neural networks	Use NN to investigate the effect of process condition (e.g. time, temperature, pH, etc.) on xylitol production	In this study, a multilayer perceptron (MLP) based feed forward neural network model with Levenberg-Marquardt back propagation (BP-MLP) algorithm was trained with 339 experimental data points. The model could predict the optimal harvest time in xylitol production.	[52]
Neural networks	Optimize the fermentation process of cyclodextrin glycosyltransferase production.	They first found the key influential factors using Plackett-Burman Design (PBD) and then optimized by NN. The NN contains one hidden layer.	[53]
Neural networks	Optimization of fermentation parameters of rapamycin production by *Streptomyces hygroscopicus* NRRL 5491	The authors applied Plackett–Burman design (PBD) method, artificial neural networks (ANN), and genetic algorithms (GA). The ANN was used to further optimize the key factors found in PBD method.	[54]
SVM, Neural networks	Predict the yield of glutamic acid from fermentation process parameters (pH, temperature, carbon source concentration, aeration)	They choose SVM method because it is suitable for small datasets (which is usually the case for production data). They also determined that SVM was more accurate in predicting yield than NN.	[55]
Gaussian process model,	Estimate the probability of a given enzyme to catalyze a given reaction	The authors created a semi-supervised Gaussian model to predict if a given enzyme is able to catalyze the desired	[56]

SVM		reaction. Furthermore, the Michaelis constant was also predicted by Gaussian progress regression to quantify the affinity between enzyme and the reaction. The results shows the ML can be a powerful tool to speed up the application of synthetic biology.	
Decision tree	Develop a data-driven model to accurately design CRISPR-based transcription regulator.	The authors used pairwise datasets of guideRNAs and gene expression to build a predictive model	[57]
SVM	Predict the essential genes in *E. coli* metabolism	The authors proposed a strategy of data curation and feature selection to improve the performance of SVM model. Instead of performing flux balance analysis, which are condition specific, to obtain flux features, they applied flux coupling analysis to get the higher sensitivity and specificity of the model.	[58]
PCA	Identify specific enzymes that limiting the production of target molecules in a pathway	Based on the PCA distribution, they manipulated the gene expression level of mevalonate pathway enzymes in *E. coli* to improve the production of limonene up to 40%.	[59]

A recent work used traditional supervised learning methods to predict bacterial central metabolism[43]. In that study, experimental data of 37 bacteria species from over 100 ^{13}C-MFA papers were extracted and converted into structured data. Three supervised algorithms, Support Vector Machine (SVM), k-Nearest Neighbors (kNN), and Decision Tree were employed to train regressors to predict fluxes using features (substrate types, genetic modifications, and cultivation methods). ML results can generate reasonable flux boundaries for FBA models and reduce solution space. ML has also been employed for a priori estimation of chemical productivity from

engineered *E. coli* and *S. cerevisiae*, given a set of model inputs (biosynthesis steps, nutrient supplementation, bioreactor modes) [60], [61]. Such models via linear regressions correctly predict that the product synthesis using long pathways unavoidably gives poor production yield and titer. ML models are useful for manufacturers to decide whether a product should be produced via engineered microbial cell factories or via a chemical synthesis route. Moreover, ML can help improve the fidelity of metabolic network reconstructions used for genome scale modeling [62].

1.4 Perspectives on applying machine learning in computational strain design

Figure 1.6 shows possible paradigms for utilizing data-driven techniques in systems metabolic engineering. The earlier applications of ML in fermentation processes usually involved data from bioprocess studies. These studies aim to link influential factors (e.g., bioreactor conditions) to cell productivity via linear/nonlinear regressions or neural network (paradigm #1). Most of the applications listed in Table 1.1 are of this kind. The advantage of this scheme is that the data formats of inputs/outputs are relatively simple (usually from one set of study). Because the dataset size is usually small, model scope is fairly limited. Another type of efforts has sought to decode complexity in cellular networks by using omics dataset as well as details of synthetic biology constructs (paradigm #2). These frameworks learn system behaviors at different regulation layers and decode key genes that control desired cellular functions, which enable design-build-test-learn cycle during strain improvements [63]. They can also improve the fidelity of metabolic network reconstructions used for genome scale modeling [62]. A limitation of such

frameworks is that they do not usually consider the bioprocess conditions or engineering strategies. Researchers may potentially combine the benefits of the first two paradigms. Via knowledge engineering to generate a database that contains structured input (species, nutrient, culture conditions, genetic tools, strain tolerance and stability) and outputs (yield/rate/titer), ML can capture microbial physiologies in response to various genetic and fermentation conditions. For example, ML models were developed for a priori estimation of chemical productivity from engineered *E. coli* and *S. cerevisiae*, given a set of model inputs (e.g., biosynthesis steps, nutrient supplementation, bioreactor modes) [60], [61]. Such models via linear regressions correctly predict that the product synthesis using long pathways unavoidably gives poor production yield and titer. These models are useful for manufacturers to decide whether a product should be produced via engineered microbial cell factories or via a chemical synthesis route.

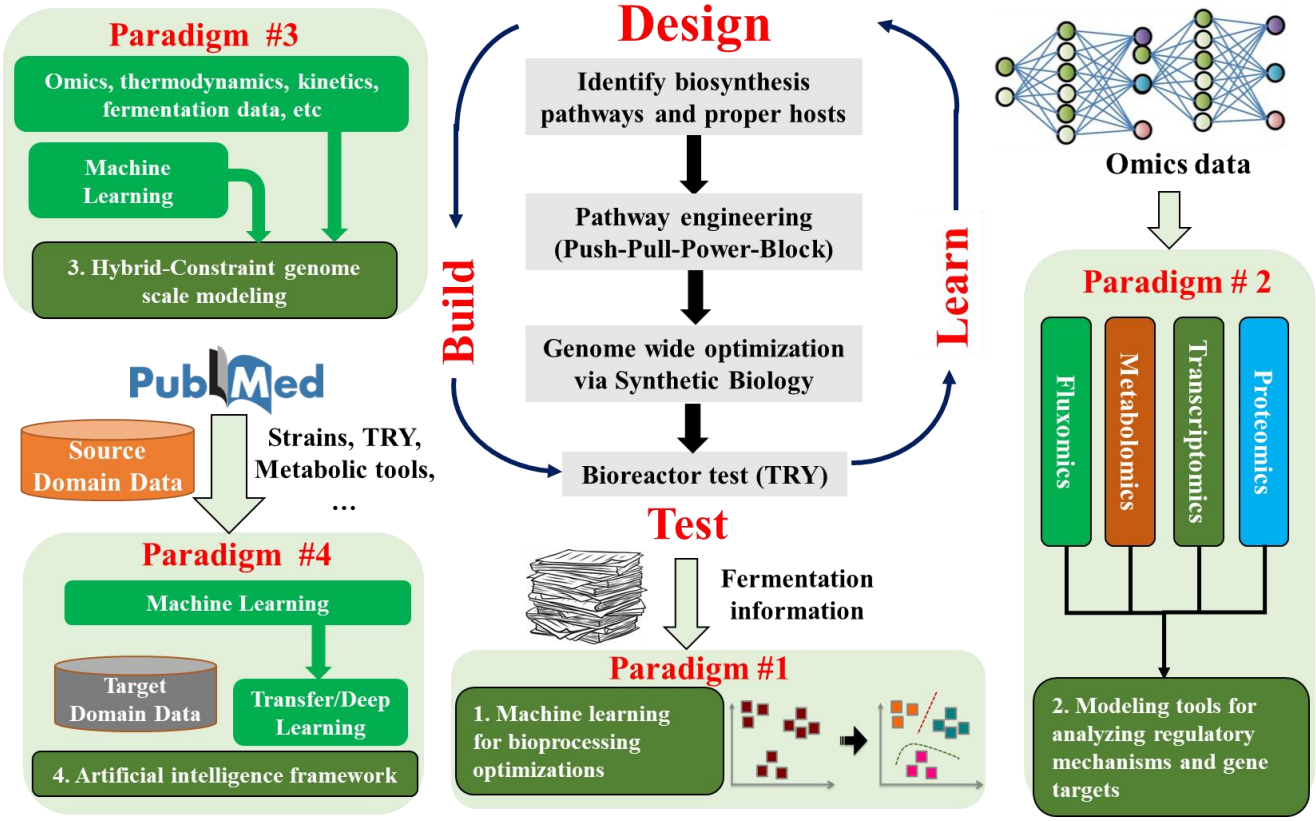


Figure 1.6 Paradigms of data-driven techniques in systems metabolic engineering

The advantages of GSM/FBA over ML lie in their interpretable and biologically meaningful solutions. On the contrary, ML models rely purely on statistics, thus may generate predictions that violate some biological constraints or lie out of reasonable ranges. In this regard, ML models are expected to gain great improvement when combined with GSM/FBA models. GSMs can help identify whether ML outcomes are biologically feasible, within biological reasonable ranges, or directly place upper bounds for ML outcomes. ML, FBA algorithm and constraint logic programming can be integrated to offer an expressive way to represent knowledge that involves statistics, constraints (usually on integers or real numbers) and logics (paradigm #3). Such hybrid models take into account the metabolic network, genetic design of the microbial host system, and the "suboptimal" conditions under which the fermentation process occurs. For example,

supervised learning methods and FBA have been used together to predict bacterial central metabolism[43]. In that study, experimental data of 37 bacteria species from over 100 ^{13}C-MFA papers were extracted and converted into structured data. Three supervised algorithms, Support Vector Machine (SVM), k-Nearest Neighbors (kNN), and Decision Tree were employed to train regressors to predict fluxes using features (substrate types, genetic modifications, and cultivation methods). The ML can generate reasonable flux boundaries for FBA models to reduce solution space during flux predictions of nonmodel microbial species. In summary, paradigm #3 binds the ML predictions with the GSM optimizations, which can not only predict production metrics (like yield, titer and rate) but also can suggest optimal genetic engineering strategies to employ (like what kind of plasmid to use, promoter strength, etc.) during design-build-test-learn cycle.

Finally, metabolic engineering is a rapid-developing field. The new high-throughput technologies can quickly generate large amount of data, such as high throughput mass spectrometry [64] and microfluidics [65], [66]. These data allow extensive validation of ML platforms and parameter estimations. Even those failed experimental data are valuable for training ML. For example, combinatorial synthesis and screening approaches create vast numbers of off-target phenotypes that can be used to study engineered metabolism by supervised learning. On the other hand, many input/output variables are not continuous or complete among different datasets. Advanced Deep Learning (DL) can investigate noisy but large biological data [67], [68]. Due to its nonlinear mapping power, DL can unify incomplete inputs/outputs. Small dataset sizes (which is usually the case for metabolic engineering data) can be tackled by strategies such as unsupervised pre-training [69]. During the learning process, noisy and incomplete data will be automatically “flattened” in their new representation space. Furthermore, DL can solve one system and apply the knowledge gained to a different but related

new system [70], [71], which may offer systems design or a priori estimation of broad-scope microbial factories. Subsequently, advanced mechanistic models, knowledge engineering, and machine learning lead to ever-improving artificial intelligence framework that relies less and less on the intuition of human engineers (Paradigm #4).

1.5 Hindrances and possible solutions to successful application of machine learning

Despite the promise of ML for synthetic biology and metabolic engineering, several hurdles still need to be tackled. A key challenge for applying ML is the lack of formatted, high-quality, and high quantity data. For example, DL will need ~ 10000 conditions to be effective. Large research groups are devoting increasingly time and manpower to establish and standardize systems biology database that will facilitate the validation and improvement of ML frameworks in the near future [47]. However, most existing publications contain data with no unified format and these datasets have to be manually curated from non-standardized reports. It is quite challenging to extract the information from a large amount of publications, because the data could be noisy and each paper contains large amounts of variables. Errors can arise from the original authors of the paper or researchers attempting to extract the information. This opens up the need of automatic and semi-automatic tools for collecting experimental data from literature. Natural language processing (NLP) may enable the automatic extraction of relevant data from thousands of publications, which can perform text summarization, evaluate paper quality, and minimize the impact and occurrence of human errors. On the other hand, transfer learning is a ML technique which alleviates the data insufficiency problem by transferring knowledge in one domain (typically with lots of data) to another domain where data are scarce [72] (paradigm #4). For

example, data and models on *E. coli* are relatively abundant. This knowledge can be transferred to the non-model microbial platforms, which have few available data by well-tuned transfer learning algorithms. Such practices will not only facilitate the specific task of microbial prediction, but also build a unified viewpoint of representation learning and domain adaptation through the study on practical biological data [73].

Another major concern is the fact ML models do not generalize well to data points representing conditions not present in the training data. For instance, the training datasets are enormous to identify gene targets for engineering a new host while optimizing bioreactor conditions for typical fermentations requires far less data. This challenge underscores the importance of creating hybrid data-driven and mechanistic models. The success of such hybrid frameworks has been demonstrated in recent efforts [43], [74], [75]. One study showed the possibility of using data-driven approaches to guide future developments of mechanistic-based models[76]. Furthermore, there has been a rapid increase in metabolic engineering data, while the influential factors (e.g., genetic tools, basic microbial pathways and hosts) have remained limited. Specifically, the variability of key upstream pathways towards biosynthesis is unchanged (Figure 1.7), and most bio-manufacturing comes from a few precursors (such as acetyl-CoA and pyruvate). Proper feature extraction from existing metabolic engineering data might result in rather robust coverage of possible conditions. Therefore, the number of model parameters may not increase as the size of the training database grows, which ensures the predictive fidelity of the ML platform.

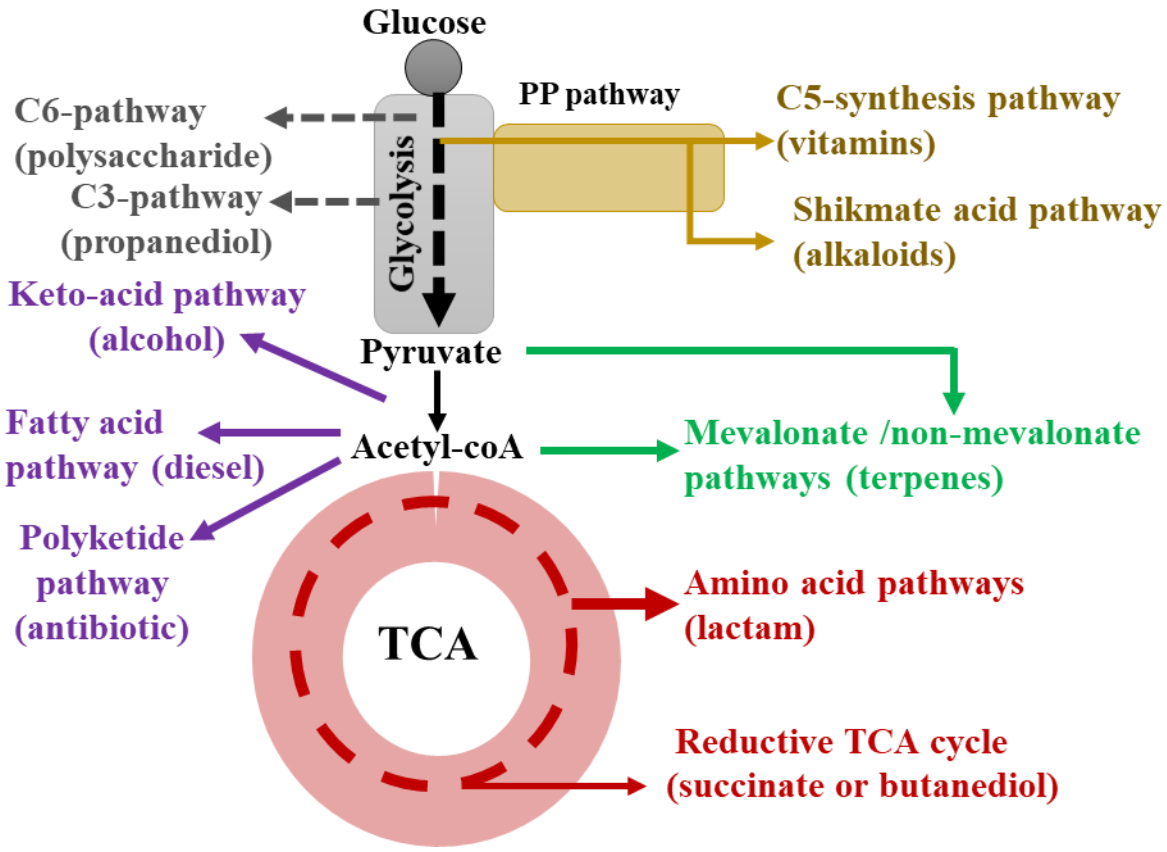


Figure 1.7 Common biosynthesis pathways from the central metabolic network

Chapter 2:Refining genome-scale metabolic network reconstructions[2]

2.1 Introduction

Genome-scale metabolic reconstructions are the basis of constraint-based analyses, which are finding ever increasing applications in metabolic engineering for industrial, medical and environmental purposes[77]. One of the major reasons for inconsistencies between genome-scale model predictions and experimental measurements is the presence of gaps in the network reconstruction [1]. Knowledge gaps are the result of missing information on genes, proteins, or reactions, while scope gaps occur due to the fact the metabolic network is only one of several integrated cellular networks (e.g. signaling networks). Thus, the consumption and production of a metabolite might not be fully captured by metabolism alone. Moreover, some microbes that depend on communal support of other organisms actually have gaps in their metabolism. Therefore, automated gap filling tools are merely hypotheses generators whose predictions need to be verified experimentally. Two general approaches to tackle the challenge of network gaps have been reviewed[78]. The first involves the use of algorithms based on network topology and genomic data. These are mostly concerned with finding gene candidates for orphan reactions. The second seeks to find missing reactions by minimizing the difference between computation and experiments. Gap-filling algorithms serve a dual benefit of model refinement and discovery

[2] This chapter is adapted from my publication: Oyetunde, T., Zhang, M., Chen, Y., Tang, Y., & Lo, C. (2016). BoostGAPFILL: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods. **Bioinformatics**, 33(4), 608-611.

of new biological capabilities [78]. Thus, efficient and robust gap-filling algorithms would prove invaluable in the development of high fidelity metabolic network reconstructions[79]. Newer approaches have sought to uncover inherent patterns in metabolic networks and have shown promise in predicting diverse network functions [80]. However, some of the predictions based on these methods might not be biologically realizable. Constraint-based methods, on the other hand, may not capture the information embedded in the network topology. It is difficult to test the accuracy of gap filling algorithms because verification usually involves experimentation to examine the biological relevance of suggested reactions. Thus, it is important to develop benchmark tests for gap filling algorithms to increase confidence in their use. In this work, we present a novel gap-filling framework, BoostGAPFILL, which integrates constraint-based and pattern-based methods [81] for metabolic network refinement. Our framework is inspired by machine learning methods developed for the Netflix prize [82]. We test the robustness of the gap-filling algorithms using artificial gaps (i.e. metabolites that cannot be produced or consumed at steady state) to simulate poorly characterized biochemistry. The gaps are introduced by randomly deleting reactions from the network. We then rank the algorithms on their ability to predict the actual deleted reactions from a universal reactions database and unblock blocked metabolites (i.e. gaps).

2.2 Methods

Our novel algorithm combines machine learning and constraint-based methods to identify possible candidates for missing reactions. We use machine learning to characterize the topology of the incomplete metabolic network and predict a set of possible reactions. The preliminary predictions are integrated with standard constraint-based gap filling in two ways: (i) using the

preliminary predictions as weighting factors in constraint-based algorithms and (ii) solving the pattern-based problem simultaneously with the standard gap filling formulation [83]. Details of this are described in the Appendix A. The basic concepts of the pattern module of our algorithm are shown in Figure 2.1.

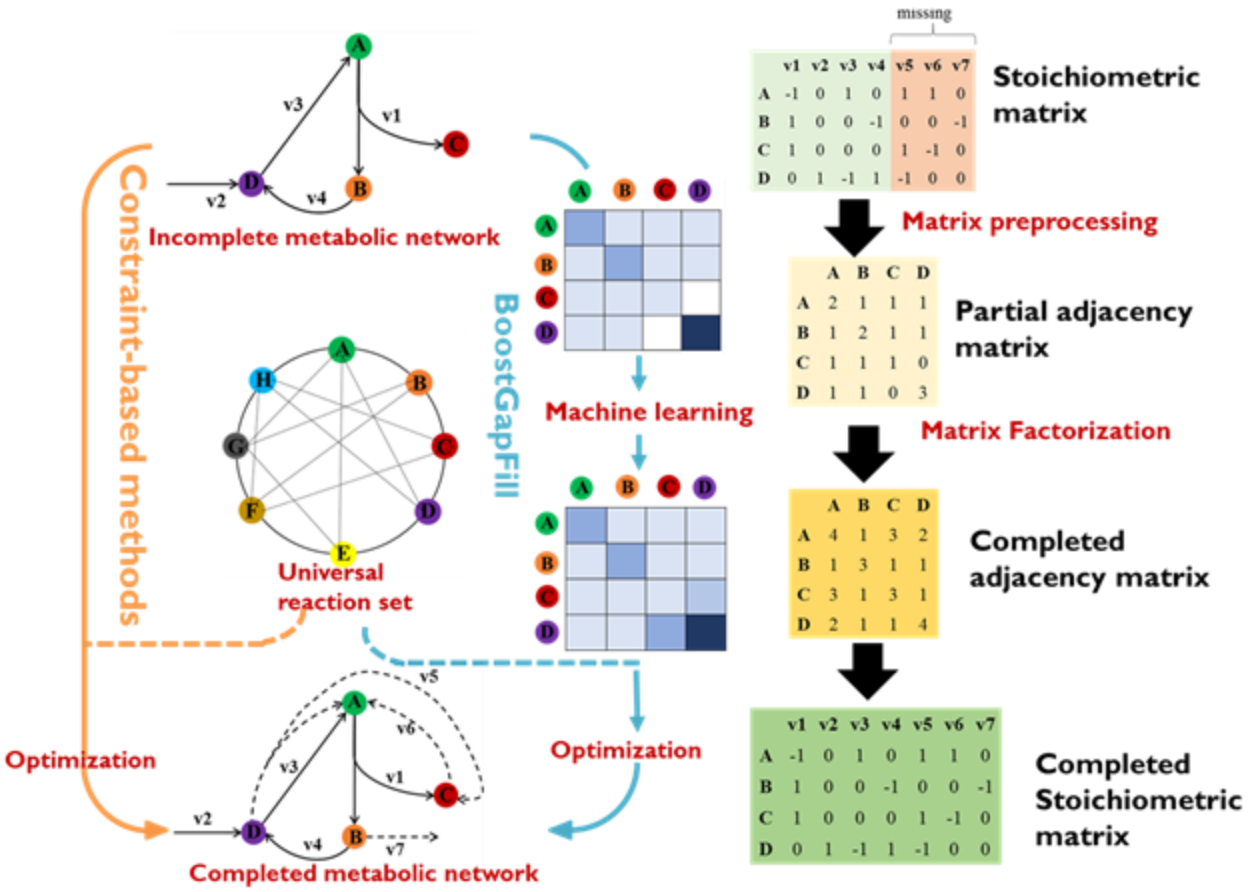


Figure 2.1 Basic concepts of the pattern-based module of BoostGAPFILL. BoostGAPFILL (right) contrasted with constraint-based procedures (left). In BoostGAPFILL, the partial adjacency matrix is derived from the incomplete stoichiometric matrix. The partial adjacency matrix is completed using matrix factorization models. Then reactions are selected from a universal database. The selection is formulated as an integer least squares problem in which the difference between the completed adjacency matrix is transformed to the stoichiometric matrix. In constraint-based procedures, the reactions are selected directly from the universal reactions database using an optimization criterion, such as minimum number of reactions required to fill the gaps in the network

2.2.1 Step A: Conversion of incomplete stoichiometric matrix to metabolite adjacency matrix

The binary incidence matrix, $\widehat{\boldsymbol{S}}$, can be derived from the stoichiometric matrix, **S**, by simply placing a one if the corresponding entry in the stoichiometric matrix is not zero, and a zero if otherwise. Post multiplying $\widehat{\boldsymbol{S}}$ with its transpose gives an m by m metabolite adjacency matrix, **A**, where m is the number of metabolites. **A** provides information about the relationship between the different metabolites. Each entry gives the number of reactions in which the two metabolites jointly participate.

2.2.2. Step B: Completion of metabolite adjacency matrix using matrix factorization

The entries of **A** conceptually represent the ranking of the relationship between metabolites. **A** is incomplete and we employ the standard matrix factorization model [82] as implemented in the free tool libFM [84] for its completion. Slight modifications are discussed in Appendix A.

2.2.3 Step C: Prediction of new reactions from a universal reaction set

Next, we attempt to recover the completed **S** by an integer least squares optimization in which we select reactions from a universal set that best match the completed **A**. The integer least squares optimization is relaxed to avoid long computational times associated with integer optimization problems. The result is a ranking of all reactions. Selections are made based on the top percent threshold or the top number of reactions. This step (of selecting reactions from a set based on

some constraints) is common to standard gap filling tools and is the step where we integrate standard constraints.

2.2.4 Modes of running BoostGAPFILL

BoostGAPFILL can be run in three modes (shown in Figure 2.2). Mode 1: the tool is run as described above. Thus, the predictions are based solely on the inherent metabolite patterns in the incomplete network. This mode is very accurate at capturing the topological information in the network as seen in Figure 2.3 but does not fill all the gaps. Mode 2: The pattern-based module is used to weight reactions in the universal database for use in FASTGAPFILL. Thus, BoostGAPFILL is used as a preprocessing step for FASTGAPFILL. This improves the fidelity of FASTGAPFILL as demonstrated in Figure 2.3. Mode 3: In this mode, we include the flux constraints (used in the standard constraint-based gap filling formulation) in step C described above. This enables BoostGAPFILL to be used for growth inconsistency reconciliation like tools such as SMILEY. Running BoostGAPFILL in mode 1 is preferred for initial screening of a large reactions database, with mode 2 and mode 3 preferred for more biologically realistic predictions. Mode 2 is best for pure gap filling while mode 3 can be used for growth data reconciliation and predicting reactions to unblock metabolites in turn. The limitations and technical implementation details are discussed in the Appendix B.

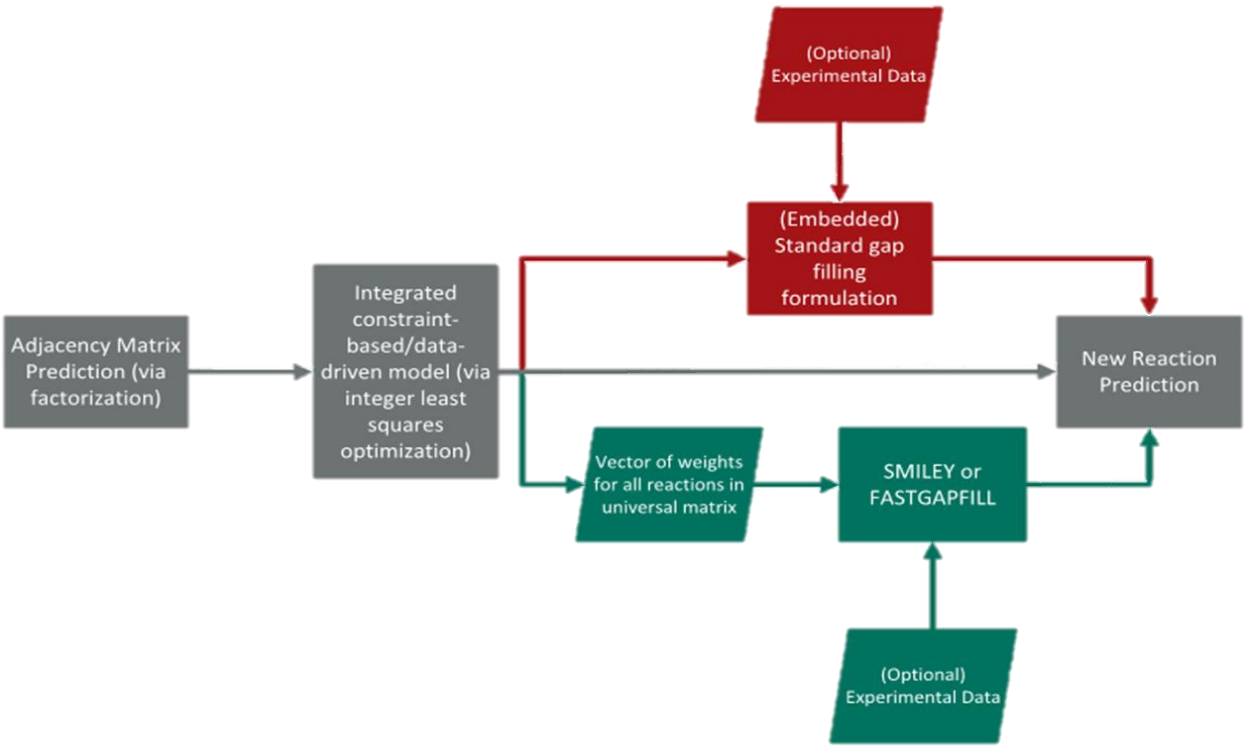


Figure 2.2 Modes of using BoostGapFill BoostGapFill seamlessly integrates existing gap filling tools and can incorporate growth or knockout data for inconsistency reconciliation.

2.3 Results and discussion

We test the performance of BoostGAPFILL on seven different metabolic network reconstructions downloaded from the BiGG database [37]. Figure 2.3 presents the comparison of the performance of BoostGAPFILL and FASTGAPFILL on the *E. coli* model iAF1260.

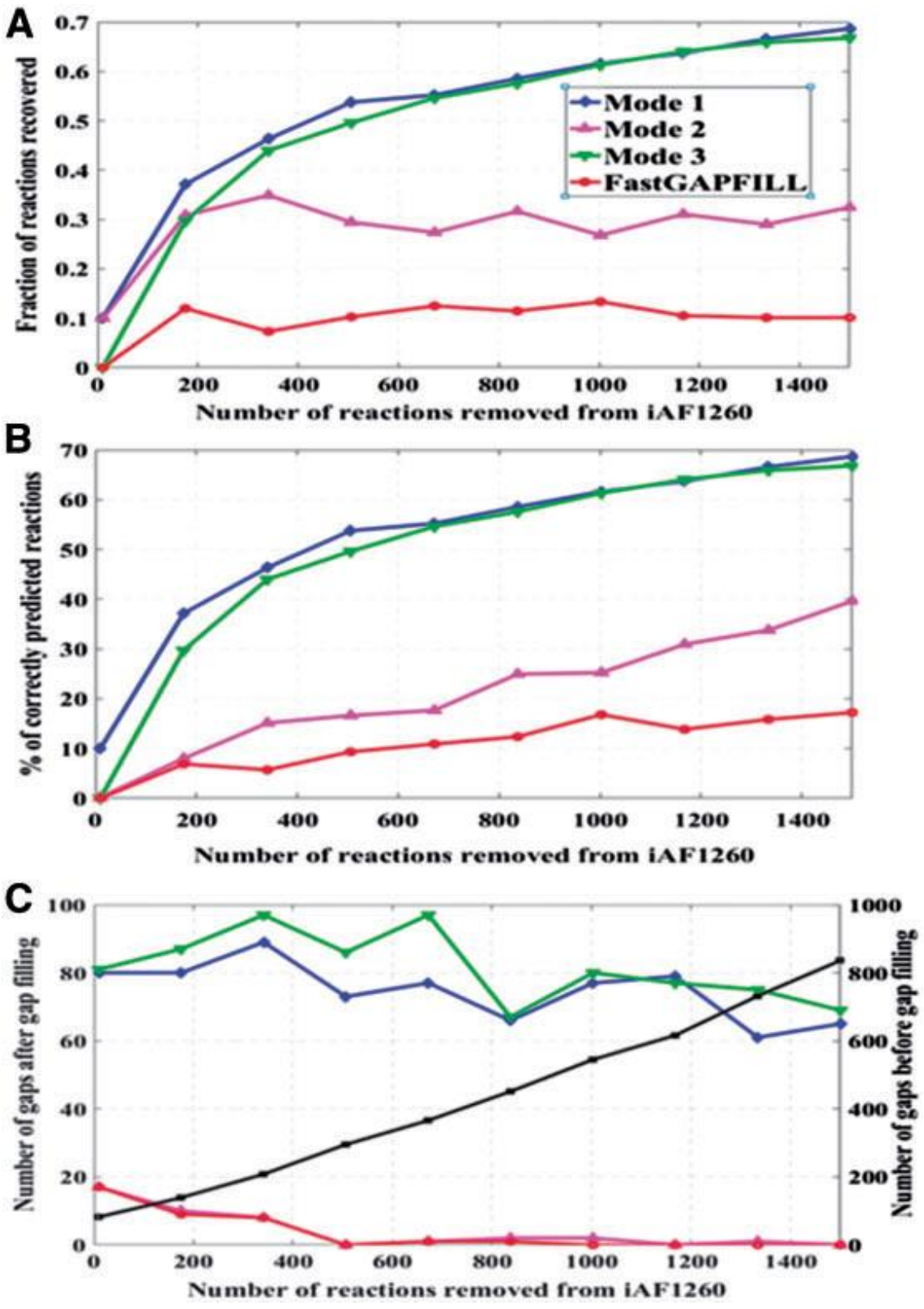

A
Fraction of reactions recovered
Mode 1
Mode 2
Mode 3
FastGAPFILL
Number of reactions removed from iAF1260
B
% of correctly predicted reactions
Number of reactions removed from iAF1260
C
Number of gaps after gap filling
Number of gaps before gap filling
Number of reactions removed from iAF1260

Figure 2.3 Comparison of the performance of gap-filling algorithms on E.coli model iAF1260 (A) Reactions are selected at random and deleted from iAF1260. The number of reactions deleted is shown on the x axis. Gap-filling algorithms are then used to predict possible candidates to complete the network. The number of reactions correctly predicted as a fraction of the number of reactions deleted is shown on the y axis. For BoostGAPFILL run in mode 1 and 3, the number of predicted reactions is the same as the number of deleted reactions (this can be manually set in the algorithm). For other algorithms the number of reactions predicted vary and cannot be directly set. **(B)** For the same simulation described above, the number of reactions removed from iAF1260 is shown on the x-axis, and the number of correctly predicted reactions is shown as a percentage of the total number of reactions predicted by the algorithm. **(C)** The number of gaps in the network before (shown as a black line) and after gap filling is shown on the right and left y axes respectively. Note that the model before gap filling has a certain number of reactions deleted (as seen on the x axis). Both mode 2 of BoostGAPFILL and FASTGAPFILL completely fill all the gaps

BoostGAPFILL automatically fixes gaps (see Figure 2.4). It also appears to perform well even when a large number of reactions are missing.

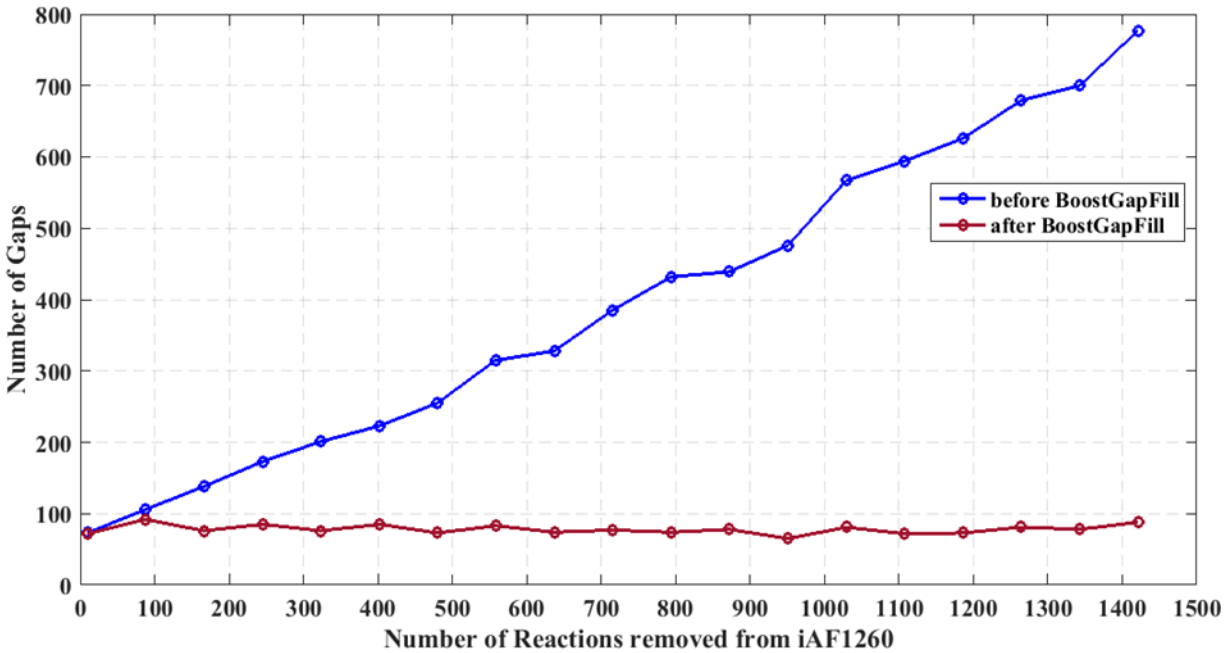


Figure 2.4 Simulation of artificial gaps The number of gaps (blocked metabolites) in the network before and after gap filling is shown on the y axes. Note that the 'before' model corresponds has a certain number of reactions deleted (as seen on the x axis). Reactions are selected at random and deleted from iAF1260. The number of reactions deleted is shown on the x-axis. The number of gaps of the resulting model is computed. BoostGapFill (mode 1) is then used to predict possible candidates to complete the network. The x axis shows the number of reactions randomly removed from the *E. Coli* model iAF1260. The y axis shows the number of gaps before and after the BoostGapFill.

The algorithm was able to predict several new reactions added in iJO1366 (the latest *E. coli* model at the time of this work) from an earlier version (iAF1260) including new content (15 gap filling reactions and 4 new content reactions), as shown in Figure 2.5.

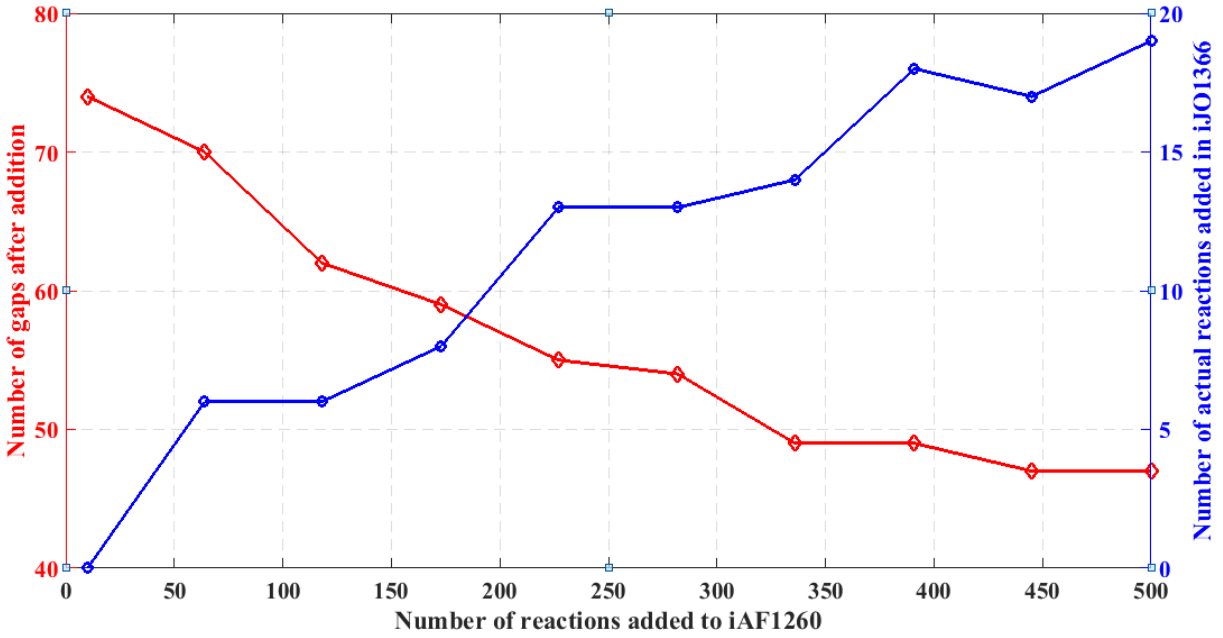


Figure 2.5 Effect of added reactions on number of gaps BoostGapFill (run in mode 1) is used to suggest reactions for the *E. Coli* model iAF1260. The number of reactions added is shown on the x axis. The graph shows that the predicted reactions by BoostGapFill lead to a reduction in gaps (After the addition of 500 reactions, the number of gaps is down to 47). Some of the reactions predicted (19) were actually added in the latest metabolic reconstruction of *E. Coli*, iJO1366 – 15 of them were gap filling reactions while 4 represent new content added to the model. When FASTGAPFILL is run on iAF1260 all the unblocked metabolites are unblocked but none of the reactions predicted is

present in iJO1366. When BoostGapFill is run in mode 2 (reaction weights are fed into FASTGAPFILL), all the gaps are closed and 23 of the new predicted reactions are present in iJO1366. Note it is not possible to directly vary the number of reaction predictions (for BoostGapFill run in mode 2 and FASTGAPFILL)

While tools like FASTGAPFILL [85] and SMILEY [86] perform well in predicting reactions that close as many gaps as possible (Figure 2.3 C), BoostGAPFILL outperforms them in terms of preserving the network topology (Figure 2.3). This illustrates the fact that constraint-based techniques can sometimes fail to capture the embedded patterns in metabolic networks and thus their predictive fidelity is compromised. BoostGAPFILL provides that missing functionality and easily integrates with the existing gap filling tools. Similar performance was observed in other metabolic network reconstructions as seen in Figure 2.6.

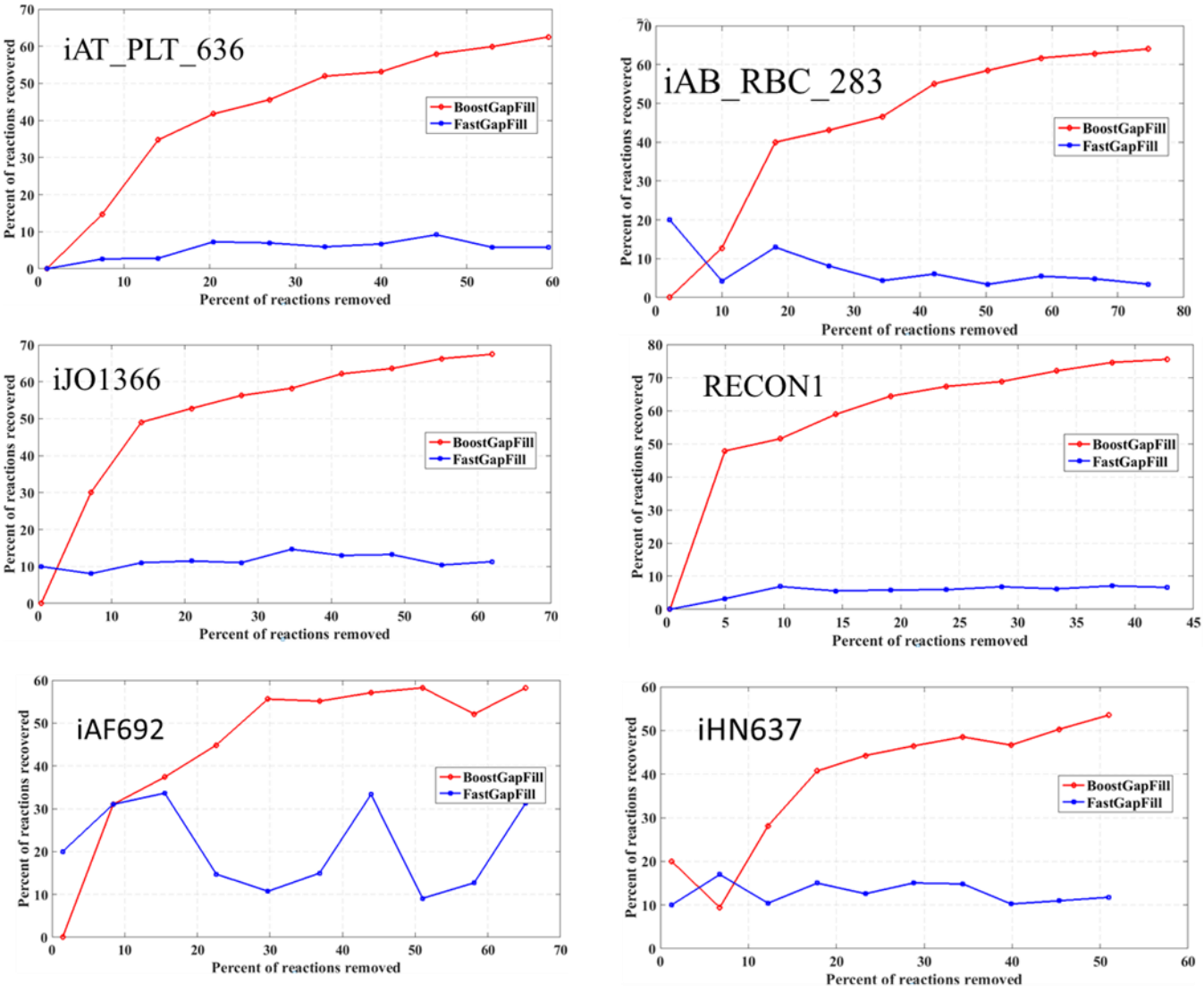


Figure 2.6 Comparison of BoostGapFill and FastGapFill across different metabolic network reconstructions

iAT_PLT_636 – human platelet metabolism (1008 reactions and 738 metabolites); **iAB_RBC_283** - erythrocyte metabolism (469 reactions and 342 metabolites); **iAF692** - *Methanosarcina barkeri* (str. Fusaro) (690 reactions and 628 metabolites); **iJO1366** – *E. coli* (2583 reactions and 1805 metabolites); **RECON1** - *H. sapiens* (3742 reactions and 2766 metabolites); **iHN637** - *Clostridium ljungdahlii* (DSM 13528) (785 reactions and 698 metabolites)[37]

BoostGAPFILL can also make predictions of reactions containing metabolites not in the original network (Figure 2.7 and see Appendix B for discussion).

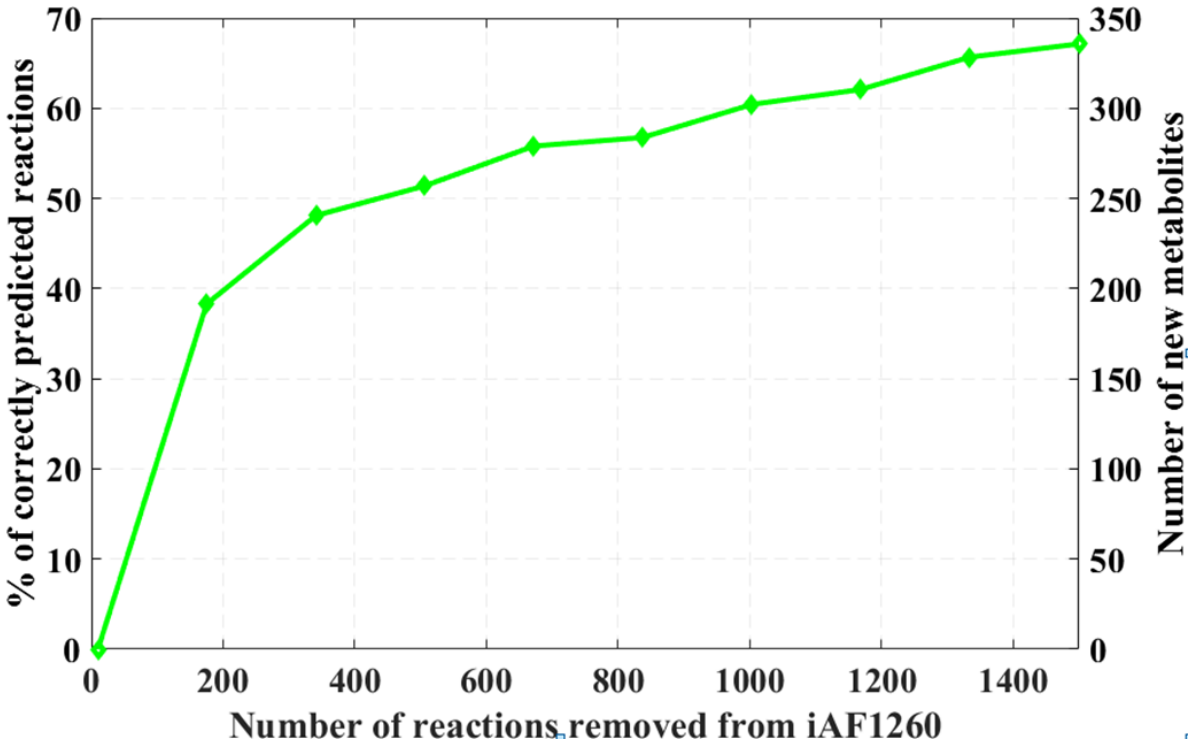


Figure 2.7 BoostGapFill with newMet option set to 'true' The number of reactions deleted is shown on the x-axis. The number of correctly predicted reactions is shown as a percentage of the total number of reactions predicted by the algorithm is shown on the left y-axis. The number of new metabolites is shown on the right y -axis.

Chapter 3: Data-driven computational strain design[3]

3.1 Introduction

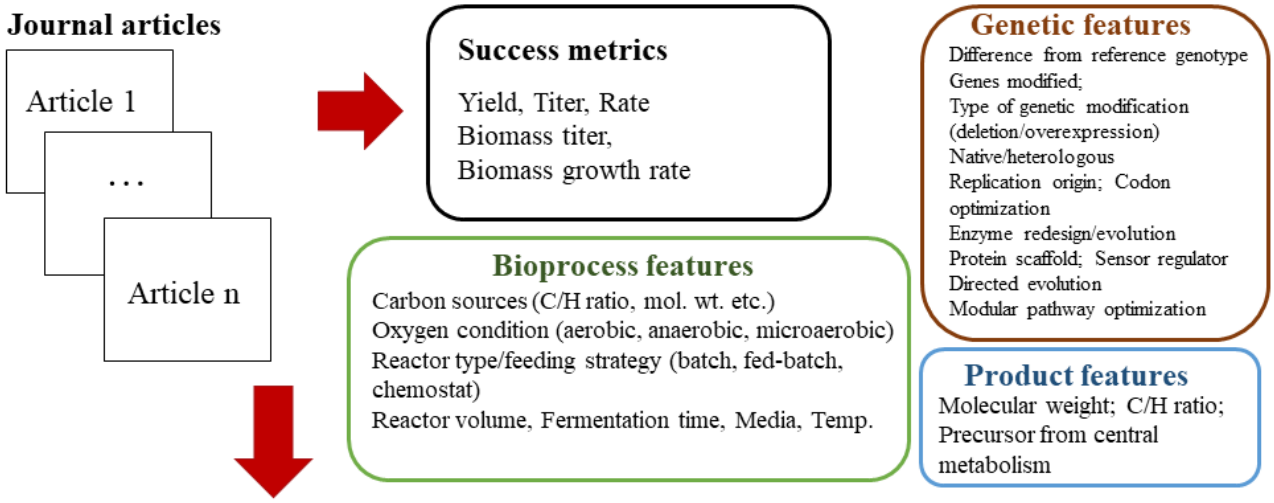
Despite the rapid advances in designing synthetic biological systems for various important applications, prediction of cellular behavior remains a challenge [87]. High fidelity predictive tools are critical for enabling rational strain design. The earlier tools developed were steady-state constraint-based methods but newer tools utilizing kinetic information [14] and integrating omics data [88] have been developed to improve model prediction accuracy. However, the practical utility of these tools has not been extensively demonstrated, and the majority of metabolic engineering efforts are still currently based on experience, intuition, and laborious testing of large numbers of designs. This is because that a mechanistic model cannot account for complete bioprocess variables or metabolic regulatory interactions, while hidden physiological constraints (such as metabolite channeling, metabolic burdens, strain stability, changes in enzyme expression in different phases of cell growth, and cell maintenance loss) lead to suboptimal cell metabolisms [89], [90]. Quantitative modeling of these phenomena is critical for the success of metabolic engineering designs. Since mechanistic models may not be comprehensive enough to guarantee accurate predictions, data-driven approaches have shown promise for accounting for nontrivial factors without knowledge of cellular processes [8]. Given the extensive microbial researches to produce variety of bio-products, there has been a lot of interests in utilizing

[3] This chapter is adapted from my publication: Oyetunde, T., Liu D., Martin H.G., and Tang Y.J Machine learning framework for robust assessment of microbial factory performance. **PLoS ONE** (in revision).

published metabolic engineering data to facilitate new designs and shorten the 'design-build-test-learn' paradigm of strain improvement [91]. Currently, metabolic engineering case studies are rapidly growing. Databases for strain development and related omics studies are being developed [39], [42], [43], [45]–[47], [87]. These databases provide genomic information to gain insights into cellular processes and their regulations. On the other hand, there are still few knowledge engineering efforts to extract and standardize holistic bioinformatics from the published papers including genetic modification strategies, cell physiological responses, and bioprocess conditions. In fact, these published papers may contain wealthy resources and lessons to support machine learning for strain designs, and thus leverage published data may assist metabolic model to predict cell realistic performances and tradeoffs among TRY (titer, rate and yield) under realistic conditions (e.g., product inhibitions and suboptimal pathway functions, etc.).

Nerveless, the use of literature data for computer learning strain design and performance predictions still faces difficulties: 1) Lack of standardizations of data reports from different research labs, 2) Incomplete production metrics (titer, yield, and rate) and experimental parameters; 3) Sparse data coverage (most of the available data are focused on a few popular products and designs). To digest the noisy information from thousands of metabolic engineering publications, data collections, curations, and feature categorizations must be performed to make sufficiently large datasets assessable to machine learning tools. Such knowledge engineering requires extreme large amount of manpower. To resolve this problem, this proof-of-concept study has manually extracted data from over 100 published *E. coli* biomanufacturing papers over the past decade (Fig. 3.1). Advanced machine learning techniques (data augmentation, ensemble learning) are employed to alleviate the challenges of sparse and small datasets. Constraint-based modeling is used to provide additional features for training the ensemble machine learning

models (Fig. 3.2). The hybrid platform provides reasonable estimations of *E.coli* TRY performance, which may open a new direction for metabolic modeling and strain design.



Design No.	Bioprocess			Genetic			Product			Success metrics		
	Media	...	Oxygen condition	Diff from ref. genotype	...	Genes modified	No_C	...	mol. wt of product	Yield	...	Rate
1	1	...	2	iLvG ,rfb, rph,fadD		tesA, ipdV, bkdAA, bkdAB, bkdB, fadR, ilvGM, ilvCD, ilvA, rfp	6	...	256	0.06	...	0.18
2	1	...	1	ilvG , rfb, rph, fadD, sucC, fabR		vcfadR	8	...	270	1.41	...	0.04
...	...	...	...					...	228	...	...	...
m	3	...	2	rph, hsdR, metA, tdh, ilvB, ilvI, fadE		NA	4	...	74	0.19	...	0.28

Figure 3.1 Database curation and feature extraction methodology

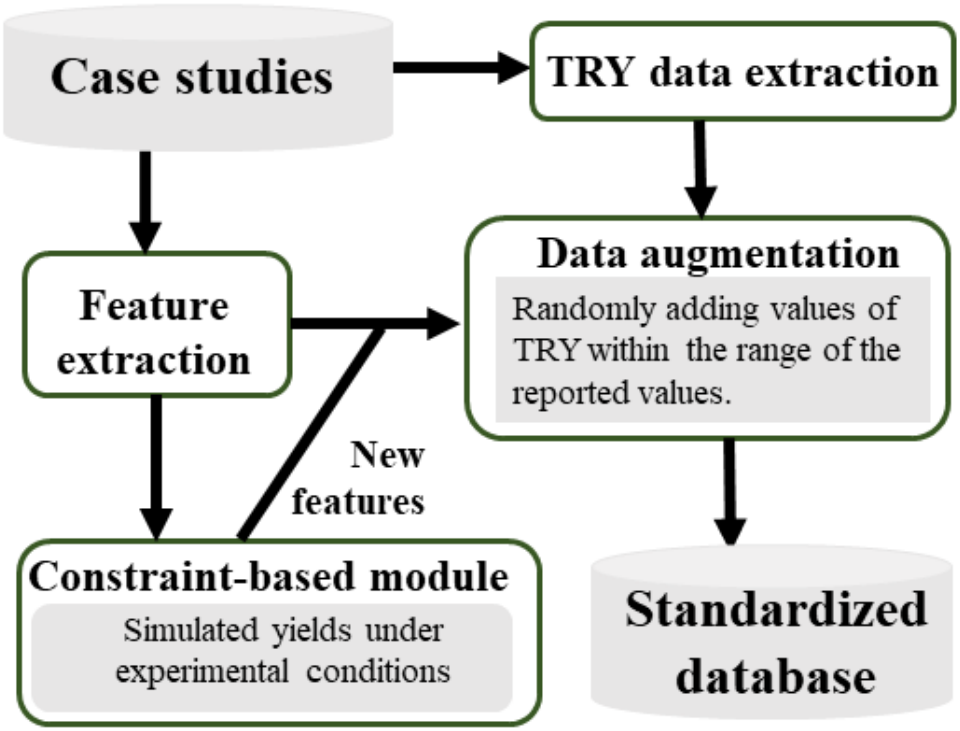


Figure 3.2 Feature additions via genome scale model simulations and data augmentation based on case studies described in the literatures

3.2 Methodology

3.2.1 Database curation

E. coli is the most common platform for metabolic engineering. The database is manually curated from metabolic engineering literature on the production of diverse chemicals from *E. coli* grown on different substrates. The data curation strategy is based on previous work [92]. This involves identifying possible influential factors a priori (shown in Fig. 3.1 and Table 3.1). The full list of papers is shown in the supplementary file. A sample of feature extraction from a journal paper is shown in Table 3.1. The list of features is iteratively updated based on model performance. Because of incomplete experimental details described in some papers,

comprehensive data extraction may be difficult. Two additional features are used to describe whether or not all the genetic and experimental conditions have been fully included by the feature list.

Metabolic engineering design factors template used for feature extraction. Sample values are taken from [93]. Features that refer to a list of genes are entered as a vector of ones and zeros. For example, in the sample values, 'het_gene' (whether the gene inserted/overexpressed was heterologous) is entered as 1,0,0 meaning alsS is heterologous while ilvC, ilvD are not. YE stands for yeast extract.

		Feature	Description	Sample value
carbon source characterization	1	cs1	first carbon source	1
	2	cs1_mw	first carbon source molecular weight	180.16
	3	cs_conc1	first carbon source concentration (mM)	111.0124334
	4	CS_C1	mol C in first carbon source	6
	5	CS_H1	mol H in first carbon source	12
	6	CS_O1	mol O in first carbon source	6
Bioprocess conditions	7	reactor_type	type of reactor (continuous, batch or fed-batch)	1
	8	rxt_volume	working volume of reactor (L)	2
	9	media	media used for fermentation (M9,AM1,AM2, M9+ yeast extract,LB,NBS,TB,other rich media)	YE
	10	temp	temperature of medium used for fermentation (oC)	37
	11	time	total time for fermentation	36
Genetic modifications	12	oxygen	oxygen condition in reactor (aerobic, anaerobic, microaerobic,extra aerobic)	2
	13	sbg_ref	reference strain in the study	BFA7.001(DE3) PCT01
	14	s_ref_gen	genes modified from the strain MG1655	lacI, rrnB, lacZ, hsdR514, araBAD, rhaBAD, zwf, mdh, frdA, ndh, pta, poxB, ldhA,T7 RNA polymerase
	15	s_gen_mod	type of gene modification: insertion/deletion	0,0,0,0,0,0,0,0,0,0,0,0,0,1
	16	gene_mod	genes modified from reference strain of study	alsS, ilvC, ilvD
	17	gene_del	whether or not the gene was deleted	0,0,0
	18	gene_ovr	whether or not the gene was overexpressed	1,1,1
	19	het_gene	is the gene heterologous? (yes/no)	1,0,0
	20	rep_origin	plasmid copy numbers	5,5,5
	21	codon_opt	codon optimization?	0,0,0
	22	sen_reg	sensor regulator?	0,0,0
	23	enz_design	enzyme redesign evolution?	0,0,0
	24	protein_scaffold	protein scaffolding?	0,0,0
	25	dir_evo	direction evolution?	0
	26	Mod_path_opt	modular pathway optimization?	0

Product characterization	27	prod_name	name of the product	Isobutanol
	28	no_C	mol C in product	4
	29	no_H	mol H in product	10
	30	no_O	mol O in product	1
	31	no_N	mol N in product	0
	32	mw	molecular weight of product	74
	33	precursor	precursor from central metabolism	6
	34	enz_steps	number of enzyme steps from precursor	5
	35	atp_cost	number of atp molecules needed from precursor to product	0
	36	na_cost	number of nadh/nadph molecules needed from precursor to product	2
Production metrics	37	yield_1	yield in gProduct/g Carbon source fed	0.0405
	38	yield_2	yield in gProduct/g Carbon source consumed	NA
	39	yield_3	yield in gProduct/g Biomass	0.623076923
	40	titer	concentration of product in g/L	0.81
	41	rate	maximum productivity in g Product/ L /h	0.0225
	42	bio_titre	biomass concentration (g/L)	1.3
	43	bio_grw_rate	biomass growth rate in exponential phase (/h)	0.45
other	44	gen_info	are all the genetic modifications in the paper fully captured by the above categories? (yes/no)	1
	45	env_info	are all the reactor conditions in the paper fully captured by the above categories? (yes/no)	1

3.2.2 Constraint-based simulations

Given the genetic and environmental background, the most recent *E. coli* genome-scale metabolic reconstruction, iML1515 [94] is used to simulate theoretical microbial yields based on reaction stoichiometry. First, iML1515 flux network is modified based on each case study (e.g., gene knockouts), while inflow and outflow fluxes are constrained based on bioprocess conditions (such as carbon sources, aeration level in the reactor, growth rate, etc.) by setting the upper and lower bounds of the associated reactions to zero. A flux balance analysis (FBA) simulation (maximize biomass growth objective) is then performed to test if the resulting model is feasible. Then, the further genetic interventions (in form of knockouts or overexpression) are simulated

similarly so that the *in-silico* model represents the actual experimental conditions as closely as possible (Eqn.3.1). To simulate overexpression of a biosynthesis pathway, the lower boundary of the associated flux is set to 10% of the theoretical maximum flux through this pathway. To characterize the metabolic capacity of the network after genetic modification under the applied process conditions (feature engineering), we have computed the product and biomass yield under different constraints. These are maximum biomass growth and product yield, maximum biomass growth at 50% maximum product yield, maximum product yield at 50% biomass growth) (Eqns. 3.2-3.5). FBA results are used as additional features used in training the various machine learning models employed, which captures the metabolic network capabilities (in terms of feature variables) for data driven models. For cases, iML1515 model (with the experimental genetic and bioprocess conditions imposed) can predict feasible solution spaces. The corresponding FBA can be constrained based on biomass growth, the number of genes modified, and the fraction of those genes that are overexpressed or deleted. The FBA simulation outcomes (simulated yields under presumed experimental conditions) are fed into machine learning pipelines as additional features from Table 3.1 for model training (Figure 3.2 and 3.3).

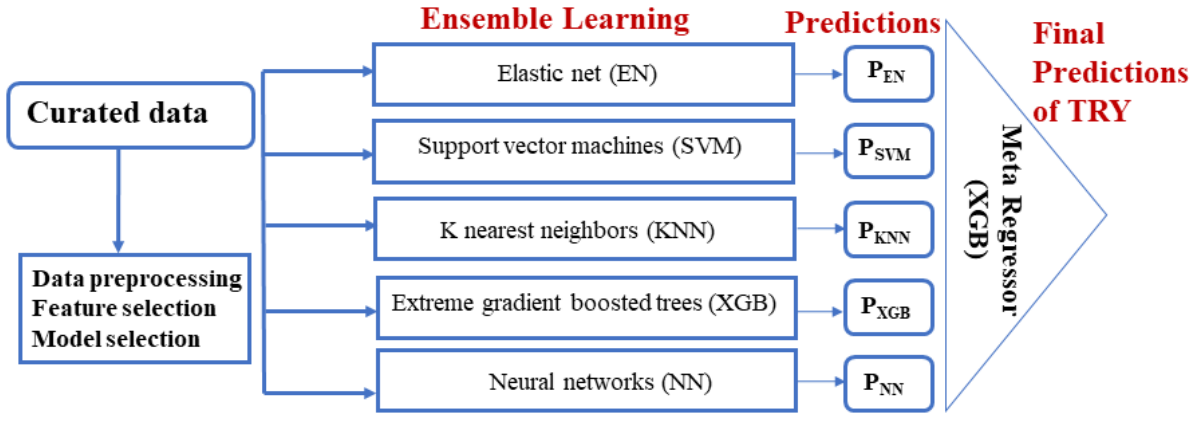


Figure 3.3 Machine learning pipeline. Ensemble learning using stacked regressors.

$$\max \boldsymbol{c_b v}$$

$$subject\ to \begin{cases} \boldsymbol{S.v} = 0 \\ lb_j^e \leq v_j \leq ub_j^e \end{cases} \quad (3.1)$$

where c_b *is a vector of zeros with one for the biomass flux variable*

lb_j^e *and* ub_j^e *are the flux bounds adjusted based on the bioprocess condtions and genetic modifications* (*using the gene* $-$ *to* $-$ *protein relationships*)

$$\max \boldsymbol{c_p v}$$

$$subject\ to \begin{cases} \boldsymbol{S.v} = 0 \\ lb_j^e \leq v_j \leq ub_j^e \end{cases} \quad (3.2)$$

where c_p *is a vector of zeros with one for the desired product flux variable*

$$\max \boldsymbol{c_b v}$$

$$subject\ to \begin{cases} \boldsymbol{S.v} = 0 \\ lb_j^e \leq v_j \leq ub_j^e \\ \boldsymbol{c_p v} = 0.5 v_p^* \end{cases} \quad (3.3)$$

where v_p^* *is a maximum product flux computed by Eqn.* 2

$$\max \boldsymbol{c_p v}$$

$$subject\ to \begin{cases} \boldsymbol{S.v} = 0 \\ lb_j^e \leq v_j \leq ub_j^e \\ \boldsymbol{c_b v} = 0.5 v_b^* \end{cases} \quad (3.4)$$

where v_b^* *is a maximum product flux computed by Eqn.* 1

$$y_b^{max} = \frac{v_b^*}{v_c^*},\ \ y_p^{max} = \frac{v_p^*}{v_c^p}, y_b^{50p} = \frac{v_b^{50p}}{v_c^{50p}}, y_p^{50b} = \frac{v_p^{50b}}{v_c^{50b}} \quad (3.5)$$

$where\ \ v_c^*, v_c^p, v_c^{50p}, v_c^{50b} are\ \ carbon\ source\ uptake\ rates\ from\ Eqns\ 1-4\ respectively$

$v_p^*, v_p^{50b}\ are\ the\ product\ fluxes\ from\ \ Eqns\ 2\ and\ 4\ respectively$

$v_b^*, v_b^{50p}\ are\ the\ biomass\ growth\ rates\ \ from\ \ Eqns\ 1\ and\ 4\ respectively$

$y_b^{max}\ is\ the\ maximum\ biomass\ yield$

$y_b^{50p}\ is\ the\ biomass\ yield\ at\ 50\%\ of\ the\ maximum\ product\ flux$

$y_p^{max}\ is\ the\ maximum\ product\ yield$

$y_p^{50b}\ is\ the\ product\ yield\ at\ 50\%\ of\ the\ maximum\ biomass\ growth\ rate$

3.2.3 Data pre-processing and augmentation

Principal component analysis and data standardization (using mean and standard deviation) are used to transform the input data (The first 40 components of the PCA are used in training the model). The data set is divided into training, validation, and test sets (test set is 10% of the whole dataset). The test set is handled separately to prevent the data leakage (where some properties of the test distribution are inadvertently used in tune the model resulting in overly optimistic prediction accuracies). For the training and validation sets, data augmentation (a popular technique used in computer vision)[95] was employed as follows: for each data the point, n number of points where generated by randomly adjusting the values of titer, rate and yield within t % of the reported value. A grid search is used to tune hyperparameters n and t. n ranged from 10 to 90 and t ranged from 0.1% to 1%. Final values of n and t used are 50 and 0.1% respectively. Data augmentation improved the cross validation and test set accuracies.

3.2.4 Ensemble learning and hyperparameter tuning

An overview of the machine learning pipeline is shown in Figure 3.3. Different machine learning models are tested. Support vector machines, elastic nets, random forest, gradient boosted trees, k nearest neighbors, and neural network models (densely connected, 5 hidden layers (100 neurons

each) with batch normalization and dropout between layers) are trained separately on the training set. The results (test scores, cross validation and learning curves) of each of the ML models are shown in the supplementary file. Ensemble learning is then performed using the output of the different ML models. This is done with a stacked regressor (using gradient boosted trees as a meta regressor). This helps to combine the best effects of the different machine learning models to higher predictive accuracies. Hyper parameter tuning for each machine learning model and final stacked regressor was based on grid search with five-fold cross validation. The modeling framework was implemented in Python. Scikit-learn [96], XGBoost [97] and Keras [98] machine learning libraries were used in the supervised learning module. COBRApy [99] implementations of constraint-based methods were used. Visualizations generated with Matplotlib[100] and Bokeh (http://bokeh.pydata.org) libraries.

3.3 Results and discussion

3.3.1 Description of curated database

This study focuses on *E. coli* platforms with native or heterologous pathways for producing small molecules. About 1200 metabolic engineering designs for producing more than 20 compounds have been manually extracted and estimated from ~100 journal articles. The genetic strategies and microbial fermentation conditions were extracted based on Table 3.1, as proposed by the previous paper [60], [92]. In brief, data are organized as six categories, including carbon sources, bioprocess conditions (e.g., medium types), genetic modification strategies, product features (e.g., molecular weight, enzyme steps from central pathways, etc.), production metrics TRY, and other unaccountable factors. To summarize extracted data, the distribution of titer (the most commonly reported metric) for the different compounds is shown in Fig 3.4, where native

products (naturally synthesized by *E.coli*) often have higher titer than non-native products (synthesis via heterologous pathways).

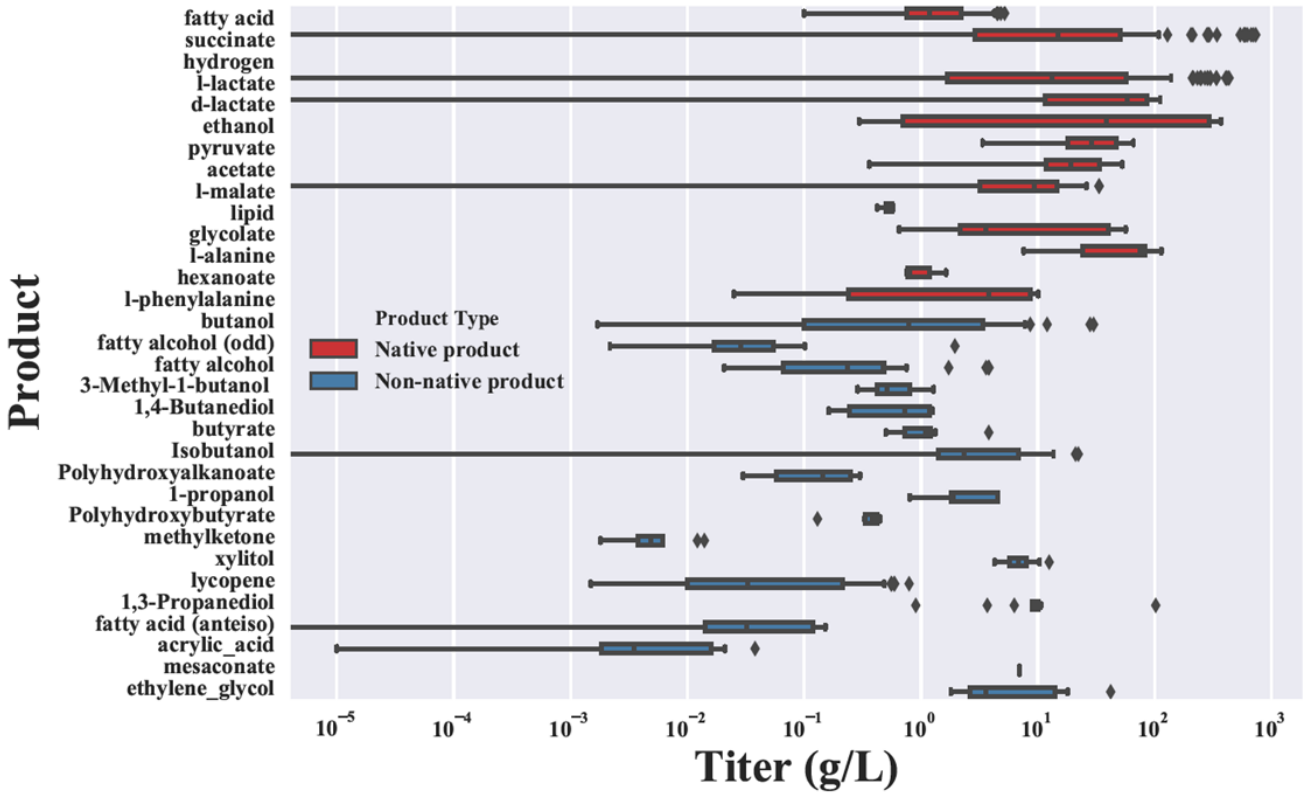


Figure 3.4 Summary of curated database showing distribution of titers (units in g/L) for 25 different products from the bacterium E. coli.

Biomanufacturing requires cell factories to achieve desired TRY. Figure 3.5 provides correlations among the three metrics as well as product molecular weight (mol. wt). There appears to be positive correlations between titer and yield (i.e., the increase of feedstock conversions improves product concentrations). However, production rate can be impaired by very high production yield/titer (i.e., elevation of yield reduces carbon resource to generate ATP and biomass for cell well-being, while the high titer may stress cell physiologies). In general, it is difficult to maximize all three biomanufacturing metrics due to the imbalance of carbon/energy

metabolisms and product inhibitions. Figure 3.5 shows that these maximal production rates from published case studies are in the medium ranges of titer (6~10g/L) and yield (0.45g/g~0.75g/g), while some products (e.g., succinate) achieve very high yield (>1g /g substrate) due to cellular carbon fixations. These extracted datasets can be used as the base for machine learning to predict fermentation performance and tradeoffs.

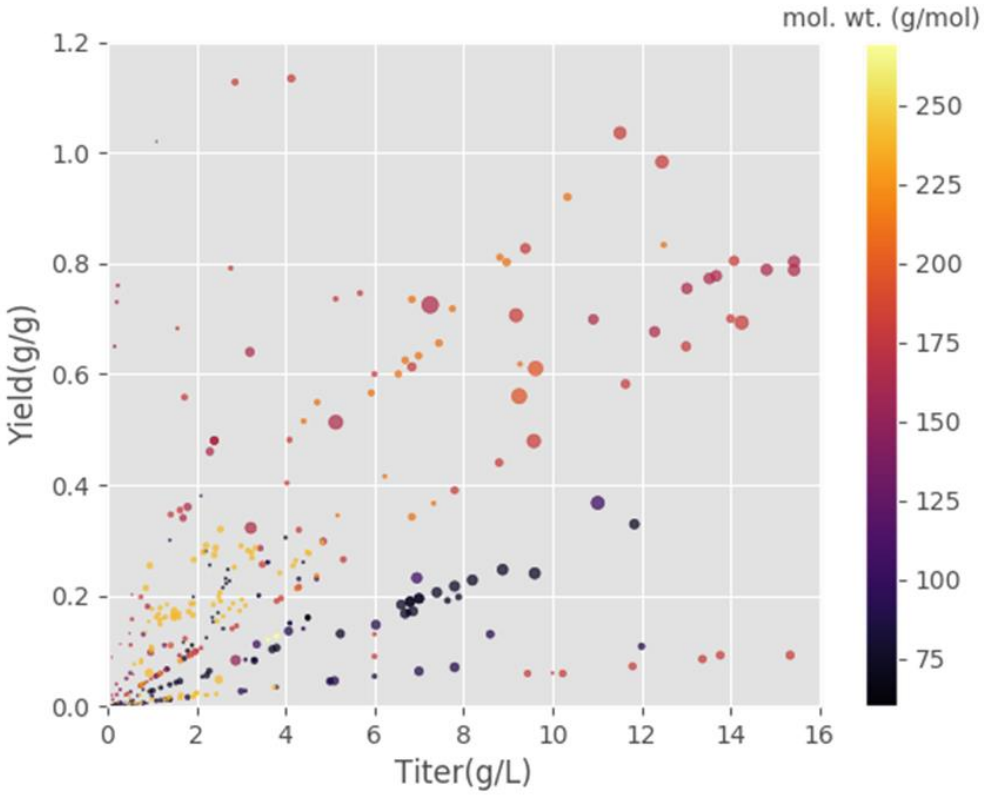
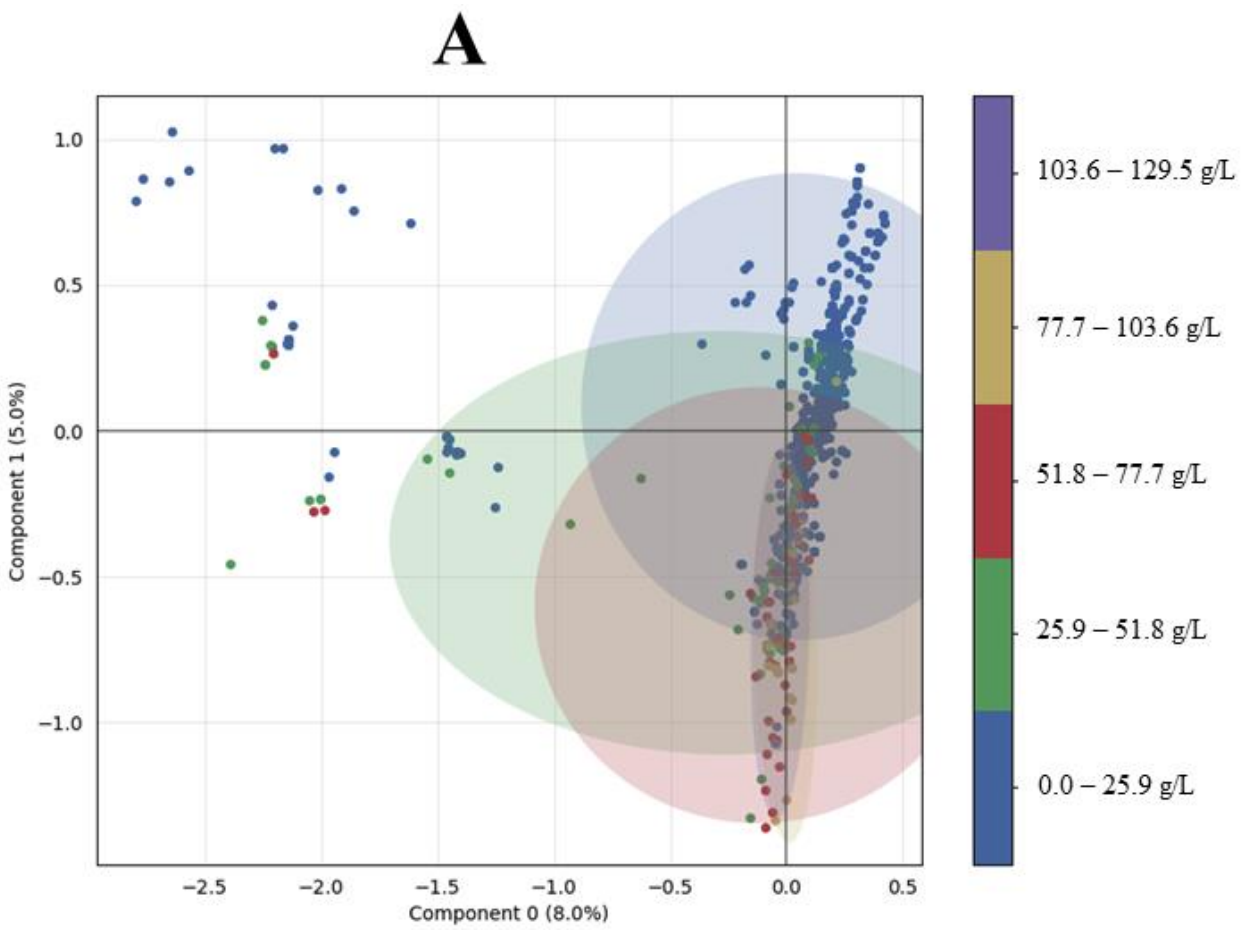


Figure 3.5 Comparison of production metrics (titer, rate, and yield) The size of the dots corresponds to the rate values (in g/L/h scaled by the minimum and maximum value – 0.000043 and 10.83 g/L/h respectively). Molecular weight of each product (g/mol) is shown by the color gradient of the dots (color bar).

3.3.2 Identification of critical metabolic engineering factors

Many factors may play roles in optimal metabolic engineering design. To analyze the data based on our custom-designed features, we utilized the complementary approaches of multiple correspondence analysis (MCA) [101] and principal component analysis (PCA) [102]. MCA is

more suited for categorical data while PCA works best with continuous data. Interestingly, both techniques yielded similar results (clustering of the high titer values around the zero of the first principal component and along the second principal component). Fig. 3.6A shows the plot of the first two principal components of the MCA with the titer values superimposed. Regions of high titers are clustered along the second principal component and most have a value of zero for the first principal component. This indicates that the factors that make up the second principal component are critical for high titers. The contributions of different factors to the first two principal components of the PCA are shown in Fig. 3.6 B and are indicative of their relative influence on microbial cell performance. Bioprocess factors such as reactor volume, temperature, oxygen conditions (anaerobic or aerobic), medium types, substrate characteristics (molecular weight, C, H, O composition) have impacts on cell performance. Therefore, further categorization and addition of bioprocess conditions as model inputs can improve machine learning accuracy. On the other hand, outcomes from genetic factors/modifications are more-uncertain due to complex genomic nature and metabolic responses to engineered pathways. To overcome this problem, the *E. coli* genome scale metabolic network reconstruction (iML1515) is simulated to estimate metabolic network capabilities (subject to the experimental genetic modifications and bioprocess conditions) (Equation 3.1~3.5). The results of the simulations are used as additional features for training the machine learning models. The hybrid of constraint-based simulation with machine learning provides more realistic estimation of cell performance.


A
1.0
0.5
0.0
−0.5
−1.0
Component 1 (5.0%)
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
Component 0 (8.0%)
103.6 – 129.5 g/L
77.7 – 103.6 g/L
51.8 – 77.7 g/L
25.9 – 51.8 g/L
0.0 – 25.9 g/L

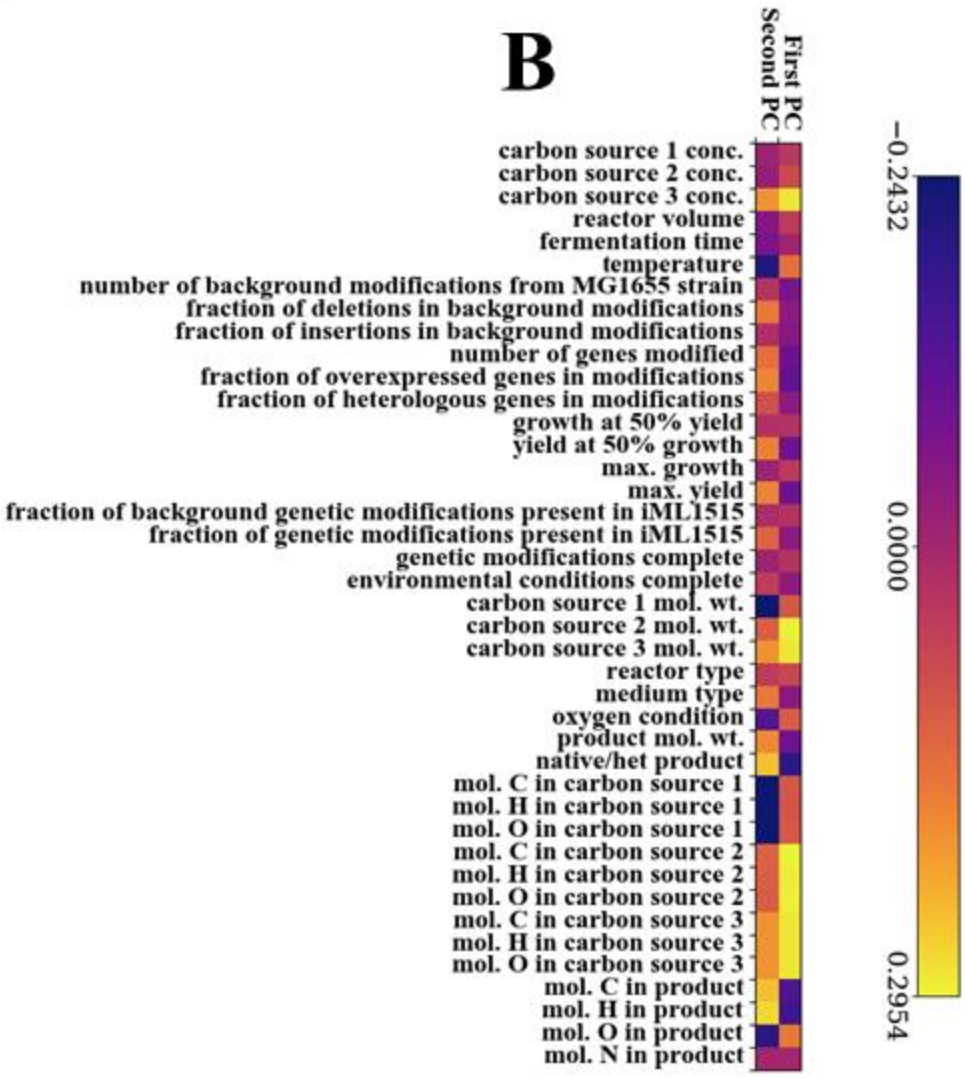


Figure 3.6 Inferring possible influential factors on metabolic engineering design performance A. First two principal components from multiple correspondence analysis (MCA). The labels correspond to titer values in g/L.The shaded areas for each point show the predicted area within which all points have a high probability of belonging to the specified titer range. **B. Impact of different influential factors on first two principal components from principal component analysis (PCA)**. Carbon source 1, 2 and 3 are used to capture the cases in which more than one carbon source was used. If only one was used, corresponding entries of carbon source 2 and 3 were set to zero. *E.coli* MG1655 was taken as the reference strain and all modifications done to get the background strain used in each study were captured as 'background modifications'. The scores describe the relative contribution of each feature to the principal components.

3.3.3 Model performance validation

The predictive ability of the machine learning model on the test dataset (no previously seen by the model) is shown Fig. 3.7 and 3.8. Despite the small dataset size (~1200) from many different studies (~120), the predictive performance of the model is remarkably high for native and non-native *E. coli* products. The use of techniques such as data augmentation and stacked regression (discussed in the methods section) significantly improve model performance. The model also does well for products with wide ranges of titer, rate, or yield values (for example, L-lactate and succinate). The use of extra features from constraint-based simulations as well as ensemble learning of different machine learning models improves predictive performance (Fig. 3.9). Some models (like Extreme Gradient boosted trees, which is itself an ensemble technique) give good performance for one metric but not others. Other like Support Vector Machines (SVMs) give high test scores but the cross-validation accuracies are not robust, showing the model might not generalize well to new data not seen by the model. The final model (stacked regressor) gives a balanced performance across all metrics TRY.

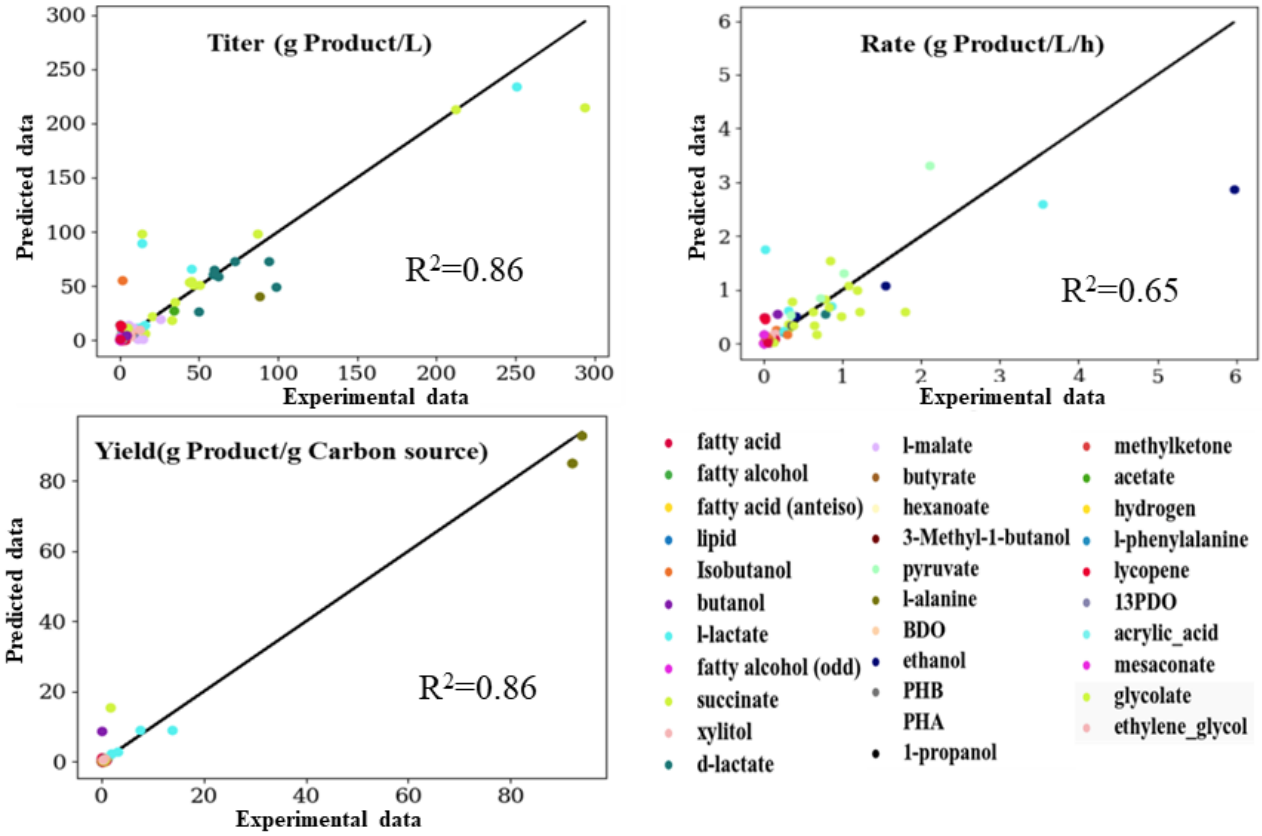


Figure 3.7 Prediction of production metrics TRY R^2: coefficient of determination. Solid lines are shown on the diagonal that represent where all the points would fall for perfect prediction. A scaled version of this figure is presented in Fig. 3.8 (enabling the fit to visualized without the outlier effects). The data points are scaled based on the maximum value (titer, rate or yield) for the particular product in our curated database.

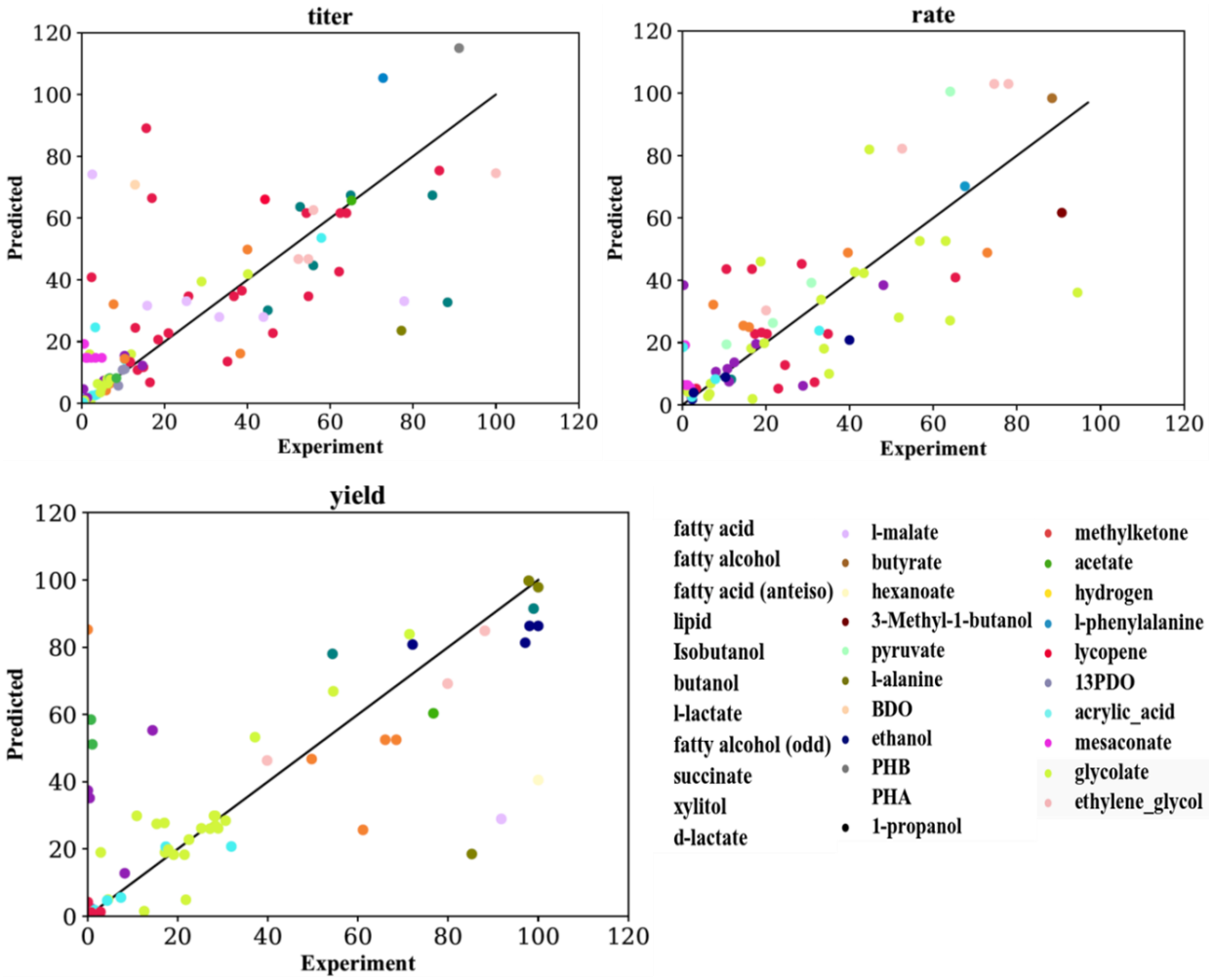
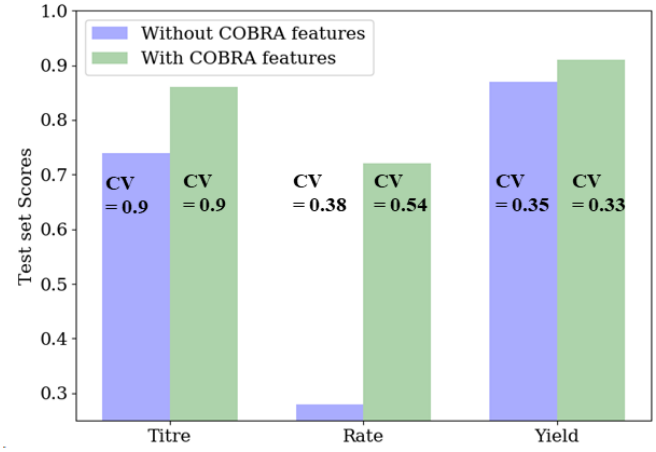


Figure 3.8 Prediction of production metrics (titer, yield and rate) The yield, titer and rate are scaled by the maximum reported values for each product in our curated database.

A



B

Model		Titer	Rate	Yield
Elastic net (EN)	CV	0.2	0.22	0.14
	TS	0.27	0.51	0.55
K nearest neighbors(kNN)	CV	0.76	0.29	0.19
	TS	0.91	0.66	0.88
Support vector machines (SVM)	CV	0.85	0.02	-0.19
	TS	0.88	0.77	0.73
Extreme Gradient Boosted Trees (XGB)	CV	0.83	0.17	0.23
	TS	0.75	0.21	0.52
Neural networks (NN)	CV	0.49	0.24	0.4
	TS	0.69	0.33	0.79
Ensemble model	CV	0.9	0.54	0.33
	TS	0.82	0.64	0.46

Figure 3.9 Model performance analyses A. Quantification of the effect of COBRA (Constraint-Based Reconstruction and Analysis) - based features on model performance. CV stands for the best cross validation accuracy (R^2 values). Higher scores imply a better fit.**B. Comparing individual machine learning performance with ensemble model.** TS stands for Test Scores (R^2 values). CV stands for the best cross validation accuracy (R^2 values). Higher scores imply a better fit.

3.3.4 Model improvement

While there is a good correlation between experiment and model prediction, cross validation analyses reveal variability in model predictions. There are three limitations for machine learning approaches. First, data extractions and curations from published data are prohibitively time-consuming. This is because metabolic engineering papers do not have standard reports of yield/titer and cell productivity can be strikingly different under different growth stages. Manual estimation of production metrics from incomplete published datasets contains human or subjective errors. Second, fermentation media are often undefined (with significant amount yeast extract or other secondary substrates), which make yield calculation inaccurate (i.e., the model predictions on production rate and yield are subpar to titer). Third, our data size and extracted features are still limited, and there are other influential factors (such as waste byproduct secretion during fermentation and strain stability) that are ignored during data curations. Therefore, high-accuracy computational methods for predicting complex cellular phenomena under bioprocess conditions remain challenging. Much efforts and resources must be devoted to data curation, feature extractions, and tailoring of machine learning techniques for application to metabolic engineering data. For example, learning curves demonstrate the possibility of more robust model predictions with larger datasets (Fig. 3.10). Learning curves for yield and rate are shown in Figs 3.11 and 12.

Figure 3.10 Titer learning curve as the function of size of training data set The training scores (R^2) and cross validation (CV) scores (also R^2) are shown. Below 800 training examples, the cross-validation accuracies variation were too large. The hybrid model can fit the training data set (red points) well irrespective of the number of training examples. The cross-validation scores improve slightly with more data points. This implies that more feature engineering (and not necessarily more data) would be necessary to significantly improve model performance.

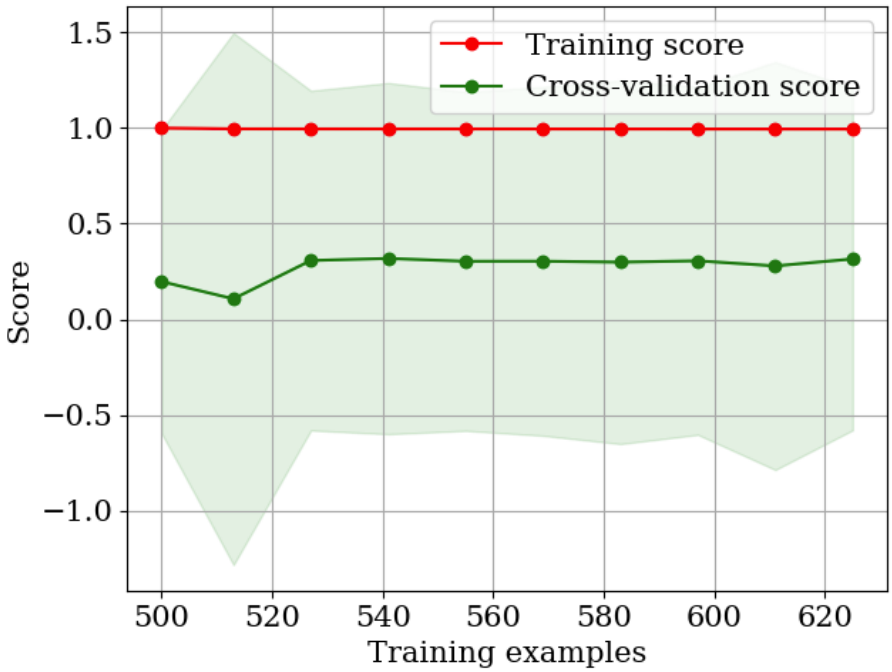


Figure 3.11 Rate learning curve

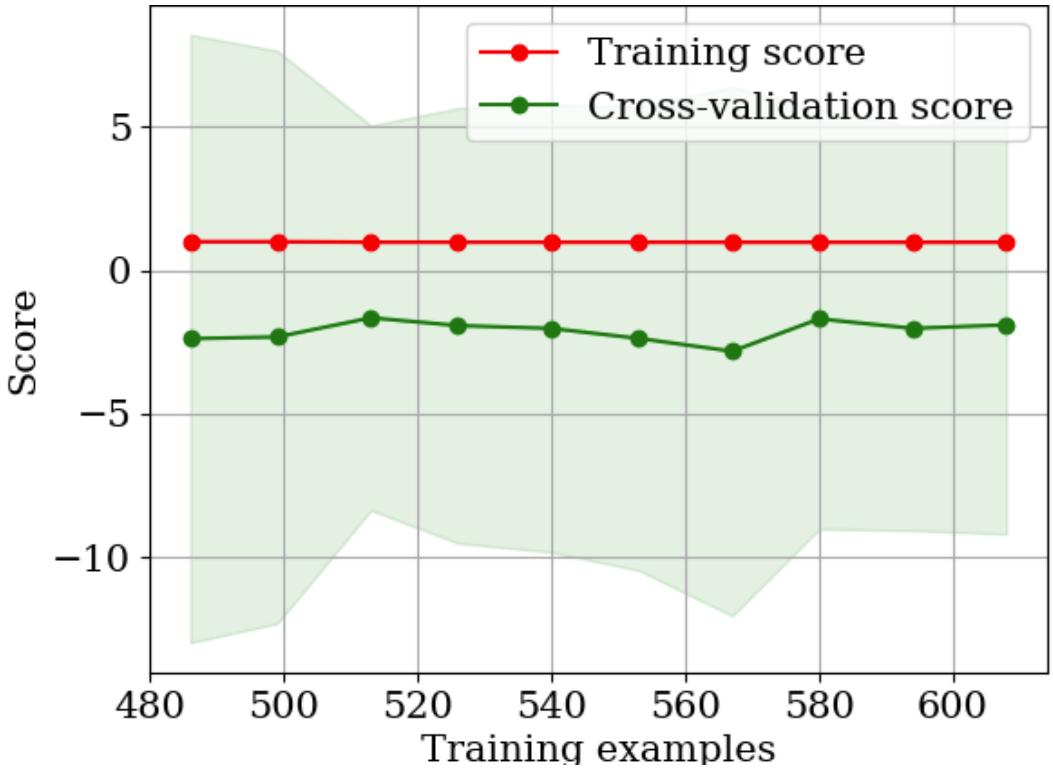


Figure 3.12 Yield learning curve

Chapter 4: Thermodynamic framework for mutant phenotype prediction[4]

4.1 Introduction

It is critical for metabolic engineers to gain a detailed understanding of the cellular regulatory systems involved in routing of matter and energy through the different metabolic pathways. Such understanding would include the role that cellular events (like transcription and posttranscriptional regulation, structural modifications of enzymes, and feedback inhibition) play in control of flux [103]. Genetic and environmental perturbations have been employed to generate insights into transcriptional regulation [104], adaptive evolution responses [105], and metabolic network robustness [106]. For instance, the construction of the Keio library, which contains flux information on single gene knockout (KO) *E. coli* mutants [107], is helping to guide these efforts. However, intracellular flux distributions in microbes have complex responses to genetic and environmental conditions [108]. To facilitate determination of the metabolic flux redistribution within mutants, computational methods have been developed. The most prominent computational tools used are constraint-based reconstruction and analysis (COBRA) techniques. COBRA-based techniques require only the metabolic network stoichiometry and a defined 'objective function'[109] to predict cellular fluxes and have been extensively used to guide metabolic engineering [110], drug discovery [111], and adaptive evolution studies [112]. A

[4] This chapter is adapted from my manuscript: Oyetunde, T., Fatehi A.,Czajka J., and Tang Y.J Thermodynamic framework for mutant phenotype prediction **BMC Systems Biology** (submitted)

variety of COBRA techniques have been developed, including flux balance analysis (FBA), minimizations of metabolic adjustment (MOMA) [113], regulatory on/off minimization of metabolic fluxes (ROOM) [114], and relative optimality in metabolic networks (RELATCH) [115] (Fig. 4.1). All these techniques rely on the hypothesis that the goal of cellular regulation is to maintain a flux distribution as close as possible to a desired state. FBA is based on the assumption that cellular metabolism has evolved to favor some predefined objective function (usually optimal biomass growth). MOMA and ROOM attempt to improve upon the FBA by utilizing experimental ^{13}C MFA measurements of the wild-type flux distribution as the desired metabolic state. MOMA and ROOM algorithms attempt to minimize the Euclidean and Hamming distances, respectively, between the mutant flux distribution and the wild-type [114]. MOMA tends to favor small changes in the mutant's metabolic flux network, while ROOM minimizes the number of significant changes. RELATCH uses wild type gene expression data to improve the characterization of the desired metabolic state [115]. FBA predictions have been reasonably accurate for mutant strains that have undergone adaptive evolution and which are growing under optimal conditions [116], [117].

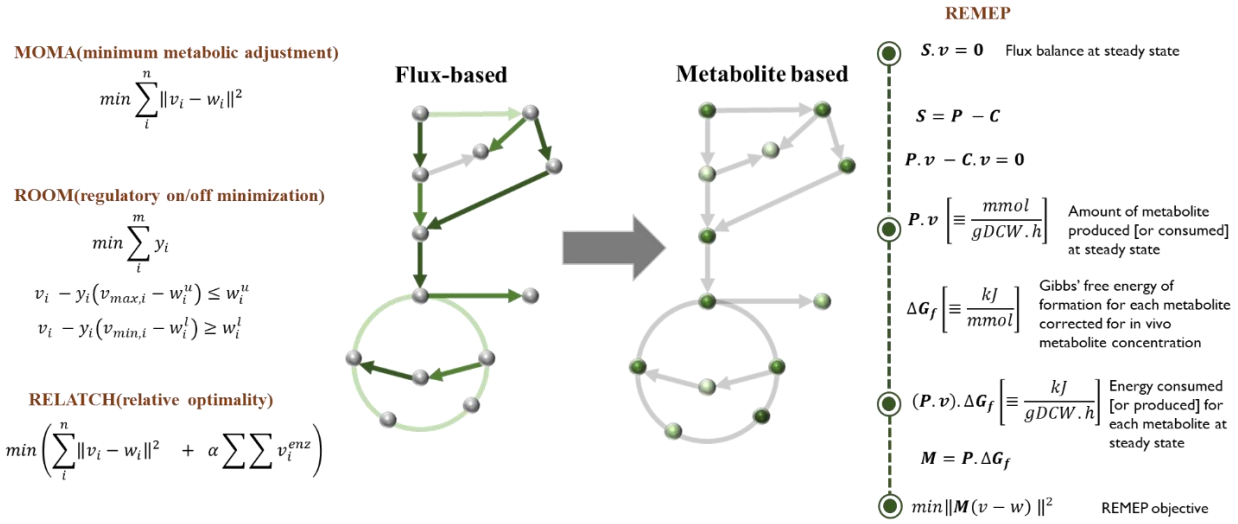


Figure 4.1 Comparison of flux-based to metabolite based mutant prediction algorithms MOMA, RELATCH and ROOM abstract cellular regulation as an attempt to conserve flux patterns (A). MOMA in particular minimizes the Euclidean norm between wild type and mutant flux distributions, ROOM minimizes the number of largest changes in flux distribution and RELATCH minimizes flux distributions with an additional constraint/objective function term based on gene expression B) REMEP hypothesizes that metabolite patterns based on thermodynamics provide additional information for understanding cellular regulation. See methods section for detailed explanation of the method. Arrows represent the fluxes through the enzyme, circles represent the metabolite pool, and the colors represent the conserved portion in each method.

MOMA, ROOM and RELATCH have shown further improvement in predictive accuracy. However, there are still discrepancies between experimentally determined and computationally predicted flux distributions [108]. These discrepancies imply that these models do not capture all mechanisms for organizing fluxomes such as transcriptional, translational and allosteric regulations. Arguments from metabolic control theory have demonstrated that environmental perturbations tend to result in small changes in metabolite concentrations [118] as shown in Fig. 4.2. The distribution of absolute percent changes in gene expression levels, metabolite concentrations and fluxes are plotted for *E. coli* mutants under different conditions[27], [106] where environmental perturbations affect metabolite concentrations much less than genetic perturbations. Furthermore, it has been noted that absolute metabolite concentrations and Gibbs free energies are conserved across species and that the metabolite concentrations are usually larger than the associated kinetic parameters which corresponds to an evolutionary drive to utilize enzymes efficiently [119]. Taken together, this suggests that significant changes in metabolite levels only occur when the perturbation hampers the ability of the cell to modulate its enzyme levels in such way as to minimize changes in metabolite concentrations (usually because of a gene deletion or a severe change in environmental conditions). Network-embedded

thermodynamic analysis (NET) [120] show that genetic knockouts that leads to significant changes in Gibbs free energy of intracellular reactions may induce strong perturbations of metabolite levels (e.g., pgi). On the other hand, switching carbon sources results in small impacts on Gibbs free energy of intracellular reactions leading to minimal changes in metabolite concentrations (Fig. 4.2). Therefore, thermodynamic analyses can be exploited to gain insights into metabolic reorganization upon perturbation [121]–[123]. Here, we present REMEP method for prediction of flux distributions in perturbed cells. REMEP relies on the assumption that Gibbs' free energy profiles for metabolite turnovers, in addition to flux patterns, are informative of cellular regulatory mechanisms, and thus would prove useful in predicting the phenotypic effects of genetic and environmental perturbations. Therefore, the REMEP algorithm is proposed and compared with different methods on experimental knockout data of *E. coli* and *S. cerevisiae* grown in batch and continuous cultures.

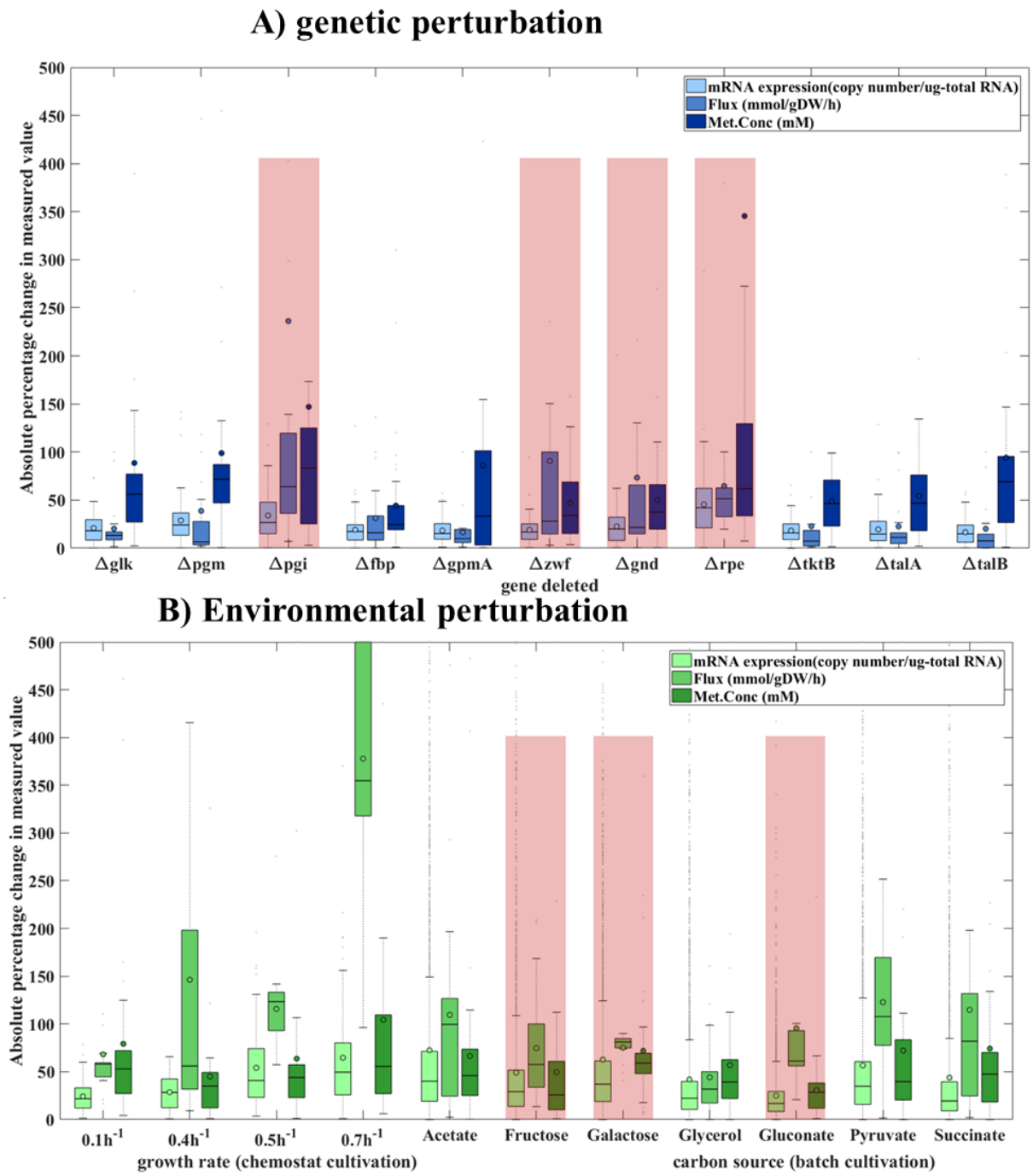


Figure 4.2 Comparing the effects of genetic and environmental perturbations on gene expression, metabolite concentrations and intracellular flux The distribution of the absolute percent changes in experimentally measured quantities is shown as box plots. Data in A) taken from [106]. A plot of all genetic knockouts studied is shown in Figure 4. The genetic knockouts with the highest number of significant changes in the range of Gibbs free energy of

reaction are highlighted (pgi, zwf, gnd and rpe) For B), data on growth rate perturbations is taken from [106] while that for carbon source perturbations is taken from [27]. The carbon source perturbations with the least number of significant changes in the range of Gibbs free energy of reaction are highlighted (fructose, galactose and gluconate). The ranges of feasible Gibbs free energy of reaction are computed by network-embedded thermodynamic analysis as described in [120].

4.2 Methodology

4.2.1 Mathematical formulation of REMEP

Consider a $\boldsymbol{m}$ by $\boldsymbol{n}$ stoichiometric matrix S* representing the metabolism of an organism with $\boldsymbol{m}$ metabolites and $\boldsymbol{n}$ reactions such that at steady state the following equation is fulfilled:

$$\boldsymbol{S}^{*}.\boldsymbol{v}^{*} = 0 \qquad (4.1)$$

Where $\boldsymbol{v}^{*}$ is the vector of reactions (fluxes), including both reversible and irreversible ones. We can rewrite each reversible flux in $\boldsymbol{v}^{*}$ as the difference between two irreversible fluxes and expand $\boldsymbol{S}^{*}$ accordingly, so we have:

$$\boldsymbol{S}.\boldsymbol{v} = 0 \qquad (4.2)$$

Where $\boldsymbol{S}$ is m by ($\boldsymbol{n}$+$\boldsymbol{r}$) matrix and $\boldsymbol{w}$ is ($\boldsymbol{n}$+$\boldsymbol{r}$) vector, $\boldsymbol{r}$ being the number of reversible reactions.

Furthermore, for each metabolite i, we can write a vector $\boldsymbol{Pi}$ consisting of only the positive elements in the row $\boldsymbol{i}$ of $\boldsymbol{S}$ (that is, reactions producing the metabolite). We could thus construct a matrix $\boldsymbol{P}$, such that

$$\boldsymbol{P}.\boldsymbol{v} = \boldsymbol{d} \qquad (4.3a)$$

Where each element in vector $\boldsymbol{d}$ represents the total amount of producing flux through a metabolite.

If each row in matrix P is multiplied by the Gibbs' free energy of formation of the corresponding metabolite (ΔG_f), then vector d, corresponds to the energy per unit biomass required to produce and consume each metabolite. Thus, we construct matrix M of energy flows as follows:

$$\boldsymbol{M} = \boldsymbol{P}.\Delta\boldsymbol{G_f} \tag{4.3b}$$

REMEP minimizes the difference between metabolite energetic requirements (i.e., energy flows) for mutant and wild type strains by solving the following nonlinear optimization problem:

$$min\|\boldsymbol{M}.\boldsymbol{v} - \boldsymbol{d}^*\|^2$$

Subject to:

$$\boldsymbol{S}.\boldsymbol{v} = 0 \tag{4.4}$$

$$\boldsymbol{0} \leq \boldsymbol{v} \leq \boldsymbol{ub}$$

Where ub is the upper bound vector for the set of irreversible fluxes. **d*** refers the **d** computed from the wildtype flux distribution. Scaled versions of the objective function could be used such as:

$$min \left\| \frac{\boldsymbol{M}.\boldsymbol{v}}{\sum \boldsymbol{M}.\boldsymbol{v}} - \frac{\boldsymbol{d}^*}{\sum \boldsymbol{d}^*} \right\|^2 \tag{4.5}$$

Minimization of the difference between biomass growth of wild type and mutant strains could also be added as an extra row in M. The values in the upper bound vector ***ub*** can be set based on experimental information. For example, if a reaction was knocked out, the corresponding

element in ***ub*** would be set to zero. Details of the REMEP algorithm and solution procedure are described in Figs 4.3 and 4.4.

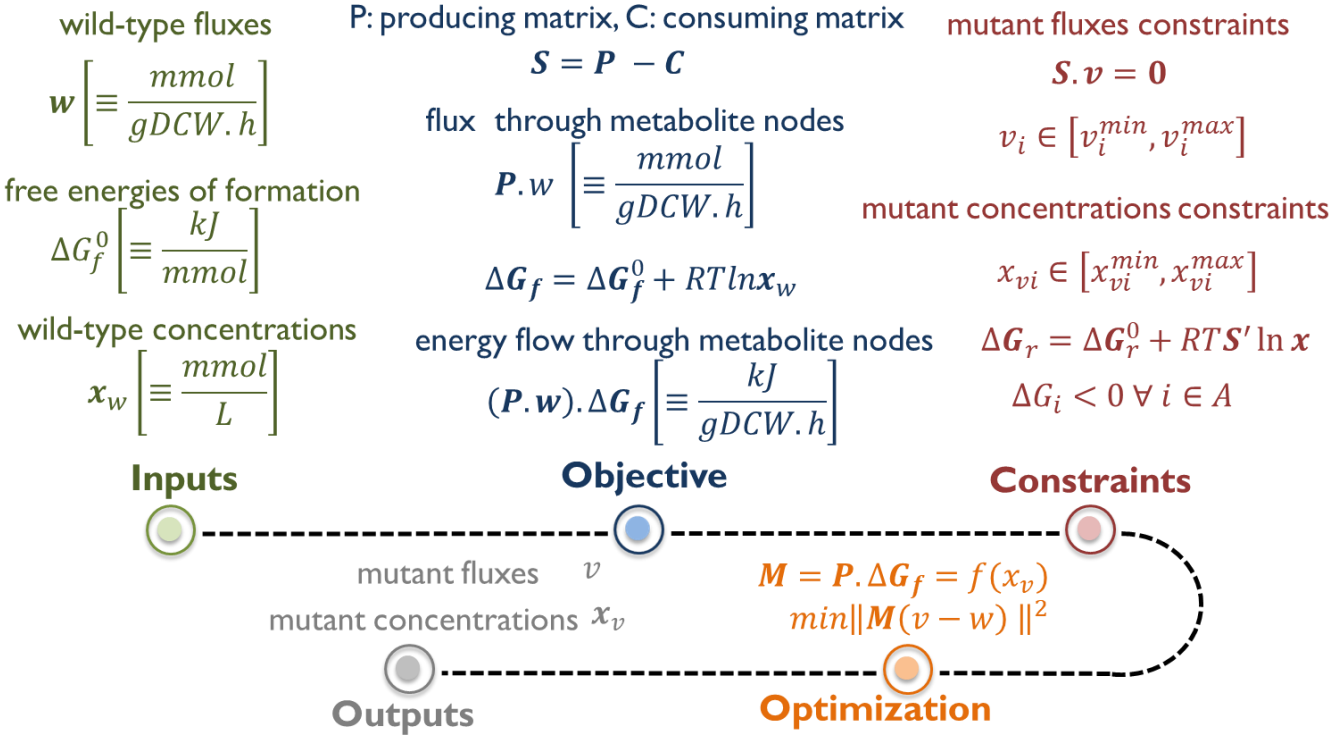


Figure 4.3 The REMEP algorithm workflow

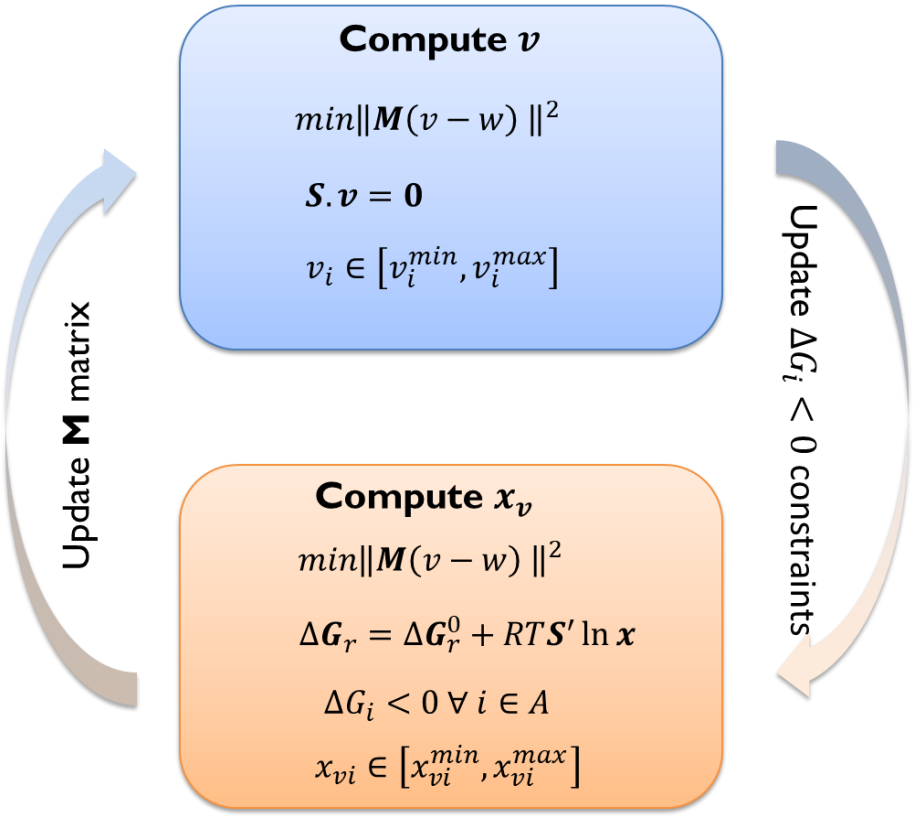


Figure 4.4 REMEP's two-step iterative solution procedure To avoid long computational times with the original nonlinear formulation of REMEP, the above procedure is used. In the first step, the fluxes are computed by minimizing the objective function and assuming no change in metabolite concentrations. In the second step, a metabolite concentration profile consistent with the thermodynamic constraints implied by the fluxes is computed.

Conceptually, REMEP generalizes earlier frameworks by providing a rational basis for weighted minimization of the differences between mutant and wild type flux distribution. Thus, all fluxes are equal, but some are 'more equal' than others based on their contributions to the underlying metabolite patterns that represent the cellular regulatory structure.

4.2.2 Description of computational experiments

We compared the predictions of the REMEP method to existing algorithms (FBA, MOMA and RELATCH) using knockout datasets of *E. coli* and *S. cerevisiae* strains. ROOM was not used because its performance is not significantly better than MOMA (for unevolved mutants) or FBA (for adaptively evolved mutants or mutants grown in chemostats). Genome-scale models of *E. coli* (iAF1260) and *S. cerevisiae* (iMM904) were downloaded from the BiGG database [37]. The gene expression data for *E. coli* [124] and *S. cerevisiae* [125] needed for RELATCH computations were obtained from previously published work. Mutant flux distributions for *E. coli* [104], [106], [109], [126], [127] and *S. cerevisiae* [128] were obtained from literature. All simulations were performed in MATLAB 2016a. The COBRA toolbox implementations of FBA and MOMA were used to obtain predictions for the models. The RELATCH program was downloaded from http://reedlab.che.wisc.edu/ [115].

4.3 Results

4.3.1 E. coli mutants

FBA, MOMA, RELATCH and REMEP can predict flux re-organizations after gene knockouts. The flux data for four single gene knockouts in *E. coli* grown on glucose in a batch reactor was previously reported based on ^{13}C metabolic flux analysis [126]. Fig. 4.5 compares the qualitative behavior of four phenotype prediction algorithms on the *pgi* mutant (all the mutants from the paper are shown in Fig. 4.6).

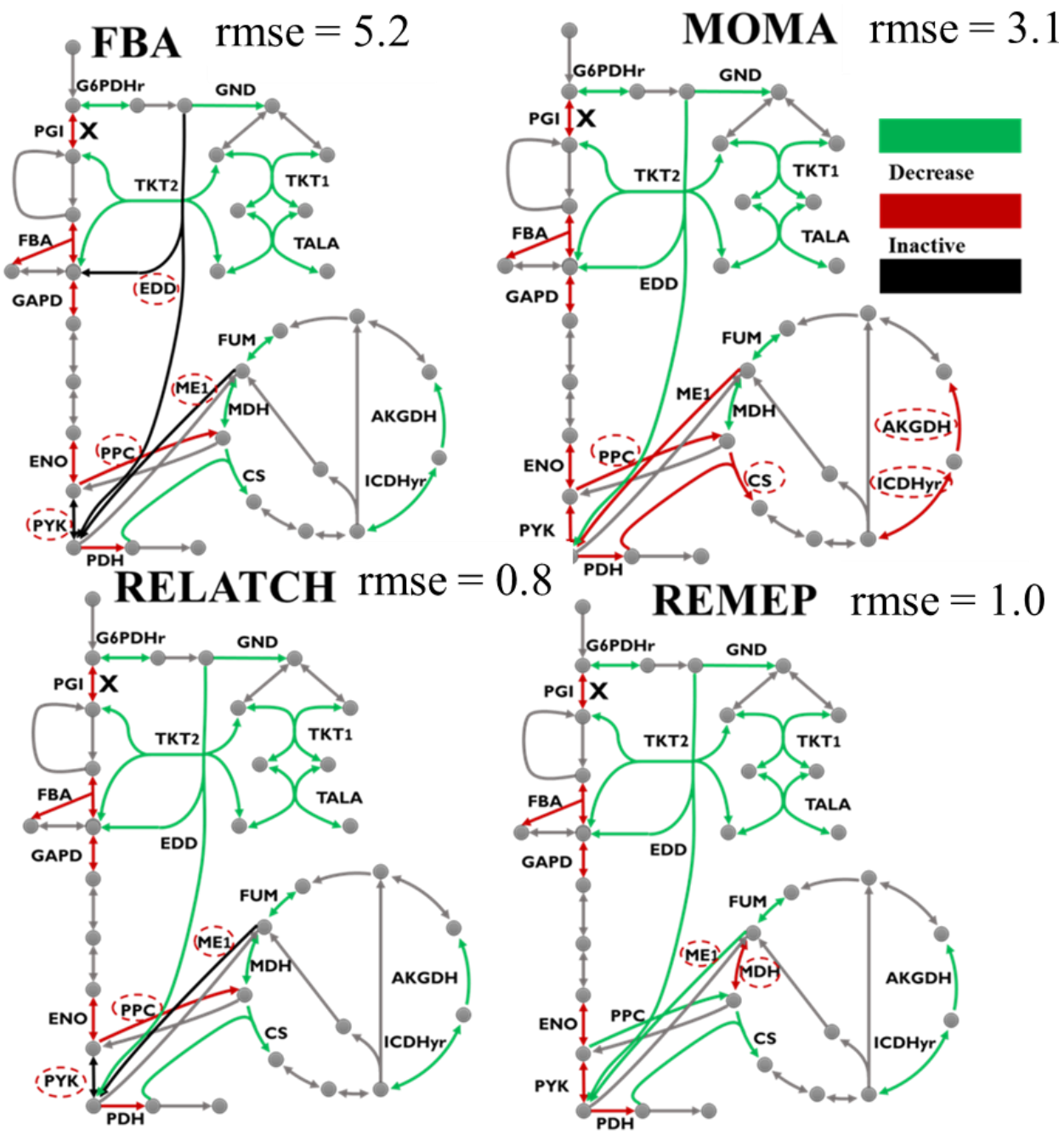


Figure 4.5 Predicted changes in *E. coli*'s central metabolism upon knockout of pgi gene The color code shows if there is a percentage increase/decrease in the fractional usage of the reaction relative to the wild type flux distribution. Reactions are circled in red dashes when the qualitative change is not in agreement with experimental data. Experimental information is not available for reactions colored grey. Other knockouts studied in the same experimental paper are presented in Figure 4.6.

Almost all the algorithms correctly predict the reallocation of flux in pathways near the gene knocked out: the oxidative pentose phosphate pathway and the Entner-Doudoroff (ED) pathway (The FBA model does not capture the increase in flux through the ED pathway). The challenge for the algorithms is predicting reaction fluxes further downstream from the point of genetic knockout. For example, MOMA does not capture the decrease of flux in some reactions in the tricarboxylic acid cycle (TCA), as it is only trying to reallocate flux in order to minimize the difference between wild type and mutant fluxes. RELATCH, which uses gene expression data, shows a better performance although it also makes incorrect predictions on a few downstream reactions. REMEP performs well in predicting downstream fluxes except for the increase and decrease in ME1 (malic enzyme) and MDH (malate dehydrogenase) fluxes. Interestingly, both fluxes pass through the same node (malate), which is a key branched node in the TCA cycle and anaplerotic pathways (e.g., glyoxylate shunt and malic enzyme reactions). REMEP is also the only algorithm to correctly predict the increase in PPC flux. REMEP shows consistently high correlation with experimental data. The REMEP prediction is better than the FBA and MOMA models, and on par with the RELATCH predictions using experimental measurements of fluxes in the central metabolism. By focusing on metabolite patterns rather than flux patterns, REMEP can capture subtleties in cellular regulation that are not possible with the earlier methods, which are based on the conservation of flux patterns between mutant and wild type strains.

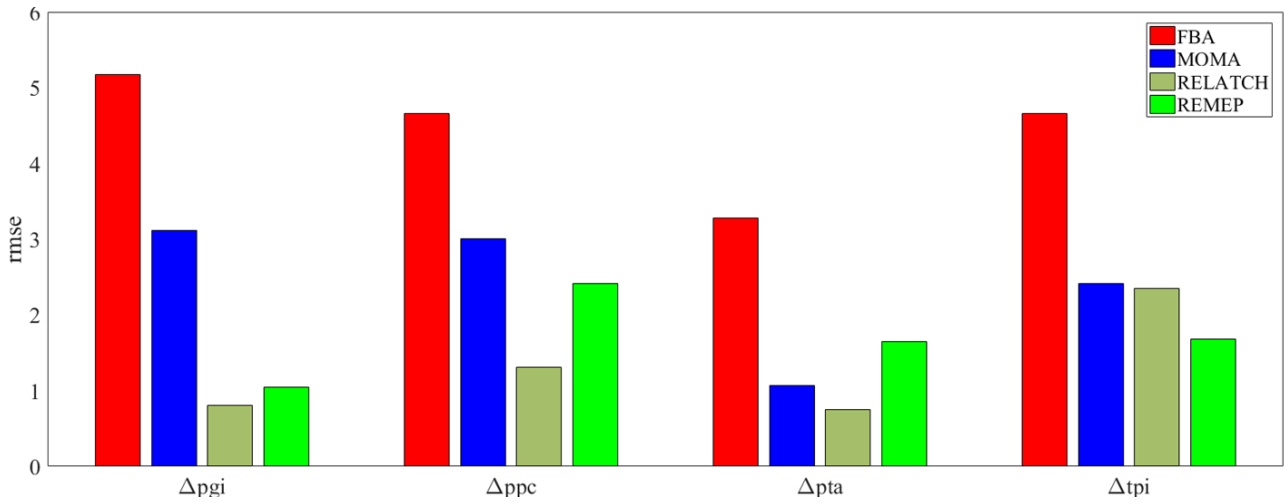


Figure 4.6 Comparison of different phenotype prediction algorithms on *E. coli* mutant strains rmse is the root mean square error.

Fig. 4.7 shows how the algorithms compare when predicting intracellular flux profiles of *E. coli* grown in a batch reactor with different carbon sources [27].

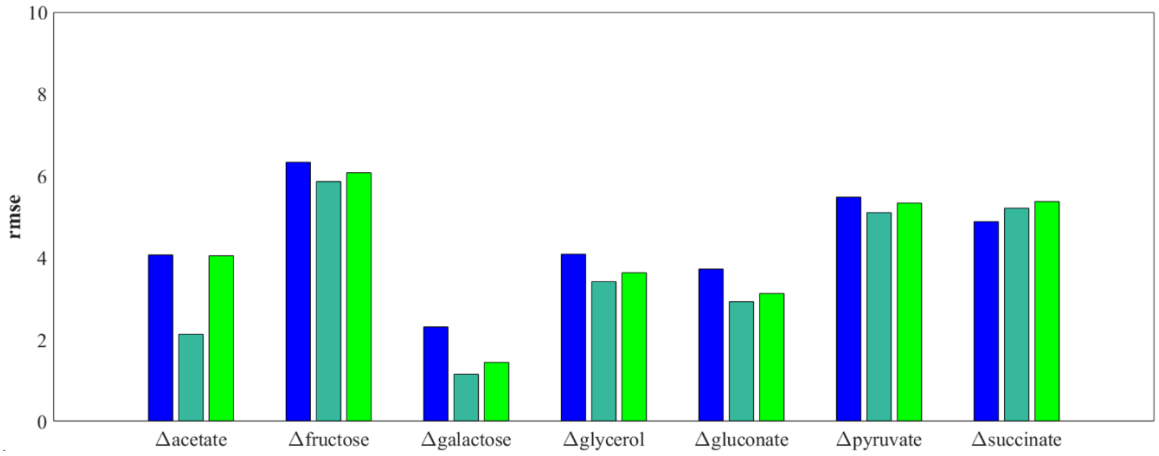


Figure 4.7 Predicting the effect of changing carbon sources rmse is the root mean square error.

4.3.3 S. cerevisiae mutants

REMEP also works well for knockout predictions of eukaryotic strains as shown in Fig. 4.8. Fig. 4.6 shows the comparison of the RELATCH and REMEP algorithms' flux predictions for single

gene knockouts performed in *S. cerevisiae* [125]. In general, REMEP model predictions are better or on par with RELATCH, this indicates that

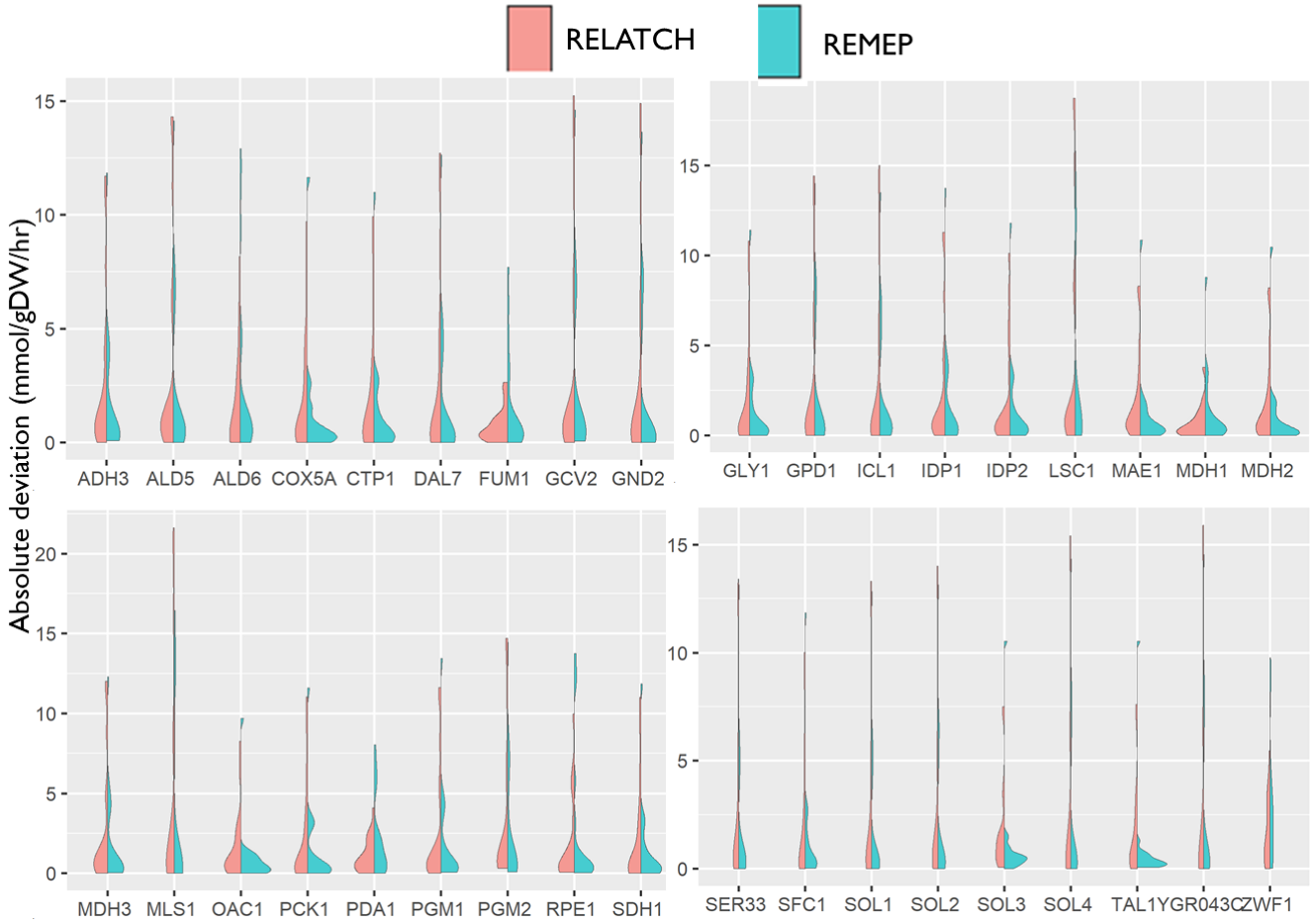


Figure 4.8 Beanplot comparison of RELATCH and REMEP on S. cerevisiae mutant strains The plots show the distribution of the deviations between measured and predicted flux values.

4.4 Discussion

Microbes catabolize carbon sources into a few essential metabolites as intermediates for the synthesis of building blocks through central metabolic pathways. The flux split ratio around those metabolites demonstrates relative robustness upon genetic variations [129]. Based on such observations, a few methods have attempted to characterize the regulatory behavior of cellular

metabolism [130]–[134]. As many computational strain design tools rely on mutant prediction algorithms, it is important to have an algorithm that accurately reflects the cellular regulatory structure. REMEP aims to fulfill that objective by capturing cellular regulatory behavior encoded in fluxes through metabolite nodes and patterns based on energetic requirements, which have been shown to contain useful information about cellular function and evolutionary trends [80]. Moreover, metabolite-centric (rather than pathway-centric) approaches to study metabolic networks (27) have gained attention in recent years. These approaches have been shown to be informative in guiding strain improvement (41) and identification of drug targets (28). For example, it has been observed that the summation of all incoming (or outgoing) fluxes around essential metabolites are relatively conserved under severe perturbations (42). We demonstrated the utility of metabolite patterns to the classic problem of gap filling of genome-scale metabolic network reconstructions [62]. We have employed REMEP for predicting the effects of genetic and environmental perturbations. We also highlight the fact that the hypothesis made by different mutant prediction algorithms implies a cellular regulatory structure pattern. This is demonstrated in Fig.4.9 where we show the models' percentage change in the usage of selected reactions in central metabolism of *E. coli* and *S. cerevisiae* after genetic knockout (based on experimental ^{13}C MFA data from [106]). A key difference between *E. coli* and *S. cerevisiae* is that the flux distribution changes more significantly in *E. coli* than in *S. cerevisiae* upon genetic modification. Thus, the cellular regulatory structure of prokaryotes is predicted to be more flexible (network plasticity) to genetic knockouts than eukaryotes that have cellular compartments and complex regulations (network rigidity). Moreover, we note a similarity between RELATCH and REMEP even though REMEP does not make use of gene expression data. This observation suggests that

most of the information embedded in gene expression may also exist within metabolite patterns. This is possible based on known physicochemical principles (Fig 4.9)

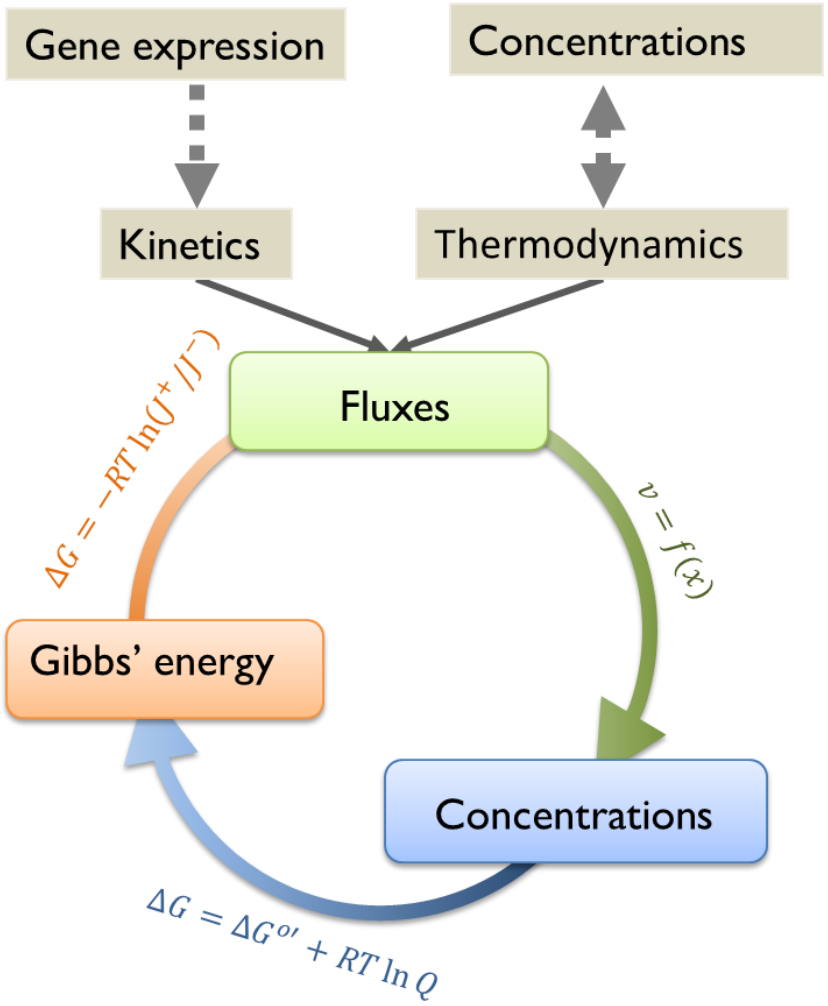


Figure 4.9 Basic physicochemical principles constraining key players in cellular metabolism

Thus, REMEP serves as a useful substitute for RELATCH when gene expression data is absent. REMEP also has a simpler computational layout and it can be easily incorporated into computational strain design tools [43], [135]–[140]. Comparison of heat maps shown in Fig. 4.10 with experimentally generated ones can help pinpoint areas of improvement and refinement for mutant prediction tools.

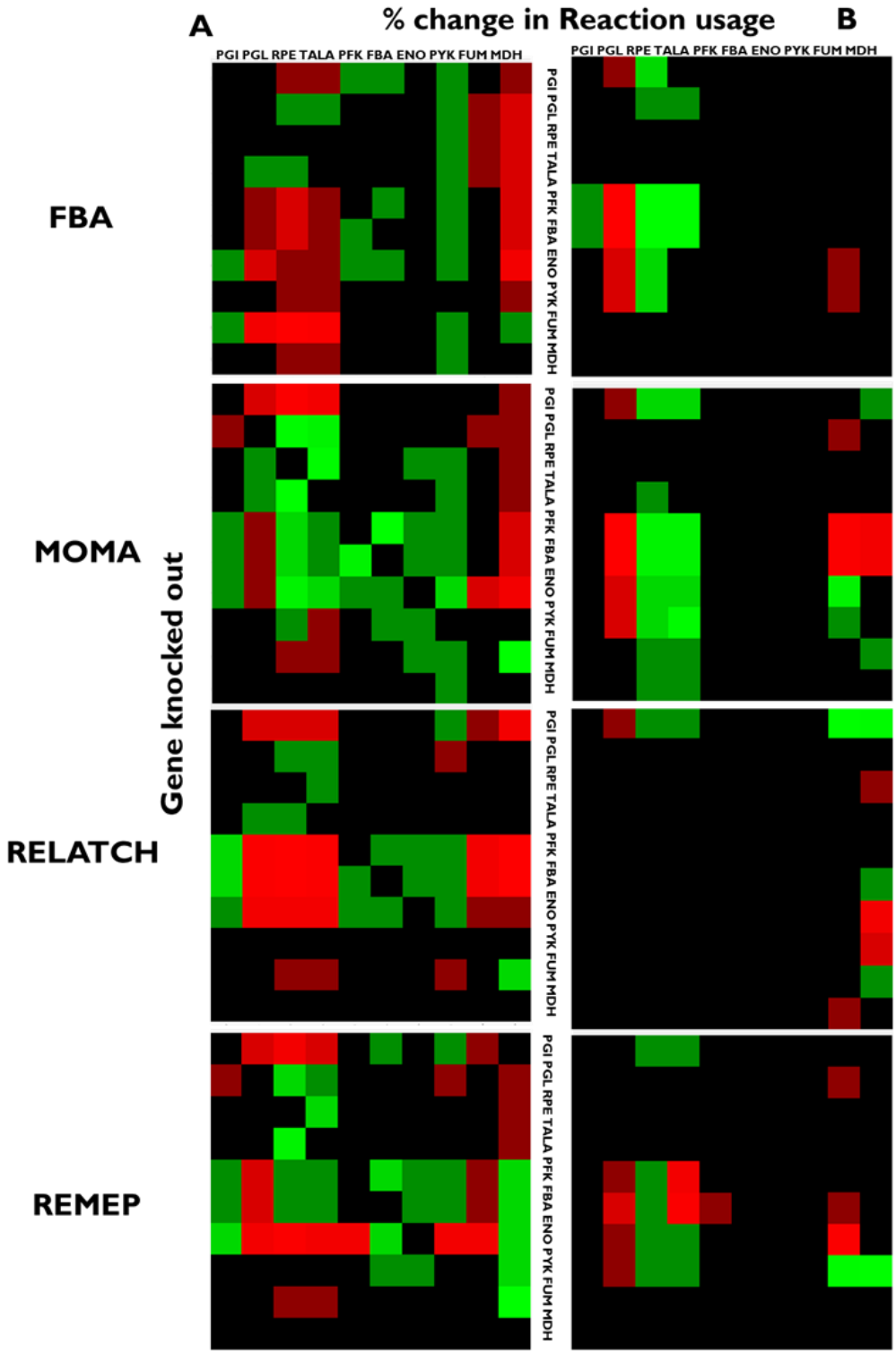


Figure 4.10 Heat Map showing percentage change in selected reactions of central metabolism A) *E. coli* and B) *S. cerevisiae* upon gene knockout. For each simulation, all the genes associated with the metabolic reaction were silenced

Chapter 5: Conclusions

In this thesis, I have demonstrated the benefits of coupling mechanistic modeling with data-driven techniques for enhanced predictive fidelity of complex biological systems. These hybrid frameworks also provide a platform for generating and testing hypotheses about the underlying logic or 'rules' governing all living systems. Below, I highlight the key contributions of the individual projects and mention interesting directions for future work. I also briefly describe the work done during a 6-month data science co-op at Monsanto company (now Bayer Crop Science) which was presented in an internal Monsanto conference.

5.1 Gap filling of metabolic networks

Metabolic network reconstructions are often incomplete. Constraint-based and pattern-based methodologies have been used for automated gap filling of these networks, each with its own strengths and weaknesses. Moreover, since validation of hypotheses made by gap filling tools require experimentation, it is challenging to benchmark performance and make improvements other than that related to speed and scalability.

We developed BoostGAPFILL, an open source tool that leverages both constraint-based and machine learning methodologies for hypotheses generation in gap filling and metabolic model refinement. BoostGAPFILL uses metabolite patterns in the incomplete network captured using a matrix factorization formulation to constrain the set of reactions used to fill gaps in a metabolic network. We formulated a testing framework based on the available metabolic reconstructions and demonstrated the superiority of BoostGAPFILL to state-of-the-art gap filling tools. We randomly delete a number of reactions from a metabolic network and rate the different

algorithms on their ability to both predict the deleted reactions from a universal set and to fill gaps. For most metabolic network reconstructions tested, BoostGAPFILL shows above 60% precision and recall, which is more than twice that of other existing tools.

Approaches that combine machine learning models and pure mechanistic models to describe biological phenomena will prove useful in decoding complex interactions that exist in living systems. Integrating pattern-based methods with constraint-based techniques can potentially enhance their predictive fidelity in computational strain design for metabolic engineering.

5.2 Data-driven computational strain design

Metabolic models can estimate intrinsic product yields from microbial factories, but such frameworks struggle to predict cell performances (including product titer or rate) under suboptimal metabolisms and complex bioprocess conditions. On the other hand, machine learning, complementary to metabolic modeling, relies on having sufficient data. Building such a database for metabolic engineering designs requires significant manpower and is subject to human errors and bias. We proposed an approach to integrate data-driven methods with genome scale metabolic model for assessment of microbial bio-production (yield, titer and rate). Using engineered *E. coli* as an example, we manually extracted and curated dataset of about 1200 experimentally realized cell factories from over 100 papers. We furthermore augment the key design features (e.g., genetic modifications and bioprocess variables) extracted from literature with additional features derived from running genome-scale metabolic model iML1515 simulations with constraints that match the experimental data. Then, data augmentation and ensemble learning (e.g., support vector machines, gradient boosted trees, and neural networks in a stacked regressor model) are employed to alleviate the challenges of sparse, non-standardized,

and incomplete datasets, while multiple correspondence analysis/principal component analysis are used to rank influential factors on bio-productions. The hybrid framework demonstrates a reasonably high cross-validation accuracy for prediction of *E.coli* factory performance metrics under presumed bioprocess and pathway conditions (Pearson correlation coefficients between 0.8 and 0.93 on new data not seen by the model). The learning curve of the hybrid framework can be improved by larger curated data size from references, more feature extractions, and standardized genetic and bioprocess factors from literatures. This proof-of-concept study points a promising direction for designing microbial chemical productions using both mechanistic and data driven models, which can be broadly extended to other platform species.

5.3 Thermodynamic framework for mutant phenotype prediction

Metabolic engineers mainly employ genetic modifications to redirect cellular metabolism towards desired ends. Mutant flux prediction algorithms are the basis of computational strain design tools which help drive rational metabolic engineering. Mutant flux prediction algorithms often have two components: (1) a metric to characterize the cell's desired metabolic state (for example flux or gene expression profiles) and (2) a metric to describe the distance from the desired state (for example, Euclidean distance). The mutant flux profile is computed as the closest possible to the wild type state (which is usually determined experimentally, for example, by ^{13}C-metabolic flux analysis) subject to the constraints of genetic or environmental perturbations).

To improve the fidelity of knockout predictions and subsequent computational strain design, we developed a metabolite-centric approach RElative MEtabolite Patterns (REMEP). REMEP

hypothesizes that the optimum metabolic state is reflected in the energetic requirements to sustain flux through each metabolite node, and thus cell fluxomes adapt to perturbations from a reference state by preserving relative pattern of metabolite energy flows (energy dissipation rates). REMEP performs better than comparable algorithms across different experimental datasets for *E. coli* and *S. cerevisiae* (in terms of lower root mean square errors and higher Pearson's correlation coefficients).

These improvements support the REMEP assumption that cellular mechanisms of response to genetic and environmental perturbations leaves signatures that can be inferred from thermodynamics-derived metabolite patterns. The findings provide a new paradigm for genotype to phenotype mapping and insights into microbial flux network plasticity. REMEP provides an will prove useful for computational strain design tools as well as for understanding cellular regulation.

5.4 Monsanto co-op experience

I designed and implemented a novel marker picking algorithm for the molecular breeding pipeline. The algorithm had two parts: 1) a framework for condensing available information from probabilistic marker genotypes into an optimization metric. 2) a scheme for multi-objective, multi-germplasm optimization based on marker informativeness, quality and cost. The tool enables efficient utilization of information from genotyping experiments as well as ensures the cost-effectiveness of marker-assisted selection and back-crossing.

5.5 Recommendations for future work

A lot of further opportunities exist for integrating mechanistic modeling with machine learning techniques to enable practical biotechnological applications. Below are a few suggestions.

5.4.1 Automatic knowledge extraction from metabolic engineering literature

Given the loads of valuable information embedded in thousands of metabolic engineering articles, it would be very beneficial to extract the data without the drudgery and significant manhours associated with manual curation (This will also limit bias and errors that could potentially arise). Interesting progress has been made in automated information extraction from printed text and the metabolic engineering field can leverage these advances.

5.4.2 Multi-omics data integration in a thermodynamic framework

Integrating data from different cellular networks is gaining increasing attention as a means to decipher cellular regulation. Thermodynamics of cellular metabolism integrates fluxomics, metabolomics and kinetics and could potentially provide clues to understand the evolution of cellular functioning and regulation. Several challenges (including the incompleteness and inaccuracies associated with thermodynamic quantities such as Gibbs' free energies of reaction and formation for all the participants in cellular metabolism; inherent stiffness in modeling thermodynamic changes) exist. Nonetheless, recent efforts in literature have highlighted the promise of thermodynamic frameworks in elucidating cellular logic.

5.4.3 Machine learning techniques for 'small' data

Most of the machine learning tools and techniques are designed to take advantage of the explosion of big data in various fields. However, biological data, especially metabolic

engineering data, is usually ‘small’ by the current standards of big data. It is imperative to look at ways to analyze and extract insights from small, disparate and often incomplete biological data. Concepts such as transfer learning and data augmentation will also prove useful in the bid to take advantage of our fragmented understanding and data on biological systems.

5.6 Publications and conference presentations

5.6.1 Publications

1. **Oyetunde, T**., Bao, F. S., Chen, J. W., Martin, H. G., & Tang, Y. J. (2018). Leveraging knowledge engineering and machine learning for microbial bio-manufacturing. **Biotechnology advances.**

2. **Oyetunde, T**., Liu D., Martin H.G., and Tang Y.J Machine learning framework for robust assessment of microbial factory performance. **PLoS ONE** (accepted).

3. **Oyetunde, T**., Zhang, M., Chen, Y., Tang, Y., & Lo, C. (2016). BoostGAPFILL: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods. **Bioinformatics**, 33(4), 608-611.

4. Liu, Z., **Oyetunde, T.,** Hollinshead, W. D., Hermanns, A., Tang, Y. J., Liao, W., & Liu, Y. (2017). Exploring eukaryotic formate metabolisms to enhance microbial growth and lipid accumulation. **Biotechnology for biofuels**, 10(1), 22.

5. Shopera, T., He, L., **Oyetunde, T**., Tang, Y. J., & Moon, T. S. (2017). Decoupling resource-coupled gene expression in living cells**. ACS synthetic biology**, 6(8), 1596-1604.

6. Wu, S. G., Wang, Y., Jiang, W., **Oyetunde, T**., Yao, R., Zhang, X., ... & Bao, F. S. (2016). Rapid prediction of bacterial heterotrophic fluxomics using machine learning and constraint programming. **PLoS computational biology**, 12(4), e1004838.

5.6.2 Conference presentations

7. **Oyetunde, T.,** Czajka J., and Tang Y.J A Deep Learning Framework Decodes Coordination of Microbial Metabolism Under Genetic and Environmental Perturbations presented at the **2017 AIChE Annual Conference**, Minneapolis, MN. (Oct. 29 – Nov. 3, 2017).

8. **Oyetunde, T.,** and Tang Y.J Thermodynamic framework for mutant phenotype prediction to be presented at the **2018 COBRA Conference**, Seattle, WA. (Oct. 14 – Oct. 16, 2018).

9. **Oyetunde, T**, Tang, Y.J. and Lo C.S " Thermodynamic Analysis of the Rigidity of Metabolic Nodes Via a Dynamic Flux Balance Approach" presented at the **2016 AIChE Annual Conference**, San Francisco, CA. (Nov 13-18, 2016).

References

[1] B. Ø. Palsson, *Systems biology: constraint-based reconstruction and analysis*. Cambridge University Press, 2015.

[2] D. Machado, K. H. Zhuang, N. Sonnenschein, and M. J. Herrgård, *Current Challenges in Modeling Cellular Metabolism*, vol. 3. 2015.

[3] N. Razavian, "Applications of Machine Learning in Computational Biology," 2004.

[4] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (80-.).*, vol. 349, no. 6245, 2015.

[5] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nat Rev Genet*, vol. 16, no. 6, pp. 321–332, 2015.

[6] C. Sommer and D. W. Gerlich, "Machine learning in cell biology – teaching computers to recognize phenotypes," *J. Cell Sciience*, vol. 126, no. Pt 24, pp. 5529–5539, 2013.

[7] A. L. Tarca, V. J. Carey, X. Chen, R. Romero, and S. Drăghici, "Machine learning and its applications to biology.," *PLoS Comput. Biol.*, vol. 3, no. 6, p. e116, 2007.

[8] C. Angermueller, T. Pärnamaa, L. Parts, and S. Oliver, "Deep Learning for Computational Biology," *Mol. Syst. Biol.*, no. 12, p. 878, 2016.

[9] S. Y. Lee and H. U. Kim, "Systems strategies for developing industrial microbial strains," *Nat. Biotechnol.*, vol. 33, no. 10, 2015.

[10] S. Parekh, V. a Vinci, and R. J. Strobel, "Improvement of microbial strains and

fermentation processes," *Appl. Microbiol. Biotechnol.*, vol. 54, no. 3, pp. 287–301, 2000.

[11] J. D. Orth, I. Thiele, and B. Ø. Palsson, "What is flux balance analysis?," *Nat. Biotechnol.*, vol. 28, no. 3, pp. 245–8, Mar. 2010.

[12] R. Schuetz, L. Kuepfer, and U. Sauer, "Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli.," *Mol. Syst. Biol.*, vol. 3, no. 119, p. 119, 2007.

[13] S. S. Fong and B. Ø. Palsson, "Metabolic gene-deletion strains of Escherichia coli evolve to computationally predicted growth phenotypes.," *Nat. Genet.*, vol. 36, no. 10, pp. 1056–1058, 2004.

[14] M. R. Long, W. K. Ong, and J. L. Reed, "Computational methods in metabolic engineering for strain design," *Curr. Opin. Biotechnol.*, vol. 34, pp. 135–141, 2015.

[15] R. J. P. van Berlo, D. de Ridder, J.-M. Daran, P. A. S. Daran-Lapujade, B. Teusink, and M. J. T. Reinders, "Predicting metabolic fluxes using gene expression differences as constraints," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 8, no. 1, pp. 206–216, 2011.

[16] S. a. Becker and B. O. Palsson, "Context-specific metabolic networks are consistent with experiments," *PLoS Comput. Biol.*, vol. 4, no. 5, 2008.

[17] H. Zur, E. Ruppin, and T. Shlomi, "iMAT: an integrative metabolic analysis tool," *Bioinformatics*, vol. 26, no. 24, pp. 3140–3142, 2010.

[18] E. J. O'Brien, J. a Lerman, R. L. Chang, D. R. Hyduke, and B. Ø. Palsson, "Genome-scale

models of metabolism and gene expression extend and refine growth phenotype prediction.," *Mol. Syst. Biol.*, vol. 9, no. 693, p. 693, 2013.

[19] C. Colijn, A. Brandes, J. Zucker, D. S. Lun, B. Weiner, M. R. Farhat, T.-Y. Cheng, D. B. Moody, M. Murray, and J. E. Galagan, "Interpreting expression data with metabolic flux models: predicting Mycobacterium tuberculosis mycolic acid production," *PLoS Comput. Biol.*, vol. 5, no. 8, p. e1000489, 2009.

[20] D. Machado and M. Herrgård, "Systematic Evaluation of Methods for Integration of Transcriptomic Data into Constraint-Based Models of Metabolism," *PLoS Comput. Biol.*, vol. 10, no. 4, p. e1003580, 2014.

[21] A. Chowdhury, A. R. Zomorrodi, and C. D. Maranas, "k-OptForce: Integrating Kinetics with Flux Balance Analysis for Strain Design," *PLoS Comput. Biol.*, vol. 10, no. 2, 2014.

[22] T. M. Hoehler and B. B. Jørgensen, "Microbial life under extreme energy limitation.," *Nat. Rev. Microbiol.*, vol. 11, no. 2, pp. 83–94, 2013.

[23] T. Nishizaki, K. Tsuge, M. Itaya, N. Doi, and H. Yanagawa, "Metabolic engineering of carotenoid biosynthesis in Escherichia coli by ordered gene assembly in Bacillus subtilis," *Appl. Environ. Microbiol.*, vol. 73, no. 4, pp. 1355–1361, 2007.

[24] L. Poshyvailo, E. von Lieres, and S. Kondrat, "Does metabolite channeling accelerate enzyme-catalyzed cascade reactions?," *PLoS One*, vol. 12, no. 2, p. e0172673, 2017.

[25] K. Kochanowski, U. Sauer, and V. Chubukov, "Somewhat in control-the role of transcription in regulating microbial metabolic fluxes," *Current Opinion in Biotechnology*, vol. 24, no. 6. 2013.

[26] K. Tummler, T. Lubitz, M. Schelker, and E. Klipp, “New types of experimental data shape the use of enzyme kinetics for dynamic network modeling,” *FEBS J.*, vol. 281, no. 2, pp. 549–571, 2014.

[27] L. Gerosa, B. R. B. Haverkorn Van Rijsewijk, D. Christodoulou, K. Kochanowski, T. S. B. Schmidt, E. Noor, and U. Sauer, “Pseudo-transition Analysis Identifies the Key Regulators of Dynamic Metabolic Adaptations from Steady-State Data,” *Cell Syst.*, vol. 1, no. 4, pp. 270–282, 2015.

[28] S. R. Hackett, V. R. T. Zanotelli, W. Xu, J. Goya, J. O. Park, D. H. Perlman, P. A. Gibney, D. Botstein, J. D. Storey, and J. D. Rabinowitz, “Systems-level analysis of mechanisms regulating yeast metabolic flux,” *Science (80-.).*, vol. 354, no. 6311, 2016.

[29] J. Utrilla, E. J. O’Brien, K. Chen, D. McCloskey, J. Cheung, H. Wang, D. Armenta-Medina, A. M. Feist, and B. O. Palsson, “Global Rebalancing of Cellular Resources by Pleiotropic Point Mutations Illustrates a Multi-scale Mechanism of Adaptive Evolution,” *Cell Syst.*, vol. 2, no. 4, pp. 260–271, 2016.

[30] J. M. Monk, A. Koza, M. A. Campodonico, D. Machado, J. M. Seoane, B. O. Palsson, M. J. Herrg??rd, and A. M. Feist, “Multi-omics Quantification of Species Variation of Escherichia coli Links Molecular Features with Strain Phenotypes,” *Cell Syst.*, vol. 3, no. 3, p. 238–251.e12, 2016.

[31] G. Q. Chen, “Omics Meets Metabolic Pathway Engineering,” *Cell Syst.*, vol. 2, no. 6, pp. 362–363, 2016.

[32] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and

methods," *Data Knowl. Eng.*, vol. 25, no. 1–2, pp. 161–197, 1998.

[33] J. F. Sowa, *Knowledge representation: logical, philosophical, and computational foundations*, vol. 13. MIT Press, 2000.

[34] L. S. Jing, F. F. M. Shah, M. S. Mohamad, N. L. Hamran, A. H. M. Salleh, S. Deris, and H. Alashwal, "Database and tools for metabolic network analysis," *Biotechnol. Bioprocess Eng.*, vol. 19, no. 4, pp. 568–585, 2014.

[35] M. Kanehisa, "The KEGG database," *silico Simul. Biol. Process.*, vol. 247, pp. 91–103, 2002.

[36] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," *Nucleic Acids Res.*, vol. 44, no. D1, 2016.

[37] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D515–D522, 2016.

[38] R. Alcántara, K. B. Axelsen, A. Morgat, E. Belda, E. Coudert, A. Bridge, H. Cao, P. De Matos, M. Ennis, and S. Turner, "Rhea—a manually curated resource of biochemical reactions," *Nucleic Acids Res.*, vol. 40, no. D1, pp. D754–D760, 2011.

[39] Z. Zhang, T. Shen, B. Rui, W. Zhou, X. Zhou, C. Shang, C. Xin, X. Liu, G. Li, J. Jiang, C. Li, R. Li, M. Han, S. You, G. Yu, Y. Yi, H. Wen, Z. Liu, and X. Xie, "CeCaFDB: a curated database for the documentation, visualization and comparative analysis of central

carbon metabolic flux distributions explored by 13C-fluxomics," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D549–D557, 2014.

[40] R. Caspi, R. Billington, L. Ferrer, H. Foerster, C. A. Fulcher, I. M. Keseler, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, P. Subhraveti, D. S. Weaver, and P. D. Karp, "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases," *Nucleic Acids Res.*, vol. 44, no. D1, 2016.

[41] P. D. Karp, R. Billington, R. Caspi, C. A. Fulcher, M. Latendresse, A. Kothari, I. M. Keseler, M. Krummenacker, P. E. Midford, and Q. Ong, "The BioCyc collection of microbial genomes and metabolic pathways," *Brief. Bioinform.*, 2017.

[42] J. D. Winkler, A. L. Halweg-Edwards, and R. T. Gill, "The LASER database: Formalizing design rules for metabolic engineering," *Metab. Eng. Commun.*, vol. 2, pp. 30–38, 2015.

[43] G. Wu, Y. Wang, W. Jiang, T. Oyetunde, R. Yao, X. Zhang, K. Shimizu, Y. J. Tang, and F. S. Bao, "Rapid Prediction of Bacterial Heterotrophic Fluxomics Using Machine Learning and Constraint Programming," *PLoS Comput. Biol.*, vol. 12, no. 4, 2016.

[44] W. Morrell, G. Birkel, M. Forrer, T. Lopez, T. Backman, M. Dussault, C. J. Petzold, E. E. K. Baidoo, Z. Costello, D. Ando, J. Alonso Gutierrez, K. George, A. Mukhopadhyay, I. Vaino, J. D. Keasling, P. D. Adams, N. J. Hillson, and H. Garcia Martin, "The Experiment Data Depot: a web-based software tool for biological experimental data storage, sharing, and visualization," *ACS Synth. Biol.*, p. acssynbio.7b00204, 2017.

[45] C. Allan, J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W.

J. Moore, C. Neves, A. Patterson, M. Porter, A. Tarkowska, B. Loranger, J. Avondo, I. Lagerstedt, L. Lianas, S. Leo, K. Hands, R. T. Hay, A. Patwardhan, C. Best, G. J. Kleywegt, G. Zanetti, and J. R. Swedlow, "OMERO: flexible, model-driven data management for experimental biology," *Nat. Methods*, vol. 9, no. 3, pp. 245–253, 2012.

[46] T. R. Maarleveld, J. Boele, F. J. Bruggeman, and B. Teusink, "A data integration and visualization resource for the metabolic network of Synechocystis sp. PCC 6803," *Plant Physiol.*, p. pp-113, 2014.

[47] A. P. Arkin, R. L. Stevens, R. W. Cottingham, S. Maslov, C. S. Henry, P. Dehal, D. Ware, F. Perez, N. L. Harris, and S. Canon, "The DOE Systems Biology Knowledgebase (KBase)," *bioRxiv*, p. 96354, 2016.

[48] F. Caschera, M. A. Bedau, A. Buchanan, J. Cawse, D. de Lucrezia, G. Gazzola, M. M. Hanczyc, and N. H. Packard, "Coping with complexity: Machine learning optimization of cell-free protein synthesis," *Biotechnol. Bioeng.*, vol. 108, no. 9, pp. 2218–2228, 2011.

[49] J. M. Dale, L. Popescu, and P. D. Karp, "Machine learning methods for metabolic pathway prediction," *BMC Bioinformatics*, vol. 11, no. 1, p. 15, 2010.

[50] J. Kludas, M. Arvas, S. Castillo, T. Pakula, M. Oja, C. Brouard, J. Jäntti, M. Penttilä, and J. Rousu, "Machine learning of protein interactions in fungal secretory pathways," *PLoS One*, vol. 11, no. 7, pp. 1–20, 2016.

[51] Y. Wang, J. Song, T. T. Marquez-lago, A. Leier, C. Li, G. I. Webb, and H. Shen, "Knowledge-transfer learning for prediction of matrix metalloprotease substrate-cleavage sites," no. January, pp. 1–15, 2017.

[52] J. S. M. Pappu and S. N. Gummadi, “Modeling and simulation of xylitol production in bioreactor by Debaryomyces nepalensis NCYC 3413 using unstructured and artificial neural network models,” *Bioresour. Technol.*, vol. 220, pp. 490–499, 2016.

[53] A. Amiri, R. Mohamad, R. A. Rahim, R. M. Illias, F. Namvar, J. S. Tan, and S. Abbasiliasi, “Cyclodextrin glycosyltransferase biosynthesis improvement by recombinant Lactococcus lactis NZ: NSP: CGT: medium formulation and culture condition optimization,” *Biotechnol. Biotechnol. Equip.*, vol. 29, no. 3, pp. 555–563, 2015.

[54] R. Sinha, S. Singh, and P. Srivastava, “Studies on process optimization methods for rapamycin production using Streptomyces hygroscopicus ATCC 29253,” *Bioprocess Biosyst. Eng.*, vol. 37, no. 5, pp. 829–840, 2014.

[55] G. Wang, B. Xu, and W. Jiang, “SVM Modeling for Glutamic Acid Fermentation Process,” pp. 5551–5555, 2016.

[56] J. Mellor, I. Grigoras, P. Carbonell, and J. L. Faulon, “Semisupervised Gaussian Process for Automated Enzyme Search,” *ACS Synth. Biol.*, vol. 5, no. 6, pp. 518–528, 2016.

[57] J. Sheng, W. Guo, C. Ash, B. Freitas, M. Paoletti, and X. Feng, “Data-Driven Prediction of CRISPR-Based Transcription Regulation for Programmable Control of Metabolic Flux,” *arXiv Prepr. arXiv1704.03027*, 2017.

[58] S. Nandi, A. Subramanian, and R. Sarkar, “An integrative machine learning strategy for improved prediction of essential genes in Escherichia coli metabolism using flux-coupled features,” *Mol. BioSyst.*, 2017.

[59] J. Alonso-Gutierrez, E.-M. Kim, T. S. Batth, N. Cho, Q. Hu, L. J. G. Chan, C. J. Petzold,

N. J. Hillson, P. D. Adams, J. D. Keasling, H. Garcia Martin, and T. S. Lee, “Principal component analysis of proteomics (PCAP) as a tool to direct metabolic engineering,” *Metab. Eng.*, vol. 28, pp. 123–133, 2015.

[60] P. F. Colletti, Y. Goyal, A. M. Varman, X. Feng, B. Wu, and Y. J. Tang, “Evaluating factors that influence microbial synthesis yields by linear regression with numerical and ordinal variables,” *Biotechnol. Bioeng.*, vol. 108, no. 4, pp. 893–901, 2011.

[61] A. M. Varman, Y. Xiao, E. Leonard, and Y. J. Tang, “Statistics-based model for prediction of chemical biosynthesis yield from Saccharomyces cerevisiae.,” *Microb. Cell Fact.*, vol. 10, no. 1, p. 45, 2011.

[62] T. Oyetunde, M. Zhang, Y. Chen, Y. Tang, and C. Lo, “BoostGAPFILL: Improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods,” *Bioinformatics*, p. btw684, 2016.

[63] R. T. Gill, A. L. Halweg-Edwards, A. Clauset, and S. F. Way, “Synthesis aided design: The biological design-build-test engineering paradigm?,” *Biotechnol. Bioeng.*, vol. 113, no. 1, pp. 7–10, 2016.

[64] T. Fuhrer and N. Zamboni, “High-throughput discovery metabolomics,” *Curr. Opin. Biotechnol.*, vol. 31, pp. 73–78, 2015.

[65] J. Heinemann, K. Deng, S. C. C. Shih, J. Gao, P. D. Adams, A. K. Singh, and T. R. Northen, “On-chip integration of droplet microfluidics and nanostructure-initiator mass spectrometry for enzyme screening,” *Lab Chip*, vol. 17, no. 2, pp. 323–331, 2017.

[66] J. Heinemann, B. Noon, D. Willems, K. Budeski, and B. Bothner, “Analysis of raw

biofluids by mass spectrometry using microfluidic diffusion-based separation," *Anal. Methods*, vol. 9, no. 3, pp. 385–392, 2017.

[67] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2014, pp. 533–540.

[68] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.

[69] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[70] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 193–200.

[71] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.

[72] Y. LeCun, Y. Bengio, G. Hinton, L. Y., B. Y., and H. G., "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[73] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[74] M. Kogadeeva and N. Zamboni, "SUMOFLUX: a generalized method for targeted 13C

metabolic flux ratio analysis," *PLoS Comput. Biol.*, vol. 12, no. 9, p. e1005109, 2016.

[75] A. Khodayari, A. R. Zomorrodi, J. C. Liao, and C. D. Maranas, "A kinetic model of Escherichia coli core metabolism satisfying multiple sets of mutant flux data," *Metab. Eng.*, vol. 25, pp. 50–62, 2014.

[76] Z. A. King, E. J. O'Brien, A. M. Feist, and B. O. Palsson, "Literature mining supports a next-generation modeling approach to predict cellular byproduct secretion," *Metab. Eng.*, vol. 39, no. August 2016, pp. 220–227, 2017.

[77] A. Bordbar, J. M. Monk, Z. a King, and B. O. Palsson, "Constraint-based models predict metabolic and associated cellular functions.," *Nat. Rev. Genet.*, vol. 15, no. 2, pp. 107–20, 2014.

[78] J. D. Orth and B. Palsson, "Systematizing the generation of missing metabolic knowledge," *Biotechnol. Bioeng.*, vol. 107, no. 3, pp. 403–412, 2010.

[79] M. Latendresse, M. Krummenacker, M. Trupp, and P. D. Karp, "Construction and completion of flux balance models from pathway databases," *Bioinformatics*, vol. 28, no. 3, pp. 388–396, 2012.

[80] M. Ganter, H.-M. Kaltenbach, and J. Stelling, "Predicting network functions with nested patterns.," *Nat. Commun.*, vol. 5, p. 3006, 2014.

[81] M. Zhang, Z. Cui, T. Oyetunde, Y. Tang, and Y. Chen, "Recovering Metabolic Networks using A Novel Hyperlink Prediction Method," *arXiv Prepr. arXiv1610.06941*, 2016.

[82] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender

Systems," *Computer (Long. Beach. Calif).*, vol. 42, no. 8, pp. 42–49, 2009.

[83] C. D. Maranas and A. R. Zomorrodi, *Optimization Methods in Metabolic Networks*. John Wiley & Sons, 2016.

[84] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, p. 57, 2012.

[85] I. Thiele, N. Vlassis, and R. M. T. Fleming, "FASTGAPFILL: Efficient gap filling in metabolic networks," *Bioinformatics*, vol. 30, no. 17, pp. 2529–2531, 2014.

[86] J. L. Reed, T. R. Patel, K. H. Chen, A. R. Joyce, M. K. Applebee, C. D. Herring, O. T. Bui, E. M. Knight, S. S. Fong, and B. O. Palsson, "Systems approach to refining genome annotation.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 46, pp. 17480–17484, 2006.

[87] W. Morrell, G. Birkel, M. Forrer, T. Lopez, T. Backman, M. Dussault, C. J. Petzold, E. E. K. Baidoo, Z. Costello, and D. Ando, "The Experiment Data Depot: a web-based software tool for biological experimental data storage, sharing, and visualization," *ACS Synth. Biol.*, 2017.

[88] D. Machado and M. Herrgård, "Systematic Evaluation of Methods for Integration of Transcriptomic Data into Constraint-Based Models of Metabolism," *PLoS Comput. Biol.*, vol. 10, no. 4, 2014.

[89] G. Wu, Q. Yan, J. A. Jones, Y. J. Tang, S. S. Fong, and M. A. G. Koffas, "Metabolic Burden: Cornerstones in Synthetic Biology and Metabolic Engineering Applications," *Trends in Biotechnology*. 2016.

[90] G. Wu, L. He, Q. Wang, and Y. J. Tang, “An ancient Chinese wisdom for metabolic engineering: Yin-Yang,” *Microb. Cell Fact.*, vol. 14, no. 1, p. 39, 2015.

[91] J. Nielsen and J. D. Keasling, “Engineering Cellular Metabolism,” *Cell*, vol. 164, no. 6, pp. 1185–1197, 2016.

[92] S. G. Wu, K. Shimizu, J. K.-H. Tang, and Y. J. Tang, “Facilitate Collaborations among Synthetic Biology, Metabolic Engineering and Machine Learning,” *ChemBioEng Rev.*, vol. 3, no. 2, pp. 45–54, 2016.

[93] C. T. Trinh, J. Li, H. W. Blanch, and D. S. Clark, “Redesigning Escherichia coli metabolism for anaerobic production of isobutanol,” *Appl. Environ. Microbiol.*, vol. 77, no. 14, pp. 4894–4904, 2011.

[94] J. M. Monk, C. J. Lloyd, E. Brunk, N. Mih, A. Sastry, Z. King, R. Takeuchi, W. Nomura, Z. Zhang, and H. Mori, “iML1515, a knowledgebase that computes Escherichia coli traits.,” *Nat. Biotechnol.*, vol. 35, no. 10, p. 904, 2017.

[95] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[96] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[97] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*,

2016, pp. 785–794.

[98] F. Chollet, “Keras deep learning library for python. convnets, recurrent neural networks, and more. runs on theano and tensorflow,” *GitHub Repos.*, 2013.

[99] A. Ebrahim, J. A. Lerman, B. O. Palsson, and D. R. Hyduke, “COBRApy: COnstraints-Based Reconstruction and Analysis for Python.,” *BMC Syst. Biol.*, vol. 7, no. 1, 2013.

[100] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[101] H. Abdi and D. Valentin, “Multiple correspondence analysis,” *Encycl. Meas. Stat.*, pp. 651–657, 2007.

[102] L. I. Smith, “A tutorial on Principal Components Analysis Introduction,” *Statistics (Ber).*, vol. 51, p. 52, 2002.

[103] S. Zadran and R. D. Levine, “Perspectives in metabolic engineering: Understanding cellular regulation towards the control of metabolic routes,” *Appl. Biochem. Biotechnol.*, vol. 169, no. 1, pp. 55–65, 2013.

[104] B. R. B. Haverkorn van Rijsewijk, A. Nanchen, S. Nallet, R. J. Kleijn, and U. Sauer, “Large-scale 13C-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in Escherichia coli.,” *Mol. Syst. Biol.*, vol. 7, no. 477, p. 477, 2011.

[105] P. Charusanti, T. M. Conrad, E. M. Knight, K. Venkataraman, N. L. Fong, B. Xie, Y. Gao, and B. Ø. Palsson, “Genetic Basis of Growth Adaptation of Escherichia coli after Deletion

of pgi, a Major Metabolic Gene," *PLoS Genet.*, vol. 6, no. 11, p. e1001186, Nov. 2010.

[106] N. Ishii, K. Nakahigashi, T. Baba, M. Robert, T. Soga, A. Kanai, T. Hirasawa, M. Naba, K. Hirai, A. Hoque, P. Y. Ho, Y. Kakazu, K. Sugawara, S. Igarashi, S. Harada, T. Masuda, N. Sugiyama, T. Togashi, M. Hasegawa, Y. Takai, K. Yugi, K. Arakawa, N. Iwata, Y. Toya, Y. Nakayama, T. Nishioka, K. Shimizu, H. Mori, and M. Tomita, "Multiple high-throughput analyses monitor the response of E. coli to perturbations.," *Science*, vol. 316, no. 5824, pp. 593–597, 2007.

[107] 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.," *Mol. Syst. Biol.*, vol. 2, p. 2006.0008, 2006.

[108] C. P. Long and M. R. Antoniewicz, "Metabolic flux analysis of Escherichia coli knockouts: Lessons from the Keio collection and future outlook," *Curr. Opin. Biotechnol.*, vol. 28, pp. 127–133, 2014.

[109] C. P. Long, J. E. Gonzalez, N. R. Sandoval, and M. R. Antoniewicz, "Characterization of physiological responses to 22 gene knockouts in Escherichia coli central carbon metabolism," *Metab. Eng.*, vol. 37, pp. 102–113, 2016.

[110] G. Plata, T.-L. Hsiao, K. L. Olszewski, M. Llinás, and D. Vitkup, "Reconstruction and flux-balance analysis of the Plasmodium falciparum metabolic network," *Mol. Syst. Biol.*, vol. 6, Sep. 2010.

[111] Y. Shen, J. Liu, G. Estiu, B. Isin, Y.-Y. Ahn, D.-S. Lee, A.-L. Barabasi, V. Kapatral, O.

Wiest, and Z. N. Oltvai, “Blueprint for antimicrobial hit discovery targeting metabolic networks,” *Proc. Natl. Acad. Sci.*, vol. 107, no. 3, pp. 1082–1087, Jan. 2010.

[112] T. M. Conrad, N. E. Lewis, and B. Ø. Palsson, “Microbial laboratory evolution in the era of genome-scale science.,” *Mol. Syst. Biol.*, vol. 7, no. 509, p. 509, 2011.

[113] D. Vitkup and G. M. Church, “Analysis of optimality in natural and perturbed,” no. Track II, 2002.

[114] T. Shlomi, O. Berkman, and E. Ruppin, “Regulatory on/off minimization of metabolic flux,” vol. 102, no. 21, 2005.

[115] J. Kim and J. L. Reed, “RELATCH: relative optimality in metabolic networks explains robust metabolic and regulatory responses to perturbations,” *Genome Biol.*, vol. 13, no. 9, p. R78, 2012.

[116] 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.

[117] S. S. Fong and B. Ø. Palsson, “Metabolic gene – deletion strains of Escherichia coli evolve to computationally predicted growth phenotypes,” vol. 36, no. 10, pp. 1056–1058, 2004.

[118] D. a. Fell, “Enzymes, metabolites and fluxes,” *J. Exp. Bot.*, vol. 56, no. 410, pp. 267–272, 2005.

[119] J. O. Park, S. A. Rubin, Y.-F. Xu, D. Amador-Noguez, J. Fan, T. Shlomi, and J. D.

Rabinowitz, “Metabolite concentrations, fluxes and free energies imply efficient enzyme usage,” vol. advance on, no. May, 2016.

[120] A. Kümmel, S. Panke, and M. Heinemann, “Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data.,” *Mol. Syst. Biol.*, vol. 2, p. 2006.0034, Jan. 2006.

[121] S. Bordel and J. Nielsen, “Identification of flux control in metabolic networks using non-equilibrium thermodynamics.,” *Metab. Eng.*, vol. 12, no. 4, pp. 369–77, Jul. 2010.

[122] N. Tepper, E. Noor, D. Amador-Noguez, H. S. Haraldsdóttir, R. Milo, J. Rabinowitz, W. Liebermeister, and T. Shlomi, “Steady-State Metabolite Concentrations Reflect a Balance between Maximizing Enzyme Efficiency and Minimizing Total Metabolite Load,” *PLoS One*, vol. 8, no. 9, pp. 1–13, 2013.

[123] A. Hoppe, S. Hoffmann, and H.-G. Holzhütter, “Including metabolite concentrations into flux balance analysis: thermodynamic realizability as a constraint on flux distributions in metabolic networks.,” *BMC Syst. Biol.*, vol. 1, p. 23, Jan. 2007.

[124] 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, no. 6987, pp. 92–96, 2004.

[125] A. B. Canelas, N. Harrison, A. Fazio, J. Zhang, J.-P. Pitkänen, J. Van den Brink, B. M. Bakker, L. Bogner, J. Bouwman, and J. I. Castrillo, “Integrated multilaboratory systems biology reveals differences in protein metabolism between two reference yeast strains,” *Nat. Commun.*, vol. 1, p. 145, 2010.

[126] S. S. Fong, A. Nanchen, B. O. Palsson, and U. Sauer, “Latent pathway activation and increased pathway capacity enable Escherichia coli adaptation to loss of key metabolic enzymes,” *J. Biol. Chem.*, vol. 281, no. 12, pp. 8024–8033, 2006.

[127] J. Zhao and K. Shimizu, “Metabolic flux analysis of Escherichia coli K12 grown on 13C-labeled acetate and glucose using GC-MS and powerful flux calculation method,” *J. Biotechnol.*, vol. 101, no. 2, pp. 101–117, 2003.

[128] L. M. Blank, L. Kuepfer, and U. Sauer, “Large-scale 13C-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast.,” *Genome Biol.*, vol. 6, no. 6, p. R49, 2005.

[129] U. W. E. Sauer, D. R. Lasko, J. Fiaux, M. Hochuli, R. Glaser, T. Szyperski, K. Wüthrich, and J. E. Bailey, “Metabolic flux ratio analysis of genetic and environmental modulations of Escherichia coli central carbon metabolism,” *J. Bacteriol.*, vol. 181, no. 21, pp. 6679–6688, 1999.

[130] S. A. Riemer, R. Rex, and D. Schomburg, “A metabolite-centric view on flux distributions in genome-scale metabolic models.,” *BMC Syst. Biol.*, vol. 7, p. 33, 2013.

[131] B. K. S. Chung and D.-Y. Lee, “Flux-sum analysis: a metabolite-centric approach for understanding the metabolic network.,” *BMC Syst. Biol.*, vol. 3, p. 117, 2009.

[132] T. Y. Kim, H. U. Kim, and S. Y. Lee, “Metabolite-centric approaches for the discovery of antibacterials using genome-scale metabolic networks,” *Metab. Eng.*, vol. 12, no. 2, pp. 105–111, 2010.

[133] U. Sauer, V. Hatzimanikatis, J. E. Bailey, M. Hochuli, T. Szyperski, and K. Wuethrich,

"Metabolic fluxes in riboflavin-producing Bacillus subtilis," *Nat. Biotechnol.*, vol. 15, no. 5, pp. 448–452, 1997.

[134] M. J. McAnulty, J. Y. Yen, B. G. Freedman, and R. S. Senger, "Genome-scale modeling using flux ratio constraints to enable metabolic engineering of clostridial metabolism in silico," *BMC Syst. Biol.*, vol. 6, no. 1, p. 42, 2012.

[135] C. Cotten and J. L. Reed, "Constraint-based strain design using continuous modifications (CosMos) of flux bounds finds new strategies for metabolic engineering," *Biotechnol. J.*, vol. 8, no. 5, pp. 595–604, 2013.

[136] S. Ranganathan, P. F. Suthers, and C. D. Maranas, "OptForce: An optimization procedure for identifying all genetic manipulations leading to targeted overproductions," *PLoS Comput. Biol.*, vol. 6, no. 4, 2010.

[137] K. Zhuang, L. Yang, W. R. Cluett, and R. Mahadevan, "Dynamic strain scanning optimization : an efficient strain design strategy for balanced yield , titer , and productivity . DySScO strategy for strain design," *BMC Biotechnol.*, vol. 13, no. 1, p. 1, 2013.

[138] A. R. Zomorrodi, M. M. Islam, and C. D. Maranas, "D-OptCom: Dynamic Multi-level and Multi-objective Metabolic Modeling of Microbial Communities," *ACS Synth. Biol.*, vol. 3, no. 4, 2014.

[139] A. R. Zomorrodi and C. D. Maranas, "OptCom: A multi-level optimization framework for the metabolic modeling and analysis of microbial communities," *PLoS Comput. Biol.*, vol. 8, no. 2, 2012.

[140] 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 Syst. Biol.*, vol. 4, p. 45, Jan. 2010.

[141] M. Zhang, Z. Cui, T. Oyetunde, T. Yinjie, and Y. Chen, “Beyond Pairwise Relations: Hyperlink Prediction in Complex Networks,” in *The 16th IEEE International Conference on Data Mining*, 2016.

[142] J. D. Orth and B. Palsson, “Gap-filling analysis of the iJO1366 Escherichia coli metabolic network reconstruction for discovery of metabolic functions,” *BMC Syst. Biol.*, vol. 6, no. 1, p. 30, 2012.

Appendix

Appendix A: Mathematical formulation of the metabolic network refinement problem in BoostGAPFILL

Inputs:

(Required)

- Incomplete Stoichiometric matrix, S
- Universal set of reactions, U

(Optional)

- Set of blacklisted reactions, B
- Growth/Knockout experimental data

The optional data are used by setting the upper and lower bounds on the reactions.

Step 1: Predict adjacency matrix (matrix factorization model)[141]

$$\boldsymbol{A = \widehat{S}\widehat{S}^{T}\ where\ \widehat{S}\ is\ the\ binary\ incidence\ matrix}\ (\boldsymbol{obtained\ from\ S})$$

The entry in the i-th row and j-th column of A is the number of reactions that metabolite i and metabolite j both participate in. Since S is incomplete, A is incomplete. We proceed to complete A by viewing it as a ranking of the weight of relationship between metabolites. This is similar to the matrix of ratings given to a set of items by a set of users. Completing this ratings matrix is a standard machine learning problem**.** y_{ui} is the rating of item i by user u. In our case, it is the 'rating' of the relationship between metabolite 'u' and 'i'. w_o, w_o, w_i refer to the overall average, user bias and item bias. Vector v_u and v_i (of length k) are used to characterize user u and item i.

Their dot product is a measure of the correlation between user u and item i. Finding the elements of these vectors for each user and item is usually done by least-squares type technique where the difference between the actual rating and predicted rating is minimized.

In our case, since the 'user' and 'item' are indistinguishable, A is symmetric so only the entries above the diagonal (i<j) are used in the matching step. Moreover, we only predict ratings for zero entries in the A matrix.

Thus, the complete adjacency matrix is computed as follows:

$$\underset{\Theta}{minimize} \sum_{i<j, A_{ij}>0} \left\| A_{ij} - y_{ij} \right\|_2^2 + \gamma R(\Theta)$$

$$y_{ui} = w_o + w_u + w_i + \sum_{f=1}^{k} v_{uf} v_{if}$$

$$\hat{A}_{ij} = \begin{cases} w_o + w_u + w_i + \boldsymbol{v_i^T v_j}, & if\ A_{ij} = 0 \\ A_{ij}, & if\ A_{ij} > 0 \end{cases}$$

$\hat{A}$ is the completed adjacency matrix.

The regularization term $\gamma R(\Theta)$ is automatically determined by an MCMC based technique [84].

Step2: Integrated constraint-based and data-driven model

$$\min_{\Lambda} \left\| \boldsymbol{U\Lambda U}^T - \widehat{\boldsymbol{A}} \right\|_F^2$$

$$\lambda_i \epsilon \{0,1\}, i \notin \boldsymbol{B} \{set\ of\ blacklisted\ reactions\}$$

$$\lambda_i = 0, i \in \boldsymbol{B}$$

BoostGapFill Mode 1 and 2

$$\sum_{k \in \boldsymbol{R}} U_{ik} \lambda_k w_k + \sum_{j \epsilon \boldsymbol{P}} S_{ij} v_j = 0 \ \forall\ i \in \boldsymbol{M} \{set\ of\ universal\ metabolites\}$$

$$lb_j \leq v_j \leq ub_j \ \forall\ j \in P \{set\ of\ reactions\ in\ incomplete\ metabolic\ network\}$$

$$-N \leq w_k \leq N \ \forall\ k \in R \{set\ of\ universal\ reactions\}$$

$$v_{biomass} \geq threshold$$

BoostGapFill Mode 3

Notes:

- $\Lambda\ is\ the\ diagonal\ matrix\ of \lambda i$ s.
- The biomass threshold indicates the smallest value for which the cell is considered viable. Set at 0.05/h for all simulations according to [142]. N is an arbitrarily chosen large number (set at 1000 mmol/g/h for all simulations presented).
- Individual blocked reactions (reactions that cannot carry flux) can be unblocked by setting the lower bound on the reaction above a small threshold. The algorithm then predicts reactions to be added that unblocks the selected reaction (using BoostGapFill mode 3).
- In BoostGapFill Mode 2, the pattern-based module is used to weight reactions in the universal database for use in FastGapFill.

Appendix B: Technical implementation details and limitations of BoostGapFill

B.1 Stochasticity of algorithm

In BoostGapFill, the Adjacency matrix is completed with a matrix factorization methodology which has some element of randomness. Moreover, the transformation of the completed adjacency matrix to the stoichiometric matrix is not unique. Therefore, we resort to select a batch of reactions from the universal reaction pool that best match A as the predicted S, which is done by solving an integer least square problem (we provide options for solving the relaxed version which gives very similar results to solving the integer version). It is well-known that least square problems are convex and thus unique solutions will be found each time given delta(A) and U. The we iteratively carry out this computation (A to S and then S to A) until the solutions of the integer least square problem converges or fixed number of iterations (this can be manipulated by the user. The default number of maximum iterations is 10).

B.2 Prediction of reactions with new metabolites

Running BoostGapFill with the newMet option set to ‘true’ allows the possibility of new reactions with metabolites not present in the original metabolic network. This is done by including a partial stoichiometry of such reactions (the reactions are represented by the coefficients of existing metabolites) in the universal reactions matrix in the formulation of the integer least square problem. We also include a penalty term in the objective function to regulate

the number of such reactions. This penalty weight can be adjusted by setting the 'newMetPenalty' option. The results are very similar to when the newMet option is set to 'false'.

B.3 Options available

In addition to the three different modes of running BoostGapFill several options are available to be set by the user depending on the stage of reconstruction of the metabolic network. These include running in integer or relaxed mode, the amount of time for each iteration when running the integer mode, the option to include reactions with new metabolites, the penalty weight for reactions with new metabolite, maximum number of iterations to run, the number of alternative solutions to generate, the solver to use, the reaction weighting and threshold (when BoostGapFill is run in mode2), the solver to use and the list of blacklisted reactions.

B.4 Timing

Most of the simulations were run on a Windows PC with 64GB RAM using the IBM CPLEX solver. One run of BoostGapFill (using default option settings on the iAF1260 *E. Coli* model) takes on average 500 secs, 200 secs and 1800 secs for modes 1, 2 and 3 respectively.

B. 5 Notes on the computational methodology

The same universal matrix (derived from the set of universal reactions on the BiGG database [37]) was used to ensure a fair comparison (provided with the source code). The gapFind[83] algorithm was used before and after using each tool to determine the number of gaps (root blocked metabolites) present. The SMILEY[86] algorithm was run 25 iterations. We update the set of predicted reactions with the new predictions after each iteration. If a reaction has already been predicted by an earlier prediction, it is not selected. We stop once we have the same number

of reactions as we randomly removed from the original model. Thus the process is entirely random The number of iterations was chosen based on that used in an earlier study [142].

B.6 Running BoostGapFill

Requirements: MATLAB with COBRA toolbox installed, IBM CPLEX or GUROBI solver (both have free fully functional academic licenses) and any version of Python.

Download the latest version of BoostGapFill from https://github.com/Tolutola/BoostGAPFILL

To see a demo, change the MATLAB working directory to BoostGAPFILL and type the following in the command line:

'BoostGAPFILL_example1' to see a simple demo run on iAF1260 model

'BoostGAPFILL_example2' to see an extended comparison of BoostGapFill and FASTGAPFILL

The optional settings can be changed in the example scripts. All scripts and function files are in the 'code' sub folder. The COBRA models are in the 'data' sub folder. The universal reactions, metabolites and stoichiometric matrix are stored as variables in the COBRA model structure.

CV

TOLUTOLA TIMOTHY OYETUNDE

St. Louis, MO 63130 | (314) 814 2793 | tolutoyo@gmail.com | www.linkedin.com/in/tolutolaoyetunde |https://github.com/Tolutola

PROFILE

Analytical PhD Candidate, who has lived and worked on three different continents, seeking to leverage extensive education and academic / industry research experience in chemical engineering, mathematical modeling, data science, machine learning, computational biology, and process design/optimization/control.

EDUCATION

WASHINGTON UNIVERSITY IN ST. LOUIS — St. Louis, MO

Ph.D. Chemical Engineering (Metabolic Engineering and Systems Biology) — **2018**

Thesis: Decoding complexity in metabolic networks using integrated mechanistic and machine learning approaches

MASDAR INSTITUTE OF SCIENCE AND TECHNOLOGY (IN COLLABORATION WITH MIT) — Abu Dhabi, UAE

MSc, Chemical Engineering (Environmental biotechnology) — **2014**

Thesis: Modelling microbial electrochemical technologies for wastewater treatment and bioenergy recovery

FEDERAL UNIVERSITY OF TECHNOLOGY — Owerri, Nigeria

BEng, Chemical Engineering — **2007**

Thesis: Modeling and simulation of a vacuum distillation column for the recovery of spent engine oil

TECHNOLOGY SKILLS SUMMARY

Programming:	Python (expert), R(expert), C / C++ (intermediate), and java (intermediate)
Data science and machine learning:	Application of machine learning algorithms and libraries (scikitlearn, tensorflow, keras) in different domains; knowledge of big data technologies, data mining and visualization.
Engineering/Optimization Software:	AutoCAD, MATLAB / Simulink, IBM CPLEX, Gurobi, Mosek, Mathematica, ASPEN PLUS, AQUASIM, SuperPro, and COMSOL
Technologies:	version control (Git), cloud computing (AWS, Domino), database management (SQL and MongoDB), Docker

EXPERIENCE SUMMARY

MONSANTO COMPANY — Saint Louis, MO

Data Scientist Co-op — **Jan – June 2018**

Developed and implemented genetic marker selection algorithm for the molecular breeding pipeline

TANG LAB, DEPARTMENT OF ENERGY, ENVIRONMENTAL, AND CHEMICAL ENGINEERING AT WASHINGTON UNIVERSITY IN ST. LOUIS — St. Louis, MO

Research Assistant / PhD Graduate Student — **2014 – 2018**

- Conducted research in computational biology and machine learning
- Served as Teaching Assistant in *Chemical Process Dynamics and Control* (Fall 2015, Fall 2016) and *Computational Modeling in Energy, Environmental and Chemical Engineering* (Spring 2016).
- Six journal publications and 2 conference presentations
- Led a team of four graduate students to curate a high-quality metabolic engineering design database from more than 450 journal articles

FRIESLANDCAMPINA WAMCO NIGERIA PLC — Lagos, Nigeria

Project Engineer — **2011 – 2012**

- Initiated and executed energy conservation projects.
- Prepared capital expenditure engineering budget and monitored / controlled CAPEX projects.

FRIESLANDCAMPINA WAMCO NIGERIA PLC — Lagos, Nigeria

Management Trainee — **2010 – 2011**

- Member of Leadership Potential Team (highest honor for management trainees)
- Initiated and executed process improvement projects across Finance/Accounts, Powder Factory, and Engineering departments.

PUBLICATIONS / PRESENTATIONS

1. **Oyetunde, T.**, Bao, F. S., Chen, J. W., Martin, H. G., & Tang, Y. J. (2018). Leveraging knowledge engineering and machine learning for microbial bio-manufacturing. **Biotechnology advances.**

2.**Oyetunde, T.**, Sarma, P. M., Ahmad, F., & Rodríguez, J. (2017). A Multiple Reaction Modelling Framework for Microbial Electrochemical Technologies. International journal of molecular sciences, 18(1), 86.

3. **Oyetunde, T.**, Zhang, M., Chen, Y., Tang, Y., & Lo, C. (2016). BoostGAPFILL: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods. **Bioinformatics**, 33(4), 608-611.

4. Shopera, T., He, L., **Oyetunde, T.**, Tang, Y. J., & Moon, T. S. (2017). Decoupling resource-coupled gene expression in living cells. **ACS synthetic biology**, 6(8), 1596-1604.

5. Liu, Z., **Oyetunde, T.**, Hollinshead, W. D., Hermanns, A., Tang, Y. J., Liao, W., & Liu, Y. (2017). Exploring eukaryotic formate metabolisms to enhance microbial growth and lipid accumulation. **Biotechnology for biofuels**, 10(1), 22.

6. Wu, S. G., Wang, Y., Jiang, W., **Oyetunde, T.**, Yao, R., Zhang, X., ... & Bao, F. S. (2016). Rapid prediction of bacterial heterotrophic fluxomics using machine learning and constraint programming. **PLoS computational biology**, 12(4), e1004838.

7. **Oyetunde, T.**, Liu D., Martin H.G., and Tang Y.J Machine learning framework for robust assessment of microbial factory performance. **PLoS ONE** (in revision)

8. **Oyetunde, T.**, Fatehi A., Czajka J., and Tang Y.J Thermodynamic framework for mutant phenotype prediction. (In preparation)

Professional Conference Presentations (= Presenter)*

Oyetunde, T., and Tang Y.J Thermodynamic framework for mutant phenotype prediction to be presented at the **2018 COBRA Conference**, Seattle, WA. (Oct. 14 – Oct. 16, 2018).

Oyetunde, T., Czajka J*., and Tang Y.J A Deep Learning Framework Decodes Coordination of Microbial Metabolism Under Genetic and Environmental Perturbations presented at the **2017 AIChE Annual Conference**, Minneapolis, MN. (Oct. 29 – Nov. 3, 2017).

Oyetunde, T*, Tang, T. and Lo C.S " Thermodynamic Analysis of the Rigidity of Metabolic Nodes Via a Dynamic Flux Balance Approach" presented at the **2016 AIChE Annual Conference**, San Francisco, CA. (Nov 13-18, 2016).

Oyetunde, T., Sarma, P. M, Lema, J. M., and Rodríguez, J.*, "Modelling the bioelectrochemical conversion of VFAs to alcohols: Impact of operational variables," presented at the **2014 IWA Science Summit on Urban Water**, Harbin, China (Jul 13-17, 2014).

Rashid, K., **Oyetunde, T.**, Alassali, A.*, Rodríguez, J., and Thomsen M.H., "Biofuels from the desert: the sustainability case for a Salicornia bigelovii-based biorefinery," presented at the **Pacific Rim Summit on Industrial Biotechnology and Bioenergy**, San Diego, CA (Dec 8-13, 2013).

Oyetunde, T.*, Ofiteru, I.D., and Rodríguez. J., "Modeling Bioelectrochemical Systems for (waste)water Treatment and Bioenergy Recovery with COMSOL," presented at the **COMSOL 2013 Conference**, Boston, MA (Oct 9-11, 2013).

Oyetunde, T.*, González-Cabaleiro, R., Ahmad, F., and Rodríguez, J., "A Generalized Excel/C-compatible Simulink-based implementation architecture for fast model development and simulations," presented at the **11th IWA conference on instrumentation control and automation.** Narbonne, France (Sep 18-20, 2013).

Oyetunde, T.*, González-Cabaleiro, R., Ahmad, F., and Rodríguez, J., "Modeling multiple electrode reactions in bioelectrochemical systems," presented at **the 4th International Microbial Fuel Cell Conference**, Cairns, Australia (Sep 1-4, 2013).

Alassali, A., **Oyetunde, T.***, Rashid, K., Baldwin, R., Thomsen, M.H., and Rodríguez, J., "Anaerobic digestion as key process in the biorefinery on Salicornia plant biomass: Simulation study using Super Pro Designer," presented at the **13th IWA World Congress on Anaerobic Digestion**, Santiago de Compostela, Spain (Jun 25-28, 2013).