COMPUTATIONAL METHODS FOR INTEGRATING METABOLOMICS DATA WITH METABOLIC ENGINEERING STRAIN DESIGN

A Thesis
Presented to
The Academic Faculty

By

Robert A. Dromms

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in Chemical & Biomolecular Engineering

Georgia Institute of Technology

August 2017

COMPUTATIONAL METHODS FOR INTEGRATING METABOLOMICS DATA WITH METABOLIC ENGINEERING STRAIN DESIGN

Approved by:

Dr. Mark P. Styczynski, Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Melissa Kemp
Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Rachel Chen
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Matthew Realff
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Pamela Peralta-Yahya
School of Chemical & Biomolecular Engineering
School of Chemistry and Biochemistry
Georgia Institute of Technology

Date Approved: May 15, 2017

ACKNOWLEDGEMENTS

Completing my PhD has been a long and sometimes winding road, and I owe a great deal to quite a few people who have helped me see it through to the end.

I'd like to thank my adviser Dr. Mark Styczynski for all of his support and guidance over the last 6 years. And for the ice cream.

I'd like to thank my committee, Dr. Melissa Kemp, Dr. Rachel Chen, Dr. Matthew Realff, and Dr. Pamela Peralta-Yahya, for their feedback and suggestions.

I'd like to thank the members of the Styczynski Lab, especially Katie, Sugantha, Daniel, Francisco, Monica, McKenzie, and Khaldoon. And above all, Amy, who has been a great friend and an ever better rubber ducky.

I'd like to thank Dave, Wes, Mason, and Matt for making the first years of grad school so great.

I'd like to thank Mike, Santosha, Ace, Sabrina, Jay, Brent, Sam, Paul, Saul, Brian, and the rest of the Wings crew for all the chicken wings, noodly goodness, and super happy fat times.

I'd like to thank Lauren, Michael, Jess, and Jonathan for the games days, holiday parties, picnics and lunches, and other random fun things.

I'd like to thank Peter and Andrew for being such great roommates.

I'd like the thank my friends from Cornell and from Liverpool, who I may not see or talk to as often, but somehow that never seems to matter when we do.

I'd like to thank Kate and IBK for being such awesome friends despite the long distance.

I'd like to thank my family, who have been there all along, whether I like it or not. Especially John and Rebecca for their snark and ridiculousness.

And most of all,

My parents.
Without their support through the years,
this would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AIC: Akaike Information Criterion

ANN: Artificial Neural Network

BLP: Bi-linear Program

BN: Bayesian Network

BST: Biochemical Systems Theory

CBM: Constraint-Based Model

CE-MS: Capillary Electrophoresis-Mass Spectrometry

CHO: Chinese Hamster Ovary

COBRA: COnstraints Based Reconstruction and Analysis

CoV: Coefficient of Variance

DAG: Directed Acyclic Graph

DF: Direct Fit

DFBA: Dynamic Flux Balance Analysis

DFE: Dynamic Flux Estimation

DMFA: Dynamic Metabolic Flux Analysis

DOA: Dynamic Optimization Approach

EMUs: Elementary Metabolite Units

FBA: Flux Balance Analysis

GA: Genetic Algorithm

GC-MS: Gas Chromatography-Mass Spectrometry

GMA: Generalized Mass Action

HCA: Hierarchical Clustering Analysis

HPLC: High-Performance Liquid Chromatography

idFBA: Integrated-Dynamic Flux Balance Analysis

iFBA: Integrated Flux Balance Analysis

IOMA: Integrative “Omics”-Metabolic Analysis

kFBA: kKnetic Flux Balance Analysis

LC-MS: Liquid Chromatography-Mass Spectrometry

LK-DFBA: Linear Kinetics-Dynamic Flux Balance Analysis

LP: Linear Program

LR: (Linear) Regression

LR+: (Linear) Regression-Plus

MASS: Mass Action Stoichiometric Simulation

MFA: Metabolic Flux Analysis

MM: Michaelis-Menten

MOMA: Minimization of Metabolic Adjustment

NET: Network-Embedded Thermodynamic Analysis

NLP: Non-Linear Program

NMR: Nuclear Magnetic Resonance

nT: Number of Time Intervals

ODE: Ordinary Differential Equation

OMNI: Optimal Metabolic Network Identification

PC: Peter-Clark

PCA: Principal Components Analysis

PLS: Partial Least Squares regression

PLS-DA: Partial Least Squares Discriminant Analysis

QP: Quadratic Program

rFBA: Regulatory Flux Balance Analysis

RM: Resampling Method

RMSD: Root-Mean-Square-Displacement

SBRT: Systems Biology Research Tool

SSE: Sum-of-Squares Error

SSR: Sum-of-Squares Residual

TCA: Tricarboxylic acid

TMFA: Thermodynamic Metabolic Flux Analysis

uFBA: Unsteady-state Flux Balance Analysis

VHG: Very High Gravity

SUMMARY

The genome-scale analysis of cellular metabolites, “metabolomics”, provides data ideal for applications in metabolic engineering. The goals of metabolic engineering are well-served by the biological information provided by metabolomics: information on how the cell is currently using its biochemical resources is perhaps one of the best ways to inform strategies to engineer a cell to produce a target compound. In the first chapter, I review the most common systematic approaches for integrating metabolite data with metabolic engineering and discuss some of the most common approaches for computational modeling of cell-wide metabolism, including constraint-based models (CBMs). This overview provides the motivation and the context for the contributions presented in this thesis.

In the second chapter, I present several improvements to current approaches for smoothing metabolite time course data using defined functions. First, I use a biologically-inspired mathematical model function taken from transcriptional profiling and clustering literature that captures the dynamics of many biologically relevant transient processes. I demonstrate that it is competitive with, and often superior to, previously described fitting schemas, and may serve as an effective single option for data smoothing in metabolic flux applications. I also implement a resampling-based approach to buffer out sensitivity to specific data sets and

allow for more accurate fitting of noisy data. I find that this method, as well as the addition of parameter space constraints, yields improved estimates of concentrations and derivatives (fluxes) in previously described fitting functions. These methods have the potential to improve the accuracy of existing and future dynamic metabolic models by allowing for the more effective integration of metabolite profiling data.

In the third chapter, I address this problem of accounting for metabolite levels in CBMs by discussing the main contribution of this thesis: an improved constraint-based modeling framework I refer to as Linear Kinetics-Dynamic Flux Balance Analysis (LK-DFBA). I describe and assess a modeling framework based on dynamic FBA (DFBA) that tracks metabolite concentrations and uses them to constrain system fluxes using strictly linear equations describing the kinetics and regulation of metabolism. I discuss procedures to identify model parameters using both regression and global parameter optimization. With these methods, I show that we were able to produce fitted models that performed comparable or better than Ordinary Differential Equation models fitted to Generalized Mass Action and Michealis-Menten rate laws. I also implement a larger, biologically relevant model in LK-DFBA and further discuss the consequences and benefits of two different parameterization structures.

In the analysis of the third chapter, the correct regulatory structure of the network was known and provided to the LK-DFBA simulation. I continue to assess the LK-DFBA framework in the fourth chapter by exploring the impact of modeling different regulatory connections on model performance. In a small test case with two true regulatory connections, I use a brute-force approach to fit a series of LK-DFBA models with differing regulatory structures to noisy data, and find that I can robustly detect the contribution of one connection to model fitting performance, but have difficulty with the other, suggesting that models implemented using LK-DFBA has in principle some ability to distinguish between correct and incorrect model regulatory structures, but some care must be taken when interpreting the results of this task.

The contributions presented in this thesis are a series of strategies and methods for working with metabolomics data and constructing working, biologically relevant metabolic models that will aid strain design. These models capture the dynamics and regulation of metabolism in a structure that is easily compatible with existing tools build around FBA. While this approach has promise, it also has not yet been used for actual strain design or experimentally validated. In the fifth chapter, I discuss the steps necessary to see this ultimate aim to fruition, and describe some complementary strategies that may help further improve the model or use metabolomics data in interesting new ways.

Chapter 1: Background—Metabolomics in Metabolic Engineering

1.1 Introduction

Organisms such as *Saccharomyces cerevisiae* and *Aspergillus niger* have a long history of commercial use in natural fermentation processes to produce chemicals such as ethanol and citric acid. Traditional bioprocess engineering entails the design and optimization of the equipment and procedures necessary to efficiently manufacture these and other biologically derived products. The development of recombinant DNA technologies enabled the direct manipulation and expansion of the metabolic capabilities of *S. cerevisiae* and *A. niger* (as well as other organisms such as *Escherichia coli* and *Bacillus subtilis*), which resulted in the emergence of metabolic engineering as a field distinct from bioprocess engineering[2]. Metabolic engineering is the (usually genetic) control of the metabolic activities of a living organism to establish and optimize the production of desirable metabolites—the class of small molecules that comprise the primary resources and intermediates of all cellular activity. With widespread and growing interest in environmentally sustainable industrial technologies, metabolic

engineering is poised to provide an effective and efficient means for producing various small molecule chemicals from clean and renewable sources, such as biofuels derived from lignocellulosic feedstocks[3-13].

Frequently, metabolic engineering studies use targeted analysis of a few carefully selected intracellular or secreted extracellular compounds to drive or assess the progress of their efforts[3,10,14-25]. High-Performance Liquid Chromatography (HPLC) and enzymatic assays have typically been the methods of choice to generate this data, used in engineering *S. cerevisiae*[3,14,22,24], *E. coli*[17,20], *Clostridium acetobutylicum*[15,19], and other organisms. These measurements may be direct readouts of the performance of an engineered strain[3], or they may be interpreted as performance and response characteristics (for example, trehalose as a marker for stress response in yeast[22,24]). These analyses are typically focused on effects at the level of individual pathways[3,20,22,26].

Another technique used to characterize metabolic pathways during metabolic engineering is Metabolic Flux Analysis (MFA). MFA provides more information than measurement of just a few metabolites, and is a staple technique of many who work in metabolic engineering[15,19,21,23,25,27-33]. In MFA, isotopically labeled metabolites (typically using ^{13}C labels) are leveraged to calculate fluxes—the rate at which material is processed through a metabolic pathway—from knowledge of carbon-carbon transitions in each reaction and the measured isotopomer

distribution in each metabolite[2]. Ongoing research in MFA includes continued improvement of ^{13}C protocols and analytical platforms[34-37], improvements to software for MFA calculations[33,38], use of network stoichiometry to determine the minimal set of required metabolite measurements[39], and study of Elementary Metabolite Units (EMUs) for more efficient analysis of flux patterns[32,40,41].

Metabolic engineering seeks to maximize the production of selected metabolites in a cell, whether produced by the organisms' natural metabolic activities or by entire exogenous pathways introduced through genetic engineering. Strategic, small-scale measurements and flux calculations have to date been indispensable tools for metabolic engineering. However, the development of systems-level analyses—precipitated by whole-genome sequencing and the rapid accumulation of data on RNA, protein and metabolite levels—has provided new opportunities to more completely understand the effects of strain manipulations. Genetic modifications often have additional effects outside the immediately targeted pathway, and a better understanding of the nature and extent of these perturbations would lead to more effective strategies for redesigning strains, as well as improved ability to understand why a proposed design may fail to achieve its predicted performance.

Aided by recent advancements in analytical platforms that allow for the simultaneous measurement of a wide spectrum of metabolites, metabolomics

(the analysis of the total metabolic content of living systems) is approaching the level of maturity of preceding "global analysis" fields like proteomics and transcriptomics[42,43]. Metabolomics approaches have already found some success in clinical applications, where studies have demonstrated their efficacy in identifying clinically relevant biomarkers in diseases such as cancer[44-46]. Surprisingly though, the application of metabolomics approaches to problems in metabolic engineering has been somewhat scarce.

Here, we review examples of recent strategies to integrate metabolomics datasets into metabolic engineering. First, we briefly cover the fundamentals of metabolomics. We then discuss strategies for assessing metabolic engineering strain designs, and how metabolomics methods can extend these strategies. We follow with discussion of computational tools for metabolic engineering, with an emphasis on how these methods are used to design strains and predict their performance as well as how metabolomics datasets are currently applied to computational modeling. We conclude with a brief summary of the state of the field and the potential that integrating metabolomics presents.

1.2 Metabolomics Background

The development of metabolomics, the newest of the global analysis methods, has much in common with its predecessor fields of genomics, transcriptomics, and proteomics[42,43]. The analytical platforms used for metabolomics have now

developed to the point that metabolomics datasets can serve as an excellent complement to standard metabolic engineering approaches. The goals of metabolic engineering ultimately focus on producing desired metabolites, and metabolomics offers a means of broadly and directly assessing how well a strain meets those goals. What follows is a cursory overview of metabolomics technologies and the most common ways that metabolomics data are interpreted and analyzed, provided as context for how metabolomics data can be used towards metabolic engineering efforts.

1.2.1 Analytical Platforms

One of the primary difficulties facing the development of metabolomics has been the staggering diversity of metabolites. Metabolites are substantially more chemically diverse than the subunit-based chemistries of DNA, RNA, and proteins, impeding the progress of metabolomics as a truly "omics" field that measures all metabolites. The entire genome and transcriptome can be (at least theoretically) surveyed using single platforms, from simple PCR to more exhaustive sequencing and microarrays, whereas metabolomics requires multiple analytical platforms to achieve complete coverage of all metabolites.

Common approaches involve coupling of a chromatographic separation to mass spectrometry, including gas chromatography-mass spectrometry (GC-MS)[7,26,29,30,35,47-59], liquid chromatography-mass spectrometry (LC-

MS)[27,34,35,50,53,55,57,60-64], and capillary electrophoresis-mass spectrometry (CE-MS)[35,64-66]. Other common platforms include nuclear magnetic resonance (NMR)[29,44,67-70] and an assortment of direct injection-mass spectrometry methods[44,49,52,54,68]. Protocols for using these platforms are under constant development, and span sample processing and work-up[52,57,71], efforts to improve the quantitative reliability of measurements[34,57,62], and data processing software[72-84]. These referenced software tools, along with those presented in subsequent sections of this chapter, are summarized in Table 1 (though we emphasize that this list is far from exhaustive). A more extensive review of these platforms is available from Dunn *et al.*[85].

1.2.2 Data Analysis

As the youngest of the global analysis methods, metabolomics has drawn heavily from the data analysis techniques developed for transcriptomics and proteomics. Like these two fields, the datasets generated by metabolomics suffer from a "curse of dimensionality," where there are far more variables than there are samples. Methods taken from transcriptomics and proteomics, as well as some derived from the field of chemometrics, have been used extensively to analyze metabolomics data as a result[42,43] (some examples given in Figure 1.1). Though metabolomics datasets are of high dimension, the connected nature of biochemical pathways and networks can often lead to strong underlying patterns in the data; multivariate techniques have proven effective at identifying these

underlying factors even if individual effect sizes are too small to be detected by univariate analyses (e.g., Figure 1.1A), and so here we highlight several of the methods most widely adopted for metabolomics studies.

One of the most prominent methods for analysis of metabolomics data is Principal Components Analysis (PCA) (Figure 1.1B). This technique identifies the natural "axes" of variation in the dataset by constructing a series of orthogonal component axes from the original metabolite features. Each component is a weighted combination of the original metabolite measurements that provides the maximum possible variance in a single composite variable; the components are all mutually orthogonal. The weights of the original features for each component ("loadings") and the projections of the samples onto the components ("scores") can reveal putative biomarkers or lead to simplified separation between biological sample classes, respectively[42,49,50,58,71,86-92]. Notably, this is an unsupervised technique; PCA uses no information about sample classes in its calculations, and the user can try to identify clusters of data points before projecting class information onto the score plot.

A few examples of using PCA to reveal underlying patterns in metabolomics datasets include the characterization of extracellular culture conditions in Chinese Hamster Ovary (CHO) cell batch cultures[91], a study of the response of *S. cerevisiae* to very high gravity (VHG) fermentations[86], comparisons of

metabolomes across mutant strains of *S. cerevisiae*[43,49,71,88], and analyses of *Pseudomonas putida* growth on various carbon sources[50,87]. In this context, the loadings from the components that capture the separation between sample classes (e.g., culture condition or strain) on the score plot provide information about which metabolites are important to each class. The magnitude of each metabolite's loading coefficient and the groups of metabolites with high loadings in components that capture separation can be used to infer biological significance.

Much of the value of PCA comes from its dimensional reduction capabilities: typically the first few components contain biologically relevant information, and higher components contain variance due to noise or biological variability. The number of components that are "significant" is an open question, and depends predominantly on the dataset or even the specific downstream processing and applications[93]. Since the principal component scores are "optimal" lower-dimension projections of the original data, they can be used in place of the original data in subsequent analysis, such as Hierarchical Clustering Analysis (HCA, Figure 1.1C)[91]. For example, Barrett *et al.* performed PCA on a flux balance analysis solution space to identify a lower-dimension set of key reactions that form the underlying basis of the solution space[90].

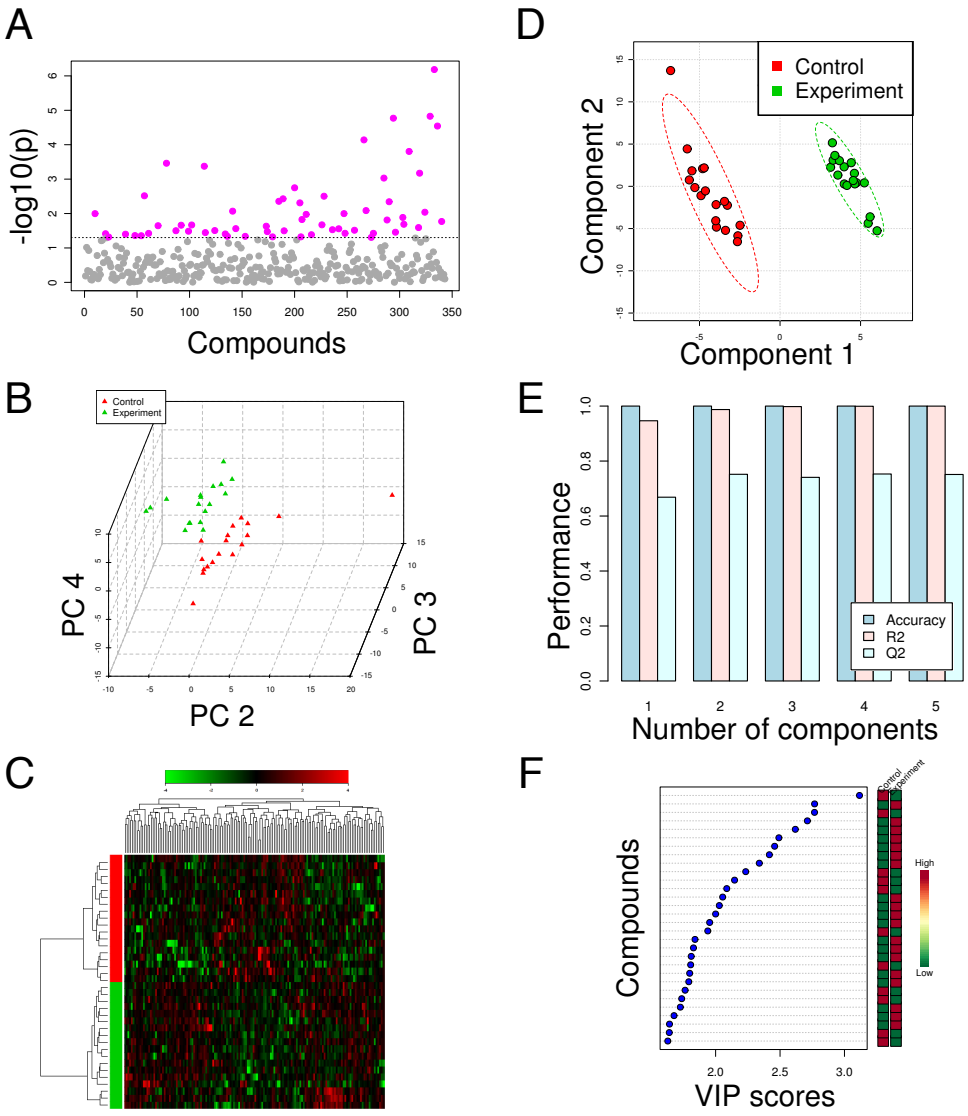

Figure 1.1. Examples of data analysis techniques for metabolomics

The effects of glucose deprivation on a cancer cell line were measured with GC-MS and analyzed in MetaboAnalyst[81] (unpublished data).

A. Pairwise t-tests of metabolites identify statistical significance of differences in individual compounds between control and experiment. The dotted line indicates $p < 0.05$ (no multiple hypotheses testing corrections).

B. PCA score plot reveals separation between control and experiment samples in components 2, 3, and 4. Component 1 (not shown) corresponds to analytical batch separation.

C. HCA (Ward method, Pearson's correlation) and heatmap using the 150 most significant compounds as determined by t-test. Compounds along top, samples along left side.

D. PLS-DA score plot shows separation achieved using components 1 and 2. Dashed circles indicate the 95% confidence interval for each class.

E. Leave-one-out cross-validation shows that the majority of the predictive capacity is derived from the first two PLS-DA components. R^2 and Q^2 denote, respectively, the goodness of fit and goodness of prediction statistics.

F. Contribution of individual compounds to PLS-DA component 1. The 30 most important compounds and their relative abundance in control and experiment are shown, sorted by the Variable Importance in the Projection (VIP) for the first component.

Partial Least Squares (PLS) regression and discriminant analysis (PLS-DA, Figure 1.1D-F) are also common tools in metabolomics analysis. They are multivariate analogs of linear regression and linear discriminant analysis, respectively. They are constructed in a manner similar to PCA, but require response variables (e.g. titer, viability, or conversion) or a class label, respectively, to determine the component axes[94]. Again, assessment of metabolite loading coefficients in PLS-DA axes allows biological interpretability. In one representative application, Cajka *et al.* used PLS-DA to identify a set of compounds that could discriminate between different beers by their origin[68]. Kamei *et al.* used OPLS-DA, a variant of PLS-DA that constructs distinct predictive and orthogonal components that describe between-class and within-class variance, respectively,[95,96], to assess the effects of knockouts related to replicative lifespan in *S. cerevisiae*[92]. They found that a component corresponding to separation between short-lived and long-lived strains identified differences in TCA cycle metabolites as predictors of longevity. These are just a few examples of the increasingly prevalent applications of PLS-based techniques in the field.

Complete and effective use of a metabolomics dataset necessitates not only careful design of experiment and data processing methods, but also a thorough validation of conclusions from data analysis (e.g. apparent clusters in principal component space). For example, discussion of p-value distributions by Hojer-

Pedersen *et al.* touches upon the importance of multiple hypothesis testing corrections in metabolomics studies, such as Bonferroni or false discovery rate corrections[88,97]. As a supervised method, PLS-DA is particularly susceptible to over-fitting, and so cross-validation is critical[98]. Statistical issues aside, non-biological factors can also lead to separation in principal component space, with sources of variance potentially including derivatization protocols[71,86,87], analytical platform[49], chromatographic drift or batch effects[99] and data processing methods[87]. Broadhurst & Kell review other potential pitfalls in greater detail[100].

1.3 Applications of Metabolomics in Metabolic Engineering

Metabolomics continues to be exploited for numerous biomedical applications, ranging from the study of differences between clinically isolated and industrial yeast strains[89], to blood or urine-based biomarkers for many human diseases, including diabetes[65,101], gallstone diseases[102], sepsis[70], and multiple types of cancer[44-46] (Blekherman *et al.* provide a more comprehensive review of the applications of metabolomics to cancer biomarker discovery[103]). Metabolomics also has the potential for a significant biotechnological impact in metabolic engineering: as the goal of metabolic engineering is to manipulate metabolite production, metabolomics naturally lends itself to that goal. Moreover, organisms such as *S. cerevisiae* and *E. coli* have been studied extensively, providing a rich biological context in which the metabolome of strains derived from both rational design and directed evolution strategies can be interpreted and understood.

Nonetheless, the use of metabolomics in metabolic engineering is not as prevalent as one might expect.

1.3.1 Metabolomics Data as an Extension of Small-scale, Targeted Analysis

The simplest and most direct use of metabolomics datasets is as an extension of existing small-scale metabolite analyses; metabolomics inherently enables a more comprehensive assessment of a strain than a handful of narrowly selected measurements. Studies employing this approach typically either compare strains and culture conditions, or seek to monitor the time-course evolution of many metabolite concentrations in parallel. These studies use a combination of measured growth and production parameters in conjunction with direct examination of the metabolomics data (e.g. significant increases or decreases in metabolite levels) in the context of known biochemical pathways to determine the effects of mutations and culture conditions. For example, if one overexpresses the enzyme that is the first step in a linear biosynthetic pathway and finds that the first few metabolites accumulate significantly but subsequent metabolites do not, this may suggest a rate-limiting step further down the pathway that needs to be upregulated. Broader knowledge of metabolite levels beyond the target pathway can serve to determine the wider-ranging effects of a given metabolic engineering perturbation and can suggest candidate supplementary perturbations (to address, for example, cofactor imbalances).

One example of a strain- or condition-comparison approach is a study of an arcA mutant in *E. coli* by Toya *et al.*, which compared parent and mutant responses to aerobic, anaerobic and nitrate-rich media conditions[66]. Through analysis of fold-changes in the metabolome, transcriptome, and ^{13}C MFA-derived fluxes, they found significant differences in tricarboxylic acid (TCA) cycle metabolism and ATP production among conditions. Similarly, Christen *et al.* compared the metabolomic profiles of seven yeast species to assess differences in aerobic fermentation on glucose[63]. While ^{13}C MFA suggested differences between TCA cycle fluxes and consistent flux through glycolysis, there was a much wider variation of metabolite levels across species–especially in amino acid pool compositions. They also found that across species, these values correlated poorly with fluxes.

In an example of time-course analysis, Hasunuma *et al.* studied the effects of acetic and formic acid, chemicals commonly found in lignocellulosic hydrolysates, on xylose-utilizing strains of *S. cerevisiae*[11]. A separate similar study by Klimacek *et al.* also explored differences in xylose-utilizing strains[10]. Continued work in time course analysis has also exploited improved collection of time-course metabolomics data. Link *et al*. developed a procedure for measuring over 300 metabolites at 15 to 30 second intervals, and demonstrated it in *E. coli*[104]. With this, they were able to identify feedback in amino acid biosynthesis and efficient recycling for purine that avoided expensive biosynthesis[104].

Other work with *S. cerevisiae* has investigated the transient effects of redox perturbations[105] relief from glucose limitation[27,106], diauxic shift[107], as well as differences between *S. cerevisiae* and *Pichia pastoris*[57]. However, examples branch out in to a span a variety of organisms and culture conditions, from xylose utilization in *A. niger*[7] to the effects of extended culture periods[108] and low phenylacetic acid conditions after key pathway knockouts[30] on penicillin biosynthesis in *Penicillium chrysogenum*. Work by Sevin *et al.* explored the impact of osmtic stress via salt shock on the metabolome of twelve bacteria, two yeast, and two human cell lines, finding glycolysis, TCA, branch-chain amino acid synthesis, and heme biosynthesis as key pathways across organisms[109].

One area of particular interest involves identifying changes in regulation, whether at the transcriptional or metabolic level. Work by Goncalves *et al*. used time-course metabolomics data to search for regulatory connections such as transcription factors and kinases/phosphatases, and were able to identify and experimentally validate several regulatory connections without resorting to gene knockout strains[110]. Other work by Zampieri *et al.* used metabolomics to identify mechanisms behind the development of antibiotic resistance in E. coli, and identified potential fragility as well, such as hypersensitivity to fosfomycin in ampicillin-resistant strains[111].

1.3.2 General Strategies for Integrating Metabolomics into Metabolic Engineering

The simple approaches used to exploit the results of targeted measurements can be scaled up to metabolomics datasets, but they often do not take full advantage of structures or patterns in the data at the systems level. Many in the field of metabolic engineering have used multivariate techniques to interrogate metabolomics datasets on more complicated questions about strain performance and metabolite allocation. Due to the complexity of biological systems, the answers to these questions are often non-intuitive and increasingly difficult to identify without taking such a systems-scale approach.

1.3.2.1 Adaptive Evolution and High Throughput Libraries: Locating the Cause of Improved Phenotypes

While rational design approaches were the original driving force in metabolic engineering, directed evolution and high throughput screens of mutant libraries have since become increasingly commonplace[4,5,9,12,13,112-119]. One of the main difficulties involved in these two "inverse metabolic engineering" approaches is the identification of the underlying behaviors responsible for the improved phenotype[120]. These frequently non-intuitive changes can often best be characterized with a direct, systems-scale readout of the metabolic state[117].

Common techniques employed in such approaches include HCA, PCA, and PLS-DA. These methods generate clusters or loadings that identify key metabolite

differences, which in turn suggest what genetic changes may have been selected for. For example, Hong *et al.* used PCA and clustering analysis of metabolomics data, supplemented with transcriptional data, to investigate strains of *S. cerevisiae* that had been selected via directed evolution for improved galactose uptake[117]. A study by Yoshida *et al.* examined metabolic differences between *S. cerevisiae* and *Saccharomyces pastorianus* in regards to SO_2 and H_2S production[121]. Similarly, Wisselink *et al.* investigated a xylose-utilizing strain of *S. cerevisiae* developed by introduction of L-arabinose pathway genes to an existing xylose-utilizing strain, followed by directed evolution to improve L-arabinose utilization[122].

Other examples of using metabolomics for after-the-fact assessment of engineered strains include studying the effects of repeated exposure to vacuum fermentation conditions on *S. cerevisiae*[123], comparing evolved strains with knockouts proposed by the OptKnock algorithm[48], and identifying the differences between several yeast species during aerobic fermentation on glucose[63].

1.3.2.2 Other Global Analysis Approaches: Harnessing Proteomics, Transcriptomics, and Genomics for Metabolic Engineering

While the goal of metabolic engineering is to introduce a change on the metabolic level, many of these changes are necessarily implemented by introducing genetic modifications to affect transcriptional levels. As such, analysis of biological layers

beyond the metabolome, such as the transcriptome and proteome, can provide further, and sometimes crucial, insight into the wide-reaching effects of an alteration.

A number of techniques widely used in metabolomics (such as PCA and HCA) are also well-established for many of these other "omics" datasets, though there are a number of other techniques that until recently were more specific to transcriptional or proteomic analyses. One of the most prominent examples of this is enrichment analysis, originally developed for transcriptomics datasets. Enrichment analysis uses information about the frequency of occurrence or the ranking of sets of gene names or functions in a given list of genes to examine the biological relevance of observed changes[124]. For example, the number of genes from a given pathway occurring in a list of interest (say, high-importance variables in PCA or a cluster from HCA) is assessed to see if that number of genes would be expected to be found in an arbitrary list of genes purely at random; this comparison is made using a hypergeometric distribution. If a list is statistically significantly "enriched" for a set of genes, one then may hypothesize that the list of genes plays an important role in the underlying biological process. This technique has recently been extended to metabolomics datasets[81,125]. A combination of enrichment, multivariate, and univariate analyses comprise the bulk of the strategies currently used in metabolic engineering to analyze "omics" datasets in parallel.

In metabolic engineering, use of metabolomics is comparatively much less common than using other global analysis approaches, perhaps attributable to the maturity of fields like transcriptomics compared to metabolomics. Proteomics, transcriptomics and genomics have frequently been combined with small-scale metabolite measurements for metabolic engineering purposes. Examples of this include functional genomics with targeted metabolite measurements for isoprenoid production in *E. coli*[126], as well as the combination of the proteome, transcriptome and targeted metabolite measurements for *E. coli* carbon storage regulation[64], penicillin production in *P. chrysogenum*[108], glucose repression in *S. cerevisiae*[53], and relief from glucose deprivation in *S. cerevisiae*[106].

The above examples at most used small-scale metabolite measurements, but a handful of studies have combined analysis of full metabolomics datasets with other "omics" datasets. Previously described analyses of adaptations from directed evolution generally fit this category: transcriptional measurements using microarrays[48,117,121,122] and genomic analysis[89] have each been combined with metabolomics to pinpoint the source of the observed phenotype.

In other applications, Piddock *et al.* assessed high gravity beer brewing conditions to determine the effect of the protease enzyme Flavourzyme on the free amino nitrogen content of the wort [56]. A collaborative study by Canelas *et al.* investigated the growth characteristics of two strains of *S. cerevisiae* under two

standard growth conditions.[67]. Work by Dikicioglu *et al.* examined the combined metabolomic and transcriptomic response of *S. cerevisiae* to transient perturbations in glucose and ammonium concentrations[59].

The emergence of genome-scale investigations has led to a deluge of information about all molecular layers in the cell. This in turn has provided a broader context in which metabolic engineering strategies can be evaluated. However, we note that many of the techniques discussed so far have focused on systematic *assessment* of the results of metabolic engineering strategies, rather than on systematic methods of *designing* strains to begin with. While some of these studies have used the insight gained from their evaluations to in turn design new strains, more systematic approaches to strain design are being developed – some of which are now capable of exploiting metabolomics datasets.

1.4 Computational Methods for Combining Metabolomics and Metabolic Engineering

One of the difficulties in applying metabolomics datasets to strain design is the volume of data produced in a metabolomics experiment. Computational approaches are well suited to systematically integrate large volumes of biochemical knowledge and data. As shown in Figure 1.2, they serve dual purposes: they can combine existing biochemical knowledge with strain design

objectives to execute putative metabolic engineering strategies *in silico* before taking the time and expense to execute them *in vivo*, and they can close the loop on experimental design by producing hypotheses that, when tested, can be used iteratively to refine broader biochemical knowledge and models. This in turn leads to improved predictive power for subsequent rounds of metabolic engineering design.

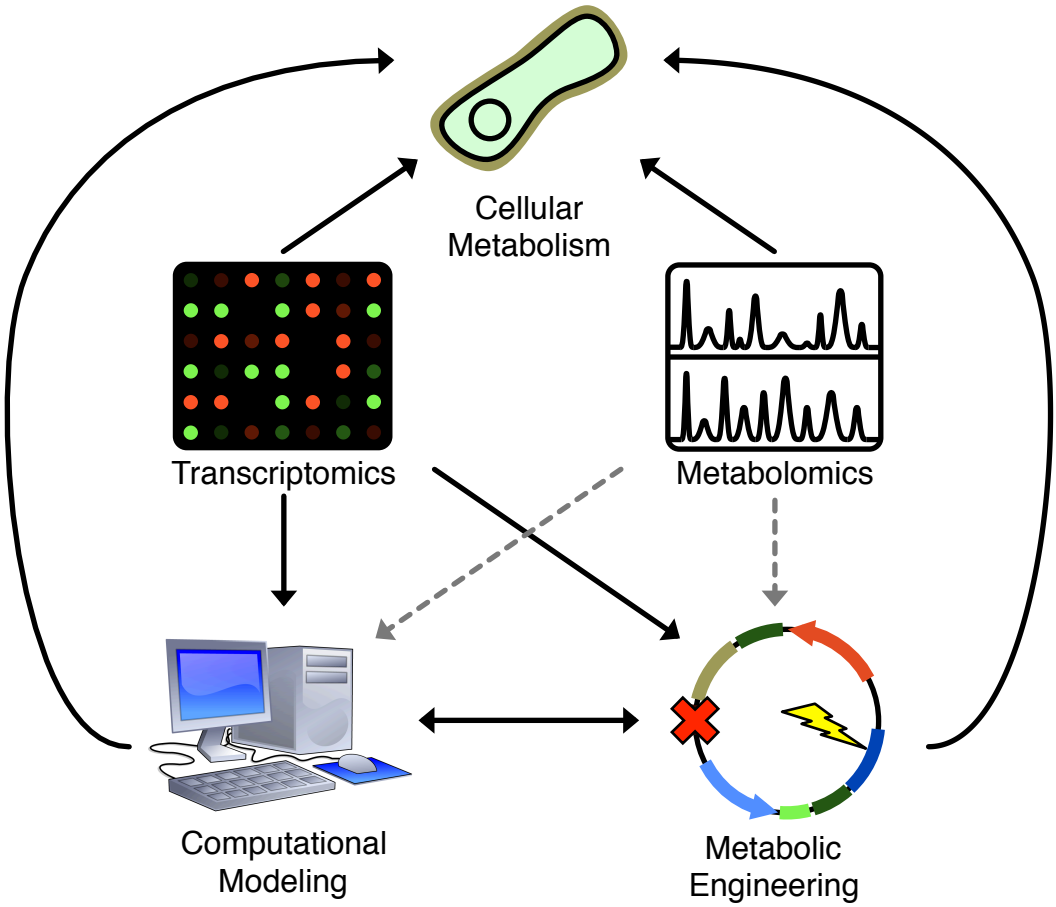

Figure 1.2. Applications of various techniques to understanding and manipulating cellular metabolism
Solid lines represent widely used strategies, dotted lines represent underused strategies. Both metabolomics and transcriptional profiling provide a direct readout that helps enable a deeper understanding of cellular metabolism, but only transcriptional profiling has seen widespread application to enhance standard computational modeling and metabolic engineering strategies. Integrating metabolomics data into metabolic engineering and computational modeling strategies would help bridge gaps in biochemical knowledge and improve our ability to control cellular metabolism.

One of the most powerful ways to extend this concept would be to include metabolomics data in the design and fitting of computational models. While there are many well-developed metabolic modeling strategies, most of these approaches have not yet been adapted to effectively leverage the additional information that metabolomics can offer. Nonetheless, these strategies have made substantial contributions to metabolic engineering. We discuss these computational approaches to establish how they have been used to date in metabolic engineering, to suggest how metabolomics can contribute to their effectiveness, and to highlight current efforts to integrate the two.

1.4.1 Constraint-based Models

The most basic models for metabolic engineering use simplified equations for bioreactor kinetics to empirically fit relationships between characteristics such as metabolite uptake or secretion and specific growth rate. While these models are useful as tools for investigating specific behaviors of existing strains, their small-scale and coarse-grained nature precludes broader application to directing engineering strategies, as well as the possibility of substantively integrating metabolomics data even when available[14].

Early biochemical modeling strategies initially sought to move beyond such simplistic approaches by compiling knowledge of metabolic pathways and enzyme kinetics into detailed mechanistic models to predict the dynamic behavior

of metabolite concentrations[127]. However, numerous issues hampered these efforts. Incomplete knowledge of the regulatory structure or the form of reaction rate equations can limit the accuracy of these types of models[128]. More importantly, the necessary kinetic parameters are for the most part unknown, or have only been measured *in vitro* for specific organisms (although recent efforts have sought to develop methods to determine kinetic parameters relevant to *in vivo* conditions via selected intracellular metabolite measurements[129,130]). Additionally, many models in systems biology also exhibit "sloppy" behaviors in regard to parametric sensitivity, where model performance is sensitive only to certain parameter combinations and consistent parameter estimation is difficult even with sufficient data[131].

To attempt to overcome these issues, "constraint-based" approaches that calculate metabolic fluxes primarily from stoichiometry were developed. This change of focus from dynamic metabolite levels to fluxes made sense, as the idea of optimizing and controlling metabolic fluxes has long been a fundamental part of metabolic engineering. These approaches allow flux calculations without the difficulties arising from parametric uncertainty by predicting flux distributions from the structure of the biochemical network and constraints on the feasible range of fluxes[132,133].

1.4.1.1 Flux Balance Analysis: The Prototypical Constraint-based Model

The prototypical CBM is Flux Balance Analysis (FBA)[132,133]. It is a modeling technique that uses metabolic network stoichiometry, a set of feasible flux ranges, and a cellular objective function to calculate an optimal flux distribution for a metabolic network[132,133].

First, mass balances are constructed around the metabolic network, describing the relationship between metabolite concentration changes and the metabolic reaction rates (fluxes). A key assumption of FBA is that the system is at steady-state (i.e., metabolites are neither accumulating or depleting). As a result, the non-linear differential equations describing the stoichiometric metabolite mass balances are reduced to a set of linear equations in terms of unknown reaction fluxes and a stoichiometric matrix describing the connectivity between metabolites and fluxes. Based on the fact that metabolic reactions occur more quickly than upstream cellular processes like intracellular signal transduction, gene transcription, and RNA translation, this steady-state assumption may often be a reasonable simplification.

Since the number of reactions exceeds the number of metabolites in biologically relevant metabolic networks, the system is underdetermined and there are multiple flux distributions that satisfy the stoichiometric equations. Additional information is used to select from this solution space. Upper and lower bounds

are set for each of the system fluxes, reflecting saturation rates (e.g. for transporters), enzyme reversibility (to specify irreversible reactions), or large nominal values necessary to bound the solution space (typical for intracellular fluxes). Then, an objective function representing the cell biology is constructed from a linear combination of the system fluxes. The most common objective functions in FBA seek to maximize biomass accumulation[132] or ATP production[132,134], or to minimize redox potential[132].

These constraints and objective are used to solve the FBA optimization problem:

$$\max_{\vec{v}} \vec{c}^T \vec{v}$$

$$\boldsymbol{S}\vec{v} = 0$$

$$\vec{v}_{LB} < \vec{v} < \vec{v}_{UB}$$

where $\vec{v}$ is the flux distribution vector, $\vec{c}$ is the vector of objective weights, $\boldsymbol{S}$ is the stoichiometric matrix, and $\vec{v}_{LB}$ and $\vec{v}_{UB}$ are lower and upper bounds on $\vec{v}$, respectively.

A key feature of FBA is that due to its linear constraints and objective function, the problem specifies a Linear Program (LP) and can be solved very efficiently with freely available computational libraries and tools.

A few examples of basic FBA for metabolic engineering include estimation of flux distributions from MFA measurements[32], prediction of knockout performances to assess proposed strain designs[6], use of artificial (virtual) metabolites to better capture flux ratios from ^{13}C MFA[135], systematic evaluation of different objective functions in *E. coli*[136], and evaluation of Elementary Flux Modes from flux measurements[137] and transcription datasets[138]. Tools such as the COnstraints Based Reconstruction and Analysis (COBRA) toolbox for MATLAB[139], CellNetAnalyzer[140], the Systems Biology Research Tool (SBRT)[141], and OptFlux[142] are available for simplified implementation of constraint-based modeling methods (and are listed in Table 1.1). In particular, multiple tools have been developed to augment the COBRA toolbox, such as visualization tools MapMaker and PathTracer[143], flux space sampling tool optGpSampler[144], and a port to Python, CobraPy[145].

1.4.1.2 Model Reconstructions

A prerequisite step in FBA is the reconstruction of genome-scale metabolic networks for the organism of interest. Reviews by Fiest *et al.*[146] and by Thiele & Palsson[147] describe this process in detail. The reconstruction process involves the synthesis of a genome-scale model from established biochemical and genomic knowledge, stored in publically accessible databases. These databases span various organisms and layers of biological information (e.g., biochemical pathways, transcription factors, and nutrient transport mechanisms). Examples of

relevant databases for this process (as well as several for metabolomics in general) are also listed in Table 1.1.

New strategies for efficiently performing parts of this process are being continuously developed (listed in Table 1.1)[148-151]. For example, the Model SEED[151] pipeline can automatically generate a metabolic model from annotated gene sequence data. Subsequent steps automatically determine biomass data via GrowMatch[152] by progressively adding and removing connections in the model with GapFill[148] and GapGen[151], respectively, until the model matches the available Growth/NoGrowth data. Other tools such as fastGapFill have been developed to better handle gap filling [153], and many such methods are available in the COBRA toolbox[139].

A few recent genome-scale reconstructions particularly relevant to metabolic engineering and metabolomics include models of *A. niger*[154], *C. acetobutylicum*[149], *Clostridium beijerinckii*[155], updated reconstructions of *E. coli*[156,157], and addition of lipid metabolism to a model for *S. cerevisiae*[158]. Notably, a reconstruction of *Mycoplasma genitalium*[159] has recently been incorporated into a whole-cell computational model by Karr *et al.*[160], who have suggested development of a similar model for *E. coli* as a possible next step. Network reconstructions for several dozen species are publicly available, and

programs such as MetRxn have been developed to aid comparison across model reconstructions[161]. Some examples of these are shown in Table 1.1.

1.4.1.3 Applications of Constraint-based Models in General to Metabolic Engineering

The original FBA framework has been supplemented with dozens of refinements broadly referred to as constraint-based models. While these models retain the optimization problem framework based on stoichiometric constraints, the flux constraints or objective function are altered. We direct the reader to reviews on the topic of FBA by Lee, Gianchanani, and others for more complete discussion of these methods[162,163], though it is instructive to analyze a few representative classes relevant to metabolic engineering. Further, we note that these approaches do not generally make use of metabolite measurements.

The most basic refinements are straightforward extensions of FBA, from adding a simplified representation of transcriptional regulatory constraints, to integrating uptake/effluxes and comparing against extracellular concentration profiles. Examples from this family include regulatory FBA (rFBA)[164-166], dynamic FBA (DFBA)[167,168], GIM3E[169], integrated FBA (iFBA)[168,170], and integrated-dynamic FBA (idFBA)[171]. Similar in application to idFBA, one CBM refinement by Vardi *et al.* modeled steady-state intracellular signal transduction using flux balancing, with added proportionality conditions to tie certain flux values together to act as

regulators. While these refinements demonstrate improved accuracy over basic FBA, most dynamic examples only make use of targeted extracellular concentration profiles as a means of constraining their dynamic elements. Using metabolomics data to constrain FBA solutions could provide these strategies direct information about intracellular metabolite levels in place of relying purely on the calculated fluxes to infer intracellular behaviors.

In the case of DFBA, one approach discussed in that work has been adapted for use in kinetic ODE modeling via DFBALab[172,173]. This method performs FBA to describe dynamics rapid enough to become steady-state compared to the outer model to calculate the RHS of an ODE model. However, this requires adaptations to the LP to make it compatible with the ODE solver, which may fail if the LP at a given intermediate step is infeasible[172,173]. This method has been applied in spatial bioreactor models[174] and to design a strain of *Dunaliella salina* for beta-carotene overproduction[175]

Another class of refinements to FBA comprises methods intended to better reduce the discrepancy between model predictions and experimental observations. Optimal metabolic network identification (OMNI) is used to identify discrepancies between measured and predicted fluxes, and then determine changes that need to be made to the model to better match the measurements[176]. This (and other[178,179]) methods would also directly benefit from

the additional information that metabolomics datasets can provide about the intracellular state of metabolism.

More directly relevant to metabolic engineering applications is a class of refinements focused on predicting the result of metabolic network alterations. An early and well-known example of this is Minimization of Metabolic Adjustment (MOMA), which formulates a quadratic programming (QP) problem to find the feasible flux distribution nearest to the original FBA solution in response to a gene knockout[180]. OptKnock uses a bi-level optimization framework to balance an FBA objective function with a desired overproduction target[181] (the work by Hua *et al.* compares the results of an evolved strain against an OptKnock prediction[48]).

Recent extensions to this work by Tervo and Reed take advantage of information about the feasibility space and shadow prices (marginal impact of moving a constraint) to allow OptKnock and related techniques to satisfy additional design criteria[182]. They demonstrated the tractability of these extensions by using it to engineer the ethanol production of *E. coli* on glucose[182]. OptGene uses a genetic algorithm to generate metabolic engineering strategies[183], and was used by Asadollahi *et al.* to design a strain exhibiting improved sesquiterpene production in *S. cerevisiae*[184]. Another extension is OptForce, which uses flux measurement data to generate a minimal set of engineering interventions required to guarantee

the desired overproduction target[185]. In addition to recapitulating already proven strategies for succinate production in *E. coli*, it also identified several other successful and nonintuitive strategies. The CosMos method developed by Cotton *et al.* uses a similar approach, but instead allows more flexibility in the constraints by allowing them to be selected by the algorithm rather than before run-time via e.g. Flux Variability Analysis[186]. An extension of Opt-Force, k-Opt-Force, was developed to incorporate targeted metabolite data into strain design by refining constraints using kinetics information, which was not accounted for previously[187]. A similar tool for performing strain design in the context of microbial communities is d-OptCom, which uses multi-level, multi-objective optimization to account for extracellular dynamics and interactions between multiple species[188]. Again, while many of these methods do not integrate metabolomics data into their calculations, the trend is moving towards incorporating them more and more frequently, with the goal of realizing significant improvements by harnessing such data.

1.4.1.4 Integrating Metabolomics Data into Constraint-based Models

As reviewed above, many constraint-based modeling strategies make negligible use of systems-scale metabolite data in their calculations. The requirement that organisms adhere not only to stoichiometric mass conservation but also to thermodynamic restrictions on energy and entropy provides one means of introducing metabolite concentrations into the constraints. Several constraint-

based model techniques make use of metabolite or metabolomics data in this fashion.

Table 1.1 Summary of Software Tools Presented in Chapter 1

Tool Name	*Reference*	*Description*
Metabolomics Data Processing		
ChromA	[75]	GC-MS Peak Alignment
Metab	[78]	GC-MS Data Statistical Analysis Package
MetaboAnalyst 2.0	[81]	Web-based Metabolomics Data Processing Pipeline
MetAlign	[79]	GC-MS and LC-MS Data Processing Pipeline
Mzmine 2	[77]	MS Data Processing Pipeline
SpectConnect	[74]	GC-MS Peak Alignment
Xalign	[72]	LC-MS Data Pre-processing
XCMS Online	[80]	Web-based Untargeted Metabolomics Pipeline
Constraint-Based Modeling		
anNET	[190]	MATLAB-based NET analysis
CellNetAnalyzer	[140]	MATLAB-based Metabolic and Signal Network Analysis
COBRA Toolbox	[139]	MATLAB-based FBA Toolbox Suite
OptFlux	[142]	Open Source, Modular Constraint-based Model Strain Design Software Toolbox
Systems Biology Research Tool	[141]	Open Source, Modular Systems Biology Computational Tool
Network Reconstruction		
GapFind, GapFill	[148]	Automated Network Gap Identification and Hypothesis Generation
GeneForce	[150]	Regulatory Rule Correction for Integrated Metabolic and Regulatory Models
MetRxn	[161]	Web-based Knowledgebase Comparison Tool
Model SEED	[151]	Web-based Generation, Optimization and Analysis of Genome-scale Metabolic Models
Databases		
BioCyc	[191]	Genome and pathway database for >2000 organisms
BRENDA	[192]	Comprehensive enzyme database, ~5000 enzymes
ChEBI	[193]	Biologically relevant small molecules and their properties
KEGG	[194]	Genomes, enzymatic pathways, and biological chemicals
MetaCyc	[195]	>1,900 metabolic pathways from >2,200 different organisms
PubChem	[196]	Biological activity and structures of small molecules

Network-embedded Thermodynamic (NET) Analysis combines pre-determined flux directions with quantitative metabolomics datasets and the metabolite Gibbs energy of formations to determine the feasible ranges of Gibbs free energy of reaction throughout the system[189]. This method can assess the internal consistency of a metabolomics dataset, predict thermodynamically feasible ranges for unmeasured metabolites, and identify putative sites of transcriptional regulation. anNET is a MATLAB implementation of the algorithm designed to facilitate straightforward application of NET analysis[190]. NET analysis of a

metabolomics dataset for *S. cerevisiae* by Canelas *et al.* revealed a thermodynamic inconsistency in modeled whole-cell NAD/NADH ratio[197]. Other studies have also used NET analysis to verify the thermodynamic consistency of their measurements in a variety of metabolic engineering contexts[10,60,198].

Henry *et al.* developed Thermodynamic Metabolic Flux Analysis (TMFA), a constraint-based modeling approach similar to NET analysis[199]. The *E. coli* network reconstruction published by Fiest *et al.* includes thermodynamic information, and the manuscript includes an assessment of thermodynamic consistency using TMFA[156]. More recently, Hamilton *et al.* used a genome-scale model of *E. coli* with experimental gene knockout and metabolomics data in an effort to validate it, finding good agreement with data and model predictions[200]. Recent improvements to group contribution methods should improve this approach's accuracy[201,202].

Several other methods that incorporate metabolite concentrations and thermodynamic constraints have been developed as well. For example, Bordel *et al.* developed a constraint-based model based on Ziegler's principle for the maximization of entropy production that uses non-equilibrium thermodynamics to identify flux bottlenecks[203]. Hoppe *et al.* designed a constraint-based model that combines thermodynamic constraints similar to TMFA and NET analysis with a penalty function for deviations from concentration measurements[204].

Other recent efforts have sought ways to take advantage of metabolomics as well. One such approach is kFBA, which uses kinetic rate laws with flux variability analysis to provide better bounds on flux values[205]. Information about metabolite levels can be incorporated into the rate law calculations.

Another set of approaches build from MFA, and seek to extend it to dynamic contexts. Several dynamic MFA approaches break time course data into intervals and use the metabolite slope to estimate the accumulation terms in the mass balances, and subsequently use this to calculate the flux distribution over that interval. These include the DMFA approach of Leighy *et al.*[206], the TremFlux approach of Kleessen *et al.*[207], the MetDFBA, and the unsteady-state FBA (uFBA) of Bordbar *et al.*[208] We discuss similar approaches for ODE models in section 1.4.2.4.

1.4.2 Kinetic Models

Constraint-based models have successfully directed numerous metabolic engineering projects. However, by construction they often ignore or have trouble dealing with dynamic metabolite behaviors that may have significant impact on final product titers, and in general they only indirectly make use of metabolite concentration measurements. Improved knowledge of network structures and strategies for dealing with parametric uncertainty have made ordinary differential equation (ODE) based models of metabolic kinetics increasingly viable tools for

strain design. These methods explicitly model intracellular concentrations, making them attractive and convenient frameworks for integrating metabolomics datasets.

1.4.2.1 Recent Developments in Kinetic Modeling Strategies

Kinetic models are built around explicit mathematical descriptions of enzyme-metabolite interactions. Natural choices for kinetic rate laws are mass action kinetics and Michaelis-Menten kinetics, but a review by Heijnen highlights several approximate rate laws that require fewer parameters and are relevant to metabolic engineering applications[209]. Included are discussions of S-systems and power-law kinetic rate laws, long established by the early efforts at kinetic modeling that developed into Biochemical Systems Theory (BST), and reviewed specifically in the context of metabolic networks recently by Voit[210-213].

Several investigators have sought to assess the properties of several of these approximate forms in the context of metabolic networks. These include studies of the glycolytic pathway in *S. cerevisiae* using a local linearization method[214] and lin-log kinetics[215,216], as well as study of central carbon metabolism in *E. coli* to compare lin-log kinetics, convenience kinetics, power law kinetics, and Michaelis-Menten kinetics[217].

1.4.2.2 Examples of Kinetic Models for Metabolic Engineering

Independent of metabolomics, kinetic models have already been applied in several recent metabolic engineering contexts. Rasler *et al.* constructed a dynamical model of cellular redox state in *S. cerevisiae* to assess response to oxidative stress[218]. Chassagnole *et al.* constructed a kinetic model of central carbon metabolism in *E. coli*.[128]. A similar study by Oh *et al.* constructed a model of lactic acid fermentation in *Lactococcus lactis*[21].

Ensemble approaches are also promising, and are in part a response to issues of parametric "sloppiness" which can preclude precise determination of kinetic parameters[131], and can be assessed with tools such as the STRIKE-GOLDD toolbox developed by Villaverde *et al*.[219] These ensemble approaches entail constructing a set of models that are structurally identical, but each using a different parameter set. Each model fits the training data comparably well, and the behavior of the whole ensemble is used to make predictions. Tran *et al.* developed a model for central metabolism in *E. coli* as a proof of concept[220]. This effort led to an ensemble model constructed by Rizk *et al.* that predicted the effect of gene knockouts on the production of aromatic compounds in *E. coli*[221], and a model by Contador *et al.* to predict flux data in L-lysine-producing *E. coli*[222]. More recent work has expanded to the genome-scale with a model of *E. coli* by Khodayari *et al.*[223] that used extensive metabolite measurements to refine their previous work in this system[224].

1.4.2.3 Integrating Metabolomics Datasets into Kinetic Models

The aforementioned ODE models generally do not explicitly look to integrate metabolomics data into their analyses, but some other efforts have. For example, Klimacek *et al.* used published kinetic parameters with time-course metabolomics measurements to assess metabolic control in xylose-fermenting *S. cerevisiae*[10]. Similarly, a model of nitrogen assimilation in *E. coli* developed by Yuan *et al.* used kinetic parameters from the literature together with a genetic algorithm to fit undefined parameters to metabolomics data[61]. While these examples both use metabolomics data, the modeling strategies focused only on capturing the dynamics of individual pathways and modules—not the whole metabolic network.

Two early examples attempted to apply metabolomics measurements to models of an entire metabolic network. Yizhak *et al.* developed a constraint-based modeling approach they referred to as integrative "omics"-metabolic analysis (IOMA), which solves a QP problem in a constraint-based model to fit a flux distribution to proteomic and metabolomics data[225]. Their approach introduces a reaction rate model based on Michaelis-Menten kinetics, and uses proteomic data in conjunction with metabolomics data to fit the kinetic parameters and satisfy the steady-state flux requirement in the unperturbed system. This results in a defined system of ODEs. When compared against FBA and MOMA to predict the effects of gene knockouts based on an erythrocyte kinetic model and

published data for *E. coli*, IOMA demonstrated significantly improved recall, precision, and accuracy.

A mass action stoichiometric simulation (MASS) modeling strategy described by Jamshidi *et al.* follows a different scheme, by fitting thermodynamic equilibrium constants to a measured metabolomics dataset and a calculated flux distribution[226]. The kinetic rate law equations are then solved to obtain the forward reaction rate constants. As a proof of concept, they used this method to construct a human erythrocyte model and demonstrated many of the key behaviors of the original erythrocyte model.

Notably, these last two methods both take advantage of constraint-based modeling strategies, but result in ODE-based kinetic models that can subsequently be used for strain design. This reflects the complementary nature of metabolite fluxes and concentrations, especially when faced with system-wide parametric uncertainty. However, to fully capture the wide-ranging dynamics that directly and indirectly contribute to the often subtle and nonintuitive behaviors exhibited in engineered strains, additional model detail is necessary. More advanced modeling strategies will need to find ways to integrate additional information, including proteomics and transcriptomics, to meet this need.

A more recent example by Mannan *et al.* used steady-state data from the Keio multi-omics dataset to train a hybrid kinetic-FBA model of E coli metabolism[227]. Using the genome-scale model at steady state to augment the ODE model allowed them to close the open system of their ODE model, and to identify two growth phenotypes that arise due to bistability that reflected a real subpopulation difference in experimental populations[227].

1.4.2.4 Dynamic Flux Estimation

Previous we described several methods that sought to expand MFA to wider contexts, such as dynamics. Here, we discuss work that uses similar approaches, but with the emphasis typically placed on producing ODE models.

One set of tools can be described under the umbrella of Dynamic Flux Estimation (DFE), a procedure for generating fitted models from metabolite time course data and knowledge of the system stoichiometry[228]. As in DMFA, the slope of the data is estimated and used to calculate a dynamic flux distribution. In turn, model rate law equations can be identified as independent fitting problems, reducing error compensation that can occur when simultaneously solving for all model parameters[228]. This builds off previous work in the field of BST, such as the Artificial Neural Network (ANN) data smoothing working of Voit and Almeida[229], or the alternating regression method of Chou *et al.*[230] for solving for parameters in S-systems models.

This DFE framework involves three key steps. The first step involves smoothing the data and estimating the slope of the metabolite concentrations[228]. Common methods for accomplishing this include polynomials[231] and splines[232], and our own work exploring the use of an impulse function is described in the next chapter[233]. More sophisticated methods include techniques such as piecewise approximation of sub-intervals[234].

The second step is to apply the mass balance equations with the estimated slopes to determine the dynamic flux distribution. One consideration at this step is the preservation of conservation of mass when data is noisy or the model is incomplete[228,235]. The basis for calculating flux distributions may be derived from DFBA, as is the case in the MetFBA procedure of Willemsen *et al.*[236], or by breaking the flux set into dynamic and static subsets and using FBA for the static sets, as was described by Yugi *et al.*[237]. Many of the DMFA methods described previously in Section 1.4.1 also focus on this problem. Alternatively, the flux distribution may be bounded to Elementary Flux Modes plus dynamic perturbations[238]. The relationship between metabolites and fluxes may also be cast as Guassian Process models, as was done by Zurauskiene *et al*[239].

The final step involves the application of a kinetic rate law model, such as S-systems[230,240] or Generalized Mass Action (GMA) power-law kinetics[228,241]. The previous step generated the dynamic flux distribution, and individual fluxes can

be matched against the corresponding metabolites to identify kinetic rate law parameters using regression. This breaks the highly interconnected parameter-fitting problem (which often requires performing integrations to solve) down in to a series of decoupled algebraic equations. The method of Chou *et al.* identifies a subset of the flux-metabolite relationships that can be estimated from time series data by isolating one metabolite that varies when the others in the rate law hold constant, and generates piece-wise a reconstruction of this relationship[242].

While not building from the DFE framework *per se*, several parameter optimization approaches by Jia *et al.* use similar concepts[243,244]. Somewhat similar to the alternating regression procedure of Chou *et al.*[244], one approach alternates between slope and concentration error calculations to iteratively refine model parameters[243]. Another breaks the flux distribution into subsets, and uses these subsets to sequentially calculate a flux distribution using decoupled parameter fitting on one subset[244].

1.4.2.5 Whole-cell modeling strategies

One relatively recent development mentioned briefly before is the publication of a whole-cell model of *Mycoplasma genitalium*[160]. This model incorporated interacting modules for gene expression, protein synthesis, DNA synthesis, metabolism, growth, replication, and other cellular processes to produce a simulation of *M. genitalium* growth through one full division. This model was

generated from and validated against a wide variety of data types, including metabolomics, proteomics, and transcriptomics. Later efforts with this model have led to exploration of its deficiencies and have used its predictions to guide experimental efforts, such as determination of kinetics parameters[245]. Offshoots from this work include new strategies for formulating objectives and accounting for time-varying conditions and metabolite utilization across modules[246]. The resulting analysis has been made accessible at WholeCellSimDB in an effort to help develop other whole-cell modeling projects[247].

1.5 Summary

Metabolomics is the global analysis of the metabolic content of a living system. While it has found increasing application in fundamental biological research and in fields of clinical interest (e.g. disease biomarker discovery), there is surprisingly little use of metabolomics approaches to drive metabolic engineering efforts. Existing experimental approaches to supplement rational metabolic engineering efforts typically focus instead on the determination of flux with MFA techniques, or the use of enzyme assays and analytical platforms such as HPLC for highly targeted metabolite measurements.

While global analysis methods have been used to better predict and assess the effects of metabolic engineering modifications, the techniques most typically used have been transcriptomic or proteomic analyses—not metabolomics. While this

may have previously been due to the relative immaturity of metabolomics techniques, the current technology in the field should allow for easy integration of metabolomics into metabolic engineering workflows.

Direct applications of metabolomics datasets to metabolic engineering include expanding the existing narrowly targeted analysis methods to a broader scope, identifying non-intuitive mutations in strains produced by directed evolution, and adding direct metabolic context to other global analysis datasets. Computational approaches have also begun to integrate metabolomics datasets through thermodynamic constraints in constraint-based models or even more directly in the case of some kinetic models.

However, long-term strategies will need to find novel ways of incorporating the system-wide perspective provided by metabolomics and other global analysis methods. Such approaches will facilitate strain design based on increasingly detailed mechanistic descriptions and enable us to engineer strains towards any arbitrary product, not just those well-suited to high-throughput screens and directed evolution. Computational methods have a great deal of potential here.

In the case of kinetic models, combining the metabolome and proteome can help address issues of *in vivo* parameter estimation. Ensemble models are proving to be one effective method of addressing issues of parametric uncertainty and

model “sloppiness”, and metabolomics provides substantial data to better constrain feasible parameter sets. With proper alterations and structural changes, constraint-based models may be able to more explicitly incorporate metabolite concentrations into constraints to capture effects such as allosteric regulation.

It is this last case we will focus on primarily in this thesis. In the following chapters, we will discuss our contributions to computational strategies for leveraging metabolomics data for metabolic engineering. These efforts improve on gaps in existing processing methods for metabolite time course data and expand existing CBM frameworks to allow us to more completely integrate metabolomics data. The work presented is aimed ultimately at using metabolomics to improve the efficiency and accuracy of prospective strain designs; we close with a discussion of the next steps necessary to apply our contributions towards that end and explore several potential complementary strategies.

1.6 References

1 Dromms, R. A. & Styczynski, M. P. Systematic applications of metabolomics in metabolic engineering. *Metabolites* **2**, 1090-1122, doi:10.3390/metabo2041090 (2012).

2 Stephanopoulos, G. Metabolic fluxes and metabolic engineering. *Metab Eng* **1**, 1-11, doi:10.1006/mben.1998.0101 (1999).

3 Zaldivar, J. *et al.* Fermentation performance and intracellular metabolite patterns in laboratory and industrial xylose-fermenting Saccharomyces cerevisiae. *Appl Microbiol Biotechnol* **59**, 436-442, doi:10.1007/s00253-002-1056-y (2002).

4 Sonderegger, M. & Sauer, U. Evolutionary engineering of Saccharomyces cerevisiae for anaerobic growth on xylose. *Appl Environ Microbiol* **69**, 1990-1998, doi:10.1128/aem.69.4.1990-1998.2003 (2003).

5 Sonderegger, M., Jeppsson, M., Hahn-Hagerdal, B. & Sauer, U. Molecular basis for anaerobic growth of Saccharomyces cerevisiae on xylose, investigated by global gene expression and metabolic flux analysis. *Appl Environ Microbiol* **70**, 2307-2317, doi:10.1128/aem.70.4.2307-2317.2004 (2004).

6 Bro, C., Regenberg, B., Forster, J. & Nielsen, J. In silico aided metabolic engineering of Saccharomyces cerevisiae for improved bioethanol production. *Metab Eng* **8**, 102-111, doi:10.1016/j.ymben.2005.09.007 (2006).

7 Meijer, S., Panagiotou, G., Olsson, L. & Nielsen, J. Physiological characterization of xylose metabolism in Aspergillus niger under oxygen-limited conditions. *Biotechnol Bioeng* **98**, 462-475, doi:10.1002/bit.21397 (2007).

8 Panagiotou, G. *et al.* Systems analysis unfolds the relationship between the phosphoketolase pathway and growth in Aspergillus nidulans. *PLoS One* **3**, e3847, doi:10.1371/journal.pone.0003847 (2008).

9 Wisselink, H. W., Toirkens, M. J., Wu, Q., Pronk, J. T. & van Maris, A. J. Novel evolutionary engineering approach for accelerated utilization of glucose, xylose, and arabinose mixtures by engineered Saccharomyces cerevisiae strains. *Appl Environ Microbiol* **75**, 907-914, doi:10.1128/AEM.02268-08 (2009).

10 Klimacek, M., Krahulec, S., Sauer, U. & Nidetzky, B. Limitations in xylose-fermenting Saccharomyces cerevisiae, made evident through

comprehensive metabolite profiling and thermodynamic analysis. *Appl Environ Microbiol* **76**, 7566-7574, doi:10.1128/AEM.01787-10 (2010).

11 Hasunuma, T. *et al.* Metabolic pathway engineering based on metabolomics confers acetic and formic acid tolerance to a recombinant xylose-fermenting strain of Saccharomyces cerevisiae. *Microb Cell Fact* **10**, 2, doi:10.1186/1475-2859-10-2 (2011).

12 Koppram, R., Albers, E. & Olsson, L. Evolutionary engineering strategies to enhance tolerance of xylose utilizing recombinant yeast to inhibitors derived from spruce biomass. *Biotechnol Biofuels* **5**, 32, doi:10.1186/1754-6834-5-32 (2012).

13 Zhang, W. & Geng, A. Improved ethanol production by a xylose-fermenting recombinant yeast strain constructed through a modified genome shuffling method. *Biotechnol Biofuels* **5**, 46, doi:10.1186/1754-6834-5-46 (2012).

14 Kresnowati, M. T., van Winden, W. A., van Gulik, W. M. & Heijnen, J. J. Dynamic in vivo metabolome response of Saccharomyces cerevisiae to a stepwise perturbation of the ATP requirement for benzoate export. *Biotechnol Bioeng* **99**, 421-441, doi:10.1002/bit.21557 (2008).

15 Sillers, R., Chow, A., Tracy, B. & Papoutsakis, E. T. Metabolic engineering of the non-sporulating, non-solventogenic Clostridium acetobutylicum strain M5 to produce butanol without acetone demonstrate the robustness of the acid-formation pathways and the importance of the electron balance. *Metab Eng* **10**, 321-332, doi:10.1016/j.ymben.2008.07.005 (2008).

16 Song, H., Jang, S. H., Park, J. M. & Lee, S. Y. Modeling of batch fermentation kinetics for succinic acid production by Mannheimia succiniciproducens. *Biochemical Engineering Journal* **40**, 107-115, doi:10.1016/j.bej.2007.11.021 (2008).

17 Zhang, K., Sawaya, M. R., Eisenberg, D. S. & Liao, J. C. Expanding metabolism for biosynthesis of nonnatural alcohols. *Proceedings of the National Academy of Sciences of the United States of America* **105**, 20653-20658, doi:10.1073/pnas.0807157106 (2008).

18 Hou, J., Lages, N. F., Oldiges, M. & Vemuri, G. N. Metabolic impact of redox cofactor perturbations in Saccharomyces cerevisiae. *Metab Eng* **11**, 253-261, doi:10.1016/j.ymben.2009.05.001 (2009).

19 Sillers, R., Al-Hinai, M. A. & Papoutsakis, E. T. Aldehyde-alcohol dehydrogenase and/or thiolase overexpression coupled with CoA

transferase downregulation lead to higher alcohol titers and selectivity in Clostridium acetobutylicum fermentations. *Biotechnol Bioeng* **102**, 38-49, doi:10.1002/bit.22058 (2009).

20 Trinh, C. T., Huffer, S., Clark, M. E., Blanch, H. W. & Clark, D. S. Elucidating mechanisms of solvent toxicity in ethanologenic Escherichia coli. *Biotechnol Bioeng* **106**, 721-730, doi:10.1002/bit.22743 (2010).

21 Oh, E. *et al.* Dynamic modeling of lactic acid fermentation metabolism with Lactococcus lactis. *J Microbiol Biotechnol* **21**, 162-169, doi:10.4014/jmb.1007.07066 (2011).

22 Pereira, F. B., Guimaraes, P. M., Teixeira, J. A. & Domingues, L. Robust industrial Saccharomyces cerevisiae strains for very high gravity bio-ethanol fermentations. *J Biosci Bioeng* **112**, 130-136, doi:10.1016/j.jbiosc.2011.03.022 (2011).

23 Trinh, C. T., Li, J., Blanch, H. W. & Clark, D. S. Redesigning Escherichia coli metabolism for anaerobic production of isobutanol. *Appl Environ Microbiol* **77**, 4894-4904, doi:10.1128/AEM.00382-11 (2011).

24 Aboka, F. O. *et al.* Identification of informative metabolic responses using a minibioreactor: a small step change in the glucose supply rate creates a large metabolic response in Saccharomyces cerevisiae. *Yeast* **29**, 95-110, doi:10.1002/yea.2892 (2012).

25 Lu, M. *et al.* Identification of factors regulating Escherichia coli 2,3-butanediol production by continuous culture and metabolic flux analysis. *J Microbiol Biotechnol* **22**, 659-667, doi:10.4014/jmb.1112.12018 (2012).

26 Villas-Boas, S. G., Kesson, M. & Nielsen, J. Biosynthesis of glyoxylate from glycine in Saccharomyces cerevisiae. *FEMS Yeast Res* **5**, 703-709, doi:10.1016/j.femsyr.2005.03.001 (2005).

27 Wu, L. *et al.* Short-term metabolome dynamics and carbon, electron, and ATP balances in chemostat-grown Saccharomyces cerevisiae CEN.PK 113-7D following a glucose pulse. *Appl Environ Microbiol* **72**, 3566-3577, doi:10.1128/AEM.72.5.3566-3577.2006 (2006).

28 Costenoble, R. *et al.* 13C-Labeled metabolic flux analysis of a fed-batch culture of elutriated Saccharomyces cerevisiae. *FEMS Yeast Res* **7**, 511-526, doi:10.1111/j.1567-1364.2006.00199.x (2007).

29 Kleijn, R. J. *et al.* Metabolic flux analysis of a glycerol-overproducing Saccharomyces cerevisiae strain based on GC-MS, LC-MS and NMR-

derived C-labelling data. *FEMS Yeast Res* **7**, 216-231, doi:10.1111/j.1567-1364.2006.00180.x (2007).

30 Nasution, U., van Gulik, W. M., Ras, C., Proell, A. & Heijnen, J. J. A metabolome study of the steady-state relation between central metabolism, amino acid biosynthesis and penicillin production in Penicillium chrysogenum. *Metab Eng* **10**, 10-23, doi:10.1016/j.ymben.2007.07.001 (2008).

31 Moxley, J. F. *et al.* Linking high-resolution metabolic flux phenotypes and transcriptional regulation in yeast modulated by the global regulator Gcn4p. *Proceedings of the National Academy of Sciences of the United States of America* **106**, 6477-6482, doi:10.1073/pnas.0811091106 (2009).

32 Suthers, P. F., Chang, Y. J. & Maranas, C. D. Improved computational performance of MFA using elementary metabolite units and flux coupling. *Metab Eng* **12**, 123-128, doi:10.1016/j.ymben.2009.10.002 (2010).

33 Ravikirthi, P., Suthers, P. F. & Maranas, C. D. Construction of an E. Coli genome-scale atom mapping model for MFA calculations. *Biotechnol Bioeng* **108**, 1372-1382, doi:10.1002/bit.23070 (2011).

34 Wu, L. *et al.* Quantitative analysis of the microbial metabolome by isotope dilution mass spectrometry using uniformly 13C-labeled cell extracts as internal standards. *Anal Biochem* **336**, 164-171, doi:10.1016/j.ab.2004.09.001 (2005).

35 Buscher, J. M., Czernik, D., Ewald, J. C., Sauer, U. & Zamboni, N. Cross-platform comparison of methods for quantitative metabolomics of primary metabolism. *Anal Chem* **81**, 2135-2143, doi:10.1021/ac8022857 (2009).

36 Choi, J. & Antoniewicz, M. R. Tandem mass spectrometry: a novel approach for metabolic flux analysis. *Metab Eng* **13**, 225-233, doi:10.1016/j.ymben.2010.11.006 (2011).

37 Choi, J., Grossbach, M. T. & Antoniewicz, M. R. Measuring complete isotopomer distribution of aspartate using gas chromatography/tandem mass spectrometry. *Anal Chem* **84**, 4628-4632, doi:10.1021/ac300611n (2012).

38 Srour, O., Young, J. D. & Eldar, Y. C. Fluxomers: a new approach for 13C metabolic flux analysis. *BMC Syst Biol* **5**, 129, doi:10.1186/1752-0509-5-129 (2011).

39 Chang, Y., Suthers, P. F. & Maranas, C. D. Identification of optimal measurement sets for complete flux elucidation in metabolic flux analysis

experiments. *Biotechnol Bioeng* **100**, 1039-1049, doi:10.1002/bit.21926 (2008).

40 Antoniewicz, M. R., Kelleher, J. K. & Stephanopoulos, G. Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions. *Metab Eng* **9**, 68-86, doi:10.1016/j.ymben.2006.09.001 (2007).

41 Young, J. D., Walther, J. L., Antoniewicz, M. R., Yoo, H. & Stephanopoulos, G. An elementary metabolite unit (EMU) based method of isotopically nonstationary flux analysis. *Biotechnol Bioeng* **99**, 686-699, doi:10.1002/bit.21632 (2008).

42 Raamsdonk, L. M. *et al.* A functional genomics strategy that uses metabolome data to reveal the phenotype of silent mutations. *Nature Biotechnology* **19**, 45-50 (2001).

43 Allen, J. *et al.* High-throughput classification of yeast mutants for functional genomics using metabolic footprinting. *Nat Biotechnol* **21**, 692-696, doi:10.1038/nbt823 (2003).

44 Chen, H., Pan, Z., Talaty, N., Raftery, D. & Cooks, R. G. Combining desorption electrospray ionization mass spectrometry and nuclear magnetic resonance for differential metabolomics without sample preparation. *Rapid Commun Mass Spectrom* **20**, 1577-1584, doi:10.1002/rcm.2474 (2006).

45 Sreekumar, A. *et al.* Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression. *Nature* **457**, 910-914, doi:10.1038/nature07762 (2009).

46 Huang, Z. *et al.* Holistic metabonomic profiling of urine affords potential early diagnosis for bladder and kidney cancers. *Metabolomics* **9**, 119-129, doi:10.1007/s11306-012-0433-5 (2012).

47 Kromer, J. O., Sorgenfrei, O., Klopprogge, K., Heinzle, E. & Wittmann, C. In-depth profiling of lysine-producing Corynebacterium glutamicum by combined analysis of the transcriptome, metabolome, and fluxome. *J Bacteriol* **186**, 1769-1784, doi:10.1128/jb.186.6.1769-1784.2004 (2004).

48 Hua, Q., Joyce, A. R., Fong, S. S. & Palsson, B. O. Metabolic analysis of adaptive evolution for in silico-designed lactate-producing strains. *Biotechnol Bioeng* **95**, 992-1002, doi:10.1002/bit.21073 (2006).

49 Mas, S., Villas-Boas, S. G., Hansen, M. E., Akesson, M. & Nielsen, J. A comparison of direct infusion MS and GC-MS for metabolic footprinting of

yeast mutants. *Biotechnol Bioeng* **96**, 1014-1022, doi:10.1002/bit.21194 (2007).

50 van der Werf, M. J. *et al.* Comprehensive analysis of the metabolome of Pseudomonas putida S12 grown on different carbon sources. *Mol Biosyst* **4**, 315-327, doi:10.1039/b717340g (2008).

51 Kleijn, R. J. *et al.* Metabolic fluxes during strong carbon catabolite repression by malate in Bacillus subtilis. *The Journal of biological chemistry* **285**, 1587-1596, doi:10.1074/jbc.M109.061747 (2010).

52 Taymaz-Nikerel, H. *et al.* Development and application of a differential method for reliable metabolome analysis in Escherichia coli. *Anal Biochem* **386**, 9-19, doi:10.1016/j.ab.2008.11.018 (2009).

53 Kummel, A. *et al.* Differential glucose repression in common yeast strains in response to HXK2 deletion. *FEMS Yeast Res* **10**, 322-332, doi:10.1111/j.1567-1364.2010.00609.x (2010).

54 Taymaz-Nikerel, H., van Gulik, W. M. & Heijnen, J. J. Escherichia coli responds with a rapid and large change in growth rate upon a shift from glucose-limited to glucose-excess conditions. *Metab Eng* **13**, 307-318, doi:10.1016/j.ymben.2011.03.003 (2011).

55 Canelas, A. B., Ras, C., ten Pierick, A., van Gulik, W. M. & Heijnen, J. J. An in vivo data-driven framework for classification and quantification of enzyme kinetics and determination of apparent thermodynamic data. *Metab Eng* **13**, 294-306, doi:10.1016/j.ymben.2011.02.005 (2011).

56 Piddocke, M. P. *et al.* Revealing the beneficial effect of protease supplementation to high gravity beer fermentations using "-omics" techniques. *Microb Cell Fact* **10**, 27, doi:10.1186/1475-2859-10-27 (2011).

57 Carnicer, M. *et al.* Development of quantitative metabolomics for Pichia pastoris. *Metabolomics* **8**, 284-298, doi:10.1007/s11306-011-0308-1 (2012).

58 Carnicer, M. *et al.* Quantitative metabolomics analysis of amino acid metabolism in recombinant Pichia pastoris under different oxygen availability conditions. *Microb Cell Fact* **11**, 83, doi:10.1186/1475-2859-11-83 (2012).

59 Dikicioglu, D., Dunn, W. B., Kell, D. B., Kirdar, B. & Oliver, S. G. Short- and long-term dynamic responses of the metabolic network and gene expression in yeast to a transient change in the nutrient environment. *Mol Biosyst* **8**, 1760-1774, doi:10.1039/c2mb05443d (2012).

60 Fendt, S. M. *et al.* Tradeoff between enzyme and metabolite efficiency maintains metabolic homeostasis upon perturbations in enzyme capacity. *Mol Syst Biol* **6**, 356, doi:10.1038/msb.2010.11 (2010).

61 Yuan, J. *et al.* Metabolomics-driven quantitative analysis of ammonia assimilation in E. coli. *Mol Syst Biol* **5**, 302, doi:10.1038/msb.2009.60 (2009).

62 Buescher, J. M., Moco, S., Sauer, U. & Zamboni, N. Ultrahigh performance liquid chromatography-tandem mass spectrometry method for fast and robust quantification of anionic and aromatic metabolites. *Anal Chem* **82**, 4403-4412, doi:10.1021/ac100101d (2010).

63 Christen, S. & Sauer, U. Intracellular characterization of aerobic glucose metabolism in seven yeast species by 13C flux analysis and metabolomics. *FEMS Yeast Res* **11**, 263-272, doi:10.1111/j.1567-1364.2010.00713.x (2011).

64 McKee, A. E. *et al.* Manipulation of the carbon storage regulator system for metabolite remodeling and biofuel production in Escherichia coli. *Microb Cell Fact* **11**, 79, doi:10.1186/1475-2859-11-79 (2012).

65 Hirayama, A. *et al.* Metabolic profiling reveals new serum biomarkers for differentiating diabetic nephropathy. *Anal Bioanal Chem* **404**, 3101-3109, doi:10.1007/s00216-012-6412-x (2012).

66 Toya, Y., Nakahigashi, K., Tomita, M. & Shimizu, K. Metabolic regulation analysis of wild-type and arcA mutant Escherichia coli under nitrate conditions using different levels of omics data. *Mol Biosyst* **8**, 2593-2604, doi:10.1039/c2mb25069a (2012).

67 Canelas, A. B. *et al.* Integrated multilaboratory systems biology reveals differences in protein metabolism between two reference yeast strains. *Nat Commun* **1**, 145, doi:10.1038/ncomms1150 (2010).

68 Cajka, T., Riddellova, K., Tomaniova, M. & Hajslova, J. Ambient mass spectrometry employing a DART ion source for metabolomic fingerprinting/profiling: a powerful tool for beer origin recognition. *Metabolomics* **7**, 500-508, doi:10.1007/s11306-010-0266-z (2011).

69 Nikolaev, Y. V., Kochanowski, K., Link, H., Sauer, U. & Allain, F. H. Systematic Identification of Protein-Metabolite Interactions in Complex Metabolite Mixtures by Ligand-Detected Nuclear Magnetic Resonance Spectroscopy. *Biochemistry* **55**, 2590-2600, doi:10.1021/acs.biochem.5b01291 (2016).

70 Blaise, B. J., Gouel-Cheron, A., Floccard, B., Monneret, G. & Allaouchiche, B. Metabolic phenotyping of traumatized patients reveals a susceptibility to sepsis. *Anal Chem* **85**, 10850-10855, doi:10.1021/ac402235q (2013).

71 Villas-Boas, S. G., Moxley, J. F., Akesson, M., Stephanopoulos, G. & Nielsen, J. High-throughput metabolic state analysis: the missing link in integrated functional genomics of yeasts. *The Biochemical journal* **388**, 669-677, doi:10.1042/BJ20041162 (2005).

72 Zhang, X., Asara, J. M., Adamec, J., Ouzzani, M. & Elmagarmid, A. K. Data pre-processing in liquid chromatography-mass spectrometry-based proteomics. *Bioinformatics* **21**, 4054-4059, doi:10.1093/bioinformatics/bti660 (2005).

73 Smith, C. A., Want, E. J., O'Maille, G., Abagyan, R. & Siuzdak, G. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem* **78**, 779-787, doi:10.1021/ac051437y (2006).

74 Styczynski, M. P. *et al.* Systematic identification of conserved metabolites in GC/MS data for metabolomics and biomarker discovery. *Anal Chem* **79**, 966-973, doi:10.1021/ac0614846 (2007).

75 Hoffmann, N. & Stoye, J. ChromA: signal-based retention time alignment for chromatography-mass spectrometry data. *Bioinformatics* **25**, 2080-2081, doi:10.1093/bioinformatics/btp343 (2009).

76 Xia, J., Psychogios, N., Young, N. & Wishart, D. S. MetaboAnalyst: a web server for metabolomic data analysis and interpretation. *Nucleic Acids Res* **37**, W652-660, doi:10.1093/nar/gkp356 (2009).

77 Pluskal, T., Castillo, S., Villar-Briones, A. & Oresic, M. MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics* **11**, 395, doi:10.1186/1471-2105-11-395 (2010).

78 Aggio, R., Villas-Boas, S. G. & Ruggiero, K. Metab: an R package for high-throughput analysis of metabolomics data generated by GC-MS. *Bioinformatics* **27**, 2316-2318, doi:10.1093/bioinformatics/btr379 (2011).

79 Lommen, A. & Kools, H. J. MetAlign 3.0: performance enhancement by efficient use of advances in computer hardware. *Metabolomics* **8**, 719-726, doi:10.1007/s11306-011-0369-1 (2012).

80 Tautenhahn, R., Patti, G. J., Rinehart, D. & Siuzdak, G. XCMS Online: a web-based platform to process untargeted metabolomic data. *Anal Chem* **84**, 5035-5039, doi:10.1021/ac300698c (2012).

81 Xia, J., Mandal, R., Sinelnikov, I. V., Broadhurst, D. & Wishart, D. S. MetaboAnalyst 2.0--a comprehensive server for metabolomic data analysis. *Nucleic Acids Res* **40**, W127-133, doi:10.1093/nar/gks374 (2012).

82 Wei, X. *et al.* MetPP: a computational platform for comprehensive two-dimensional gas chromatography time-of-flight mass spectrometry-based metabolomics. *Bioinformatics* **29**, 1786-1792, doi:10.1093/bioinformatics/btt275 (2013).

83 Li, S. *et al.* Predicting network activity from high throughput metabolomics. *PLoS Comput Biol* **9**, e1003123, doi:10.1371/journal.pcbi.1003123 (2013).

84 Uppal, K. *et al.* xMSanalyzer: automated pipeline for improved feature detection and downstream analysis of large-scale, non-targeted metabolomics data. *BMC Bioinformatics* **14**, 15, doi:10.1186/1471-2105-14-15 (2013).

85 Dunn, W. B., Broadhurst, D. I., Atherton, H. J., Goodacre, R. & Griffin, J. L. Systems level studies of mammalian metabolomes: the roles of mass spectrometry and nuclear magnetic resonance spectroscopy. *Chem Soc Rev* **40**, 387-426, doi:10.1039/b906712b (2011).

86 Devantier, R., Scheithauer, B., Villas-Boas, S. G., Pedersen, S. & Olsson, L. Metabolite profiling for analysis of yeast stress response during very high gravity ethanol fermentations. *Biotechnol Bioeng* **90**, 703-714, doi:10.1002/bit.20457 (2005).

87 van den Berg, R. A., Hoefsloot, H. C., Westerhuis, J. A., Smilde, A. K. & van der Werf, M. J. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics* **7**, 142, doi:10.1186/1471-2164-7-142 (2006).

88 Højer-Pedersen, J., Smedsgaard, J. & Nielsen, J. The yeast metabolome addressed by electrospray ionization mass spectrometry: Initiation of a mass spectral library and its applications for metabolic footprinting by direct infusion mass spectrometry. *Metabolomics* **4**, 393-405, doi:10.1007/s11306-008-0132-4 (2008).

89 MacKenzie, D. A. *et al.* Relatedness of medically important strains of Saccharomyces cerevisiae as revealed by phylogenetics and metabolomics. *Yeast* **25**, 501-512, doi:10.1002/yea.1601 (2008).

90 Barrett, C. L., Herrgard, M. J. & Palsson, B. Decomposing complex reaction networks using random sampling, principal component analysis and basis rotation. *BMC Syst Biol* **3**, 30, doi:10.1186/1752-0509-3-30 (2009).

91 Chong, W. P. *et al.* Metabolomics profiling of extracellular metabolites in recombinant Chinese Hamster Ovary fed-batch culture. *Rapid Commun Mass Spectrom* **23**, 3763-3771, doi:10.1002/rcm.4328 (2009).

92 Kamei, Y. *et al.* GABA metabolism pathway genes, UGA1 and GAD1, regulate replicative lifespan in Saccharomyces cerevisiae. *Biochem Biophys Res Commun* **407**, 185-190, doi:10.1016/j.bbrc.2011.02.136 (2011).

93 Peres-Neto, P. R., Jackson, D. A. & Somers, K. M. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis* **49**, 974-997, doi:10.1016/j.csda.2004.06.015 (2005).

94 Wold, S., Ruhe, A., Wold, H. & Dunn, W. J. The Collinearity Problem in Linear-Regression - the Partial Least-Squares (Pls) Approach to Generalized Inverses. *Siam Journal on Scientific and Statistical Computing* **5**, 735-743, doi:Doi 10.1137/0905052 (1984).

95 Trygg, J. & Wold, S. Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics* **16**, 119-128, doi:10.1002/cem.695 (2002).

96 Wiklund, S. *et al.* Visualization of GC/TOF-MS-based metabolomics data for identification of biochemically interesting compounds using OPLS class models. *Anal Chem* **80**, 115-122, doi:10.1021/ac0713510 (2008).

97 Storey, J. D. & Tibshirani, R. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America* **100**, 9440-9445, doi:10.1073/pnas.1530509100 (2003).

98 Westerhuis, J. A. *et al.* Assessment of PLSDA cross validation. *Metabolomics* **4**, 81-89, doi:10.1007/s11306-007-0099-6 (2008).

99 Dunn, W. B. *et al.* Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature protocols* **6**, 1060-1083, doi:10.1038/nprot.2011.335 (2011).

100 Broadhurst, D. I. & Kell, D. B. Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics* **2**, 171-196, doi:10.1007/s11306-006-0037-z (2006).

101 Culeddu, N. *et al.* NMR-based metabolomic study of type 1 diabetes. *Metabolomics* **8**, 1162-1169, doi:10.1007/s11306-012-0420-x (2012).

102 Sonkar, K., Behari, A., Kapoor, V. K. & Sinha, N. 1H NMR metabolic profiling of human serum associated with benign and malignant gallstone diseases. *Metabolomics* **9**, 515-528, doi:10.1007/s11306-012-0468-7 (2012).

103 Blekherman, G. *et al.* Bioinformatics tools for cancer metabolomics. *Metabolomics* **7**, 329-343, doi:10.1007/s11306-010-0270-3 (2011).

104 Link, H., Fuhrer, T., Gerosa, L., Zamboni, N. & Sauer, U. Real-time metabolome profiling of the metabolic switch between starvation and growth. *Nat Methods* **12**, 1091-1097, doi:10.1038/nmeth.3584 (2015).

105 Mashego, M. R., van Gulik, W. M. & Heijnen, J. J. Metabolome dynamic responses of Saccharomyces cerevisiae to simultaneous rapid perturbations in external electron acceptor and electron donor. *FEMS Yeast Res* **7**, 48-66, doi:10.1111/j.1567-1364.2006.00144.x (2007).

106 Kresnowati, M. T. *et al.* When transcriptome meets metabolome: fast cellular responses of yeast to sudden relief of glucose limitation. *Mol Syst Biol* **2**, 49, doi:10.1038/msb4100083 (2006).

107 Zampar, G. G. *et al.* Temporal system-level organization of the switch from glycolytic to gluconeogenic operation in yeast. *Mol Syst Biol* **9**, 651, doi:10.1038/msb.2013.11 (2013).

108 Douma, R. D. *et al.* Degeneration of penicillin production in ethanol-limited chemostat cultivations of Penicillium chrysogenum: A systems biology approach. *BMC Syst Biol* **5**, 132, doi:10.1186/1752-0509-5-132 (2011).

109 Sevin, D. C., Stahlin, J. N., Pollak, G. R., Kuehne, A. & Sauer, U. Global Metabolic Responses to Salt Stress in Fifteen Species. *PLoS One* **11**, e0148888, doi:10.1371/journal.pone.0148888 (2016).

110 Goncalves, E. *et al.* Systematic Analysis of Transcriptional and Post-transcriptional Regulation of Metabolism in Yeast. *PLoS Comput Biol* **13**, e1005297, doi:10.1371/journal.pcbi.1005297 (2017).

111 Zampieri, M. *et al.* Metabolic constraints on the evolution of antibiotic resistance. *Mol Syst Biol* **13**, 917, doi:10.15252/msb.20167028 (2017).

112 Santos, C. N. & Stephanopoulos, G. Melanin-based high-throughput screen for L-tyrosine production in Escherichia coli. *Appl Environ Microbiol* **74**, 1190-1197, doi:10.1128/AEM.02448-07 (2008).

113 Tyo, K. E., Zhou, H. & Stephanopoulos, G. N. High-throughput screen for poly-3-hydroxybutyrate in Escherichia coli and Synechocystis sp. strain PCC6803. *Appl Environ Microbiol* **72**, 3412-3417, doi:10.1128/AEM.72.5.3412-3417.2006 (2006).

114 Lee, S. K. *et al.* Directed evolution of AraC for improved compatibility of arabinose- and lactose-inducible promoters. *Appl Environ Microbiol* **73**, 5711-5715, doi:10.1128/AEM.00791-07 (2007).

115 Atsumi, S. & Liao, J. C. Directed evolution of Methanococcus jannaschii citramalate synthase for biosynthesis of 1-propanol and 1-butanol by Escherichia coli. *Appl Environ Microbiol* **74**, 7802-7808, doi:10.1128/AEM.02046-08 (2008).

116 Leonard, E. *et al.* Combining metabolic and protein engineering of a terpenoid biosynthetic pathway for overproduction and selectivity control. *Proceedings of the National Academy of Sciences of the United States of America* **107**, 13654-13659, doi:10.1073/pnas.1006138107 (2010).

117 Hong, K. K., Vongsangnak, W., Vemuri, G. N. & Nielsen, J. Unravelling evolutionary strategies of yeast for improving galactose utilization through integrated systems level analysis. *Proceedings of the National Academy of Sciences of the United States of America* **108**, 12179-12184, doi:10.1073/pnas.1103219108 (2011).

118 Shen, C. R. *et al.* Driving forces enable high-titer anaerobic 1-butanol synthesis in Escherichia coli. *Appl Environ Microbiol* **77**, 2905-2915, doi:10.1128/AEM.03034-10 (2011).

119 Sun, Z. *et al.* Metabolic engineering of the L-phenylalanine pathway in Escherichia coli for the production of S- or R-mandelic acid. *Microb Cell Fact* **10**, 71, doi:10.1186/1475-2859-10-71 (2011).

120 Bailey, J. E. *et al.* Inverse metabolic engineering: A strategy for directed genetic engineering of useful phenotypes. *Biotechnol Bioeng* **52**, 109-121, doi:10.1002/(SICI)1097-0290(19961005)52:1<109::AID-BIT11>3.0.CO;2-J (1996).

121 Yoshida, S. *et al.* Development of bottom-fermenting saccharomyces strains that produce high SO2 levels, using integrated metabolome and transcriptome analysis. *Appl Environ Microbiol* **74**, 2787-2796, doi:10.1128/AEM.01781-07 (2008).

122 Wisselink, H. W. *et al.* Metabolome, transcriptome and metabolic flux analysis of arabinose fermentation by engineered Saccharomyces

cerevisiae. *Metab Eng* **12**, 537-551, doi:10.1016/j.ymben.2010.08.003 (2010).

123 Ding, M. Z., Zhou, X. & Yuan, Y. J. Metabolome profiling reveals adaptive evolution of Saccharomyces cerevisiae during repeated vacuum fermentations. *Metabolomics* **6**, 42-55, doi:10.1007/s11306-009-0173-3 (2010).

124 Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 15545-15550, doi:10.1073/pnas.0506580102 (2005).

125 Xia, J. & Wishart, D. S. MSEA: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Res* **38**, W71-77, doi:10.1093/nar/gkq329 (2010).

126 Kizer, L., Pitera, D. J., Pfleger, B. F. & Keasling, J. D. Application of functional genomics to pathway optimization for increased isoprenoid production. *Appl Environ Microbiol* **74**, 3229-3241, doi:10.1128/AEM.02750-07 (2008).

127 Joshi, A. & Palsson, B. O. Metabolic dynamics in the human red cell. Part I--A comprehensive kinetic model. *J Theor Biol* **141**, 515-528, doi:10.1016/S0022-5193(89)80233-4 (1989).

128 Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. & Reuss, M. Dynamic modeling of the central carbon metabolism of Escherichia coli. *Biotechnol Bioeng* **79**, 53-73, doi:10.1002/bit.10288 (2002).

129 van Eunen, K. *et al.* Measuring enzyme activities under standardized in vivo-like conditions for systems biology. *FEBS J* **277**, 749-760, doi:10.1111/j.1742-4658.2009.07524.x (2010).

130 van Eunen, K., Kiewiet, J. A., Westerhoff, H. V. & Bakker, B. M. Testing biochemistry revisited: how in vivo metabolism can be understood from in vitro enzyme kinetics. *PLoS Comput Biol* **8**, e1002483, doi:10.1371/journal.pcbi.1002483 (2012).

131 Gutenkunst, R. N. *et al.* Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol* **3**, 1871-1878, doi:10.1371/journal.pcbi.0030189 (2007).

132 Savinell, J. M. & Palsson, B. O. Network analysis of intermediary metabolism using linear optimization. I. Development of mathematical formalism. *J Theor Biol* **154**, 421-454 (1992).

133 Varma, A. & Palsson, B. O. Metabolic capabilities of Escherichia coli: I. synthesis of biosynthetic precursors and cofactors. *J Theor Biol* **165**, 477-502, doi:10.1006/jtbi.1993.1202 (1993).

134 Meléndez-Hevia, E., Waddell, T. G., Heinrich, R. & Montero, F. Theoretical Approaches to the Evolutionary Optimization of Glycolysis. *European Journal of Biochemistry* **244**, 527-543 (1997).

135 Choi, H. S., Kim, T. Y., Lee, D. Y. & Lee, S. Y. Incorporating metabolic flux ratios into constraint-based flux analysis by using artificial metabolites and converging ratio determinants. *J Biotechnol* **129**, 696-705, doi:10.1016/j.jbiotec.2007.02.026 (2007).

136 Schuetz, R., Kuepfer, L. & Sauer, U. Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli. *Mol Syst Biol* **3**, 119, doi:10.1038/msb4100162 (2007).

137 Nookaew, I. *et al.* Identification of flux regulation coefficients from elementary flux modes: A systems biology tool for analysis of metabolic networks. *Biotechnol Bioeng* **97**, 1535-1549, doi:10.1002/bit.21339 (2007).

138 Cakir, T., Kirdar, B., Onsan, Z. I., Ulgen, K. O. & Nielsen, J. Effect of carbon source perturbations on transcriptional regulation of metabolic fluxes in Saccharomyces cerevisiae. *BMC Syst Biol* **1**, 18, doi:10.1186/1752-0509-1-18 (2007).

139 Schellenberger, J. *et al.* Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. *Nature protocols* **6**, 1290-1307, doi:10.1038/nprot.2011.308 (2011).

140 Klamt, S., Saez-Rodriguez, J. & Gilles, E. D. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst Biol* **1**, 2, doi:10.1186/1752-0509-1-2 (2007).

141 Wright, J. & Wagner, A. The Systems Biology Research Tool: evolvable open-source software. *BMC Syst Biol* **2**, 55, doi:10.1186/1752-0509-2-55 (2008).

142 Rocha, I. *et al.* OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Syst Biol* **4**, 45, doi:10.1186/1752-0509-4-45 (2010).

143 Tervo, C. J. & Reed, J. L. MapMaker and PathTracer for tracking carbon in genome-scale metabolic models. *Biotechnol J* **11**, 648-661, doi:10.1002/biot.201500267 (2016).

144 Megchelenbrink, W., Huynen, M. & Marchiori, E. optGpSampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PLoS One* **9**, e86587, doi:10.1371/journal.pone.0086587 (2014).

145 Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst Biol* **7**, 74, doi:10.1186/1752-0509-7-74 (2013).

146 Feist, A. M., Herrgard, M. J., Thiele, I., Reed, J. L. & Palsson, B. O. Reconstruction of biochemical networks in microorganisms. *Nat Rev Microbiol* **7**, 129-143, doi:10.1038/nrmicro1949 (2009).

147 Thiele, I. & Palsson, B. O. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols* **5**, 93-121, doi:10.1038/nprot.2009.203 (2010).

148 Satish Kumar, V., Dasika, M. S. & Maranas, C. D. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics* **8**, 212, doi:10.1186/1471-2105-8-212 (2007).

149 Senger, R. S. & Papoutsakis, E. T. Genome-scale model for Clostridium acetobutylicum: Part I. Metabolic network resolution and analysis. *Biotechnol Bioeng* **101**, 1036-1052, doi:10.1002/bit.22010 (2008).

150 Barua, D., Kim, J. & Reed, J. L. An automated phenotype-driven approach (GeneForce) for refining metabolic and regulatory models. *PLoS Comput Biol* **6**, e1000970, doi:10.1371/journal.pcbi.1000970 (2010).

151 Henry, C. S. *et al.* High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat Biotechnol* **28**, 977-982, doi:10.1038/nbt.1672 (2010).

152 Kumar, V. S. & Maranas, C. D. GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. *PLoS Comput Biol* **5**, e1000308, doi:10.1371/journal.pcbi.1000308 (2009).

153 Thiele, I., Vlassis, N. & Fleming, R. M. fastGapFill: efficient gap filling in metabolic networks. *Bioinformatics* **30**, 2529-2531, doi:10.1093/bioinformatics/btu321 (2014).

154 Andersen, M. R., Nielsen, M. L. & Nielsen, J. Metabolic model integration of the bibliome, genome, metabolome and reactome of Aspergillus niger. *Mol Syst Biol* **4**, 178, doi:10.1038/msb.2008.12 (2008).

155 Milne, C. B. *et al.* Metabolic network reconstruction and genome-scale model of butanol-producing strain Clostridium beijerinckii NCIMB 8052. *BMC Syst Biol* **5**, 130, doi:10.1186/1752-0509-5-130 (2011).

156 Feist, A. M. *et al.* A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol* **3**, 121, doi:10.1038/msb4100155 (2007).

157 Orth, J. D. *et al.* A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. *Mol Syst Biol* **7**, 535, doi:10.1038/msb.2011.65 (2011).

158 Nookaew, I. *et al.* The genome-scale metabolic model iIN800 of Saccharomyces cerevisiae and its validation: a scaffold to query lipid metabolism. *BMC Syst Biol* **2**, 71, doi:10.1186/1752-0509-2-71 (2008).

159 Suthers, P. F. *et al.* Metabolic flux elucidation for large-scale models using 13C labeled isotopes. *Metab Eng* **9**, 387-405, doi:10.1016/j.ymben.2007.05.005 (2007).

160 Karr, J. R. *et al.* A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389-401, doi:10.1016/j.cell.2012.05.044 (2012).

161 Kumar, A., Suthers, P. F. & Maranas, C. D. MetRxn: a knowledgebase of metabolites and reactions spanning metabolic models and databases. *BMC Bioinformatics* **13**, 6, doi:10.1186/1471-2105-13-6 (2012).

162 Lee, J. M., Gianchandani, E. P. & Papin, J. A. Flux balance analysis in the era of metabolomics. *Brief Bioinform* **7**, 140-150, doi:10.1093/bib/bbl007 (2006).

163 Gianchandani, E. P., Chavali, A. K. & Papin, J. A. The application of flux balance analysis in systems biology. *Wiley Interdiscip Rev Syst Biol Med* **2**, 372-382, doi:10.1002/wsbm.60 (2010).

164 Covert, M. W., Schilling, C. H. & Palsson, B. Regulation of gene expression in flux balance models of metabolism. *J Theor Biol* **213**, 73-88, doi:10.1006/jtbi.2001.2405 (2001).

165 Covert, M. W. & Palsson, B. O. Transcriptional regulation in constraints-based metabolic models of Escherichia coli. *The Journal of biological chemistry* **277**, 28058-28064, doi:10.1074/jbc.M201691200 (2002).

166 Meadows, A. L., Karnik, R., Lam, H., Forestell, S. & Snedecor, B. Application of dynamic flux balance analysis to an industrial Escherichia

coli fermentation. *Metab Eng* **12**, 150-160, doi:10.1016/j.ymben.2009.07.006 (2010).

167 Mahadevan, R., Edwards, J. S. & Doyle, F. J., 3rd. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophys J* **83**, 1331-1340, doi:10.1016/S0006-3495(02)73903-9 (2002).

168 Feng, X., Xu, Y., Chen, Y. & Tang, Y. J. Integrating flux balance analysis into kinetic models to decipher the dynamic metabolism of Shewanella oneidensis MR-1. *PLoS Comput Biol* **8**, e1002376, doi:10.1371/journal.pcbi.1002376 (2012).

169 Schmidt, B. J. *et al.* GIM3E: condition-specific models of cellular metabolism developed from metabolomics and expression data. *Bioinformatics* **29**, 2900-2908, doi:10.1093/bioinformatics/btt493 (2013).

170 Covert, M. W., Xiao, N., Chen, T. J. & Karr, J. R. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics* **24**, 2044-2050, doi:10.1093/bioinformatics/btn352 (2008).

171 Lee, J. M., Gianchandani, E. P., Eddy, J. A. & Papin, J. A. Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLoS Comput Biol* **4**, e1000086, doi:10.1371/journal.pcbi.1000086 (2008).

172 Hoffner, K., Harwood, S. M. & Barton, P. I. A reliable simulator for dynamic flux balance analysis. *Biotechnol Bioeng* **110**, 792-802, doi:10.1002/bit.24748 (2013).

173 Gomez, J. A., Hoffner, K. & Barton, P. I. DFBAlab: a fast and reliable MATLAB code for dynamic flux balance analysis. *BMC Bioinformatics* **15**, 409, doi:10.1186/s12859-014-0409-8 (2014).

174 Chen, J. *et al.* Spatiotemporal modeling of microbial metabolism. *BMC Syst Biol* **10**, 21, doi:10.1186/s12918-016-0259-2 (2016).

175 Flassig, R. J., Fachet, M., Hoffner, K., Barton, P. I. & Sundmacher, K. Dynamic flux balance modeling to increase the production of high-value compounds in green microalgae. *Biotechnol Biofuels* **9**, 165, doi:10.1186/s13068-016-0556-4 (2016).

176 Herrgard, M. J., Fong, S. S. & Palsson, B. O. Identification of genome-scale metabolic network models using experimentally measured flux profiles. *PLoS Comput Biol* **2**, e72, doi:10.1371/journal.pcbi.0020072 (2006).

177 Zomorrodi, A. R. & Maranas, C. D. Improving the iMM904 S. cerevisiae metabolic model using essentiality and synthetic lethality data. *BMC Syst Biol* **4**, 178, doi:10.1186/1752-0509-4-178 (2010).

178 Oh, Y. G., Lee, D. Y., Lee, S. Y. & Park, S. Multiobjective flux balancing using the NISE method for metabolic network analysis. *Biotechnol Prog* **25**, 999-1008, doi:10.1002/btpr.193 (2009).

179 Adadi, R., Volkmer, B., Milo, R., Heinemann, M. & Shlomi, T. Prediction of microbial growth rate versus biomass yield by a metabolic network with kinetic parameters. *PLoS Comput Biol* **8**, e1002575, doi:10.1371/journal.pcbi.1002575 (2012).

180 Segre, D., Vitkup, D. & Church, G. M. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 15112-15117, doi:10.1073/pnas.232349399 (2002).

181 Burgard, A. P., Pharkya, P. & Maranas, C. D. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng* **84**, 647-657, doi:10.1002/bit.10803 (2003).

182 Tervo, C. J. & Reed, J. L. Expanding Metabolic Engineering Algorithms Using Feasible Space and Shadow Price Constraint Modules. *Metab Eng Commun* **1**, 1-11, doi:10.1016/j.meteno.2014.06.001 (2014).

183 Patil, K. R., Rocha, I., Forster, J. & Nielsen, J. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics* **6**, 308, doi:10.1186/1471-2105-6-308 (2005).

184 Asadollahi, M. A. *et al.* Enhancing sesquiterpene production in Saccharomyces cerevisiae through in silico driven metabolic engineering. *Metab Eng* **11**, 328-334, doi:10.1016/j.ymben.2009.07.001 (2009).

185 Ranganathan, S., Suthers, P. F. & Maranas, C. D. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput Biol* **6**, e1000744, doi:10.1371/journal.pcbi.1000744 (2010).

186 Cotten, C. & Reed, J. L. Constraint-based strain design using continuous modifications (CosMos) of flux bounds finds new strategies for metabolic engineering. *Biotechnol J* **8**, 595-604, doi:10.1002/biot.201200316 (2013).

187 Chowdhury, A., Zomorrodi, A. R. & Maranas, C. D. k-OptForce: integrating kinetics with flux balance analysis for strain design. *PLoS Comput Biol* **10**, e1003487, doi:10.1371/journal.pcbi.1003487 (2014).

188 Zomorrodi, A. R., Islam, M. M. & Maranas, C. D. d-OptCom: Dynamic multi-level and multi-objective metabolic modeling of microbial communities. *ACS Synth Biol* **3**, 247-257, doi:10.1021/sb4001307 (2014).

189 Kummel, A., Panke, S. & Heinemann, M. Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data. *Mol Syst Biol* **2**, 2006 0034, doi:10.1038/msb4100074 (2006).

190 Zamboni, N., Kummel, A. & Heinemann, M. anNET: a tool for network-embedded thermodynamic analysis of quantitative metabolome data. *BMC Bioinformatics* **9**, 199, doi:10.1186/1471-2105-9-199 (2008).

191 Karp, P. D. *et al.* Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Res* **33**, 6083-6089, doi:10.1093/nar/gki892 (2005).

192 Schomburg, I. BRENDA, enzyme data and metabolic information. *Nucleic Acids Research* **30**, 47-49, doi:10.1093/nar/30.1.47 (2002).

193 de Matos, P. *et al.* Chemical Entities of Biological Interest: an update. *Nucleic Acids Res* **38**, D249-254, doi:10.1093/nar/gkp886 (2010).

194 Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* **28**, 27-30, doi:10.1093/nar/28.1.27 (2000).

195 Caspi, R. *et al.* The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* **36**, D623-631, doi:10.1093/nar/gkm900 (2008).

196 Bolton, E. E., Wang, Y., Thiessen, P. A. & Bryant, S. H. in *Annu. Rep. Comput. Chem.* Vol. 4 (eds A. Wheeler Ralph & C. Spellmeyer David) 217-241 (Elsevier, 2008).

197 Canelas, A. B., van Gulik, W. M. & Heijnen, J. J. Determination of the cytosolic free NAD/NADH ratio in Saccharomyces cerevisiae under steady-state and highly dynamic conditions. *Biotechnol Bioeng* **100**, 734-743, doi:10.1002/bit.21813 (2008).

198 Jol, S. J., Kummel, A., Terzer, M., Stelling, J. & Heinemann, M. System-level insights into yeast metabolism by thermodynamic analysis of elementary flux modes. *PLoS Comput Biol* **8**, e1002415, doi:10.1371/journal.pcbi.1002415 (2012).

199 Henry, C. S., Broadbelt, L. J. & Hatzimanikatis, V. Thermodynamics-based metabolic flux analysis. *Biophys J* **92**, 1792-1805, doi:10.1529/biophysj.106.093138 (2007).

200 Hamilton, J. J., Dwivedi, V. & Reed, J. L. Quantitative assessment of thermodynamic constraints on the solution space of genome-scale metabolic models. *Biophys J* **105**, 512-522, doi:10.1016/j.bpj.2013.06.011 (2013).

201 Jankowski, M. D., Henry, C. S., Broadbelt, L. J. & Hatzimanikatis, V. Group contribution method for thermodynamic analysis of complex metabolic networks. *Biophys J* **95**, 1487-1499, doi:10.1529/biophysj.107.124784 (2008).

202 Finley, S. D., Broadbelt, L. J. & Hatzimanikatis, V. Thermodynamic analysis of biodegradation pathways. *Biotechnol Bioeng* **103**, 532-541, doi:10.1002/bit.22285 (2009).

203 Bordel, S. & Nielsen, J. Identification of flux control in metabolic networks using non-equilibrium thermodynamics. *Metab Eng* **12**, 369-377, doi:10.1016/j.ymben.2010.03.001 (2010).

204 Hoppe, A., Hoffmann, S. & Holzhutter, H. G. Including metabolite concentrations into flux balance analysis: thermodynamic realizability as a constraint on flux distributions in metabolic networks. *BMC Syst Biol* **1**, 23, doi:10.1186/1752-0509-1-23 (2007).

205 Cotten, C. & Reed, J. L. Mechanistic analysis of multi-omics datasets to generate kinetic parameters for constraint-based metabolic models. *BMC Bioinformatics* **14**, 32, doi:10.1186/1471-2105-14-32 (2013).

206 Leighty, R. W. & Antoniewicz, M. R. Dynamic metabolic flux analysis (DMFA): a framework for determining fluxes at metabolic non-steady state. *Metab Eng* **13**, 745-755, doi:10.1016/j.ymben.2011.09.010 (2011).

207 Kleessen, S., Irgang, S., Klie, S., Giavalisco, P. & Nikoloski, Z. Integration of transcriptomics and metabolomics data specifies the metabolic response of Chlamydomonas to rapamycin treatment. *Plant J* **81**, 822-835, doi:10.1111/tpj.12763 (2015).

208 Bordbar, A. *et al.* Elucidating dynamic metabolic physiology through network integration of quantitative time-course metabolomics. *Sci Rep* **7**, 46249, doi:10.1038/srep46249 (2017).

209 Heijnen, J. J. Approximative kinetic formats used in metabolic network modeling. *Biotechnol Bioeng* **91**, 534-545, doi:10.1002/bit.20558 (2005).

210 Savageau, M. A. Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. *J Theor Biol* **25**, 365-369 (1969).

211 Savageau, M. A. Mathematics of organizationally complex systems. *Biomedica biochimica acta* **44**, 839-844 (1985).

212 Voit, E. O. & Savageau, M. A. Accuracy of alternative representations for integrated biochemical systems. *Biochemistry* **26**, 6869-6880 (1987).

213 Voit, E. O. Modelling metabolic networks using power-laws and S-systems. *Essays in biochemistry* **45**, 29-40, doi:10.1042/BSE0450029 (2008).

214 Steuer, R., Gross, T., Selbig, J. & Blasius, B. Structural kinetic modeling of metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 11868-11873, doi:10.1073/pnas.0600013103 (2006).

215 Nikerel, I. E., van Winden, W. A., van Gulik, W. M. & Heijnen, J. J. A method for estimation of elasticities in metabolic networks using steady state and dynamic metabolomics data and linlog kinetics. *BMC Bioinformatics* **7**, 540, doi:10.1186/1471-2105-7-540 (2006).

216 Nikerel, I. E., van Winden, W. A., Verheijen, P. J. & Heijnen, J. J. Model reduction and a priori kinetic parameter identifiability analysis using metabolome time series for metabolic reaction networks with linlog kinetics. *Metab Eng* **11**, 20-30, doi:10.1016/j.ymben.2008.07.004 (2009).

217 Costa, R. S., Machado, D., Rocha, I. & Ferreira, E. C. Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis-Menten and approximate kinetic equations. *Biosystems* **100**, 150-157, doi:10.1016/j.biosystems.2010.03.001 (2010).

218 Ralser, M. *et al.* Dynamic rerouting of the carbohydrate flux is key to counteracting oxidative stress. *J. Biol.* **6**, 10 (2007).

219 Villaverde, A. F., Barreiro, A. & Papachristodoulou, A. Structural Identifiability of Dynamic Systems Biology Models. *PLoS Comput Biol* **12**, e1005153, doi:10.1371/journal.pcbi.1005153 (2016).

220 Tran, L. M., Rizk, M. L. & Liao, J. C. Ensemble modeling of metabolic networks. *Biophys J* **95**, 5606-5617, doi:10.1529/biophysj.108.135442 (2008).

221 Rizk, M. L. & Liao, J. C. Ensemble modeling for aromatic production in Escherichia coli. *PLoS One* **4**, e6903, doi:10.1371/journal.pone.0006903 (2009).

222 Contador, C. A., Rizk, M. L., Asenjo, J. A. & Liao, J. C. Ensemble modeling for strain development of L-lysine-producing Escherichia coli. *Metab Eng* **11**, 221-233, doi:10.1016/j.ymben.2009.04.002 (2009).

223 Khodayari, A. & Maranas, C. D. A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains. *Nat Commun* **7**, 13806, doi:10.1038/ncomms13806 (2016).

224 Khodayari, A., Zomorrodi, A. R., Liao, J. C. & Maranas, C. D. A kinetic model of Escherichia coli core metabolism satisfying multiple sets of mutant flux data. *Metab Eng* **25**, 50-62, doi:10.1016/j.ymben.2014.05.014 (2014).

225 Yizhak, K., Benyamini, T., Liebermeister, W., Ruppin, E. & Shlomi, T. Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model. *Bioinformatics* **26**, i255-260, doi:10.1093/bioinformatics/btq183 (2010).

226 Jamshidi, N. & Palsson, B. O. Mass action stoichiometric simulation models: incorporating kinetics and regulation into stoichiometric models. *Biophys J* **98**, 175-185, doi:10.1016/j.bpj.2009.09.064 (2010).

227 Mannan, A. A. *et al.* Integrating Kinetic Model of E. coli with Genome Scale Metabolic Fluxes Overcomes Its Open System Problem and Reveals Bistability in Central Metabolism. *PLoS One* **10**, e0139507, doi:10.1371/journal.pone.0139507 (2015).

228 Goel, G., Chou, I. C. & Voit, E. O. System estimation from metabolic time-series data. *Bioinformatics* **24**, 2505-2511, doi:10.1093/bioinformatics/btn470 (2008).

229 Voit, E. O. & Almeida, J. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics* **20**, 1670-1681, doi:10.1093/bioinformatics/bth140 (2004).

230 Chou, I. C., Martens, H. & Voit, E. O. Parameter estimation in biochemical systems models with alternating regression. *Theor Biol Med Model* **3**, 25, doi:10.1186/1742-4682-3-25 (2006).

231 Ishii, N., Nakayama, Y. & Tomita, M. Distinguishing enzymes using metabolome data for the hybrid dynamic/static method. *Theor Biol Med Model* **4**, 19, doi:10.1186/1742-4682-4-19 (2007).

232 Zhan, C. & Yeung, L. F. Parameter estimation in systems biology models using spline approximation. *BMC Syst Biol* **5**, 14, doi:10.1186/1752-0509-5-14 (2011).

233 Dromms, R. A. & Styczynski, M. P. Improved metabolite profile smoothing for flux estimation. *Mol Biosyst* **11**, 2394-2405, doi:10.1039/c5mb00165j (2015).

234 Machina, A., Ponosov, A. & Voit, E. O. Automated piecewise power-law modeling of biological systems. *J Biotechnol* **149**, 154-165, doi:10.1016/j.jbiotec.2009.12.016 (2010).

235 Niklas, J., Schrader, E., Sandig, V., Noll, T. & Heinzle, E. Quantitative characterization of metabolism and metabolic shifts during growth of the new human cell line AGE1.HN using time resolved metabolic flux analysis. *Bioprocess Biosyst Eng* **34**, 533-545, doi:10.1007/s00449-010-0502-y (2011).

236 Willemsen, A. M. *et al.* MetDFBA: incorporating time-resolved metabolomics measurements into dynamic flux balance analysis. *Mol Biosyst* **11**, 137-145, doi:10.1039/c4mb00510d (2015).

237 Yugi, K., Nakayama, Y., Kinoshita, A. & Tomita, M. Hybrid dynamic/static method for large-scale simulation of metabolism. *Theor Biol Med Model* **2**, 42, doi:10.1186/1742-4682-2-42 (2005).

238 von Stosch, M., Rodrigues de Azevedo, C., Luis, M., Feyo de Azevedo, S. & Oliveira, R. A principal components method constrained by elementary flux modes: analysis of flux data sets. *BMC Bioinformatics* **17**, 200, doi:10.1186/s12859-016-1063-0 (2016).

239 Zurauskiene, J., Kirk, P., Thorne, T., Pinney, J. & Stumpf, M. Derivative processes for modelling metabolic fluxes. *Bioinformatics* **30**, 1892-1898, doi:10.1093/bioinformatics/btu069 (2014).

240 Vilela, M. *et al.* Parameter optimization in S-system models. *BMC Syst Biol* **2**, 35, doi:10.1186/1752-0509-2-35 (2008).

241 Voit, E. O., Goel, G., Chou, I. C. & Fonseca, L. L. Estimation of metabolic pathway systems from different data sources. *IET Syst Biol* **3**, 513-522 (2009). <http://digital-library.theiet.org/content/journals/10.1049/iet-syb.2008.0180>.

242 Chou, I. C. & Voit, E. O. Estimation of dynamic flux profiles from metabolic time series data. *BMC Syst Biol* **6**, 84, doi:10.1186/1752-0509-6-84 (2012).

243 Jia, G., Stephanopoulos, G. N. & Gunawan, R. Parameter estimation of kinetic models from metabolic profiles: two-phase dynamic decoupling method. *Bioinformatics* **27**, 1964-1970, doi:10.1093/bioinformatics/btr293 (2011).

244 Jia, G., Stephanopoulos, G. & Gunawan, R. Incremental parameter estimation of kinetic metabolic network models. *BMC Syst Biol* **6**, 142, doi:10.1186/1752-0509-6-142 (2012).

245 Sanghvi, J. C. *et al.* Accelerated discovery via a whole-cell model. *Nat Methods* **10**, 1192-1195, doi:10.1038/nmeth.2724 (2013).

246 Birch, E. W., Udell, M. & Covert, M. W. Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *J Theor Biol* **345**, 12-21, doi:10.1016/j.jtbi.2013.12.009 (2014).

247 Karr, J. R., Phillips, N. C. & Covert, M. W. WholeCellSimDB: a hybrid relational/HDF database for whole-cell model predictions. *Database (Oxford)* **2014**, doi:10.1093/database/bau095 (2014).

Chapter 2: Improved Metabolite Profile Smoothing for Flux Estimation

Portions of this chapter are reproduced under license from my publication "Improved metabolite profile smoothing for flux estimation"[1] in *Molecular Biosystems*. http://pubs.rsc.org/en/Content/ArticleLanding/2015/MB/C5MB00165J

2.1 Background

Genome-scale metabolic modeling is an area of research with the potential for significant impact on many biomedical and biotechnological applications. As discussed in Chapter 1, such models have been used to identify drug targets that specifically inhibit cancer proliferation[2], to identify genomic manipulations that can facilitate production of valuable chemicals[3], and to uncover and characterize metabolic pathways even in well-understood models[4]. This modeling approach entails using metabolic reconstructions that include all of the cataloged metabolic reactions in an organism (i.e., genome-scale reconstructions) in a defined mathematical modeling framework.

Effectively modeling biological systems at the genome scale calls for measurements and data also at the genome scale. As discussed in Chapter 1, to date very few genome-scale metabolic models have attempted to integrate metabolite profiling information, in contrast to the prominent use of transcriptomic, fluxomic, and proteomic data in such models[5-9]. In the few cases

where metabolomics data have been integrated into these models, the application of the data has typically been in setting thermodynamic constraints and estimating free energies rather than in more direct applications[10,11].

The primary reason for this omission is that most metabolic models using genome-scale metabolic reconstructions assume the cell or organism to be at a steady state, typically to simplify the model framework and associated computational complexity. While models exploiting such an assumption have shown great utility, their validity and potential for extrapolation have an intrinsic limit: while the steady state assumption may be true over short time periods, it ultimately is violated once varying forms of metabolic regulation begin to exert their influence.

The use of detailed ordinary differential equation (ODE) models would allow for the capture of dynamic behaviors and regulation, but application of ODE models on a genome-wide scale is not currently feasible due to (among other issues) the many unknown reaction rate and thermodynamic parameters[12-14], each of which would require extensive effort to be ascertained experimentally. As such, significant recent effort has focused on softening the steady state assumption in genome-scale metabolic modeling without requiring a full ODE model of the entire metabolic system[6,7,15]. These efforts hold great promise for future

biotechnological applications, and they are the background motivation for efforts described in this chapter.

Use of metabolomics data is a promising approach for bridging the gap between the steady state assumption and the dynamic intracellular reality. This data can be used to estimate the accumulation or depletion "fluxes" of certain metabolites in a system, which can then be used in place of the steady state assumption so common in genome-scale metabolic modeling. This approach has been described and implemented in multiple prior works[16-20]. The most common approach to estimating these accumulation fluxes from metabolite data is to first smooth the data or fit it to a specific mathematical function, and then use the resulting data or function to determine the flux of that metabolite at any given time (potentially between measured time points). The accuracy of these estimates has an obvious impact on the accuracy of the overall model, but effective estimation of these fluxes is a non-trivial problem given the noise inherent to measurement of metabolite levels and the limitations of the current methods for flux estimation[16].

One of the more thorough treatments of the problem of flux estimation from metabolite data for metabolic modeling was included in work by Ishii *et al.*[19] While the main focus of that work was on developing a broader metabolic model, data smoothing and flux estimation were integral parts of the data processing for

the algorithm. They fit a variety of polynomial and rational functions to simulated metabolite data and, on a metabolite-wise basis, selected as the representative function the one that minimizes the fitting error (accounting for the number of free parameters to minimize over-fitting). Of note is that none of the candidate fitting functions are derived from or selected based on biological insight. Additionally, as we show later, the fitting of an arbitrary time course can yield unphysical results. Splines, another common alternative, are sensitive to noise and outliers—this is particularly problematic when the derivative of the concentration (the accumulation flux) is the important quantity being estimated.

In this chapter, we present two approaches for improving the estimation of accumulation fluxes from metabolite time series data. First, we investigate the use of a biologically reasonable and biologically-inspired sigmoidal impulse function[21,22] as an effective and perhaps generalizable alternative to the fitting functions previously used. This functional form emulates behavior observed in known biological systems, and our work represents the first time that it has been applied in the context of metabolic modeling. Second, we investigate whether a resampling-based approach to smoothing and fitting data might yield more accurate concentration profile fits and derivative (flux) predictions than the previously used approach. In the course of these investigations, we also identified the importance of enforcing constraints on fitting equation parameter values to prevent the selection of unphysical solutions. Each of these approaches

improves the accuracy of flux estimation from metabolite time series data, providing more reliable results to be integrated into the larger metabolic modeling framework with reasonable computational expense.

2.2 Methods

2.2.1 Fitting functions

Eight functions, shown in Table 2.1, were considered as candidates to best fit the time series metabolite data. The first seven were used by Ishii *et al*.[19]. Four of these were polynomials, of order two to five. The other three were rational functions, composed of a first, second, or third order polynomial numerator and a first or second order polynomial denominator. The eighth function was the sigmoidal impulse, which was first presented in the context of filtering and clustering gene expression profiles[21,22]; it is here applied for the first time in the context of metabolic models. Unlike the other functions, it has a biologically relevant interpretation: a two-phase transition from one steady state to a (potentially new) steady state through an intermediate state. Its parameters directly correspond to features of this trajectory, representing: transition time delays; the initial, intermediate state, and steady-state metabolite levels; and the rapidity of the transitions.

Table 2.1. Fitting functions evaluated in Chapter 2

Name	Formula
P_2	$C(t) = p_1 \cdot t^2 + p_2 \cdot t + p_3$
P_2	$C(t) = p_1 \cdot t^3 + p_2 \cdot t^2 + p_3 \cdot t + p_4$
P_4	$C(t) = p_1 \cdot t^4 + p_2 \cdot t^2 + p_3 \cdot t^2 + p_4 \cdot t + p_5$
P_5	$C(t) = p_1 \cdot t^5 + p_2 \cdot t^4 + p_3 \cdot t^3 + p_4 \cdot t^2 + p_5 \cdot t + p_6$
R_{11}	$C(t) = \frac{p_1 \cdot t + p_2}{t + p_3}$
R_{22}	$C(t) = \frac{p_1 \cdot t^2 + p_2 \cdot t + p_3}{t^2 + p_4 \cdot t + p_5}$
R_{31}	$C(t) = \frac{p_1 \cdot t^3 + p_2 \cdot t^2 + p_3 \cdot t + p_4}{t + p_5}$
I	$C(t) = \frac{1}{h_1} \cdot s(t, \tau_1, h_0, \beta_1) \cdot s(t, \tau_2, h_2, \beta_2)$ $s(t, \tau, h, \beta) = h + \frac{(h_1 - h)}{1 + e^{-4\beta(t-\tau)}}$

2.2.2 Synthetic Reference Data

We tested our new methods using two different ODE models of central carbon metabolism taken from the literature, which were used to generate noise-free "gold standard" synthetic reference data for our analyses. These models were

selected because their dynamics are believed to reasonably represent *in vivo* metabolic dynamics; the fact that they are not genome-scale does not detract from their relevance as a model system, as the data smoothing/fitting step of flux estimation is independent of the scale of the model.

The first model simulates central carbon metabolism in *E. coli*[12]. While the model includes 18 metabolites, only the 17 metabolites with substantial dynamics were included in our analysis. (As implemented, metabolite 1 was a fixed value.) The second model simulates central carbon metabolism in *S. cerevisiae*[23], comprising 22 metabolites (21 of which had substantial dynamics, and were included in our analysis—changes in metabolite 17 were several orders of magnitude smaller than the concentration). While the yeast model was initially presented in the context of stable concentration oscillations, the initial conditions we used for our simulations do not produce oscillatory behaviors. To validate our implementation of the model, we used it to reproduce Figure 6 from Hynne et al.[23].

We obtained curated SBML code for both models from the BioModels Database, and solved systems of ODEs using the LSODA method in the Time Course module of Copasi 4.14, Build 89, with the default tolerances and parameters[24,25]. For each model, we solved the system of ODEs using the initial conditions specified in Table 2.2, derived from those previously reported[19], to simulate a perturbation in glucose concentration. As previously described[19] we used a

perturbation from 0.0556 mM to 1.67 mM for “Extracellular Glucose” in the *E. coli* model and a perturbation from 2.5 mM to 5.0 mM for “Mixed flow glucose” in the *S. cerevisiae* model. For the *E. coli* model, we fixed the concentrations of ATP, ADP, AMP, NAD(H), and NADP(H) at their initial values, as was done previously. The resulting gold-standard data contained concentrations at intervals of 0.01 seconds for the *E. coli* model and 0.0025 minutes for the *S. cerevisiae* model.

To generate data for parameter estimation, simulated time points were sampled at 1 second intervals from 0 seconds to 20 seconds for the *E. coli* model, and at 0.25 minute intervals from 0 minutes to 15 minutes for the *S. cerevisiae* model. The selection of different sampling rates was to be consistent with the approach taken by Ishii *et al*. for the *E. coli* model, but to account for the different time scales of the dynamics in the two mathematical models as observed in the BioModels implementations while still keeping the number of samples used for each respective model the same as that used by Ishii *et al*. By keeping the number of samples the same as in previous work for each respective model, our fitting results would be most directly comparable. We used a first-order centered finite difference approximation on the ODE output to estimate the derivatives in the synthetic reference data for each metabolite, C_i.

Table 2.2. Model Initial Conditions
Initial conditions listed here were used to generate synthetic data as a gold standard from each model as described in the Methods section.

Chassagnole (*E. coli*)		
#	Metabolite	Conc (mM)
1	Extracellular Glucose	1.670E+00
2	Glucose-6-Phosphate	3.480E+00
3	Fructose-6-Phosphate	6.000E-01
4	Fructose-1,6-bisphosphate	2.720E-01
5	Dihydroxyacetonephosphate	1.670E-01
6	Glyceraldehyde-3-Phosphate	2.180E-01
7	1,3-diphosphosphoglycerate	8.000E-03
8	3-Phosphoglycerate	2.130E+00
9	2-Phosphoglycerate	3.990E-01
10	Phosphoenol pyruvate	2.670E+00
11	Pyruvate	2.670E+00
12	6-Phosphogluconate	8.080E-01
13	Ribulose-5-phosphate	1.110E-01
14	Xylulose-5-phosphate	1.380E-01
15	sedoheptulose-7-phosphate	2.760E-01
16	Ribose-5-phosphate	3.980E-01
17	Erythrose-4-phosphate	9.800E-02
18	Glucose-1-Phosphate	6.530E-01

Hynne (*S. cerevisiae*)		
#	Metabolite	Conc (mM)
1	Extracellular glucose	3.330E-02
2	Cytosolic glucose	3.700E-03
3	Glucose-6-Phosphate	5.708E-01
4	Fructose-6-Phosphate	7.190E-02
5	Fructose 1,6-bisphosphate	5.090E-02
6	Dihydroxyacetone phosphate	2.851E-01
7	Glyceraldehyde 3-phosphate	1.240E-02
8	1,3-Bisphosphoglycerate	0.000E+00
9	Phosphoenolpyruvate	6.300E-03
10	Pyruvate	6.540E-02
11	Acetaldehyde	1.268E-01
12	Extracellular acetaldehyde	1.100E-01
13	EtOH	3.754E+00
14	Extracellular ethanol	3.210E+00
15	Glycerol	3.642E-01
16	Extracellular glycerol	1.462E-01
17	Extracellular cyanide	5.564E+00
18	AMP	6.055E-01
19	ADP	1.757E+00
20	ATP	1.571E+00
21	NAD	7.787E-01
22	NADH	2.013E-01

2.2.3 Synthetic Noisy Data

We generated sets of noisy metabolite time courses from this synthetic reference data. For each metabolite C_i, we generated a noisy time course by adding noise at each sampled time point, t_k, to the true value at that timepoint, $C_i(t_k)$, by drawing 5 simulated measurements from a normal distribution, $N_{i,k} \sim (C_i(t_k), CoV \cdot C_i(t_k))$, and then taking the mean of those 5 measurements,

called $D_i(t_k)$. We refer to each individual noisy time course as $D_{i,m}$. This approach paralleled the common experimental approach of taking biological replicate measurements and then collapsing them into one value for analyses. Here, we set the Coefficient of Variation (CoV) to 0.15, a reasonable value for many mass spectrometry-based metabolite profiling approaches. The same noisy values were used for all functions, allowing for direct comparison of the performance of each function. In total, 500 noisy time courses were generated for each metabolite in each model for the Direct Fit Method (described below), while an additional 50 time courses were used as the base data for the Resampling Method (described below).

2.2.4 Direct Fit Method

We refer to a basic nonlinear least squares fitting of parameters as the "Direct Fit" method for the purposes of this work. In this approach, we directly fitted each function of interest to each noisy time course, $D_{i,m}$, to produce the smoothed time course estimate, $f_{i,j,m}$. Best-fit parameters for a given function were selected by minimizing the root-mean-square-displacement (RMSD) of the function to the data, defined as

$$RMSD_{i,j,m} = \sqrt{\sum_k \frac{\left(D_{i,m}(t_k) - f_{i,j,m}(t_k)\right)^2}{n - p_j}}$$

where i represents a specific metabolite, j represents a function being fitted, k represents an individual time point, m represents the use of a specific noisy data

set, n is the number of sampled time points in the time course $D_{i,m}$, and p_j is the number of parameters being fit for function f_j. The denominator reflects a penalty on the number of parameters for a function, to help guard against over-fitting when comparing different functions[26].

Polynomials were fit using the built-in `polyfit()` function in MATLAB. Rational functions and the impulse function were fitted using `fmincon()` in MATLAB to allow for bounds on the parameter space, as described in sections 2.3.2.1 and 2.3.2.2 To improve the likelihood of finding globally optimal parameter sets for the rational and impulse functions, we selected optimal parameters from 20 solver runs seeded with different sets of initial conditions as described in section 2.3.2.3.

2.2.5 Resampling Method

In an approach we refer to as the "Resampling Method", we took advantage of the stabilizing effect of calculating the median of fits to multiple noisy datasets to produce more robust estimates of metabolite concentrations and derivatives.

Starting with the noisy time courses that model experimental data (described above), we generated resampled time courses by repeating the procedure used to produce the original noisy time courses, but using a noisy time course $D_{i,m}$ as input rather than the true metabolite concentration C_i. We again used a fixed CoV

of 15% for this procedure; however, in practice, a dataset-specific and/or metabolite-specific CoV could be estimated and use in place of the fixed CoV. We generated 250 such resampled noisy time courses, $R_{i,m,w}$, for each initial noisy time course $D_{i,m}$.

We used the Direct Fit Method as described above to generate a nominal parameter solution from each base noisy time course $D_{i,m}$. Then, for each resampled time course $R_{i,m,w}$ derived from that noisy time course, we fit the function of interest (once) using the parameter solution from the Direct Fit Method as the initial guess. Parameter fitting was performed as described above.

We then used the resample-derived parameters to calculate concentration and derivative trajectories for each resampled time course $R_{i,m,w}$, and calculated the median value across all resampled time courses at the time points of interest (either the original or interpolated time points, as described below). The output of the Resampling Method was this list of concentration and derivative medians.

2.2.6 Performance Calculations

The performance of each fitting function using each method (direct and resampling) on both concentration and derivative predictions was quantified for each metabolite and for each base noisy time course, $D_{i,m}$. Concentration accuracy is useful for assessing the effectiveness of smoothing, while derivative

accuracy is more relevant for downstream applications in estimating flux distributions[18]. Accuracy for each noisy time course $D_{i,m}$ was calculated using an adjusted RMSD between the synthetic reference data, C_i, and the predicted value for a given function, parameter set, and noisy data set, $f_{i,j,m}$. Specifically, we calculate accuracy as

$$RMSD_{i,j,m} = \frac{\sqrt{\sum_k \left(C_i(t_k) - f_{i,j,m}(t_k)\right)^2}}{n_I \cdot S \cdot \mu}$$

where

$$S = \sqrt{\frac{\sum_k \left(f_{i,j,m}(t_k)\right)^2}{n}}$$

$$\mu = \frac{n - p_j}{n}$$

and n_I is the number of time points used in assessing predictive accuracy. S is a scaling factor facilitating comparison and visualization by controlling for differences in the magnitude of different metabolites, and μ is a penalty factor scaling with the number of parameters in a function and the number of data points used to fit the function. For calculating derivative accuracy, the derivative values $f'_{i,j,m}(t_k)$ and $C'_i(t_k)$ are substituted in place of $f_{i,j,m}(t_k)$ and $C_i(t_k)$.

For these performance calculations, we more densely sampled metabolite concentration and derivative time courses to provide a more accurate representation of interpolation performance, relevant to the general case of dynamic genome-scale metabolic modeling. For each model, results were

sampled at time steps a factor of ten smaller than those used for the fitting data, resulting in n_I = 201 interpolated points for the *E. coli* model and n_I = 601 interpolated points for the *S. cerevisiae* model (these sets included the original sampled time points).

We ranked the functions' performance and averaged these ranks to provide a quantitative overall comparison of each function. We ranked the performance of each function for each noisy time course ($D_{i,m}$) of each metabolite and averaged the ranks for each function across all of these time courses. In both cases, a harmonic mean was used to average ranks, emphasizing the relative importance of comparing functions that perform strongly in some cases; in this way, the difference between rank 1 and rank 2 was weighted more heavily than the difference between, for example, rank 4 and rank 5.

This averaged rank approach was used to compare performance of fitting functions for the Direct Fit method only and for the Resampling Method only, as well as to compare performance between these two methods for all of the different fitting functions.

2.3 Results

2.3.1 A description of the overall approach

Two small-scale ODE metabolic models describing *E. coli* and *S. cerevisiae* metabolism were used to generate synthetic reference data for the assessment

of new methods for concentration and flux inference from metabolite data. Using this synthetic reference data as a basis, noisy time courses were generated to represent the noisy data that typically result from metabolite profiling experiments. Eight different functions, including four polynomials, three rational functions, and one impulse model function (as described in the Methods section 2.2.1 and in Table 2.1), were used as candidate fitting functions for these noisy metabolite time course data. Two different approaches were used to fit metabolite concentration curves to the noisy synthetic datasets generated from the original ODE models.

The Direct Fit Method, described in the Methods section, was a standard fitting of functions to given experimental data. The approach used to assess the effectiveness of the Direct Fit Method for each of the candidate fitting functions is outlined in Figure 2.1. Briefly, after multiple noisy time courses were generated from the synthetic reference data, each candidate function was fitted to each of the noisy time courses. Each of these fits was then assessed for their performance at recapitulating and interpolating the original data; these assessments were performed on both the fitted concentrations and the derivative values that resulted from those fitted concentrations.

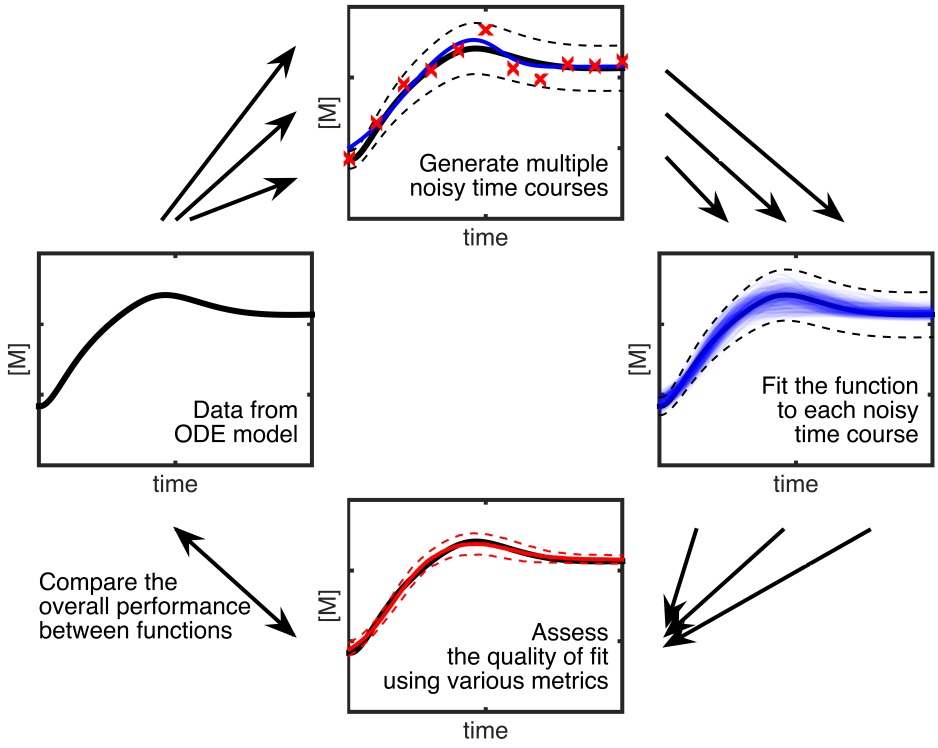

Figure 2.1. Schematic of the Direct Fit Method
Synthetic gold standard data are generated by simulating a system of ODEs over the time interval of interest. From the synthetic data, noisy time courses are generated by adding Gaussian noise with a 15% coefficient of variation to the synthetic data, to simulate experimental sources of variation in measurements. Multiple such noisy time courses are generated. A smoothing function is fit directly to a noisy time course, and the resulting fit (or its derivative) is compared against the synthetic data to determine how closely they match. The performance of each function can then be compared based on their performance relative to the initial synthetic data.

The Resampling Method, also described in the Methods section, involved fitting multiple noisy datasets generated from a single experimental (or noisy synthetic) dataset. By taking the median of these multiple fits, susceptibility to noise and outliers in the original experimental data was reduced, providing more robust estimates of metabolite concentrations and derivatives. The approach used to assess the effectiveness of the Resampling Method for each of the candidate fitting functions is outlined in Figure 2.2.

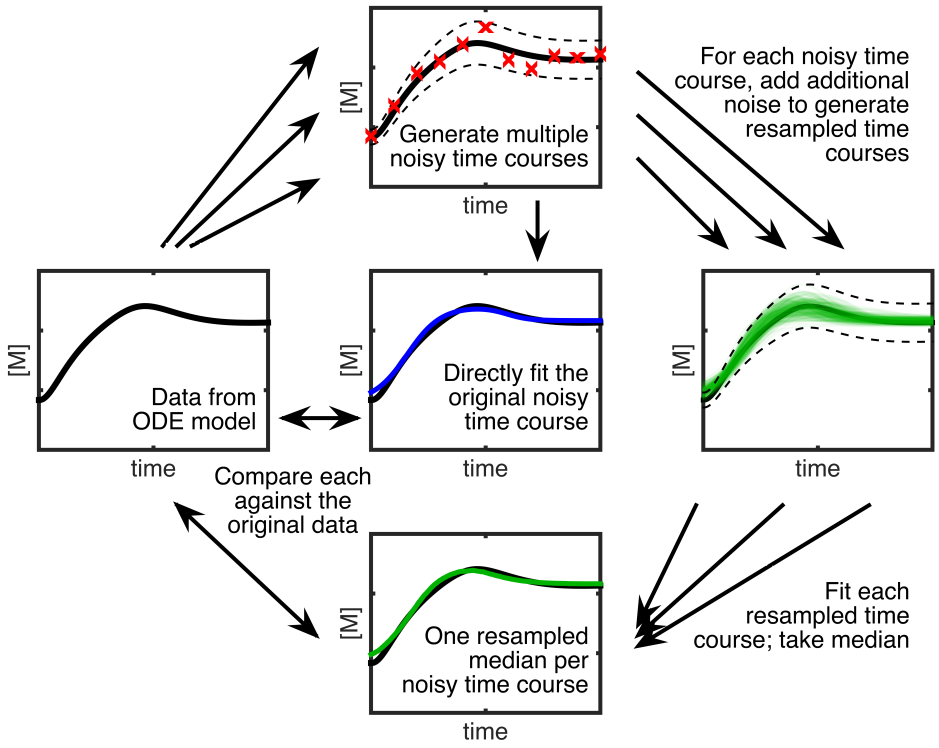

Figure 2.2. Schematic of the Resampling Method
As in the Direct Fit method, synthetic data and base noisy time courses are generated from a system of ODEs. In the Resampling Method, each base noisy time course is then used to generate a set of "Resampled" time courses, by using the same process used to generate the base noisy time courses from the synthetic data, only now with the base noisy time course as the input. The function of interest is fit to each of these resampled time courses, and the median of these functions (or their derivatives) is used to generate the resulting smoothed time course corresponding to the specific base noisy time course. As in the Direct Fit method, these median profiles can be assessed to determine accuracy and performance of the function.

Briefly, multiple "base" noisy time courses were generated from the original model to represent experimental measurements; these were fitted using the Direct Fit Method for comparison. In parallel, additional noisy time course profiles were generated ("resampled") from each of these base noisy time courses and subsequently fitted using the methods described for the Direct Fit Method—yielding a fitted concentration for each resampled noisy time course for a given base noisy time course. For each base noisy time course, the median per time point of the fitted profiles (or profile derivatives) for the resampled noisy time

courses was then used to determine the overall fitted profile. This profile, along with the Direct Fit Method profile, was compared to the original synthetic reference data to assess prediction accuracy.

2.3.2 Parameter constraints improved the behavior of fitted results

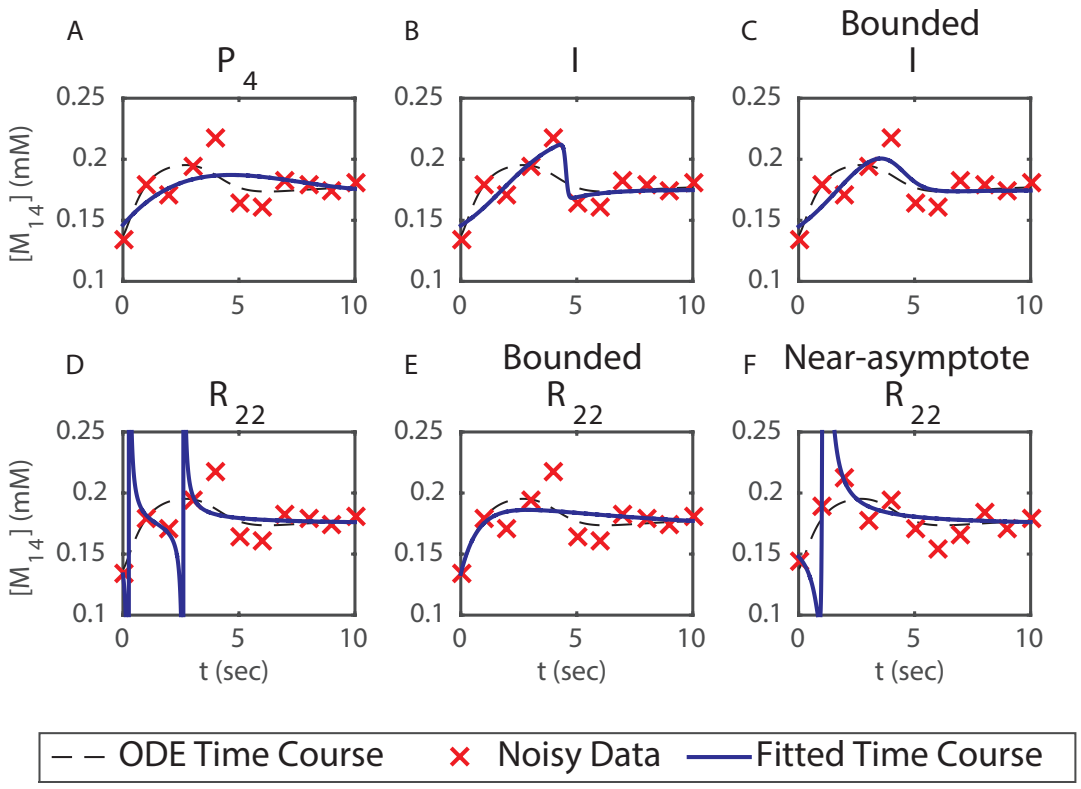

Figure 2.3. Performance of different fitting functions for fitting concentration trajectories
Thin, dotted black lines are the original synthetic data. Red crosses are the noisy time course data used to fit the functions. Solid blue lines are the function fitted to the data.
A. Polynomial curves were consistent but typically not very accurate.
B. The sigmoidal impulse function performed well but sometimes exhibited steep derivatives.
C. Constraining the parameter space for the impulse function prevented this behavior.
D. The rational function R_{22} can exhibit unphysical asymptotes in the time interval of the data due to a polynomial term in the denominator.
E. Constraining the parameter space for R_{22} prevents such asymptotes.
F. However, near-asymptote behavior can still occur in the rational functions, despite the parameter restrictions, when the value of the denominator polynomial becomes sufficiently small.
Note: A-E all use the same noisy data set.

Figure 2.3 provides representative examples of performance for different candidate fitting functions using the Direct Fit Method and the *E. coli* model. Polynomial functions provided computationally efficient data smoothing with little susceptibility to noise, but had limited abilities to qualitatively capture the dynamics present in the *E. coli* model. For certain sets of noisy data, the rational functions or the impulse function returned unphysical or unreasonable results. This result highlighted a shortcoming in the basic implementation of the rational functions and prompted the development of additional constraints for use in the optimization step of fitting the rational functions and the impulse function.

2.3.2.1 Bounding the second order denominator polynomial of the R_{22} function

We observed that for approximately 29% of noisy datasets, the R_{22} rational function produced asymptotic behavior, as shown in Figure 2.3D. The frequency of asymptote occurrence varied significantly across the different metabolites in the model, as shown in Figure 2.4A. The source of these asymptotes was selection of “optimal” parameters such that the polynomial in the denominator of R_{22} had a root over the time range of the data. Technically, such parameter selections would be optimal based on the RMSD objective function, since the RMSD only considers the ability of the function to match the data provided for fitting. However, such selections lead to clearly unphysical profiles at interpolated points that would confound any efforts to use such fitted functions in genome-scale metabolic simulations.

Accordingly, we constrained the RMSD optimization for all rational functions such that parameters could not be selected that would cause a zero in the denominator over the time range of the data. In the case of R_{22}, the denominator is a second order polynomial. Examples of these polynomials are shown in Figure 2.4B; the position of the roots relative to the interval (t_1, t_2) is determined by the parameters p_4 and p_5. By bounding the values these parameters may take, we can prevent the resulting trajectory from producing asymptotes in the interval of the data.

We construct these bounds by starting with the denominator polynomial equation,

$$D(t) = t^2 + p_4 \cdot t + p_5$$

And the equation for the roots of a quadratic polynomial,

$$r = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

In this instance, a = 1, b = p_4, and c = p_5, and hence:

$$r = \frac{-p_4 \pm \sqrt{p_4^2 - 4 \cdot p_5}}{2}$$

We isolate the square root term to get

$$2 \cdot r + p_4 = \pm\sqrt{p_4^2 - 4 \cdot p_5}$$

and square both sides to get

$$4 \cdot r^2 + 4 \cdot r \cdot p_4 + p_4^2 = p_4^2 - 4 \cdot p_5$$

We then solve for p_5, and get

$$p_5 = -(r \cdot p_4 + r^2)$$

We set r = t_1 and r = t_2 to produce the main divisions in the space spanned by p_4, and p_5; these equations are plotted in Figure 2.4C, and described mathematically in Table 2.3. The value of p_5 relative to these divisions determines the placement of the roots of the corresponding denominator polynomial. In practice, we introduce a buffer to avoid cases in which the optimization forces the roots to t_1 or t_2 and produces asymptotes at the edges of the data. By default, this buffer is calculated from the time data used to fit the function to have a magnitude of ΔT, where ΔT is the time interval between sequential data points in the time course to be fit. As a concrete example, the *E. coli* model is simulated from 0 seconds to 20 seconds, and the fitted data is sampled at 1 second intervals; the corresponding buffers lead to t_1 = -1 and t_2 = 21, which prevents the roots of the denominator from falling in the interval of [-1, 21].

In some cases, a quadratic equation may have no real roots. This situation will never lead to asymptotes in the resulting R_{22} function, and so is also acceptable. We determine the boundary of this condition by setting the discriminant $p_4^2 - 4 \cdot p_5 < 0$. Solving for p_5, we get $p_5 > \frac{1}{4} p_4^2$. Under these conditions, the resulting roots are imaginary and do not produce asymptotes, regardless of the values of t_1 and t_2. This boundary is plotted in Figure 2.4C.

Of these seven regions, four produce behavior that is acceptable for our application. Region $\mathcal{R}_4$ represents the case where the roots straddle the time

interval of the data. Regions $\mathcal{R}_{II}$ and $\mathcal{R}_{III}$ represent the cases where the roots are either both above or both below the boundaries, respectively. Region $\mathcal{R}_{IV}$ represents the case where there are no real roots. Because Regions $\mathcal{R}_{II}$ and $\mathcal{R}_{IV}$ border each other, as do Regions $\mathcal{R}_{III}$ and $\mathcal{R}_{IV}$, in practice we re-divide the regions to simplify the parameter bounds assigned to the solver to minimize the number of nonlinear constraints (the orange points in Figure 2.4C where the linear boundaries lie tangent to $\mathcal{R}_{IV}$ mark the values for p_4 we chose).

As described in section 2.3.2.3 below, we fit the data using multiple sets of random initial conditions. For each set of initial conditions, we perform four parameter optimizations, corresponding to the four (adjusted) Parameter Regions we described here and shown in Figure 2.4C. From these four regions, we choose the parameter set with the lowest RMSD to represent the parameter set for a given set of initial conditions. Parameter optimizations were performed using `fmincon()` in MATLAB.

Figure 2.3E shows the trajectory of R_{22} after adding additional constraints to the allowed parameter values in rational functions. However, this solution does not protect against near-asymptotic behavior in R_{22}, where the denominator approaches but does not reach zero; Figure 2.3F depicts such a case using a different set of noisy data for the same metabolite. Nonetheless, the results in

Figure 2.3E demonstrate significant improvement upon the results from Figure 2.3D with no parameter constraints.

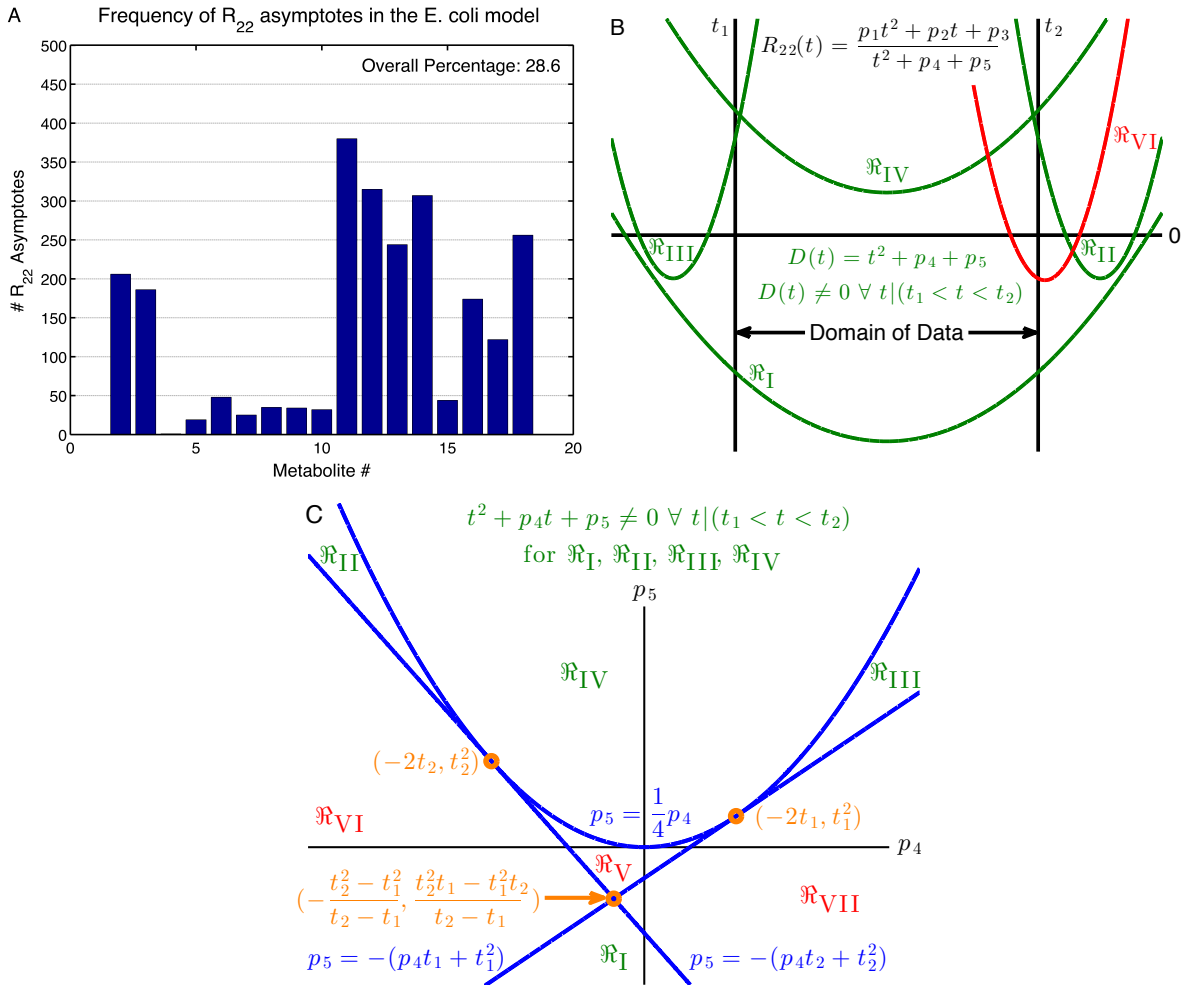

Figure 2.4. Parameter domain and bounding for the rational function denominators
A. In 28.6% of cases, the best-fit R_{22} concentration trajectory in the *E.coli* model contains an asymptote in the domain of the data, leading to the qualitatively invalid fitted concentration behavior shown in Fig 3D.
B. Some R_{22} denominator polynomials (colored red) contain roots in the time interval of the data, leading to asymptotes in the resulting concentration trajectory. Others (green) produce qualitatively valid trajectories. Polynomials are labeled by the region in panel C from which parameters p_4 and p_5 were taken.
C. Multiple regimes exist in the parameter space spanned by the R_{22} denominator polynomial, with different root positions for the denominator polynomial. Blue lines designate the boundaries between regions. Green text ($\mathcal{R}_I$, $\mathcal{R}_{II}$, $\mathcal{R}_{III}$, $\mathcal{R}_{IV}$) indicates regions that preclude problematic roots, while red text ($\mathcal{R}_V$, $\mathcal{R}_{VI}$, $\mathcal{R}_{VII}$) indicates regions that produce qualitatively invalid behaviors. In practice, the valid R_{22} denominator parameter regions are modified to simplify solver implementation by grouping sub-regions of R_{IV} instead with R_{II} and R_{III}.

Table 2.3. A description of parameter space for the rational function R_{22}
The parameters p_4 and p_5 determine the behavior of the denominator polynomial in R_{22} by determining the location of its roots. The space spanned by these parameters can be divided into 7 regions, based on the position of each of these roots relative to the interval (t_1, t_2).

Region	Constraints	Description
$\mathcal{R}_{\text{I}}$	$p_5 < -(t_1 \cdot p_4 + t_1^2)$ $p_5 < -(t_2 \cdot p_4 + t_2^2)$	$r_1 < t_1$ $r_2 > t_2$
$\mathcal{R}_{\text{II}}$	$p_5 > -(t_2 \cdot p_4 + t_2^2)$ $p_5 < \frac{1}{4}p_4^2$	$r_1 > t_2$ $r_2 > t_2$
$\mathcal{R}_{\text{III}}$	$p_5 > -(t_1 \cdot p_4 + t_1^2)$ $p_5 < \frac{1}{4}p_4^2$	$r_1 < t_1$ $r_2 < t_1$
$\mathcal{R}_{\text{IV}}$	$p_5 > \frac{1}{4}p_4^2$	$r_1 \epsilon \mathbb{C}$ $r_2 \epsilon \mathbb{C}$
$\mathcal{R}_{\text{V}}$	$p_5 > -(t_1 \cdot p_4 + t_1^2)$ $p_5 > -(t_2 \cdot p_4 + t_2^2)$ $p_5 < \frac{1}{4}p_4^2$	$t_1 < r_1 < t_2$ $t_1 < r_2 < t_2$
$\mathcal{R}_{\text{VI}}$	$p_5 > -(t_1 \cdot p_4 + t_1^2)$ $p_5 < -(t_2 \cdot p_4 + t_2^2)$	$t_1 < r_1 < t_2$ $r_2 > t_2$
$\mathcal{R}_{\text{VII}}$	$p_5 < -(t_1 \cdot p_4 + t_1^2)$ $p_5 > -(t_2 \cdot p_4 + t_2^2)$	$r_1 < t_1$ $t_1 < r_2 < t_2$

2.3.2.2 Bounding the Impulse Function

The impulse function exhibited a similar phenomenon, insofar as it yielded results that were technically correct based on the RMSD optimization function but were physically unreasonable. As depicted in Figure 2.3B, the impulse function

sometimes produced sharp shifts in concentration, which translated to sharp spikes in the derivative trajectory. In addition, we noticed that our parameter-fitting solver was prone to getting stuck in local minima when the resulting time delay parameters were outside the time span of the data. Conveniently, the direct correspondence between parameter values and features in the graph of the impulse function made it straightforward to implement effective and beneficial parameter boundaries. One fixed constraint and two new adjustable optimization parameters were created that were used to constrain the possible parameter space. These bounds were implemented as constant parameter values, and optimization was performed using `fmincon()` in MATLAB.

Our first set of bounds is on the time delay parameters. Because there is no basis from the data for modeling a transition outside the interval of the data, we restrict these values to the range of the time values in the data. This has the advantage of removing insensitive local optima from the available parameter space, and helps ensure that the fitting error remains sensitive to parameter values. In effect, we restrict τ_1 and τ_2 to the interval (t_1,t_2). Unlike the rational functions, by default we did not add a buffer to the time interval of the data; for example, in the *E.coli* model, we used t_1= 0 seconds and t_2 = 20 seconds to restrict τ_1 and τ_2.

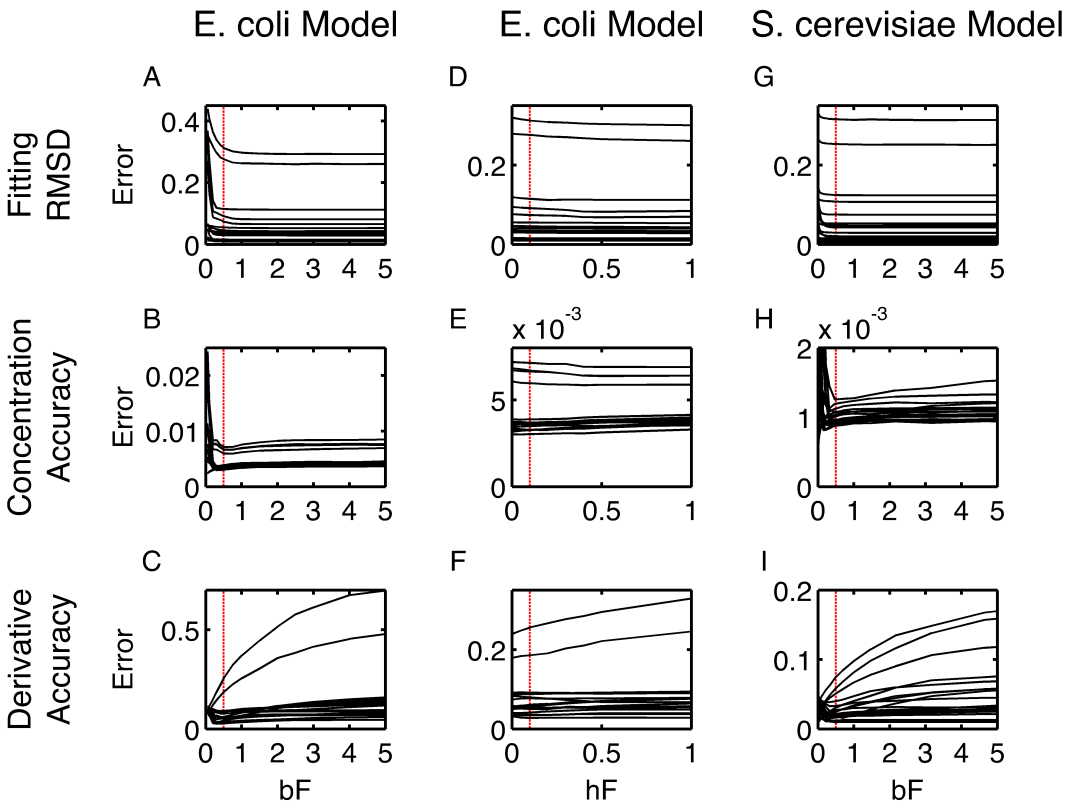

Figure 2.5. The impact of global parameters on Impulse performance
The impulse function was tested over a range of values for b_f and h_f to determine optimal values for these parameters, and to assess sensitivity to those values. Solid black lines indicate individual metabolites. Dashed red lines indicate our selected values of $b_f = 0.5$ or $h_f = 0.1$.
A-C. b_f was varied in the *E.coli* model, and we found that a value of bf = 0.5 generally worked well based on RMSD during fitting, concentration accuracy, and derivative accuracy.
D-F. h_f was varied in the *E.coli* model, and found not to strongly affect the performance of any of our metrics. We chose a value of $h_f = 0.1$ to permit small fluctuations relative to the range of the data due to expected noisiness in the data.
G-I. We verified our choice of $b_f = 0.5$ in the S. cerevisiae model, and found that this value worked reasonably well for this model as well.

The second set of bounds we introduced restrict the magnitude of the impulse function transition parameters, β_1 and β_2; this prevents the transitions from becoming too sharp. The justification for this restriction is that since the data are sampled at a given frequency, there is no justification from the data to model faster dynamics in the resulting function than is present in the data. As such, the sharpness of the impulse transitions is set to be inversely proportional to the size

of the time interval used to fit the data. The proportionality factor for this inverse relationship is a global parameter, which we label b_f. In effect, we enforce that $|\beta| < \frac{b_f}{\Delta t}$, where Δt is the time interval between sequential data points in the time course to be fit. For the *E. coli* dataset, we determined that a value of $b_f = 0.5$ was generally near optimal for the data, as shown in Figure 2.5A-C. The two exceptions to this were Metabolites 12 and 18; for these metabolites, the error in derivative accuracy decreased as the sharpness was more heavily restricted (i.e., as b_f was decreased).

The last set of bounds we introduced restrict the height parameters controlling the initial, intermediate, and steady-state values of the resulting impulse (h_0, h_1, and h_2, respectively). We restrict these parameters to a window defined by the range of the concentration data. The rationale behind this restriction is that there is no basis from the data for the function to model a change in value far outside the range seen in the data. We allowed this window to be extended by an added percentage, represented by the global parameter, h_f. For example, a value of $h_f = 0.10$ corresponds to extending the bounds an additional 10% both above and below the range of the data. As shown in Figure 2.5D-F, we found the performance of the impulse in the *E.coli* model to be generally insensitive above relatively small values of h_f; a value of $h_f = 0.1$ was selected to permit some fluctuation due to the expected noisiness of the data.

Using h_f = 0.1 and b_f = 0.5 resulted in more realistic profiles like those shown in Figure 2.3C. Importantly, in addition to the direct physical interpretation of these global parameters, the results of the parameter fitting are not highly sensitive to small changes in h_f and b_f (see Figure 4.5), and as a result the values of h_f and b_f that we used were generalizable to both model systems even though they were selected only based on their performance for the *E. coli* model.

It is important to note two features of the global parameters h_f and b_f. First, they are not parameters in the resulting fitted parameter set; they only determine bounds on those parameters. Second, they were not fit to individual noisy time courses, or even to individual metabolites. Because of this, we hypothesized that we could directly re-use these values from the *E. coli* model with the *S. cerevisiae* model without adjusting them. To validate that this was the case, we tested a range of b_f values on the *S. cerevisiae* model. As shown in Figure 2.5G-I, the value of b_f = 0.5 determined from the *E. coli* model was also an effective choice for the *S. cerevisiae* model. This suggests that these global parameters may generalize reasonably well to other models, and as such we did not penalize our metrics to account for them.

2.3.2.3 Addressing issues of local minima and solver consistency

For the rational and impulse functions, our solver routines use initial random parameter values with the built-in MATLAB function `fmincon()` to find the

parameter values with the lowest RMSD, subject to the constraints described previously. During our investigation, we found that using a single initial random parameter set would lead to inconsistent RMSD values for the resulting fitted parameters; this indicates that for a single set of initial conditions, there is a real risk of encountering local minima when minimizing the fitting RMSD. To counter this, we seeded our solvers with multiple sets of random initial parameters and selected from the resulting fitted parameters the set that produced the lowest RMSD.

To determine how many sets of random initial parameters was needed for each function, we tested values between 1 and 50 seeds for each function, repeating the procedure for each metabolite in the *E.coli* model using the dataset we produced for the Direct Fit method. The RMSD values of the resulting optimal parameter sets for each noisy time course were then compared against the RMSD values generated using 50 seeds by calculating a Pearson correlation between the RMSD values. As shown in Figure 2.6, we found that 20 seeds were sufficient across all functions to ensure consistent results; in many cases, far fewer seeds were necessary to produce RMSD values equivalent to the 50 seeds case. For the results shown in this chapter, we used 20 seeds for all four functions.

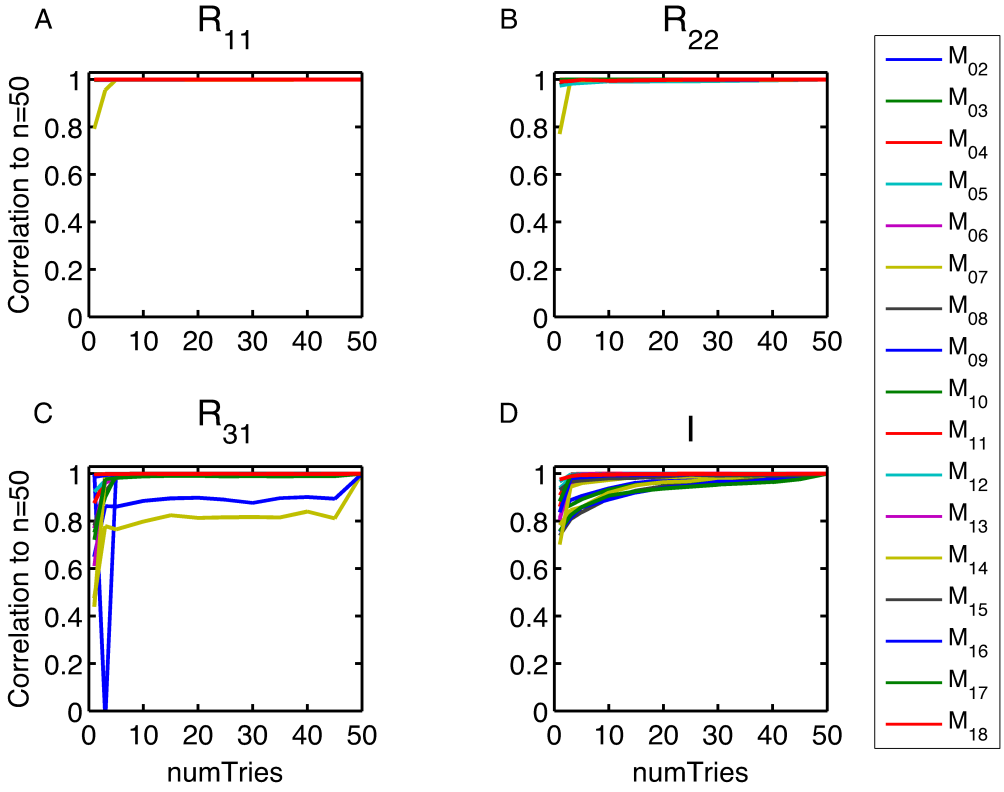

Figure 2.6. The effect of using multiple solver initial conditions on the consistency of the solver output

The parameter solver may identify only a local minimum RMSD value. In these cases, seeding the solver with multiple sets of random initial conditions may enable identification of a parameter set that produces a lower RMSD. We tested the effect of multiple optimizations with different initialization values; the parameter numTries indicates the number of initial condition sets with which we seeded the solver. Functions were fit to the noisy time courses generated for the Direct Fit method on the *E. coli* model. For each function and metabolite, numTries was varied from 1 to 50. The resulting RMSD values for each value of numTries were then compared to the RMSD values determined when numTries=50 using a Pearson correlation; a higher value indicates that the parameters found for that value of numTries is more consistent with the parameters produced when numTries=50. A) R_{11} requires very few seeds to produce consistent results. B) R_{22} also produces consistent results with few seeds. C) For most metabolites, R_{31} requires few seeds. However, this was not the case for two of the 17 metabolites (M_{07} and M_{09}). D) The Impulse required few tries for some metabolites, but some metabolties do improve as numTries is increased. The results in this chapter used numTries = 20; we note that this indicates that increasing the number of seeds would likely improve the performance of this function modestly.

2.3.3 The impulse model consistently fits metabolite data with low error

To quantitatively assess the effectiveness of the candidate fitting functions using the Direct Fit Method in the *E. coli* model, we generated 500 noisy time course

data sets for each of the 17 metabolites. The parameters resulting from fitting each noisy time course were used to calculate concentration and derivative trajectories, with the corresponding performance accuracy calculated and averaged as described in the Methods section. The results of these calculations are summarized in Table 2.4, which presents the averaged ranks for each function and each metric. Figure 2.7A and 2.7B provide a detailed quantitative comparison of each fitting function. The impulse function, I, showed the best rank averages for accuracy in both concentration and derivatives, and was almost always the best-performing function across all of the metabolites.

Table 2.4. Average rank of function accuracy using the Direct Fit method on the *E. coli* model

Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	3.68	4.13	2.50	2.94	3.94	2.33	4.83	1.74
Derivative Accuracy	3.18	3.45	2.48	3.08	3.58	2.61	3.77	2.18

The notable exceptions to the superior performance of the impulse function were on Metabolites 12 and 18. Figure 2.8 summarizes the performance of the impulse function and an average fitting function, P_4, for Metabolite 12, with representative fitted profiles in Figure 2.8A and 2.8B, and a direct comparison between the performance of P_4 and I in Figure 2.8C. P_4 consistently performed better than I. However, as is clear from Figure 2.8A and 2.8B, the total change in metabolite level was smaller than the expected range of variability of experimental measurements. Given the sparsity of samples, this metabolite's profile is likely essentially unidentifiable, and so the performance of the different functions is

likely based only on general trends of the functional forms near the ends of the time range, rather than any reliably accurate fitting.

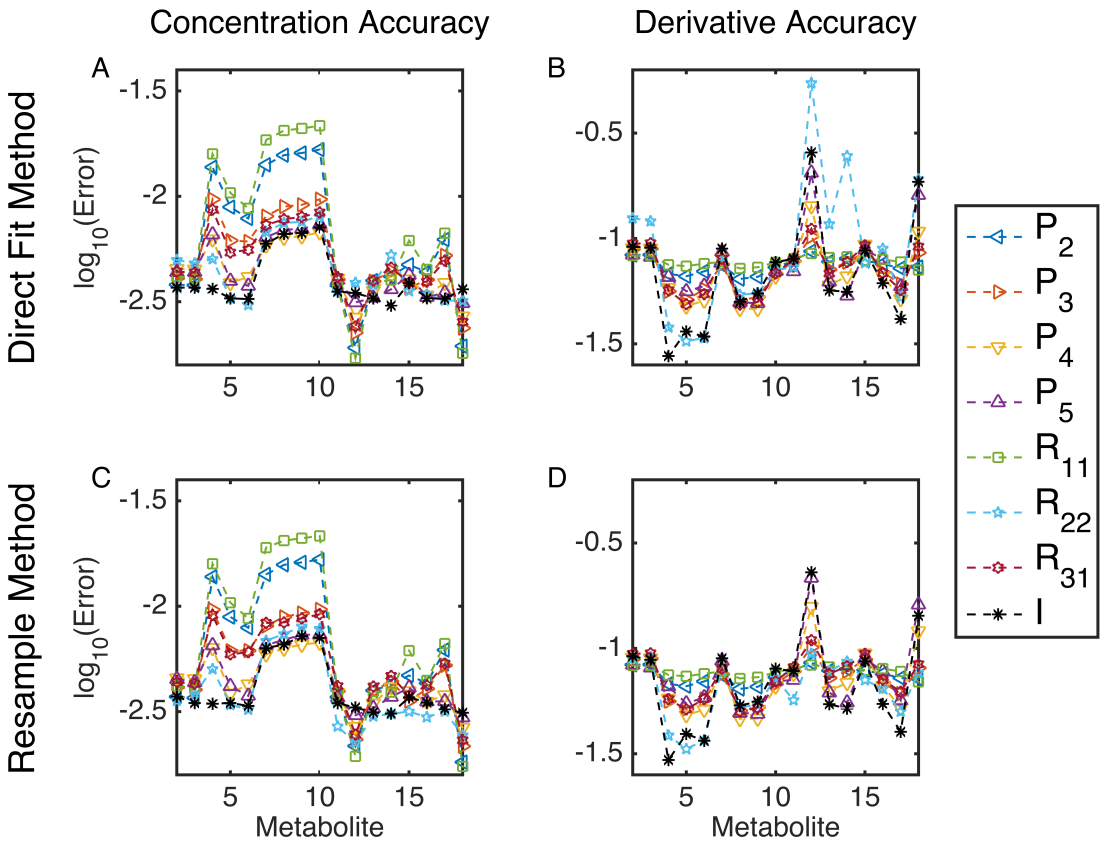

Figure 2.7. Quantitative assessment of function accuracy across metabolites in the *E. coli* model

The impulse function performs consistently well across most metabolites for both (**A**) concentration and (**B**) derivative accuracy. The resampling method improves the performance of a number of functions for both (**C**) concentration and (**D**) derivative accuracy. Error metrics are normalized to average metabolite concentrations (see Methods) for easier visualization and are presented in log-transformed format.

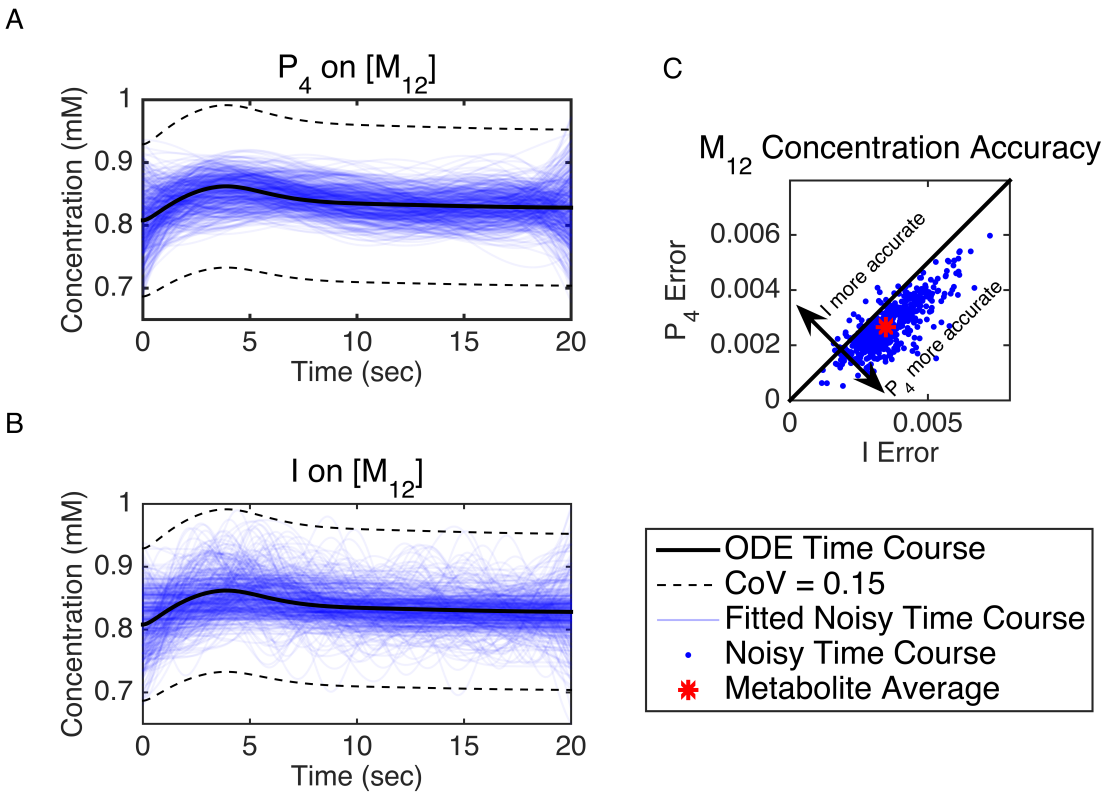

Figure 2.8. Comparison of the Impulse and P_4 on Metabolite 12 (6-Phosphogluconate) over 500 random noisy time courses

A. The P_4 polynomial function intrinsically curves upwards or downwards at the ends of the interval, which helps match the early slope in the synthetic data.

B. The impulse function exhibits greater variability across different noisy replicates due to the small dynamic concentration range in the synthetic data relative to the noise introduced. Solid black lines indicate the synthetic data. Dashed black lines indicate the 15% coefficient of variation envelope, used to generate the noisy time course data. Blue lines indicate the concentration trajectory of functional fits to individual noisy time courses.

C. As a result, the P_4 polynomial consistently fits the synthetic data concentration with lower error than the impulse. Blue dots indicate the error of each function in recapitulating the synthetic data when fit to a particular noisy time course. The red star indicates the average error of the blue dots.

2.3.4 The Resampling Method can improve fitting and predictions in the E. coli model

To quantitatively assess the performance of the Resampling Method in the *E. coli* model, we generated 50 noisy time courses from the synthetic reference data for each of the 17 metabolites, and for each noisy time course, an additional 250

resampled noisy time courses. For each noisy and resampled time course, each function was fitted as described in the Methods, and the resulting Direct Fit or Resampling Method trajectories used to calculate the performance metrics. The overall results are shown in Table 2.5. Results jointly ranking the performance of functions across both the Direct Fit Method and the Resampling Method are shown in Table 2.6. The Resampling Method had the greatest impact on the ranking of the rational function R_{22}, resulting in it being similar in accuracy and consistency to the impulse function, I. This consistently good performance is also evident in Figure 2.7C and 2.7D, which provide a detailed quantitative comparison of each fitting function.

Table 2.5. Average rank of function accuracy using the Resampling Method on the *E. coli* model

Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	4.02	4.16	2.44	3.11	4.22	1.83	5.32	1.90
Derivative Accuracy	3.38	3.40	2.50	3.07	3.68	2.16	4.66	2.20

Table 2.6. Average rank of function and method accuracy using the *E.coli* model
Results from both the Direct Fit (DF) and Resampling (RM) methods are all ranked together to facilitate direct comparison of their performance.

	P_2		P_3		P_4		P_5		R_{11}		R_{22}		R_{31}		I	
Average Rank of Metric	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM
Concentration Accuracy	6.62	6.70	7.36	7.35	3.76	3.94	5.34	5.35	7.17	6.62	3.48	2.55	8.77	10.17	2.60	2.88
Derivative Accuracy	5.40	5.50	6.20	6.21	3.98	4.02	5.12	5.09	6.49	5.85	3.76	3.12	6.33	8.96	3.30	3.17

The impacts of the Resampling Method varied across the different types of functions; representative graphs are presented in Figure 2.9, with a complete summary provided in Table 2.6. Polynomial functions showed little to no change

in results from using the Resampling Method, while rational functions show moderate to noticeable benefit. The impulse function benefited in some cases as well. Across all functions, use of the Resampling Method only infrequently caused decreased performance, and typically with very small changes relative to the magnitude of the error.

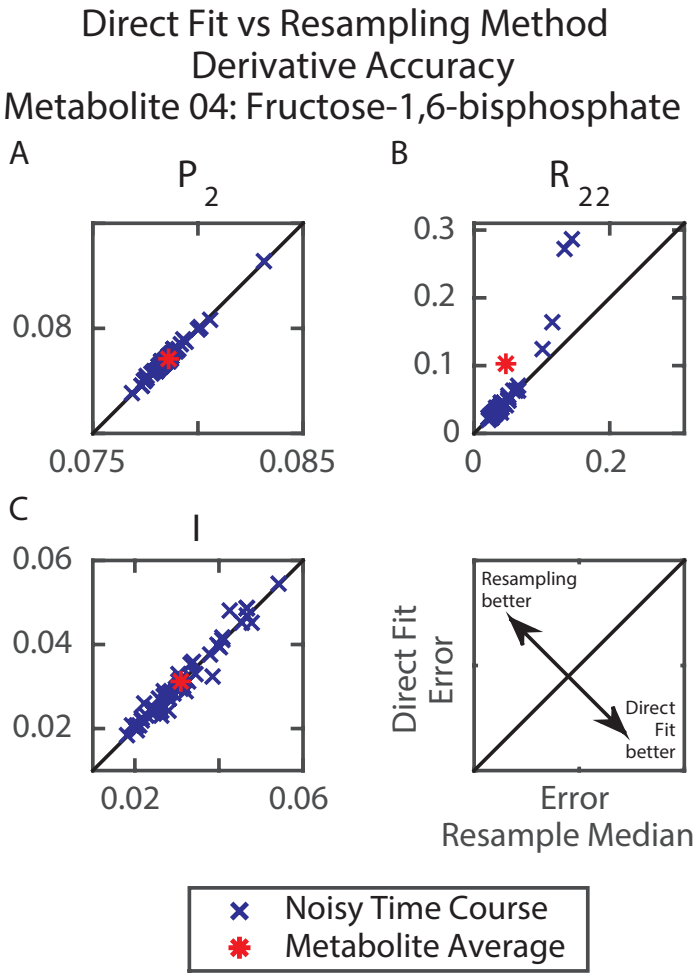

Figure 2.9. The effect of the Resampling Method on the derivative accuracy of three representative functions
The error for fitted concentration profiles was determined for both the Direct Fit and Resampling Methods and directly compared. A) For polynomial functions the Resampling Method produces results nearly identical to the Direct Fit method. B) The R_{22} rational function can produce derivative errors several orders of magnitude greater using the Direct Fit method (not shown on these axes) than when using the Resampling Method, making the Resampling Method more accurate on average. C) The impulse function is generally consistent between the Direct Fit and Resampling Methods, but does show some variability. Other metabolites exhibit modest benefits from the Resampling Method relative to the Direct Fit Method.

2.3.5 The S. cerevisiae *model results show similar trends*

We then quantitatively assessed the performance of all candidate fitting functions using both the Direct Fit Method and the Resampling Method in the *S. cerevisiae* model. We generated 500 noisy time courses for each of the 21 metabolites for use in the Direct Fit method. For use in the Resampling Method we generated 50 base noisy time courses for each of the 21 metabolites, along with an additional 250 resampled noisy time courses for each base noisy time course. Parameters were fit for each method as described in the Methods section. Tables 2.7 and 2.8 present the average ranks for the Direct Method and Resampling Method, both separately and combined, respectively. Figure 2.10 provides a detailed quantitative comparison of each fitting function. For this model, the R_{22} rational function and the impulse function, I, were usually among the best-performing fitting functions, with R_{22} performing best for concentrations and I performing best for derivatives.

Table 2.7. Average rank of function accuracy using the *S. cerevisiae* model
Here, the Direct Fit and Resampling Methods are ranked and averaged separately.

	Direct Fit Method								Resampling Method							
Average Rank of Metric	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I	P_2	P_3	P_4	P_5	R_{11}	R_{22}	R_{31}	I
Concentration Accuracy	4.28	4.00	3.83	3.22	4.81	1.34	4.45	2.07	4.48	4.15	3.90	3.33	4.82	1.24	4.79	2.10
Derivative Accuracy	3.99	3.65	3.55	2.77	4.80	1.95	4.44	1.66	4.39	4.00	3.81	2.92	4.81	1.61	5.06	1.64

Table 2.8. Average rank of function and method accuracy using the *S. cerevisiae* model
Results from both the Direct Fit (DF) and Resampling (RM) methods are all ranked together to facilitate direct comparison of their performance.

	P_2		P_3		P_4		P_5		R_{11}		R_{22}		R_{31}		I	
Average Rank of Metric	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM	DF	RM
Concentration Accuracy	7.37	7.82	7.05	7.55	7.14	7.17	5.86	6.02	7.92	7.98	1.85	1.65	7.85	8.98	3.59	3.22
Derivative Accuracy	7.52	7.41	7.16	6.75	6.64	6.74	4.79	4.85	8.23	8.10	2.95	2.14	8.34	9.43	2.72	2.15

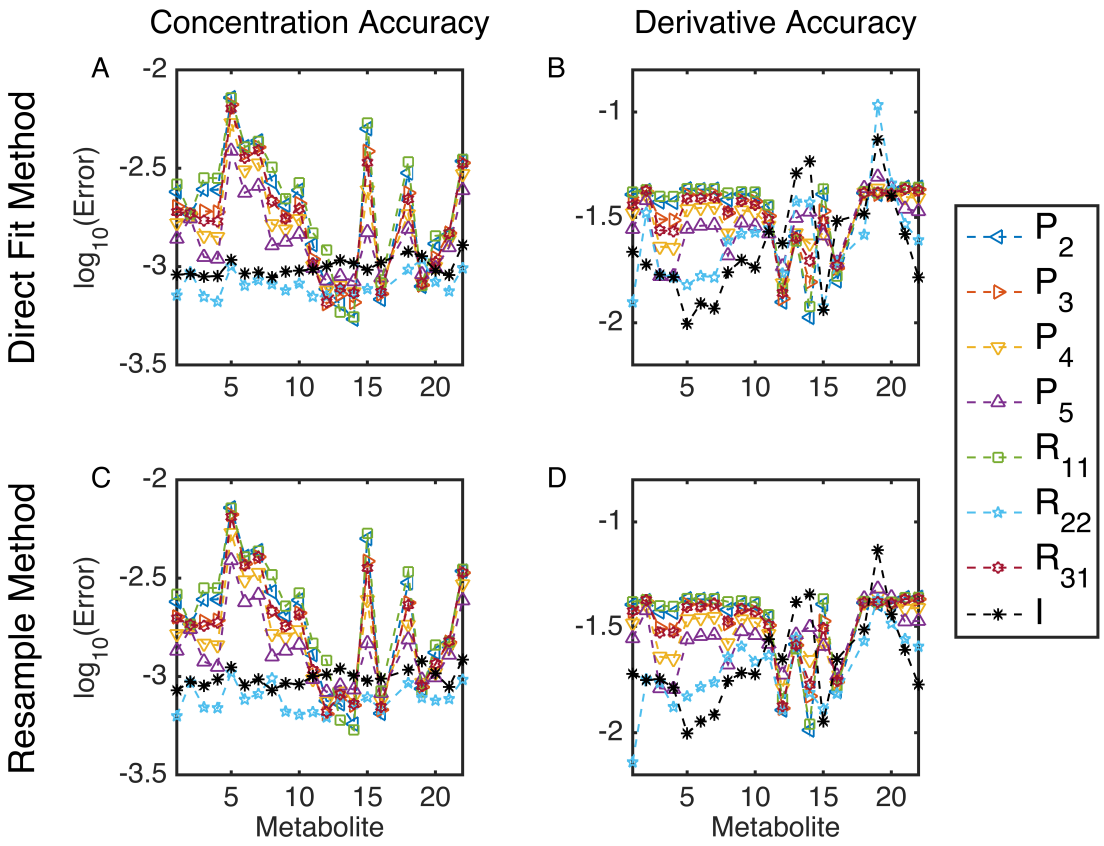

Figure 2.10. Quantitative assessment of function accuracy across metabolites in the S. cerevisiae model
Results by metric are presented for the Direct Fit Method for (A) concentration accuracy and (B) derivative accuracy, and for the Resampling Method for (C) concentration accuracy and (D) derivative accuracy. Error metrics are normalized to average metabolite concentrations (see Methods) for easier visualization and are presented in log-transformed format.

2.4 Discussion

2.4.1 Context

The goal of this effort was to improve the prediction of concentration and derivative time-course profiles derived from experimentally measured (or synthetic, noisy) metabolite data. Two small-scale model metabolic systems were used as the basis for assessing the performance of new methods to calculate and interpolate concentration and flux values based on metabolite data. These two models have different time scales and dynamics, which provided a broader

assessment of the potential utility of our approaches. These models were also used in previous work on estimating flux distributions from metabolite data[19], which allowed for direct comparison. Integrating these systems numerically provided an exact reference dataset to which we could compare fitted results. However, real metabolite concentration data contain significant variability, so we only used noisy synthetic data derived from this reference data to test the effectiveness of our approaches. In this way, we were able to generate data of defined quality and arbitrary quantity with known underlying dynamics; this allowed us to precisely and rigorously determine the performance of each approach under study.

The approach of Ishii *et al*. was to fit all of the functions to the time course in question and select the function with the lowest fitting error, once accounting for the number of fitted parameters[19]. While this is certainly a viable approach that can be extended to include the sigmoidal impulse model, here we have also investigated whether this single, biologically reasonable function can be used instead of selecting the best-fitting function from a list of arbitrary candidates. We consider the relative benefits of each function type below.

2.4.2 Polynomials are consistent but inaccurate

The polynomial functions are computationally inexpensive to fit, use few parameters (ranging from three to six), and are widely used for smoothing noisy

data. They are consistent and well-behaved, exhibiting very little sensitivity to noise. As demonstrated by their ranks in Tables 2.4, 2.5, and 2.6, they can do a reasonable job in estimating concentrations and at times even in estimating derivatives (ranking as low as 2.5 but often closer to 3.5 or 4). However, they are ill-suited to capturing dynamics that include a terminal steady state, particularly since their functional form requires them to be monotonically increasing or decreasing at the ends of the time range; this also makes them a poor choice for even limited extrapolation.

2.4.3 Resampling improves rational function accuracy

The rational functions (using three to five parameters) can exhibit a wider range of behaviors than the polynomials with the same number of parameters, and it has been reported that for many metabolite time courses, they yield better performance than the polynomials[19]. Our parameter restriction strategy was largely effective in addressing their potential to fit best with parameters that produce asymptotic behavior, though there are still lingering issues with near-asymptotes that yield spurious behavior and even negative concentrations for the R_{22} function (see Figure 2.2F). However, as shown in Table 2.5, this effect is largely ameliorated by the use of the Resampling Method to filter out asymptotic trajectories, making R_{22} one of the more effective functions we studied.

2.4.4 The impulse function is a generally effective single fitting function model

The last function, the sigmoidal impulse, is the product of two sigmoidal logistic functions[21,22]. As previously stated, it recapitulates the dynamics of a common biological process: a two-phase transition from one steady state to a (potentially new) steady state through an intermediate state. Its parameters directly correspond to features of this trajectory: the h parameters represent the initial, intermediate, and steady-state metabolite levels; the τ parameters represent the timing of the on and off transitions (accumulation and depletion driven by processes such as synthesis and degradation) in response to a perturbation; and the β parameters represent how rapidly those transition processes occur. In contrast with the work done by Chechik *et al.*, we allowed the β parameters to vary independently to reflect the fact that the on and off transitions can represent different biological processes (e.g., glucose uptake versus metabolism), which one would reasonably expect to exhibit distinct dynamics[21].

While potentially exhibiting undesirable behaviors with unrestricted parameter values, our parameter bounding strategies for avoiding broad local minima and overly sharp transitions were effective at preventing these undesirable behaviors (Figure 2.3B and 2.3C). Of particular note is that these parameters themselves typically exhibited broad local optima in performance (Figure 2.5), meaning that the fitting method was not very sensitive to the specific values selected; additionally, the default parameters we selected for the *E. coli* model generalized

well to a completely separate model, meaning that while they are technically adjustable parameters, they did not add significant risk of over-fitting to the parameter selection process.

Using the Direct Fit Method for the *E. coli* model, the impulse function performed consistently better than other functions (see Table 424) across all metabolites except for two: metabolites 12 and 18. For these metabolites, the actual dynamic range of metabolite concentrations in the synthetic reference data was substantially less than the range of the random noise used to construct the noisy time courses (see Figure 2.9). We cannot realistically expect to recover the underlying concentration in this case without either much more dense or much more accurate sampling. We suspect that the better performance of the polynomials was due in part to their tendency to swing upwards or downwards near the edges of the data, which captured the early time dynamics of each of these metabolites well; we note that the other high-performing fitting function, R_{22}, did poorly on these metabolites as well. The Resampling Method substantially improved the performance of R_{22} and slightly improved the performance of the impulse function on these metabolites (Figure 2.7), leading to qualitative behavior where the derivative effectively fluctuated around zero. Given the lack of statistically significant change over the time course of these metabolites, we argue that this is the behavior we should not only expect, but

actually be seeking given the essentially unidentifiable change in metabolite levels.

2.4.5 The Resampling Method generally improves on Direct Fit Method results

In general, the resampling method ranged from negligibly detrimental to highly beneficial. In a few cases, a very minor loss of performance was observed. Consistently, resampling provided no benefit to polynomials (Figure 2.9A); this is to be expected, since the polynomial functions are already insensitive to small changes in the data. The R_{11} and R_{31} rational functions saw minor improvements in general, while the impulse function saw improvements in cases where it performed most poorly (Figure 2.8C). The Resampling Method had the biggest effect on R_{22}; in the *E. coli* model, it moved from one of the worst performers to one of the overall best (Figure 2.7, Table 2.6). Generally speaking, then, the Resampling Method seems to be an effective way to improve accuracy at only a mild computational cost.

The Resampling Method appears to have an effect similar to parameter regularization by avoiding over-fitting due to noisy data[27]. However, we note that the Resampling Method returns a median of multiple fits, rather than a single parameter set. As a result, concentration and derivative values derived from this method need not strictly adhere to the functional form of the smoothing function; this flexibility can allow better approximation of the underlying data in cases

where the form of the particular function happens to be biased against the correct behavior.

2.4.6 S. cerevisiae *model results generally recapitulate* E. coli *model results*

The *S. cerevisiae* model generally recapitulated results from the *E. coli* model, demonstrating the potential generalizability of the Resampling Method and the impulse function (including the parameters used to restrict the fitting search space for the impulse function). For both the Direct Fit and Resampling Methods, the impulse function performed fairly well. One feature that distinguished the *S. cerevisiae* model from the *E. coli* model was the wider range of time scales present in the model's dynamics. Several metabolites (1-4,8-10,18-20) reached steady-state in several minutes, while others (12,13,14) took tens of minutes, and as a result did not reach steady-state during the time interval of the data. As the impulse function assumes long-term steady-state behavior for the time course, it did not perform as strongly for the Direct Fit Method for these metabolites. However, the Resampling Method did provide some improvement for these metabolites.

2.4.7 Selection of fitting functions should be driven by applications

In this chapter we considered the problem of data smoothing specifically in the context of genome-scale metabolic modeling. Two key factors in this application have driven our assessment of function and method performance. First, we

expect that we may need to provide flux values at points other than those for which experimental measurements are available (for instance, if a genome-scale model entails something akin to a Runge-Kutta numerical integration). This means that function accuracy should be assessed not only at the sampled points, but in between them as well. Without the inclusion of such interpolated values, some differences can be seen in apparent effectiveness; for example, previous work indicated that polynomials were more frequently optimal for the *S. cerevisiae* model[19], but in terms of practical applications they are usually inferior to R_{22} and the impulse function. Second, the main application of the metabolite concentration smoothing is for the estimation of metabolite fluxes; this means that while recapitulating the concentration profile is important, the more directly applicable metric is how accurate the derivative profile is. This distinction is most relevant for the *S. cerevisiae* model, where R_{22} more accurately recapitulates concentrations, but the impulse model more accurately recapitulates the derivatives that will be used in downstream analyses.

2.4.8 Single functions and biologically-inspired functions can be effective fitting models

While previous work selected the best-fitting of an essentially arbitrary set of functions for each individual metabolite based on the experimental data, we suggest that this may be a suboptimal approach. First, this increases the likelihood for over-fitting; it is difficult to estimate the number of effective

parameters that are introduced to the system by allowing for the variable selection of seven different models, but it suffices to say that the number of effective parameters is likely greater than the number of explicit parameters in the highest-order polynomial. As such, restricting the fitting to one function may be desirable from an information content perspective; both the R_{22} and impulse functions seem like reasonable, viable candidates for universal fitting functions. In fact, once the assessment metrics are based on a criterion more reasonable for the application (i.e., inclusion of interpolated points), there are few if any cases where the polynomials would be a desirable option. Second, there is inherent value in using biologically-inspired fitting functions. These functions, by design, recapitulate behaviors previously observed in biological systems; biasing the fit towards these results integrates prior knowledge that may help ensure that the model is closer to the underlying biology. Even though there are more parameters in these functions than the polynomials, the space of characteristic curves that can be fit is more restrictive and more relevant to expected biology, partially mitigating concerns about over-fitting due to excess parameters. In this sense, the impulse function may be the most desirable choice; either way, applying the Resampling Method ensures that the smoothing and fitting is improved over previous approaches.

2.4.9 Limitations

There are a few limitations to our analyses that bear noting. First, the number of variable parameters in the impulse function places a lower limit on the number of samples needed to fit the function well, which could stretch the experimental feasibility of acquiring a sufficient number of samples. However, our analyses have been consistent with previous work in terms of the number of samples used, and considering the possibility of using multiple biological replicates and multiple experiments to fit the same data, obtaining one or two dozen samples is often reasonable for a metabolomics experiment. Second, the impulse model assumes a steady state is reached at the end of the experiment, which may not be valid for all datasets. However, this concern is partially mitigated by the fact that many experiments would actually be continued until something more closely resembling a steady state is reached, minimizing the number of times significant non-zero derivatives were present at the end of the time range. There is also an obvious computational cost to fitting non-linearizable functions (as opposed to polynomials) and to applying the Resampling Method; however, since the data smoothing task is ultimately performed just once, not many times, we believe that the improvement in results is worth this computational cost, which is itself reasonable and does not require parallelization or even particularly long runtimes. Finally, we have not analyzed the ultimate downstream impacts in the genome-scale metabolic modeling application of the improvements we have made to assess their magnitude. Based on the tendency of functions like

polynomials to have nonzero derivatives at the end of the time range and the importance of being able to capture a steady state in a metabolic model, we expect that these improvements may be important, but will be to some extent model-specific and is thus beyond the scope of this chapter. Either way, it is often generally accepted that optimization of each intervening analysis or data processing step is desirable for complex modeling schema.

2.5 Conclusions

In this chapter, we have demonstrated two improvements to standard approaches to smooth metabolite concentration data for application to genome-scale metabolic modeling, including a Resampling Method to minimize susceptibility to experimental noise and the establishment of a single, biologically-inspired fitting function that performs well in almost all cases. In the course of this chapter, we also identified additional constraints that should be applied to existing data smoothing fitting functions to increase their robustness and activity. Taken together, these contributions have provided consistent and substantial improvements in existing methods to smooth and fit metabolite data for downstream applications, whether via a new fitting function or improvements made to existing fitting functions. We have shown these results to be generalizable across multiple models of metabolism, suggesting the potential for general utility of these improved methods to improve the accuracy of flux distributions calculated from the derivatives of their time courses.

2.6 References

1 Dromms, R. A. & Styczynski, M. P. Improved metabolite profile smoothing for flux estimation. *Molecular BioSystems* **11**, 2394-2405, doi:10.1039/C5MB00165J (2015).

2 Yizhak, K. *et al.* Phenotype-based cell-specific metabolic modeling reveals metabolic liabilities of cancer. *eLife* **3**, e03641, doi:10.7554/eLife.03641 (2014).

3 Burgard, A. P., Pharkya, P. & Maranas, C. D. Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng* **84**, 647-657, doi:10.1002/bit.10803 (2003).

4 Nakahigashi, K. *et al.* Systematic phenome analysis of Escherichia coli multiple-knockout mutants reveals hidden reactions in central carbon metabolism. *Mol Sys Biol* **5**, n/a-n/a, doi:10.1038/msb.2009.65 (2009).

5 Covert, M. W., Schilling, C. H. & Palsson, B. Regulation of Gene Expression in Flux Balance Models of Metabolism. *J Theor Biol* **213**, 73-88, doi:http://dx.doi.org/10.1006/jtbi.2001.2405 (2001).

6 Covert, M. W., Xiao, N., Chen, T. J. & Karr, J. R. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics* **24**, 2044-2050, doi:10.1093/bioinformatics/btn352 (2008).

7 Min Lee, J., Gianchandani, E. P., Eddy, J. A. & Papin, J. A. Dynamic Analysis of Integrated Signaling, Metabolic, and Regulatory Networks. *PLoS Comput Biol* **4**, e1000086, doi:10.1371/journal.pcbi.1000086 (2008).

8 Cotten, C. & Reed, J. Mechanistic analysis of multi-omics datasets to generate kinetic parameters for constraint-based metabolic models. *BMC Bioinformatics* **14**, 32 (2013).

9 McCloskey, D., Palsson, B. Ø. & Feist, A. M. Basic and applied uses of genome‐scale metabolic network reconstructions of Escherichia coli. *Mol Sys Biol* **9**, 661, doi:10.1038/msb.2013.18 (2013).

10 Henry, C. S., Broadbelt, L. J. & Hatzimanikatis, V. Thermodynamics-Based Metabolic Flux Analysis. *Biophysical J* **92**, 1792-1805, doi:10.1529/biophysj.106.093138 (2007).

11 Kümmel, A., Panke, S. & Heinemann, M. Putative regulatory sites unraveled by network – embedded thermodynamic analysis of metabolome data. *Mol Sys Biol* **2**, 2006.0034, doi:10.1038/msb4100074 (2006).

12 Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. & Reuss, M. Dynamic modeling of the central carbon metabolism of Escherichia coli. *Biotechnol Bioeng* **79**, 53-73, doi:10.1002/bit.10288 (2002).

13 Gutenkunst, R. N. *et al.* Universally Sloppy Parameter Sensitivities in Systems Biology Models. *PLoS Computat Biol* **3**, e189, doi:10.1371/journal.pcbi.0030189 (2007).

14 van Eunen, K. *et al.* Measuring enzyme activities under standardized in vivo-like conditions for systems biology. *FEBS J* **277**, 749-760, doi:10.1111/j.1742-4658.2009.07524.x (2010).

15 Mahadevan, R., Edwards, J. S. & Doyle, F. J. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophysical J* **83**, 1331-1340 (2002).

16 Voit, E. O., Goel, G., Chou, I. C. & Fonseca, L. L. Estimation of metabolic pathway systems from different data sources. *IET Syst Biol* **3**, 513-522 (2009). <http://digital-library.theiet.org/content/journals/10.1049/iet-syb.2008.0180>.

17 Chou, I.-C. & Voit, E. Estimation of dynamic flux profiles from metabolic time series data. *BMC Syst Biol* **6**, 84, doi:10.1186/1752-0509-6-84 (2012).

18 Goel, G., Chou, I.-C. & Voit, E. O. System estimation from metabolic time-series data. *Bioinformatics* **24**, 2505-2511, doi:10.1093/bioinformatics/btn470 (2008).

19 Ishii, N., Nakayama, Y. & Tomita, M. Distinguishing enzymes using metabolome data for the hybrid dynamic/static method. *Theor Biol Med Model* **4**, 19, doi:10.1186/1742-4682-4-19 (2007).

20 Yugi, K., Nakayama, Y., Kinoshita, A. & Tomita, M. Hybrid dynamic/static method for large-scale simulation of metabolism. *Theor Biol Med Model.* **2**, 42, doi:10.1186/1742-4682-2-42 (2005).

21 Chechik, G. & Koller, D. Timing of Gene Expression Responses to Environmental Changes. *J Comput Biol* **16**, 279-290, doi:10.1089/cmb.2008.13TT (2009).

22 Sivriver, J., Habib, N. & Friedman, N. An integrative clustering and modeling algorithm for dynamical gene expression data. *Bioinformatics* **27**, i392-i400, doi:10.1093/bioinformatics/btr250 (2011).

23 Hynne, F., Danø, S. & Sørensen, P. G. Full-scale model of glycolysis in Saccharomyces cerevisiae. *Biophys Chem* **94**, 121-163, doi:http://dx.doi.org/10.1016/S0301-4622(01)00229-0 (2001).

24 Le Novère, N. *et al.* BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res* **34**, D689-D691, doi:10.1093/nar/gkj092 (2006).

25 Hoops, S. *et al.* COPASI—a COmplex PAthway SImulator. *Bioinformatics* **22**, 3067-3074, doi:10.1093/bioinformatics/btl485 (2006).

26 Steel, R. G. D. & Torrie, J. H. *Principles and procedures of statistics, with special reference to the biological sciences*. (McGraw-Hill, 1960).

27 Evgeniou, T., Poggio, T., Pontil, M. & Verri, A. Regularization and statistical learning theory for data analysis. *Comput. Stat. Data Anal.* **38**, 421-432, doi:10.1016/s0167-9473(01)00069-x (2002).

Chapter 3: LK-DFBA: A Linear Programming-based modeling strategy for capturing dynamics and metabolite-dependent regulation in metabolism

3.1 Introduction

Metabolism constitutes the supply chain for all other cellular processes, such as DNA replication, transcription of RNA, and protein synthesis. It is perhaps the most immediate readout available of cellular state. An increasing focus on systems-level behavior in cellular biology coupled with the development of appropriate chemical analyses to enable such work has led to the field of metabolomics, which studies metabolism at the genome scale[1]. Metabolomics is a growing field, and the challenges associated with performing the chemical analyses and data processing necessary to harness this data are steadily improving[2,3].

As a direct readout of the state of cellular metabolism, metabolomics is a natural complement to efforts in metabolic engineering, in which an organism is genetically engineered to facilitate the overproduction of a target small molecule[4]. The diversity of chemistry in metabolism far outstrips that of proteins (polypeptides) and DNA (nucleic acids), and many of these molecules have known or potential commercial or clinical value. Some of these, such as ethanol, are known products or byproducts of primary metabolism in commonly used model organisms and can be produced in more useful quantities by careful co-

opting the metabolic machinery[5]. Other metabolites, such as many pharmaceutical precursors, derive from secondary metabolism in organisms that may be difficult to culture and can be produced in a more cost-efficient manner by exporting the corresponding metabolic pathway into a more amenable host, such as *Bacillus subtilis*, *Escherichia coli* or *Saccharomyces cerevisiae*[6].

Given how tightly tied metabolism is to so many other cellular processes and the inherent toxicity of some metabolites (which are necessary intermediates), metabolic reactions are highly connected and tightly regulated[7,8]. These regulatory effects can range from long-term changes due to transcriptional regulation of enzyme expression to short-term rapidly reversible direct regulation of enzyme reaction rates (metabolic fluxes) via allosteric mechanisms[8,9]. Genetic engineering strategies may target any of these mechanisms, but a proposed intervention derived from reductionist reasoning may lead to unintended side effects that may undermine the desired engineering goal[10]. Metabolic modeling and computational strain design tools are effective methods of anticipating these side effects, allowing metabolic engineers to more strategically allocate the significant time and resources required to produce a desired strain in the lab.

The primary methods for metabolic engineering strain modeling are constraint-based models (CBMs), of which Flux Balance Analysis (FBA) is the prototypical example[11,12]. In FBA, a stoichiometric model of metabolic reactions is combined

with a steady-state assumption, restrictions on rates of enzyme reversibility and saturation, and an objective describing the cell's preferred behavior; taken together, these specify a linear program (LP)[11]. From this, an optimal metabolic flux distribution can be calculated with relatively few data requirements.

The general class of CBMs builds on this framework through various strategies. These range from modifying the objective function (MOMA)[13], to adding further constraints on the space of valid flux distributions[14-17], to leveraging mathematical properties of LPs[10]. Due to its simplicity and the range of potential modifications, FBA has been the basis for a whole host of tools for strain design, such as OptKnock[10] and its derivatives[18-22]. A great amount of work has gone into genome-scale model reconstructions of many organisms critical to metabolic engineering as a result.

However, FBA was developed well before the advent of metabolomics, and some of its core assumptions preclude directly integrating metabolomics data into the model. This primarily arises from the steady-state assumption: decoupling the flux distribution from any consideration of metabolite concentrations produces a convenient linear system in terms of the fluxes and network stoichiometry, but comes at the expense of any consideration of metabolite concentrations. While this assumption may be valid for certain cell types under specific conditions such as exponential growth or chemostat culture, in general metabolite concentrations

will vary[23]. This is an important consideration in batch culture, where extracellular concentrations vary throughout and the state of the organism changes significantly as is evidenced by phenomena such as lag-phase and growth saturations[24].

This loss of metabolite representation also makes it more difficult to incorporate metabolite-dependent regulation directly into the model, especially if dynamics are to be accounted for. Several methods do exist to use some data for modeling regulation. Regulatory FBA (rFBA) uses transcriptome data to modulate metabolic fluxes, adjusting fluxes relative to changes in gene expression; a similar approach could be used with metabolite levels[25]. A similar approach, kinetic FBA (KFBA), uses metabolite concentrations with non-linear kinetic rate laws to constrain flux values[26]. However, these models are still limited to steady-state flux distributions, and only apply to the experimental conditions reflected in the data used for a particular calculation.

Approaches that have made more effort to account for dynamics and regulation have taken a few routes. Systems of Ordinary Differential Equations describing kinetic rate laws and mass balance equations are a well-established alternative to CBMs, but require integration of systems of equations that are highly non-linear and may represent processes at drastically different time-scales, leading to issues such as model stiffness[27-29]. In addition, large numbers of parameters are

required to construct models of even a modest scale[30]. This requires either extensive experimental effort to identify and construct appropriate kinetic rate laws if the model is to be constructed from the bottom-up, or else expensive global parameter searches with extensive time course data if the model is to be constructed top-down (a process made much more difficult by the non-linear dynamics of the kinetics rate law equations, leading to issues with parameter identifiability and error compensation)[31-36].

Much work has been done to tackle these challenges. A particularly relevant example is the Dynamic Flux Estimation (DFE) procedure described by Goel *et al.*, in which metabolite time courses are used to generate estimates of the mass balance derivatives, and subsequently a dynamic flux distribution[32]. Specific kinetic rate laws can then be fit to the appropriate combinations of flux and metabolite data from the time courses, using regression to independently solve the decoupled equations. Goel *et al*. demonstrated this with generalized mass action kinetic rate laws, but any functional form can be fit this way[32]. However, these models still require substantial additional work to integrate them into the many strain design tools built around CBMs.

Alternatively, adhering to the CBM paradigm, Mahadevan *et al.* developed Dynamic Flux Balance Analysis (DFBA)[37], an extension of FBA which discards the steady state assumption and adds non-linear constraints, such as those

describing kinetic rate laws[37]. The addition of non-linear constraints converts the LP into a non-linear program (NLP), which can be solved either over a series of independent intervals, or as a single top-down discretization. This approach allows much of the flexibility of ODE models and regulation, while retaining the basic philosophy behind CBMs.

However, because of the many non-linear constraints present in DFBA, FBA's most attractive mathematical properties are lost. The most critical of these derive from FBA's formulation as an LP: linear programs are a well-understood convex optimization (which allows results from Duality Theory to place guarantees on solution optimality and to provide insight into the solution properties) and are incredibly efficient to solve. The popular OptKnock strain development tool takes advantage of LP duality to guarantee optimality of the FBA problem while simultaneously performing an optimization on the engineering objective in question; LP duality allows the bi-level optimization to be recast as a single level optimization by incorporating constraints derived from the dual of the FBA primal problem[10].

In this work, we modified the DFBA formulation with the goal of producing a system of equations describing metabolite dynamics and regulation, but without the non-linear equations that are incompatible with an LP formulation. In this approach, metabolite stoichiometry and difference equations describing changes

in concentration are tied together by representing the metabolite accumulations term from the mass balances as part of the flux distribution (this concept is described in iFBA as “pooling fluxes” or by Karr *et al.* as “internal transport fluxes”)[30,38]. Kinetics and regulation are approximated as a set of linear equations specifying upper bounds on flux values, rather than as potentially complicated non-linear equations. As in DFBA, these equations are applied over the discretized simulation interval to produce a completely linear system of constraint equations, which are combined with the other elements of FBA to perform the usual LP optimization[37]. The result, which we call Linear Kinetics-Dynamic Flux Balance Analysis or LK-DFBA, is a system that combines many of the main advantages of FBA and DFBA, and can be directly combined with any of the strain design tools that accept FBA as input.

The lynchpin of the LK-DFBA framework is the addition of linear kinetics constraints in conjunction with pooling fluxes. On their own, pooling fluxes are not sufficient to induce biologically relevant behavior, and other information (kinetic rate law equality[30] or inequality[37] constraints; connected biological process modules[38]) must be included to incentivize accumulation and depletion.

One example of such an incentivization is the diurnal-FBA approach of Knies *et al.,* which represented day and night phases in the photosynthetic algae *Emiliania huxleyi* as two compartments of a combined metabolic model[39]. The

day module converted photons to biomass and a few storage metabolites, which were the only source available to supply biomass maintenance during the night module. However, the steady-state assumption was applied over the whole of each module, and the metabolites used for storage were only modeled as transport fluxes between the two. Modeling dynamics and regulation was beyond the intended scope of this work.

While the idea of linearized regulation has been implemented before in a CBM of intracellular signal transduction, this approach ignored concentrations and presumed steady-state behavior[40]. The resulting linear constraints simply constrained certain flux tradeoffs. Here, we combine both of these elements to both permit and incentivize metabolite dynamics in what is still an LP formulation.

These constraints have a direct impact on model dynamics, and as a result model performance depends heavily on successfully identifying appropriate parameters to describe them. Thus, a critical step to modeling correct time course dynamics is parameter optimization. We explored this in two model systems. The first model is a small *in silico* model of a Branched Pathway from Biochemical Systems Theory (BST), and the second is a model of glycolysis and pentose phosphate pathway in *E. coli*[41,42]. For each model, we generated reference time course data, and then created multiple noisy synthetic time course

data replicates by varying both the data sampling frequency and the coefficient of variance (CoV) of added noise.

We used DFE and global optimization strategies to fit model parameters for our LK-DFBA framework and compared the resulting simulations against ODE-based frameworks that use BST power-law kinetics and Michaelis-Menten rate laws. We found that LK-DFBA is able to capture the behavior of the original model systems and that for the Branched Pathway model, it is able to outperform the BST-based comparator under the conditions most relevant to metabolomics data. In the larger *E. coli* model, we explored the challenges associated with model scale-up and with structural features not present in the Branched Pathway model. We also addressed the tradeoff between two strategies for parameterizing branch points, finding that in this case the more heavily parameterized version improved performance sufficiently to justify its use.

3.2. Materials and Methods

3.2.1 Simulating regulated metabolite dynamics as a linear program

3.2.1.1 Model input

We describe here the implementation of our modified form of DFBA. It takes as input two sets of information. The first set comprises the constraints and objective from FBA: a stoichiometric matrix describing the relationship between metabolites and fluxes in the model, a set of upper and lower bounds on

metabolic fluxes, and an objective function specifying the flux(es) the network tries to locally maximize or minimize. To these, we add metabolite concentration initial conditions, a time interval, a parameter describing the number of segments into which the simulation interval is to be evenly divided, and a list of regulatory interactions (and the corresponding parameters to describe them) are required.

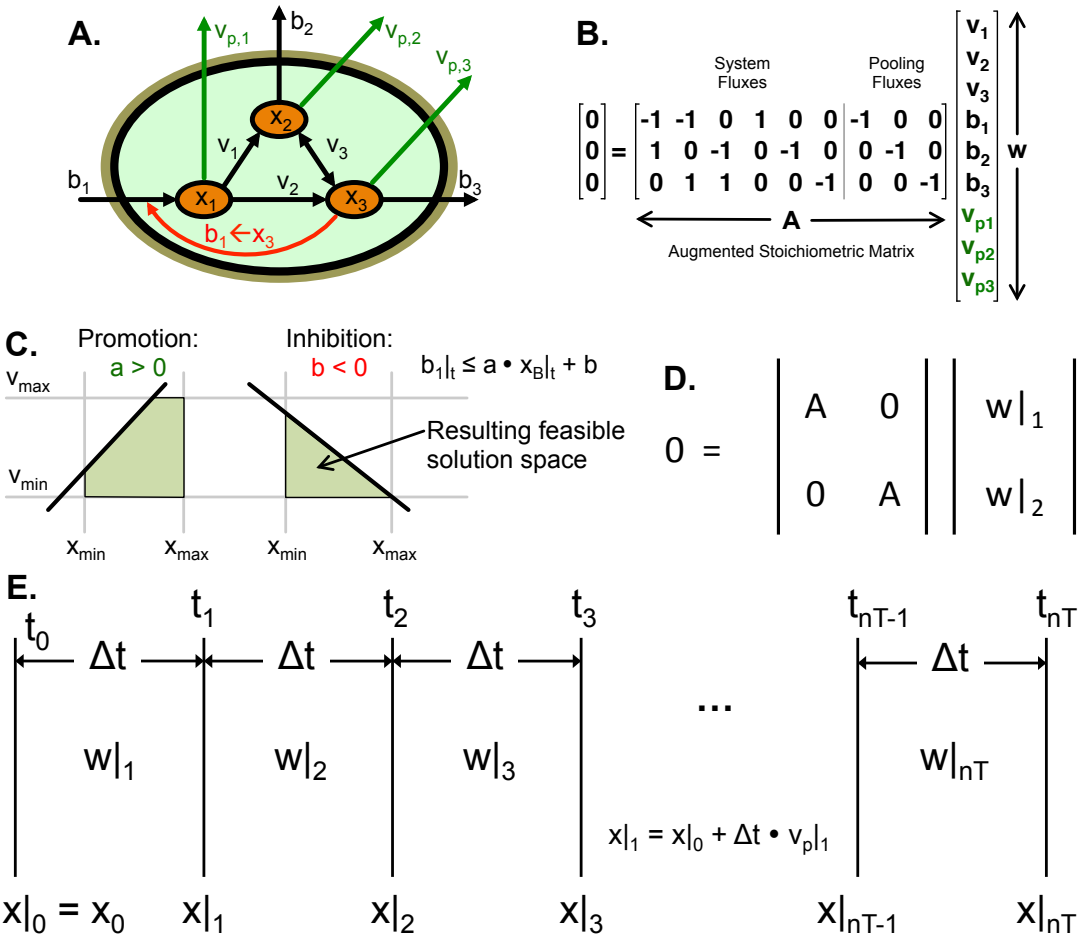

Figure 3.1. A graphical depiction of the LK-DFBA modeling framework
A. Pooling fluxes are added to capture model[43] accumulation
B. Model stoichiometry is adjusted to include pooling fluxes
C. Linear constraints describe positive and negative regulation dependent on metabolite concentrations
D. Constraints can be systematically generated as templates applied across time steps
E. The time course is discretized and accumulation integrated over time steps

3.2.1.2 Discretizing the time interval

Following the basic template of the Dynamic Optimization Approach (DOA) of DFBA[37], the simulation interval is divided into nT segments, as shown in Figure 3.1A. By our convention, metabolite concentrations are represented at the time points separating the intervals, and fluxes are represented over the interval between time points. Initial conditions for metabolite concentrations specify the concentrations at the time point prior to the first interval.

3.2.1.3 Stoichiometry and pooling fluxes

The mass balances on a set of n_m metabolites, $\vec{x}$, can be represented by the system of equations

$$\frac{d\vec{x}}{dt} = \boldsymbol{S}\vec{v}$$

where $\frac{d\vec{x}}{dt}$ is the accumulation or depletion of metabolites in the system, $\boldsymbol{S}$ is the stoichiometric matrix describing the connectivity of n_m metabolites and n_v fluxes in the metabolic network, and $\vec{v}$ is a vector of the n_v enzymatic rates through the metabolic network, i.e. the flux distribution.

In FBA, $\frac{d\vec{x}}{dt}$ is assumed to be zero, and the linear equation

$$0 = \boldsymbol{S}\vec{v}$$

applies. Combined with a linear objective function

$$z = \vec{c}^T\vec{v}$$

where c_i specifies the weight of flux v_i in the objective (e.g. to maximize growth rate, set so that $c_{biomass} = 1$ and all other $c_i = 0$) and bounds

$$\vec{v}_{LB} \leq \vec{v} \leq \vec{v}_{UB},$$

an LP can be specified as

$$\max_{\vec{v}} \; z \; = \; \vec{c}^T \vec{v}$$

$$s.t. \; 0 \; = \; \boldsymbol{S}\vec{v}$$

$$\vec{v}_{LB} \leq \vec{v} \leq \vec{v}_{UB}.$$

In LK-DFBA, we relax the steady state assumption, working from

$$\frac{d\vec{x}}{dt} \; = \; \boldsymbol{S}\vec{v}$$

Moving the $\frac{d\vec{x}}{dt}$ term to the right side and adding it to the solution vector, gives us the system shown in Figure 3.1B,

$$0 \; = \; \boldsymbol{A}\vec{w} = [\boldsymbol{S} \quad -\boldsymbol{I}] \begin{bmatrix} \vec{v} \\ \vec{v}_p \end{bmatrix}$$

where $\boldsymbol{A}$ is the (n_m × (n_m+n_v)) augmented stoichiometric matrix and $\vec{w}$ is the ((n_m+n_v) × 1) augmented flux vector, and using the “pooling flux” nomenclature of Covert *et al.* in iFBA,

$$\vec{v}_p \; = \; \frac{d\vec{x}}{dt}$$

i.e., we will describe $\frac{dx_i}{dt}$ as the pooling flux for metabolite x_i, $\vec{v}_{p,i}$. This augmented stoichiometric constraint will apply over each segment of the discretized interval, producing a set of $n_m \cdot nT$ constraint equations,

$$0 = \boldsymbol{A}\vec{w}(t_k)$$

where $\vec{w}(t_k)$ is the augmented flux vector $\vec{w}$ evaluated at the interval ending at t_k.

3.2.1.4 Difference Equations

Concentrations are explicitly represented in the model, and metabolite dynamics are incorporated by integrating metabolite concentrations over each interval using a difference equation and the corresponding pooling flux term (i.e. the $\frac{dx_i}{dt}$ term)

$$x_i(t_{k+1}) = x_i(t_k) + \Delta t \cdot v_{p,i}(t_k)$$

to produce a series of $n_m \cdot nT$ constraint equations, as shown in Figure 3.1E.

3.2.1.5 The Solution Vector

Combining the augmented flux vector over each time segment and the concentrations at each time point, the final solution vector for the LP is constructed as

$$\omega = [\vec{w}^T(t_1), \vec{w}^T(t_2), \ldots \vec{w}^T(t_{nT-1}), \vec{w}^T(t_{nT}), \vec{x}^T(t_0), \vec{x}^T(t_1), \ldots \vec{x}^T(t_{nT-1}), \vec{x}^T(t_{nT})]^T$$

and is of dimension $((n_v + n_m) \cdot nT + n_m \cdot (nT + 1) \times 1)$.

3.2.1.6 Constant constraints on Concentration and Flux values

As in FBA, lower and upper bounds on system fluxes are provided and apply to the flux distribution at each interval. Typically the upper and lower bound constraints on internal system fluxes are expected to be inactive and are set at a

large nominal value to guarantee the space is bounded. Pooling fluxes are given nominal bounds as well; due to limitations on the product $\Delta t \cdot v_p$ combined with constraints on concentrations and elsewhere in the system, it is expected that the nominal bounds will not act as active constraints.

Like bounds on flux values, constraints bounding concentrations can be set by the user, but generally it is expected that concentrations are strictly positive. If a concentration is know to be at a fixed quantity, the upper and lower bound can be set accordingly.

The initial conditions $\vec{x}_0$ are specified by setting $\vec{x}(t_0) = \vec{x}_0$.

3.2.1.7 Linearized Kinetics Constraints

The key feature of LK-DFBA is the addition of linear equations to describe constraints in which fluxes are controlled by metabolites, as is the case in circumstances ranging from mass action kinetics to allosteric regulation (on short time scales) or transcriptional regulation (on longer time scales). Any dependence of flux on metabolite concentrations is implemented in this manner, and this turns out to be a critical requirement for incentivizing relevant dynamics in the model.

These constraints are specified by a list of n_r mappings. Corresponding to each mapping is a pair of parameters (a, b) such that for mapping n between "controller" metabolites $\{x\}_n$ and "target" fluxes $\{v\}_n$,

$$\sum_i v_{i,n}(t_{k+1}) \leq a_n\left(\sum_j x_{j,n}(t_k)\right) + b_n,$$

where $v_{i,n}$ is a target flux in $\{v\}_n$, $x_{j,n}$ is a controller metabolite in $\{x\}_n$, and (a_n, b_n) are the parameters describing the linear kinetics constraint. When $a_n > 0$, this interaction produces a promotional effect, and when $a_n < 0$, this interaction has an inhibitory effect. Applied over the whole discretized time course, this produces a total of $n_r \cdot nT$ kinetics constraint equations. Examples of these mappings are shown in Figure 3.1C.

These constraints allow us to not only represent interactions such as allosteric regulation, but also to linearly approximate the dependence of enzyme activity on its substrate concentration. Consider the case of the positive regulator shown in Figure 3.1C: we note that the profile produced by simultaneously considering the effect of the constant flux bounds constraints ("v_{max}") in conjunction with the constraint produced by mapping the enzyme substrate as a "regulator" of enzyme flux $v \leq a \cdot x + b$ is comparable to the flux vs concentration profile of a simple Michaelis-Menten reaction mechanism. We refer to these types of "kinetics" constraints as "mass action" constraints, to differentiate them from "regulatory" constraints produced through mechanisms such as allostery. Our code includes a procedure to automatically generate mass action kinetics constraints from a

stoichiometric matrix, giving the user the convenience of only needing to manually specify the regulatory constraints.

We note that these regulatory interactions are implemented as bounds on the controlled fluxes, rather than as equality constraints: this is a very fundamental difference from the behavior of ODE models, in which each (non-linear) equation reduces the dimension of the solution space. For linear equations, this would often create cases in which the system of equations would produce negative concentrations, for example by forcing an efflux term to exceed an influx term when a metabolite was already depleted. However, in LK-DFBA, this just leads to a situation in which the kinetic constraint is no longer active, and the other model constrains preclude blatantly unphysical behavior. As a result, LK-DFBA has a degree of both simplicity and flexibility, which comes with advantages and disadvantages that we will explore in more depth in the Results section.

In a model with n_r regulatory constraints, each regulatory constraint adds two parameters (a, b) to the model, for a total of $(2 \cdot n_r)$ parameters. In many cases, a single controller is paired with a single target, but certain cases may allow lumping multiple species together, allowing a reduction in the number of model parameters. For example, one might reduce the number of model parameters by choosing to constrain only the sum of the effluxes from a given metabolite, such as at branch point, rather than to introduce separate constraints for each of the

individual fluxes. Our Results and Discussion in Section 3.3.5 includes an assessment of the tradeoffs between these two options to determine if this consolidation represents an improvement or an oversimplification.

To run a given simulation, the set of (a, b) parameter values is provided along with the map of controllers and targets. In practice, this will need to be determined via parameter fitting, as the linear equations in general are simplified approximations that do not directly correspond to intrinsic physical quantities. We consider this question in-depth in subsequent sections, where we provide several methods for determining these parameters and comment on their effectiveness and practicality.

3.2.1.8 Model Objective

As in FBA, LK-DFBA requires an objective. While there are several ways in which to construct this objective[37] we found that an objective that applied to the fluxes weighed equally at each time point (an "instantaneous" objective) was effective in producing stable, robust behavior. This style of objective function can be generated easily by expanding the original FBA objective to apply over each interval. We also tested an alternate "terminal" objective function (in which the objective function only looks at the concentration of a selected final time point), and found that this often led to degenerate solutions, erratic behavior, and

numerical artifacts at intermediate time points. We discuss this more in the Results section 3.3.1.

We also found it effective to add a modest penalty to the L_2 norm of the solution vector during optimization, changing the problem objective to

$$z = \vec{c}^T\vec{\omega} + \vec{\omega}^T\boldsymbol{Q}\vec{\omega}$$

where

$$\boldsymbol{Q} = -\lambda\boldsymbol{I}$$

And λ is a small penalty on the solution norm. This has the effect of imposing a parsimony preference on the solution vector $\vec{\omega}$, which has been shown to be an effective and reasonable strategy, and helped resolve some occasional observed issues with solution degeneracy[44]. While the resulting problem is technically now a (linearly-constrained) quadratic program, we observed no appreciable increase in solution time, and this particular case still specifies a convex optimization (and therefore preserves the very desirable Strong Duality features of the LP). The option to instead use the linear objective remains, but our implementation defaults to the QP formulation.

3.2.1.9 The LK-DFBA Optimization Problem

Assembling the constraints described in the previous sections produces the following linearly-constrained QP for simulating metabolic time courses which we refer to as LK-DFBA. For

$$\vec{\omega} = [\vec{w}^T(t_1), \vec{w}^T(t_2), \ldots \vec{w}^T(t_{nT-1}), \vec{w}^T(t_{nT}), \vec{x}^T(t_0), \vec{x}^T(t_1), \ldots \vec{x}^T(t_{nT-1}), \vec{x}^T(t_{nT})]^T,$$

$$\max_{\vec{\omega}} \; z = \vec{c}^T\vec{\omega} - \lambda\vec{\omega}^T\vec{\omega}$$

$$\text{s.t.} \quad 0 = \boldsymbol{A}\vec{w}(t_k) \qquad \forall\, k \in [1, nT]$$

$$\vec{w}_{LB} \le \vec{w}(t_k) \le \vec{w}_{UB} \qquad \forall\, k \in [1, nT]$$

$$\vec{x}_{LB} \le \vec{x}(t_k) \le \vec{x}_{UB} \qquad \forall\, k \in [1, nT]$$

$$\vec{x}(t_0) = \vec{x}_0$$

$$x_i(t_k) = x_i(t_{k-1}) + \Delta t \cdot v_{p,i}(t_k) \qquad \forall\, k \in [1, nT]$$

$$\sum_i v_{i,n}(t_{k+1}) \le b_n + a_n \sum_j x_{j,n}(t_k)$$

$$\forall\, k \in (1, nT), \forall\, i \in \{v\}_n, \forall\, j \in \{x\}_n, \forall\, n \in (1, n_r)$$

3.2.2 Model generation codes

We developed a procedure in MATLAB to automatically translate a standard FBA model into an LK-DFBA model, and then solve the resulting optimization problem. This procedure works by taking as input the original FBA model, plus the inputs specified above. This code has been made publically available on GitHub at https://github.com/gtStyLab/lk-dfba.

3.2.3 Test Models

To test our modeling and parameter fitting strategies, we used several models to produce “synthetic” datasets. The advantage of using these datasets as a point of comparison is that it allows us to produce idealized cases under which we can study the theoretical performance and limitations of our modeling strategy without being limited by practical concerns such as data sampling frequency, signal-to-

noise ratio in the data, or limits in the cross-section of metabolites we can measure.

3.2.3.1 The Branched Pathway Model

Our first test model is a modified version of a popular, well-established *in silico* model from BST describing a simple branched pathway with both positive and negative regulatory interactions[41]. Our modified version introduces several changes, and is shown in Figure 3.2.

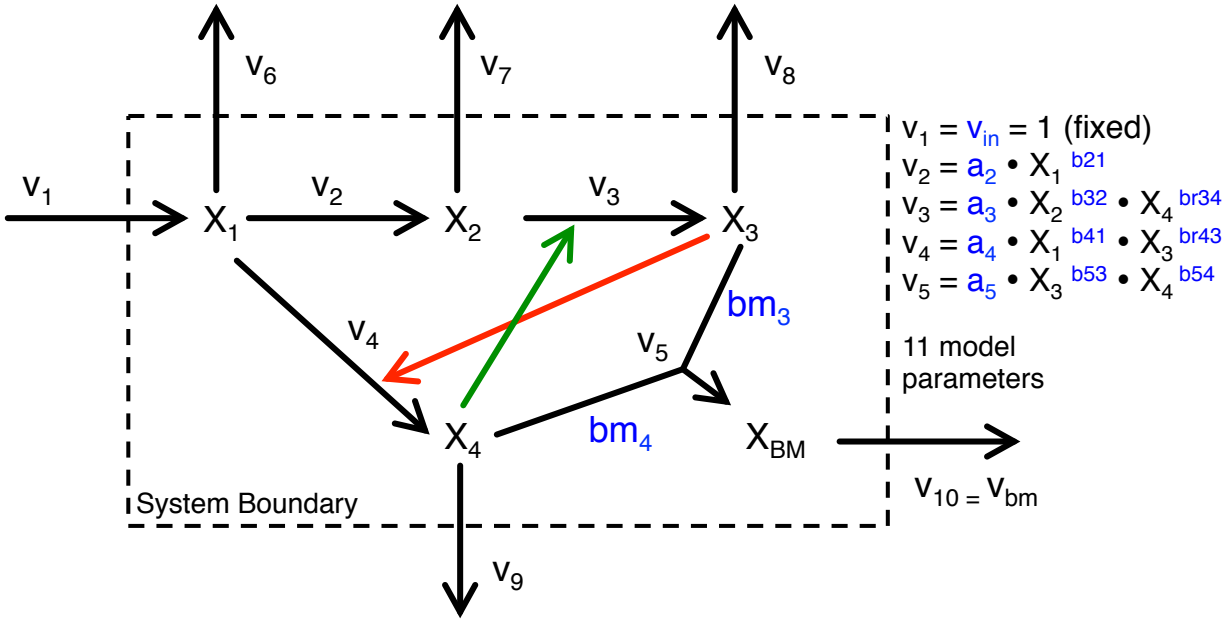

Figure 3.2. The modified Branched Pathway model used in this work, adapted from the model of Almeida *et al*.
Black arrows indicate fluxes. The green and red arrows denote positive and negative regulatory interactions. The dashed line denotes the system boundary. Metabolites are X_1, X_2, X_3, X_4, and X_{BM}. System fluxes are v_1, v_2, v_3, v_4, and v_5. Pooling fluxes for X_1, X_2, X_3, X_4, and X_{BM} are v_6, v_7, v_8, v_9, and v_{BM}, respectively. The parameters in blue specify reaction rates and stoichiometry. Not shown are initial conditions. Kinetic rate laws are implemented as generalized mass action (GMA) rate laws from BST.

First, we replaced the two effluxes in the original model with a single, fixed-stoichiometry “biomass” reaction, which produces a biomass “metabolite” subject to a mass balance equation. This introduces some additional biological relevance (such reactions are ubiquitous in genome-scale models) and allows us to define a clear objective for the system.

Second, we modified the two regulatory interactions to change their targets. The negative regulatory target of X_3 is changed from the input flux v_1 to the opposite branch’s first flux v_4, and we hold the input flux at a constant value of v_0. This allows us to simplify the model, while still producing interesting dynamics for X_1 via interactions with the branch fluxes v_2 and v_4. Second, we change the positive regulatory interaction of X_4 to target v_3, the first flux in the other branch, to avoid introducing a parameter identifiability problem by having the same metabolite acting as a controller for the same flux, v_5, twice (i.e. both as a mass action and regulator constraint).

Like the original BST model, we implement power-law kinetics, as shown in the equations of Figure 3.2. We produce several noiseless datasets by modifying the initial conditions, biomass equation stoichiometry, and kinetic rate constants; the conditions for these models are shown in Table 3.1.

Table 3.1. Parameters used to generate noise-free Branched Pathway data sets

	Stoichiometry		Kinetics											Initial Conditions				
k	bm_3	bm_4	a_2	b_{21}	a_3	b_{32}	b_{r34}	a_4	b_{41}	b_{r43}	a_5	b_{53}	b_{54}	X_1	X_2	X_3	X_4	X_5
1	0.6	0.4	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	0.1	0.2	0.3	0.4	0.5
2	0.6	0.4	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	0.1	0.1	0.1	0.1	0.1
3	0.6	0.4	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
4	0.6	0.4	0.22	0.925	0.691	0.856	0.302	0.416	0.107	-0.564	0.436	0.816	0.52	1.0	1.0	1.0	1.0	1.0
5	0.6	0.4	0.935	0.457	0.24	0.763	0.759	0.74	0.743	-0.106	0.681	0.463	0.212	1.0	1.0	1.0	1.0	1.0
6	0.6	0.4	0.52	0.725	0.791	0.656	0.402	0.816	0.807	-0.364	0.936	0.616	0.82	1.0	1.0	1.0	1.0	1.0
7	0.6	0.4	0.679	0.036	0.809	0.748	0.12	0.525	0.325	-0.546	0.398	0.415	0.18	1.0	1.0	1.0	1.0	1.0
8	0.9	0.1	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
9	0.8	0.2	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
10	0.7	0.3	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
11	0.5	0.5	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
12	0.4	0.6	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
13	0.3	0.7	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
14	0.2	0.8	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0
15	0.1	0.9	0.8	0.5	1.0	0.8	0.2	0.5	0.4	-0.8	0.5	0.5	0.8	1.0	1.0	1.0	1.0	1.0

3.2.3.2 Glycolysis and Pentose Phosphate Pathway in E. coli

While the Branched pathway model is convenient as an initial test case, it lacks physiological significance and is too simple to capture some of the challenges we expect in real metabolic networks. To explore initial scale up and to better gauge the challenges of implementing LK-DFBA, we test a model of central carbon metabolism in *E. coli*, specifically encapsulating glycolysis and the pentose phosphate pathway (PPP)[42]. The network structure is shown in Figure 3.3, and a list of model abbreviations in Tables 3.2 and 3.3. The model is a system of ODEs with empirically derived rate laws. Metabolite concentrations were generated using the procedure described in Chapter 2. Briefly, noiseless data at high resolution was generated from the default model initial conditions and parameters in Copasi 4.14 (Build 89), with the exception that moieties such as ATP, ADP, and NADH, etc. were held at constant concentrations during simulation[42,45-48].

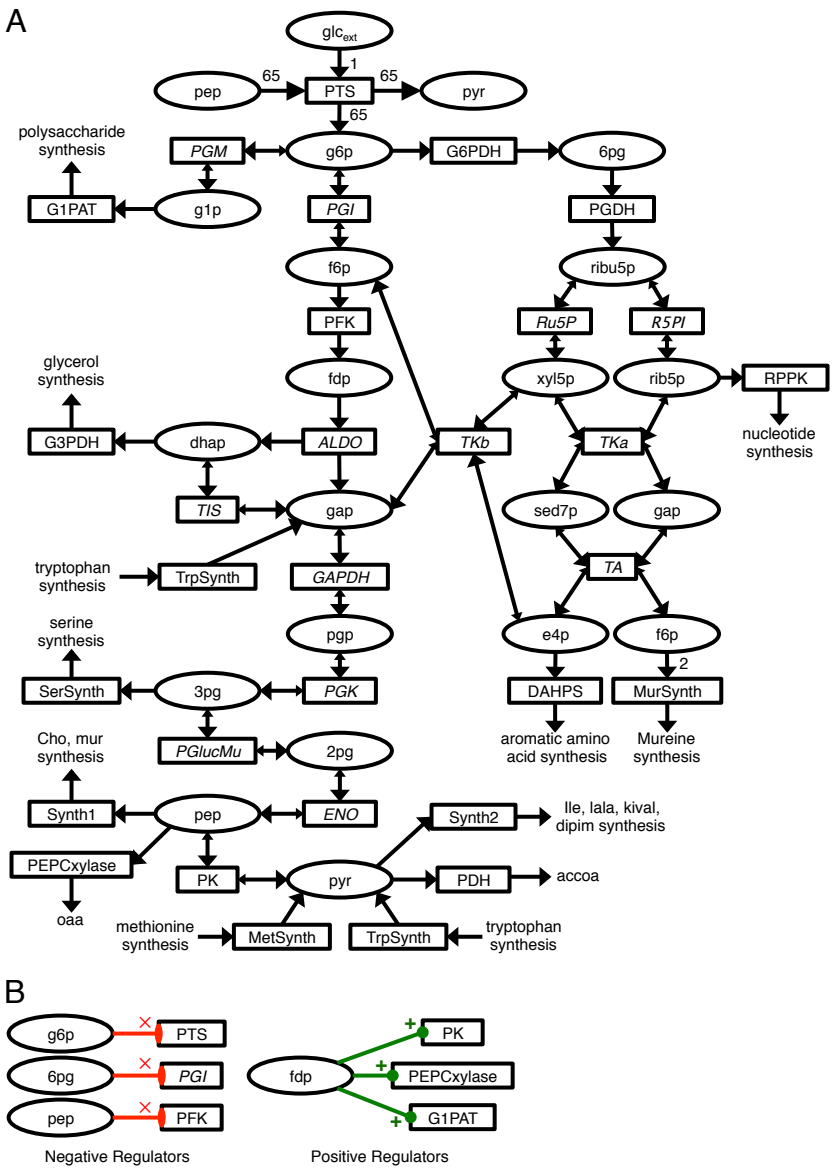

Figure 3.3. The model of E. coli central carbon metabolism
Adapted from Figure 2 of Chassagnole *et al.*[42] Abbreviations used in this figure are expanded in Tables 3.2 and 3.3.
A. The model includes glycolysis and the pentose phosphate pathway. Circles denote metabolites, and rectangles denote fluxes. Arrows with one head are irreversible reactions, and those with two heads are reversible; the larger head indicates the forward reaction direction. Numbers next to arrows denote non-unity stoichiometric coefficients. Not shown are degradation and dilution reactions.
B. The regulatory connections used in our implementation of the E. coli model. Red 'x' connections signify negative regulators, and green '+' connections represent positive regulators.

Table 3.2. Metabolite abbreviations used in the *E. coli* model

Index	Metabolite	Abbreviation
1	Extracellular Glucose	glc
2	Glucose-6-Phosphate	g6p
3	Fructose-6-Phosphate	f6p
4	Fructose-1,6-bisphosphate	fdp
5	Glyceraldehyde-3-Phosphate	gap
6	Dihydroxyacetonephosphate	dhap
7	1,3-diphosphosphoglycerate	pgp
8	3-Phosphoglycerate	3pg
9	2-Phosphoglycerate	2pg
10	Phosphoenol pyruvate	pep
11	Pyruvate	pyr
12	6-Phosphogluconate	6pg
13	Ribulose-5-phosphate	ribu5p
14	Xylulose-5-phosphate	xyl5p
15	sedoheptulose-7-phosphate	sed7p
16	Ribose-5-phosphate	rib5p
17	Erythrose-4-phosphate	e4p
18	Glucose-1-Phosphate	g1p

Table 3.3. Flux abbreviations used in the *E. coli* model

Index	Name	Abbreviation	Index	Name	Abbreviation
1	Extracellular glucose kinetics	glc_kin	25	Pyruvate dehydrogenase	PDH
2	Phosphotransferase system	PTS	26	Methionine synthesis	MetSynth
3	Glucose-6-phosphate isomerase	PGI	27	6-Phosphogluconate dehydrogenase	PGDH
4	Phosphoglucomutase	PGM	28	Ribose-phosphate isomerase	R5PI
5	Glucose-6-phosphate dehydrogenase	G6PDH	29	Ribulose-phosphate epimerase	Ru5p
6	Phosphofructokinase	PFK	30	Ribose phosphate pyrophosphokinase	RPPK
7	Transaldolase	TA	31	Glucose-1-phosphate adenyltransferase	G1PAT
8	Transketolase a	TKa	32	G6P degradation	g6p_deg
9	Transketolase b	TKb	33	F6P degradation	f6p_deg
10	Mureine synthesis	MurSynth	34	FDP degradation	fdp_deg
11	Aldolase	ALDO	35	GAP degradation	dhap_deg
12	Glyceraldehyde-3-phosphate dehydrogenase	GAPDH	36	DHAP degradation	dhap_deg
13	Triosephosphate isomerase	TIS	37	PGP degradation	pgp_deg
14	Tryptophan synthesis	TrpSynth	38	PG3 degradation	pg3_deg
15	Glycerol-3-phosphate dehydrogenase	G3PDH	39	PG2 degradation	pg2_deg
16	Phosphoglycerate kinase	PGK	40	PEP degradation	pep_deg
17	Serine synthesis	SerSynth	41	Pyruvate dilution	pyr_dil
18	Phosphoglycerate mutase	PGluMu	42	PG dilution	pg_dil
19	Enolase	ENO	43	Ribu5P dilution	ribu5p_dil
20	Pyruvate kinase	PK	44	XYL5P dilution	xyl5p_dil
21	PEP carboxylase	PEPCxylase	45	SED7P dilution	sed7p_dil
22	Synthesis 1	Synth1	46	Rib5P dilution	rib5p_dil
23	Synthesis 2	Synth2	47	E4P dilution	e4p_dil
24	DAHP synthesis	DAHPS	48	GLP dilution	g1p_dil

Reversible reactions were implemented by splitting them into two irreversible reactions representing the forward and reverse directions. For mass action constraints, the substrate of the original reaction was designated as the controller for the forward reaction, and the product was set as the controller for the reverse reaction. Fluxes such as Met and Trp synthesis, which are set to fixed values in the ODE model, were explicitly provided this information as well. We modeled the degradation reactions using equality constraints where the a parameter was known and assigned to the model based on the value in the underlying ODE model, and set $b = 0$ to produce a first-order kinetic rate law. By setting $b = 0$, we avoid creating the potential for our linear kinetics constraints to create conflicts with non-negative concentration constraints that would result in an infeasible LP.

3.2.4 Generating noise-added datasets

We generated datasets with different sampling frequencies and noise characteristics using the procedure described in section 2.2.3 of the previous chapter, allowing us to produce multiple replicates of noisy data with a specified sampling frequency and measurement noise[45].

Briefly, the noiseless data at high sampling frequency were down-sampled such that the initial conditions and nT additional time points are sampled evenly over the time interval of interest. Then, the metabolite or flux values are replaced with

a random value drawn from $N_{i,k} \sim (y_i(t_k), CoV \cdot y_i(t_k))$, where $y_i(t_k)$ is the value of species (metabolite or flux) i at time point k, and CoV is the coefficient of variance. We leave the initial time point at the original model values, and use it as unfitted input for the LK-DFBA simulation.

3.2.5 Parameter fitting

As the primary driver of system dynamics, the linear constraints play a key role in determining the performance of a model implemented in LK-DFBA. An effective and reliable method for determining appropriate parameters is critical. We pose the parameter-fitting problem as follows: given data describing a set of metabolite (and flux) time courses, determine the set of model parameters that minimize the weighted sum of squares error between the data and the time courses predicted by the model. We explored several strategies for addressing this problem.

For all methods, we assumed that the structure of the network and the regulatory interactions were known, including the signs of the interactions. In all cases, the true initial conditions (i.e. with no noise added) were provided for all metabolites.

3.2.5.1 Global Parameter Optimization

The most general strategy is a standard global optimization approach. We constructed a fitness function from the weighted sum-of-squares error (SSE) between the provided data and model predictions, subject to an L_2 regularization

penalty on the fitted parameters. The SSE weights are specified by the user, and can be used to reflect features such as differences in scale between metabolites, or heuristics to enable attempts to more effectively recapitulate the behavior of certain metabolites. These weights can potentially be applied to concentrations, fluxes, or pooling fluxes, but we only used weights on concentrations in our work. Our implementation also allows the user to specify a regularization weight and reference vector for the regularization penalty.

This SSE fitness function was used to fit FBA models for the methods based on global optimization. For the "Regression-Plus" method ('LR+'), we used the results of the Linear Regression ('LR') method (described below) as an initial starting point for the Nelder-Mead simplex solver using the MATLAB function `fmincon()`. The other method ('GA') used the `ga()` function in MATLAB to search using a genetic algorithm.

In the case of the genetic algorithm, we improved convergence of the algorithm by introducing constraints on the parameter search space to remove areas where we anticipated poor parameter sensitivity. These restrictions are described in Figure 3.4.

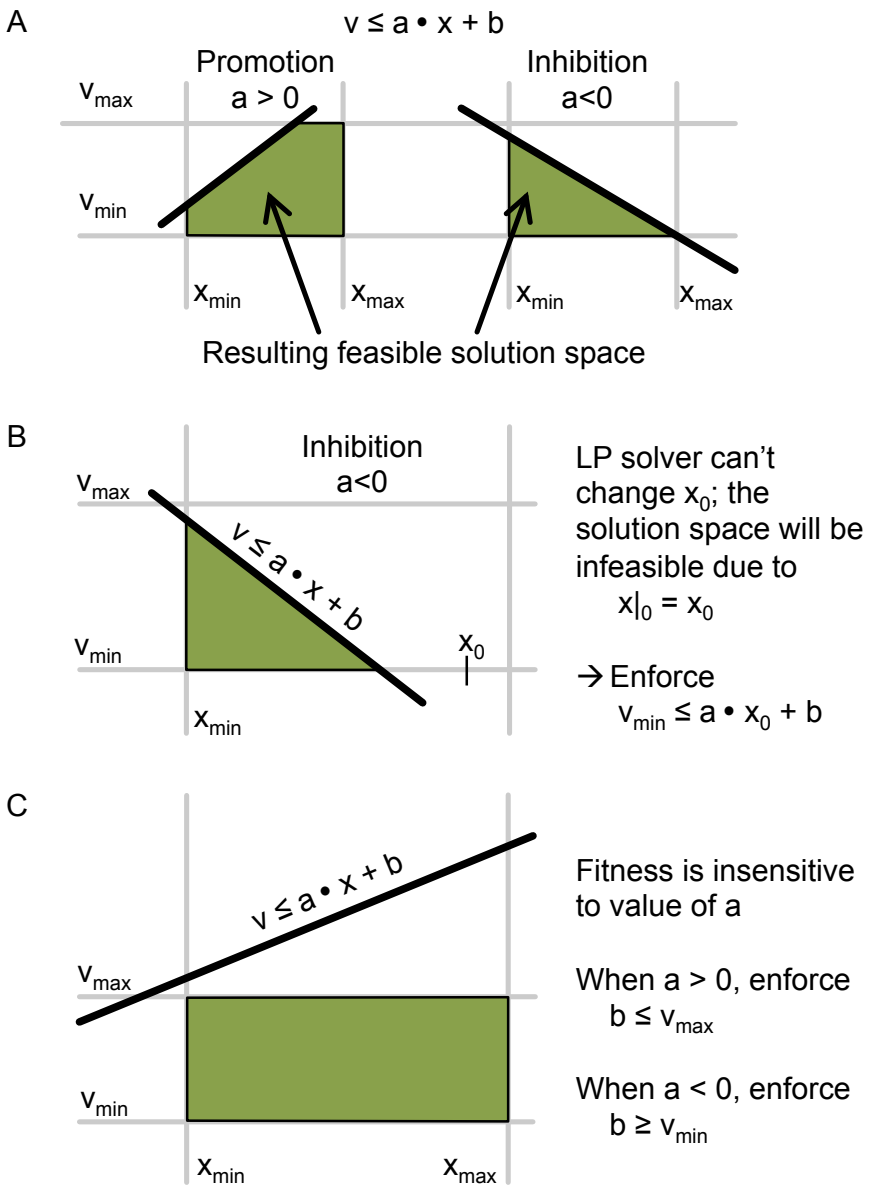

Figure 3.4. Bounding the parameter search space for the Genetic Algorithm

A. Kinetics constraints are formulated as linear inequalities. Also present are pre-specified bounds on flux values from FBA, a constraint on minimum metabolite concentrations to guarantee physically realistic concentration values, and an effective upper bound on concentration at a given time point due to mass balance constraints and the equations integrating pooling fluxes over the previous time points. This specifies a feasible solution space.

B. Under certain regions of parameter space when $a < 0$ (inhibition), the initial concentration of the metabolite, x_0, is outside the feasible space. This produces a conflicting constraint with the kinetics constraints, and the resulting LP is infeasible. We avoid this by restricting (a, b) such that $v_{min} < a \bullet x_0 + b$.

C. For certain regions of parameter space, the kinetics constraint will be guaranteed inactive. In this situation, the fitness function will be insensitive to small changes in (a, b), making it difficult to optimize these parameters. We minimize this issue by restricting b such that when $a > 0$, $b \leq v_{max}$ and when $a < 0$, $b \geq v_{min}$.

For larger systems, we found that it may be more tractable to perform multiple sequential optimization problems by fixing a subset of the parameter values and switching off between optimizing different parameters at each step. We provide an option for the user to specify multiple rounds of optimization, in which individual pairs of parameters can be set as fixed or fitted for a given round of optimization. This is accomplished by specifying a design matrix in which rows represent kinetics constraints, and columns specify individual optimization rounds. If a particular kinetics constraint (parameter pair) is fixed at its initial values for a particular round, its value for the corresponding column is set to 1; otherwise, if it is to be optimized, it is set to 0. For example, we simultaneously fit all 6 kinetics constraints in a single step by setting this matrix as a (6×1) matrix of zeros. For the *E. coli* model, we chose to optimize over individual constraints (i.e. individual (a, b) parameter pairs) in sequential order until we had cycled through fitting all kinetics constraints twice.

3.2.5.2 Dynamic Flux Estimation and parameter regression

We used a DFE scheme to fit noisy data in the Branched pathway model. We smoothed concentration time course profiles using the Impulse function as described in Chapter 2, and determined the slope (metabolite accumulation or pooling flux) from the derivative of the smoothed function[45]. From these slope values the dynamic flux distribution was calculated according to a procedure based on the method of Ishii *et al.*[47] Fluxes were divided into “static” and

"dynamic" sets, and the stoichiometric mass balance equations re-organized to solve for the "dynamic" fluxes using MATLAB's backslash pseudo-inverse. From this, we paired the resulting calculated dynamic flux distribution data with the original noisy concentration data for subsequent regression analysis.

To estimate Branched pathway model parameters in the regression-based methods ('BST','MM','LR'), we used the inferred flux data and the concentration data to fit the parameters of the individual rate law equations to the corresponding flux and metabolite data. For the BST-based generalized mass action kinetic rate law model ('BST'), we log-transformed the data to linearize the system and solved for the power-law parameters. For the Michaelis-Menten Kinetic Rate Law model ('MM'), we performed a non-linear regression by seeding the solver with 100 random initial parameters and selecting the fit with the lowest residuals. Rate law equations for the MM model were as follows:

$$v_1 = v_0$$

$$v_2 = V_2^M \frac{X_1}{V_2^K + X_1}$$

$$v_3 = V_3^M \frac{X_2}{(V_3^I + X_4)(V_3^K + X_2)}$$

$$v_4 = V_4^M \frac{X_1}{(V_4^A + \frac{1}{X_3})(V_4^K + X_1)}$$

$$v_5 = V_5^M \frac{X_3 \cdot X_4}{V_5^K + V_{5,3}^K \cdot X_3 + V_{5,4}^K \cdot X_4 + X_3 \cdot X_4}$$

where V_2^M, V_2^K, V_3^M, V_3^K, V_3^I, V_4^M, V_4^K, V_4^A, V_5^M, V_5^K, $V_{5,3}^K$, $V_{5,4}^K$ are the fitted Michaelis-Menten parameters[49].

For the Linear Regression FBA model ('LR'), we performed linear regression on the combined flux and/or concentration data for each target-controller mapping as appropriate (for example, regression on the sum (v_2 + v_4) against X_1 when controller metabolite X_1 is mapped to target fluxes v_2 and v_4).

During our analysis, we explore the impact of incomplete data in the form of missing time course data. To model this, we select a metabolite, designate it as "missing", and withhold the time course data for that metabolite from the analysis (with the exception that we provide the initial concentration of the metabolite as a means of starting the process). For the DFE procedure, we designate the pooling flux as a static flux and set its value to 0 on the basis that we have no information to justify assigning it a non-zero value. Similarly, the weight of this metabolite is set to 0 in the fitness function to preclude it from influencing global optimization.

3.2.5.3 Assessing fitted model performance: metrics and equations

Once we determined fitted parameters for each model type and noisy dataset, we simulated the time course from a particular fitted parameter set and compared it against the original noiseless data at high resolution to assess how well it recapitulated the underlying behavior. For each fit, we calculated the penalized

relative SSE (prSSE) to allow us to compare each modeling method based on the conditions used to generate the noisy synthetic data (CoV, nT, missing X_i).

First, for model m and noisy data replicate n, we calculate the resulting simulated time course data as

$$\tilde{y}_{j,k,m,n} = f_m(\vec{x}_n^0; \theta_{m,n})$$

where $\tilde{y}_{j,k,m,n}$ is the simulated value of concentration or flux j at time k for model m fitted to noisy data replicate n, and f_m is the function integrating model m over the time course with initial conditions $\vec{x}_n^0$ and fitted parameters $\theta_{m,n}$. From this, we calculated prSSE as

$$prSSE_{m,n} = w_{m,n} \sum_j w_j \frac{\sum_k^{n_k} \left(\tilde{y}_{j,k,m,n} - y_{j,k}\right)^2}{n_k}$$

where $y_{j,k}$ is the value of species j at time k in the original noiseless time course data, n_k is the number of time points in the simulation interval,

$$\mathrm{w}_j = w_*(j) \left(\max\left(\vec{y}_j\right) - \min\left(\vec{y}_j\right)\right)^{-1}$$

is the species scaling factor, $\vec{y}_j$ is the noiseless data time course for species j, $w_*(j)$ is a binary variable denoting participation in the prSSE calculation (e.g. for $j \epsilon$ pooling fluxes, we set $w_*(j)$ to 0 to exclude them from the prSSE),

$$\mathrm{w}_{m,n} = \left(\frac{n_f(m) \cdot \mathrm{nT(n)} - n_p(m)}{n_f(m) \cdot \mathrm{nT(n)}}\right)^{-1}$$

is the penalty on parameterization, $n_f(m)$ is the number of species used to fit $\theta_{m,n}$, $\mathrm{nT(n)}$ is the number of time points used to fit $\theta_{m,n}$, and $n_p(m)$ is the number of parameters in $\theta_{m,n}$.

3.3 Results

3.3.1 Simulating a time course with a nominal set of parameters

We implemented LK-DFBA in MATLAB using the Gurobi solver library[50]. These codes take a model specified by the user (including an FBA model structure and the additional information for concentrations, regulation, and simulation interval, as described in Section 3.2.1.1, generate the extended LP problem structure for the dynamic FBA problem, and solve the optimization using Gurobi. The results of this optimization are parsed into data matrices for the concentration and flux time course profiles, and are returned to the user. An example time course simulation is shown in Figure 3.5**.** One behavior we observe is a change in active constraints over the time course, leading to shifts in the resulting flux distribution. We note here that an instantaneous shift in fluxes takes time to produce changes in concentrations, due to the integration equations.

To demonstrate the necessity of including our linear kinetics constraints, we performed a simulation with a model containing no kinetics constraints. The result of this is shown in Figure 3.6. After an initial transient period in which the metabolite pools are immediately depleted, the model quickly reverted to the steady-state flux distribution one would observe from an FBA optimization with no dynamics or regulation.

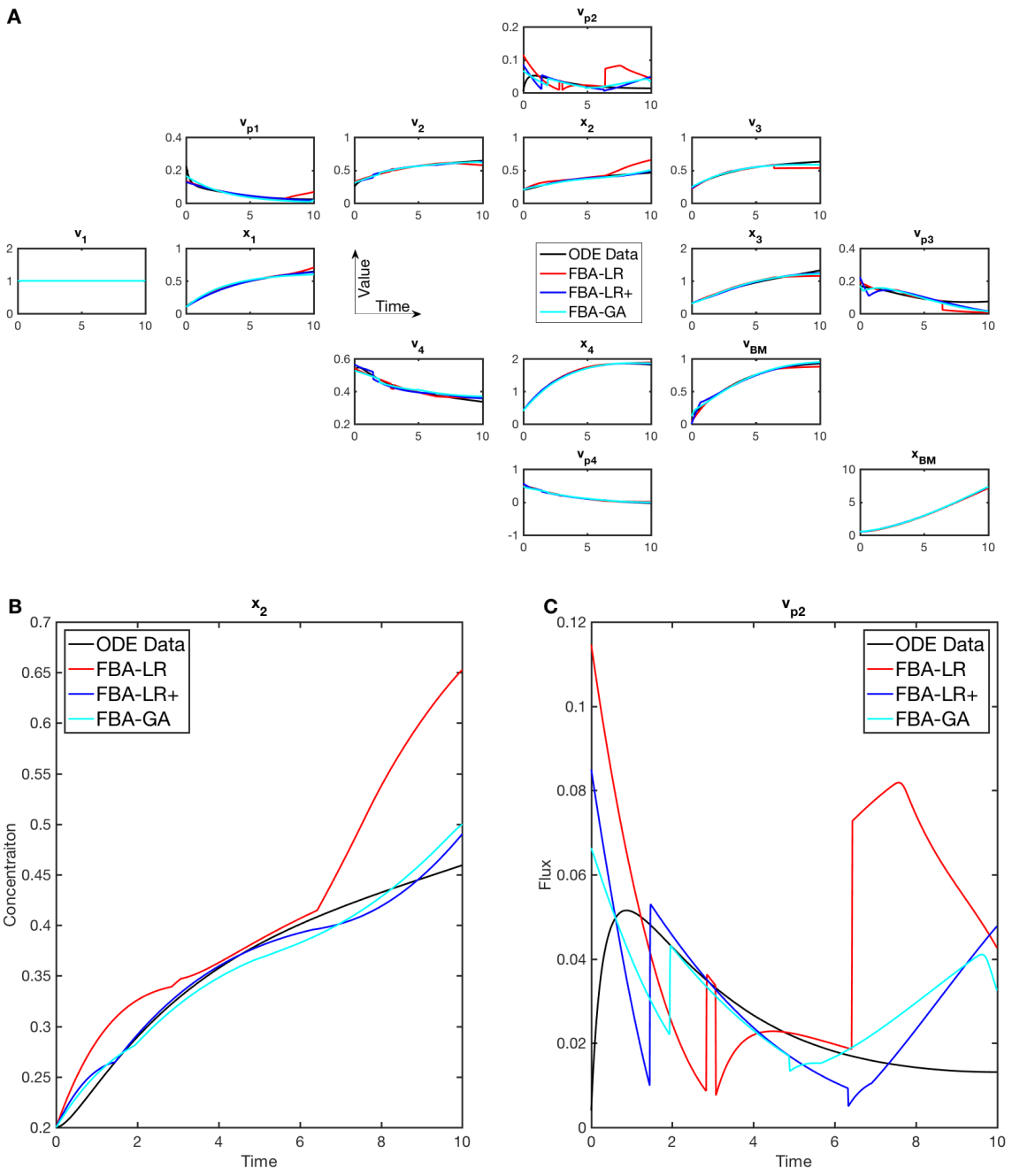

Figure 3.5. Examples of time course simulations using LK-DFBA

The ODE time course data in black were fitted with the GA, LR, and LR+ methods to identify model parameters. The resulting parameters were used with the LK-DFBA model to simulate the time course behavior.

A. The overall time course, showing metabolite concentrations, system fluxes, and pooling fluxes.

B. A closer look at metabolite X_2.

C. A closer look at pooling flux v_{p2}.

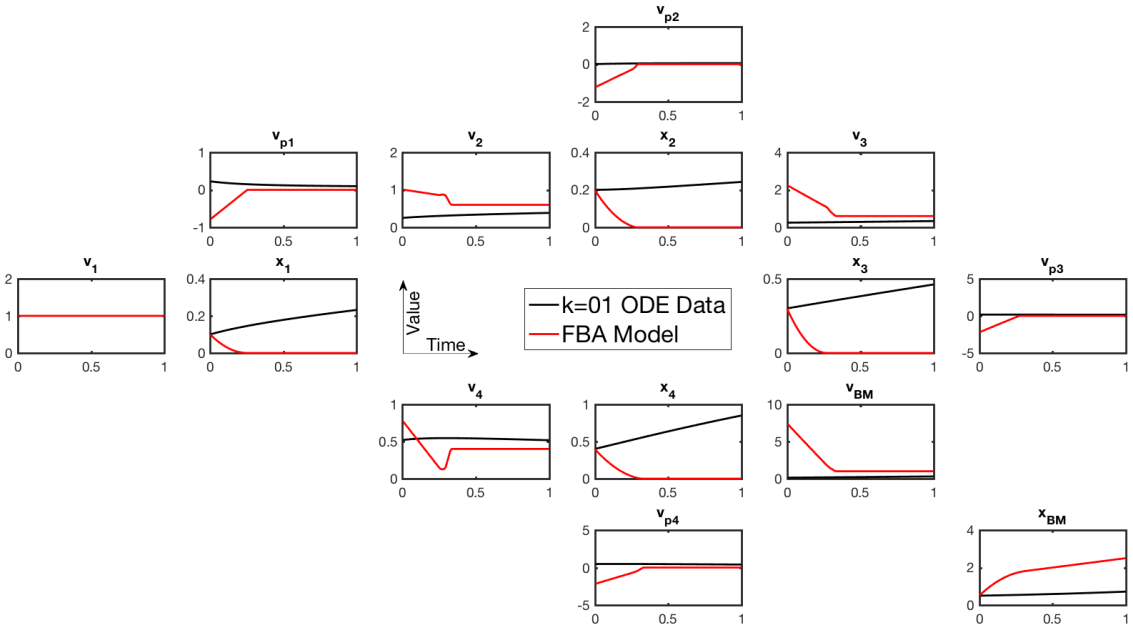

Figure 3.6. Pooling fluxes are insufficient to incentivize meaningful metabolite dynamics Additional constraints are necessary to incentivize biologically relevant behavior. When the regulatory constraints are specified as an empty set, the model exhibits an initial burst of activity as the metabolite pools are consumed. This is followed by steady-state behavior in which the model produces the same steady-state flux results that are observed in an unmodified FBA model.

To produce the stable behavior shown in Figures 3.5 and 3.6, we tested several options to determine the optimal configuration of the optimization problem. We explored a terminal and an instantaneous objective function, and determined that an instantaneous objective produced more stable behavior. The justification for this decision is shown in Figure 3.7, in which the prevalence of degenerate solutions and inconsistent time course behavior led us to abandon the terminal objective function.

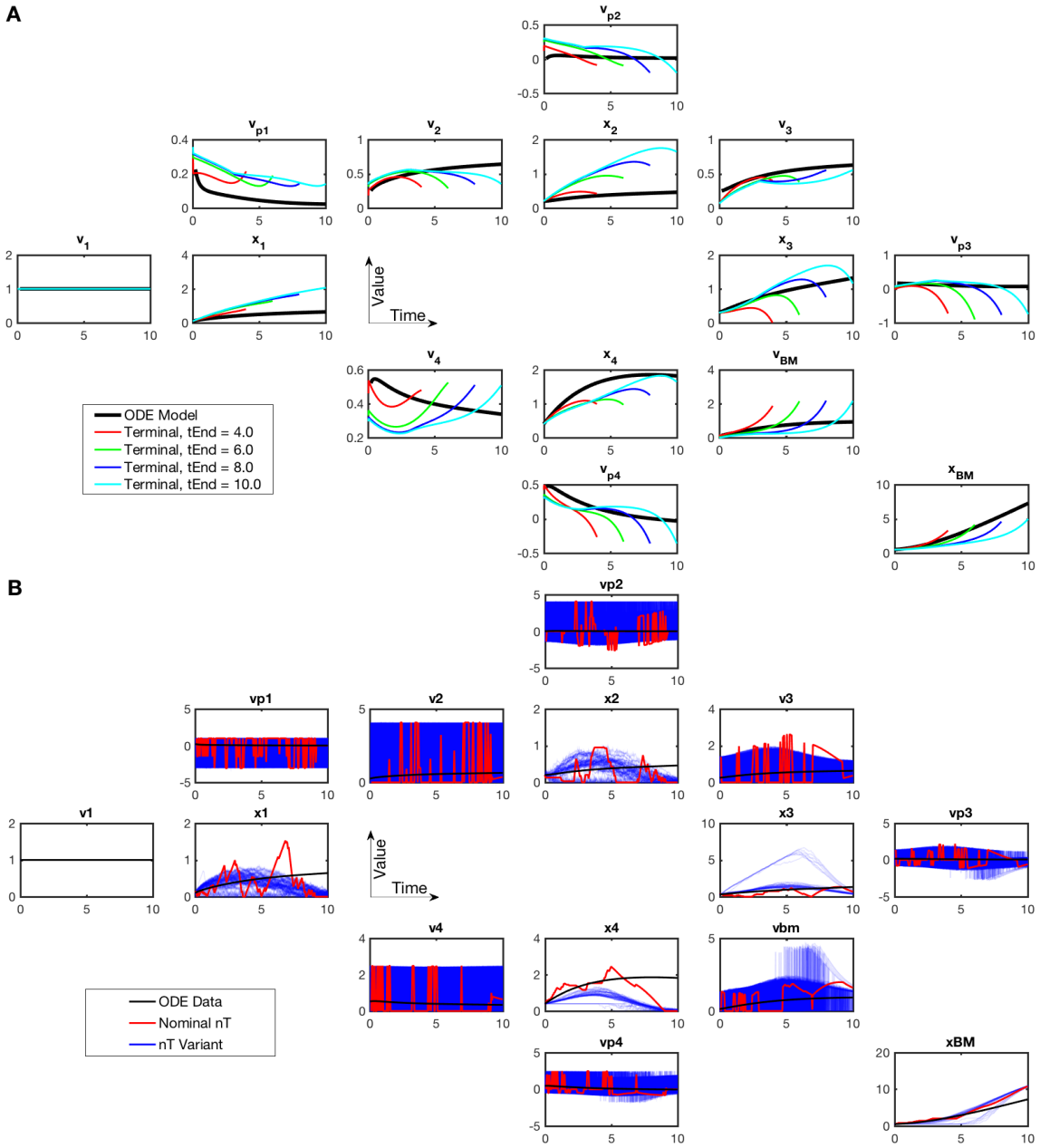

Figure 3.7. The terminal objective was prone to several serious numerical deficiencies
A. Using the same kinetics parameters and the same Δt, we simulated the model using the terminal objective under differing simulation end times. We observed that the trajectories produced under this variation were inconsistent, leading to wide variation in model behavior in the overlapping intervals. In this example, an L_2 penalty has been assigned to the solution vector to combat the degeneracy issue shown in panel B.
B. Simulations with the terminal objective without a secondary penalty were heavily sensitive to the choice of parameter nT during simulation. In this set of graphs, the genetic algorithm was used with the terminal objective at nT = 200. The resulting parameters were simulated at nT = 200, shown in red. The simulation was repeated at nT = 150, nT = 200, nT = 250, up to to nT = 2000 with the same parameters; the resulting trajectories are shown in blue. We note that the trajectories shown here represent degenerate solutions to the optimization problem: for each trajectory, the objective function (final concentration of X_{BM}) obtains the same value.

To combat degenerate solutions, we further explored penalties on the norm of the solution vector $\vec{\omega}$. These included secondary optimizations in which the optimal $z = \vec{c}^T\vec{\omega}$ was set as a constraint, and the L_1- or L_2-norm of $\vec{\omega}$ was minimized, as well as schemes penalizing $(\vec{v}(t_{k+1}) - \vec{v}(t_k))$ (not shown). The results of several regularization schemes are shown in Figure 3.8. From this analysis, we concluded that the best solution was a single optimization using the instantaneous objective with a penalty on the L_2-norm of $\vec{\omega}$, which we implemented as described in Section 3.2.1.8 as objective $z = \vec{c}^T\vec{\omega} - \lambda\vec{\omega}^T\vec{\omega}$.

In hindsight, the improved performance of the instantaneous objective function over the terminal objective is perhaps unsurprising. In a biological system, the organism lacks any foreknowledge of resource abundance, and instead is limited only to sensing the current stats of its internal and external environment. The instantaneous objective better reflects this reality, and is justified both on a theoretical basis and on the practical basis demonstrated in Figures 3.7 and 3.8.

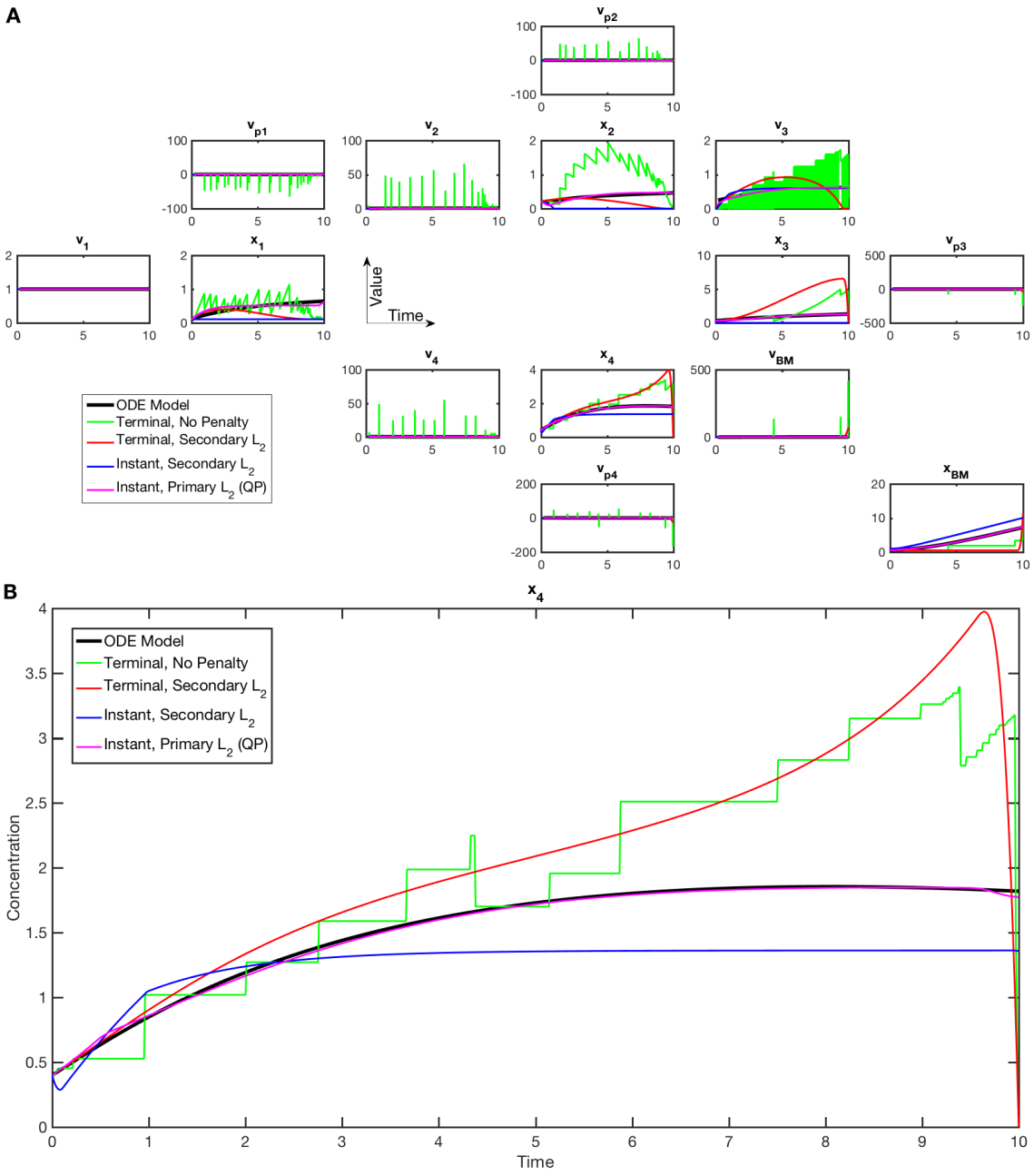

Figure 3.8. Qualitative comparison of solution-norm penalization schemes
In addition to comparing the instantaneous and terminal objective types, we explored solution-norm penalization. Model trajectories were simulated from parameters identified using the genetic algorithm method and the FBA model configuration specified on high-resolution noiseless ODE data (in black).
A. An overall comparison across the Branched Pathway model.
B. A more detailed view of metabolite X_4. The un-penalized Terminal objective (green) led to poor and inconsistent behavior, whereas the instant objective with primary L_2 minimization (formulated as a QP; magenta) gave the best fit and most robust behavior.

3.3.2 Assessment of five model types on noiseless Branched Pathway data

As described in Section 3.2.4, we generated a set of ODE time course profiles to provide a set of high-quality, high-abundance synthetic data sets with which to initially compare several modeling options. By using synthetic data, we can generate data sets at arbitrary quality and in arbitrary quantities, and have full knowledge of the true time course profiles for metabolite concentrations and metabolic fluxes. This allows us to subject LK-DFBA to a rigorous and thorough theoretical treatment. For each of these data sets, we used the overall structure of the modified Branched pathway model, but varied the initial conditions, kinetic rate law parameters, and biomass stoichiometric ratios. From this, we generated 15 sets of synthetic data. The parameters used for each data set are shown in Table 3.1.

In this initial stage, we used noise-free data at high sampling frequency (nT = 100) to fit parameters. We implemented the five methods described in the Sections 3.2.5.1 and 3.2.5.2: BST, MM, LR, LR+, and GA. The first four methods used regression between the metabolite and flux time course data directly from the data, and the last two used global optimization with the fitness function described in Section 3.2.5.1. The fitness function was configured to fit only metabolite concentrations by assigning a weight of 0 to system and pooling flux values. The results of this analysis are shown in Figure 3.9.

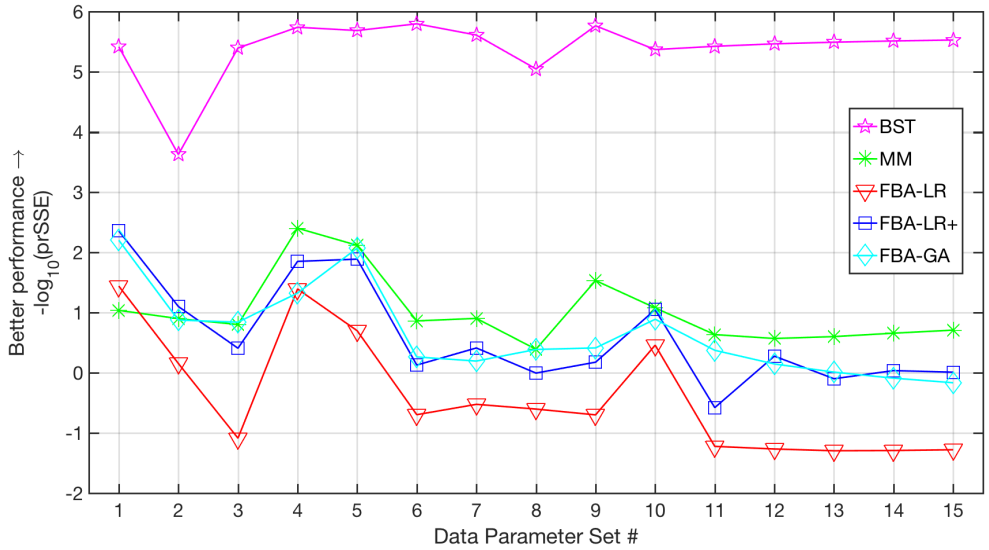

Figure 3.9. Quantitative comparison of prSSE for the BST, MM, LR, LR+, and GA methods for 15 parameterizations of the Branched Pathway model

The results of fitting these different methods to the noiseless data sets are shown in Figure 3.9. We compare the prSSE for each method across the 15 parameterizations of the ODE model described in Table 3.1. In this case, the prSSE is the sum of terms from concentrations and system fluxes (i.e. errors from pooling fluxes are omitted). A qualitative example was shown previously in Figure 3.5 for the data using the k = 01 parameters in Table 3.1.

For the noiseless, high-resolution data sets, we observe several trends. First, the BST method by far has the best performance. This is to be expected, since this method's model equations are identical to those of the underlying ODE model. Second, the LR method has the lowest performance, which is perhaps unsurprising given the number of approximations used in this method. We do note that for the k = 01 time course, it actually manages to outperform the MM

model. Third, the LR+ method substantially improves on the LR parameters, leading it to produce time course data similar in accuracy to the GA and MM methods. While the GA method outperforms the LR+ method in the majority of the cases, the differences are relatively small, and this modest improvement comes at the cost of 5-6 hours of computational time, compared to the <10 minutes required for the LR+ method (which in this case includes performing multiple fits with random perturbations to the initial LR guess). For this reason, we omit using the GA in subsequent sections.

3.3.3 Comparing the performance of methods using noisy data in the Branched pathway model

While exploring the noise-free synthetic data gives us some basic insights into the behavior of LK-DFBA under ideal conditions, there is substantial practical value in characterizing the impact of data quality on the performance of LK-DFBA. To this end, we generated datasets with different sampling frequencies and noise characteristics using the procedure described in the previous chapter. Briefly, high-resolution time course data for the modified Branched Pathway model were generated using the k = 01 parameter set. Noisy time course data were generated by sampling the high resolution data at nT = 15, 20, 30, 40, and 50, and adding Gaussian noise to data points after the initial time point with CoV = 0.05, 0.15, and 0.25. For each combination of nT and CoV, 50 replicate data sets were produced, for a total of 750 noisy data sets.

For each noisy data set replicate and each modeling method (MM, BST, LR, LR+) we employed the DFE procedure as described in detail in Section 3.2.5.2. For estimating dynamic flux profiles, we used the following steps. First, metabolite time course profiles were smoothed and slopes estimated using the Impulse smoothing function[32,51]. Then, the dynamic flux distribution was calculated using the procedure of Ishii *et al.*[32,47] From this, the inferred flux values were regressed against the original noisy metabolite concentrations as appropriate for the specific method[32]. For the LR+ model, we used the parameters from the LR model as an initial seed for a global parameter optimization. We fit parameters for each of the 4 models to each of the 750 noisy datasets. The fitted parameters were used to simulate the system time course for each case at high resolution. These simulated data were compared against the noiseless version of the data to calculate model prSSE as described in Section 3.2.5.3. The results of this analysis are shown in Figure 3.10.

In this analysis, we observe a few basic trends. First, as expected, as the quantity of data in the time courses increase, the methods all consistently achieve lower error (higher $-\log_{10}(\text{prSSE})$), with some evidence of diminishing returns in a few cases at high nT. In addition, the quality of fits decrease as the added noise increases. Across conditions, the LR+ method outperforms all other methods. We also note that the BST method performs well in cases when data quality is very good, such as at low noise (CoV = 0.05), or when there is a high sampling rate (nT = 40, nT = 50). When data is more sparse or noisy, the LR

method performs comparably or slightly better than the BST method. The MM method performs the worst, and in light of this we omit it from the analysis of cases where a metabolite time course is missing. We note that like the improvement from LR to LR+, an additional global optimization for the BST model can produce better results for this model as well; however, this improved performance (in which it outperforms LR+) is to be expected given that the BST model has the advantage of containing the true underlying system structure and kinetic rate laws.

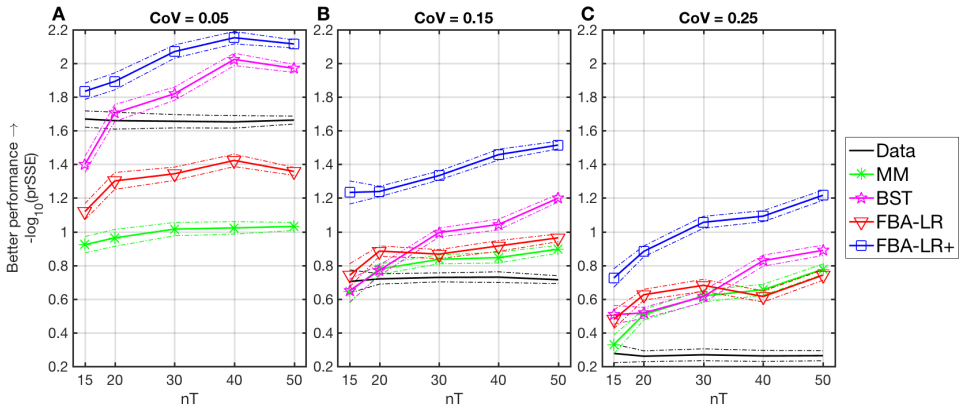

Figure 3.10. Comparison of fitting performance for MM, BST, LR, and LR+ methods
The black line for data is a benchmark comparison; each of the 750 noisy datasets was compared against the noise-free data to establish a baseline level of inaccuracy dependent on CoV; the error calculations terms are all normalized to allow a consistent comparison against this reference. A. CoV = 0.05. B. CoV = 0.15. C. CoV = 0.25.

3.3.4 The effects of withholding metabolite time courses from model performance in the Branched pathway Model

In order to test the impact of missing data on fitting performance, we repeated the analysis from the previous section, but modified the procedure by withholding information about one metabolite from the fitting pipeline to model it as "missing"

from the data (The value of the metabolite's initial condition was retained). This was accomplished by setting the pooling flux of the missing metabolite as "static" for the flux estimation step, and the corresponding regressions were performed with only the initial value as a placeholder. Each of the five metabolites in the Branched pathway were modeled as missing this way, for each of the 750 noisy datasets from the previous section. For each case, the BST, LR, and LR+ fitting methods were performed. In the case of the LR+ method, the missing metabolite was also removed from the weights of the fitness function. The results of this analysis are shown in Figure 3.11.

The position of the missing metabolite in the metabolic network leads to some dramatically different trends from Figure 3.10. These trends can be shown to derive from the quality of the performance of estimating dynamic flux distribution; by setting the pooling flux as static, the calculated system fluxes adjacent to that metabolite are skewed accordingly. This in turn affects the regression step, and the resulting parameters.

While global parameter fitting may be useful for counteracting this source of inaccuracy, it is not guaranteed to do so. The most interesting outcome from this analysis is the performance of the LR and LR+ methods in Figure 3.11D-E, in which the LR method actually outperforms the LR+ method.

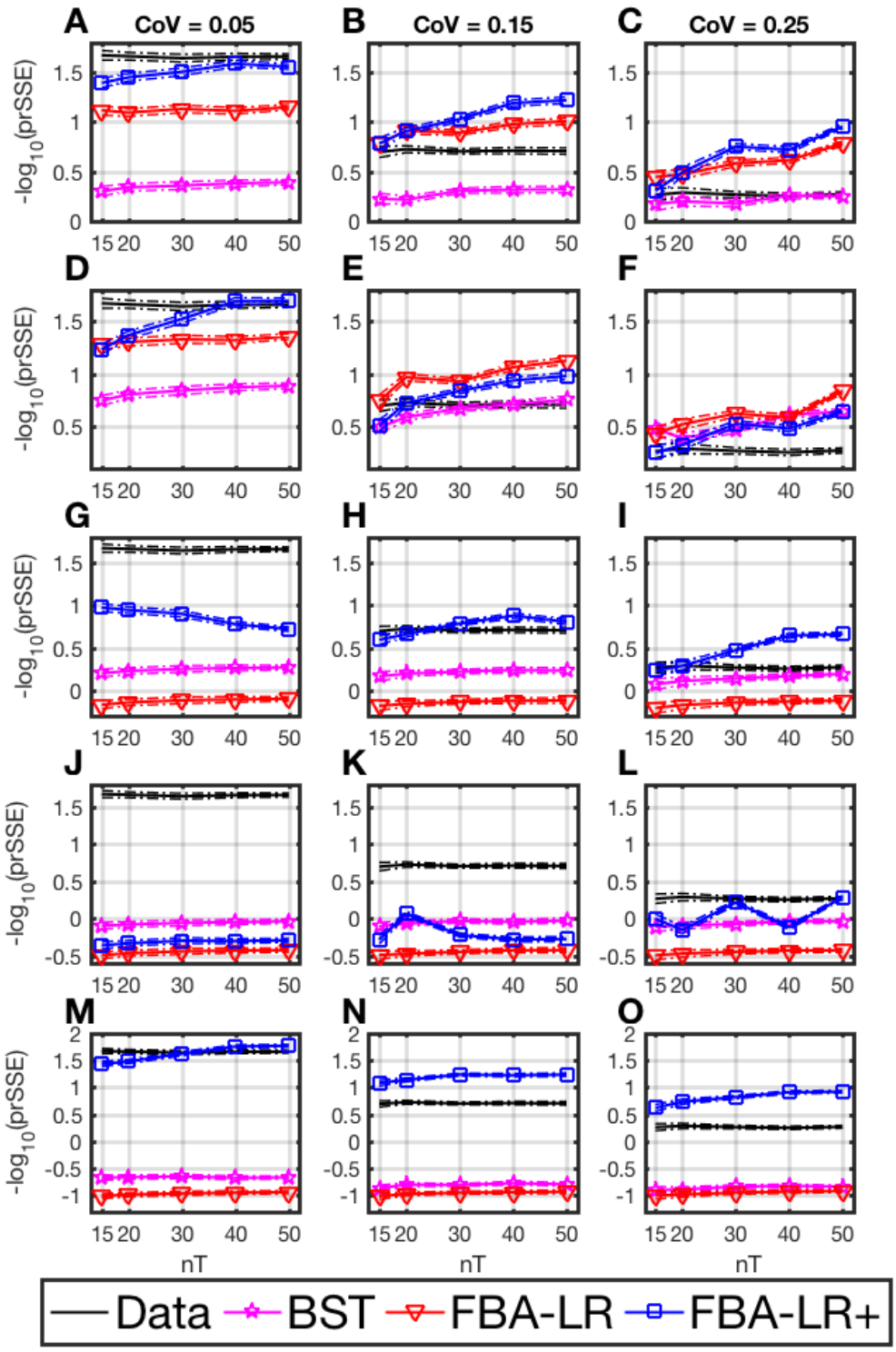

Figure 3.11. Comparison of the fitting performance of BST, LR, and LR+ when one metabolite time course is withheld from the fitting procedure
A,B,C. Performance when X_1 is missing (X_1-Missing)
D,E,F. X_2-Missing.
G,H,I. X_3-Missing.
J,K,L. X_4-Missing.
M,N,O. X_{BM}-Missing.

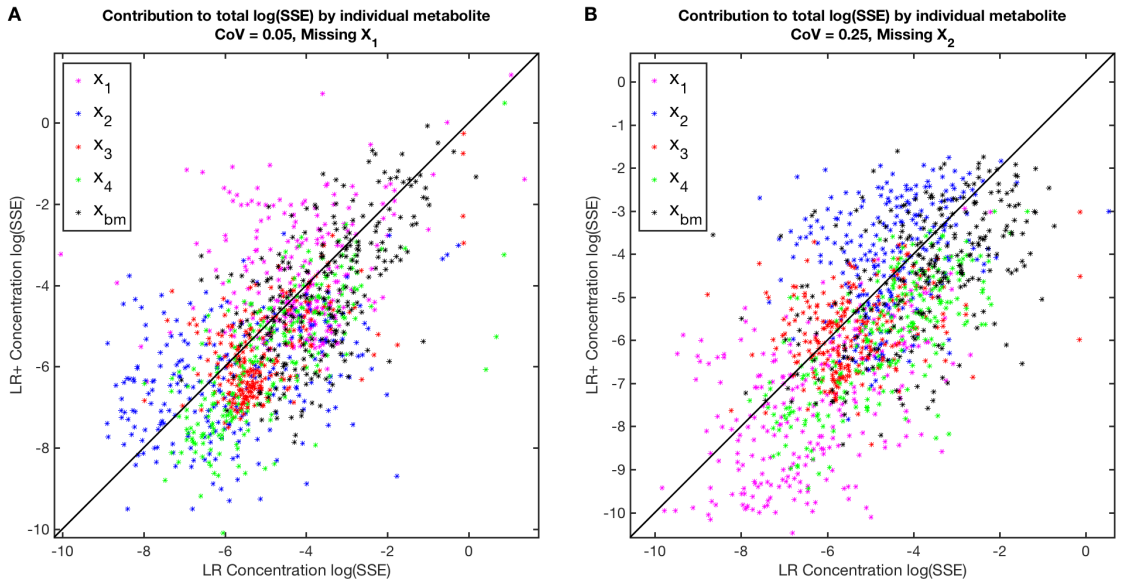

Figure 3.12. Comparing error contribution for Missing-X_1 and Missing-X_2 cases
For a given noisy dataset, the data was fit using the LR and LR+ methods. The prSSE for an individual metabolite was calculated for each fitted model, and compared as shown above. The x-axis denotes the time course total prSSE of that metabolite in the LR model, and the y-axis the corresponding prSSE in the LR+ model. Individual dots represent the error for a specific metabolite in a specific dataset, with the color indicating the metabolite. Data above the solid black line indicates that for that data noisy data set, the error in the LR+ model exceeded the data in the LR model.
A. CoV = 0.05, Missing-X_1 data sets.
B. CoV = 0.25, Missing-X_2 data sets.

In the Missing-X_2 cases, the lack of data describing X_2 dynamics led to poor optimization using the global method: the parameters that best optimized the remaining data pushed the model to poorly approximating the time course of the unmeasured metabolite (which was still included in the calculation of prSSE), as shown in Figure 3.12. Looking at the contribution of individual metabolites to the overall error, we indeed see that the largest contribution comes from the prSSE for predicting X_2**.** This serves to demonstrate a point made by Goel *et al.* regarding error compensation and the advantages of performing parameter

optimization over smaller independent subsets of the system via e.g. regression[32].

We do also note that when X_4 is withheld from flux estimation and parameter optimization (represented in Figure 3.11J-K), the LR+ model usually fails to outperform the BST model. This suggests that X_4 has a larger impact on the ability of the LR+ model to capture the correct behavior. Given that X_4 is the controller for one of the two regulatory interactions, and this interaction is a positive regulator, this serves to highlight the importance of capturing metabolite dynamics in order to incorporate regulation, and further justifies our interest in capturing these sorts of interactions.

3.3.5 Recapitulating results with the E. coli model

The branched pathway model is a useful model for exploring some basic characteristics of the approach, but lacks biologically relevant features. To introduce some of these complexities and to explore the performance of our approach with a medium-scale model with biological relevance, we generated synthetic data using the *E. coli* model of Chassagnole *et al.*[42] This larger model includes 18 metabolites of differing magnitude, and 48 fluxes; of these fluxes, 13 are reversible and 17 represent first order “degradation” reactions that act as sink terms for metabolites. The topology of this network is more complicated, with multiple branch and convergence points. Implementing this model in LK-DFBA

resulted in several modifications to our procedure, which are discussed in more detail in Section 3.2.3.2 and below.

We produced synthetic noisy data from the *E. coli* model using the procedure previously described. High quality, noise-free data for the model's 18 metabolites and 48 fluxes were generated over the interval of 10s from the ODE model and nominal parameters. From this, we produced 20 noise-added replicates each for nT = 20, 30, and 40 and CoV = 0.10 and 0.20 (for a total of 120 noisy data sets). For these datasets, we observed severe difficulties in recapitulating a qualitatively correct dynamic flux distribution using impulse smoothing and the procedure of Ishii *et al.* for dynamic flux estimation (before any LK-DFBA calculations were performed), as shown in Figure 3.13[47]. To circumvent this issue, we opted to instead use noise-added flux data directly from the ODE results for regression. We considered this to be a reasonable means of ensuring that the analysis was assessment of the modeling approach itself, rather than of the DFE procedure.

We fit each of these noisy data sets using two different implementations of LK-DFBA. In the first, a single constraint was used to limit the total efflux from a given metabolite (i.e. all effluxes were listed as targets for that constraint), adding 17 constraints (34 parameters) of this type. In the second case, we split the targets so that each metabolite-efflux mapping had only one target flux, resulting

in 49 constraints (98 parameters). We refer to the two model implementations as the “unsplit” and “split” constraint implementations, respectively. For both models, we also included 6 constraints (12 parameters) describing allosteric regulation interactions, resulting in fitting 23 constraints (46 parameters) in the unsplit implementation, and 55 constraints (110 parameters) in the split implementation. The 17 degradation and dilution reactions were modeled as first order kinetic rate laws by setting b = 0 and the a values as the rate constants from the ODE model.

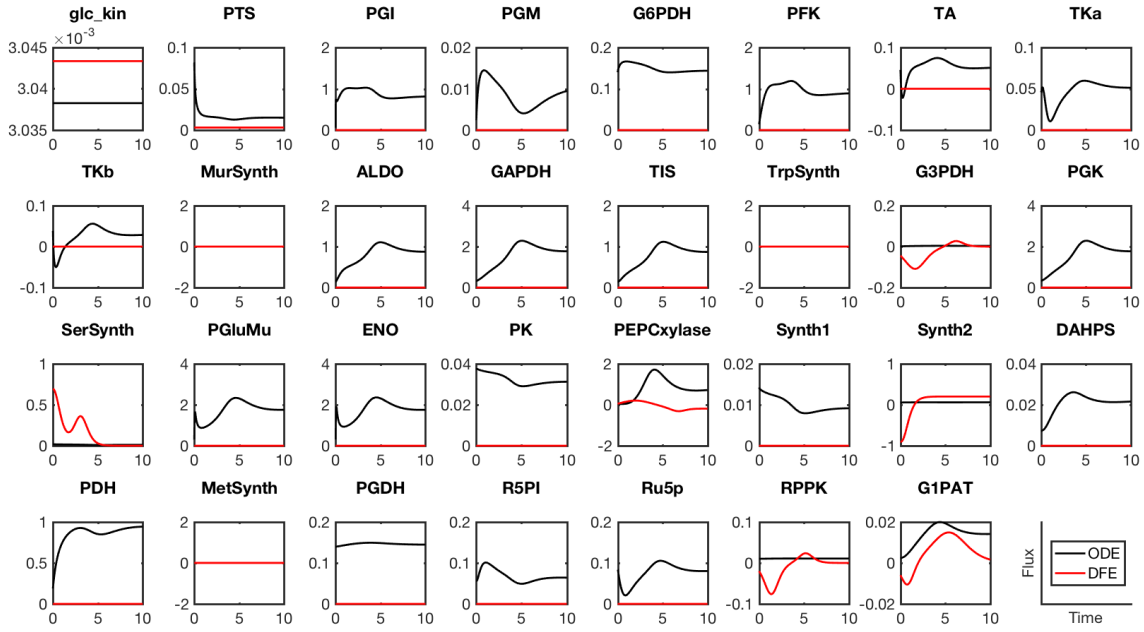

Figure 3.13. Dynamic Flux Estimation in the *E. coli* model
The black trajectories are flux profiles from the ODE model, and the red trajectories are the dynamic flux distributions calculated using the DFE procedure described in Section 3.2.3.2. In most cases, DFE failed to qualitatively capture the correct flux behaviors, making it difficult to use the regression method with any accuracy. We opted to instead use noise-added flux values from the ODE model for parameter regression.

For the *E. coli* data, we asked whether the additional parameters introduced in the split constraints implementation was justified by an improvement in model accuracy (reflected by penalizing the relative SSE value commensurate with the

additional parameters to determine prSSE); additional considerations include the increased time required to fit these extra parameters and the increased risk of producing an over-fitted model. For each implementation and noisy data set, we identified parameters both with the LR and LR+ methods, modifying the LR+ method to use the sequential parameter fitting scheme described in Section 3.2.5.1. As with the Branched pathway model, we evaluated the quality of the resulting fits by calculating a parameter-penalized relative sum-of-squares error against higher-resolution noise-free data. The results of this analysis are shown in Figure 3.14.

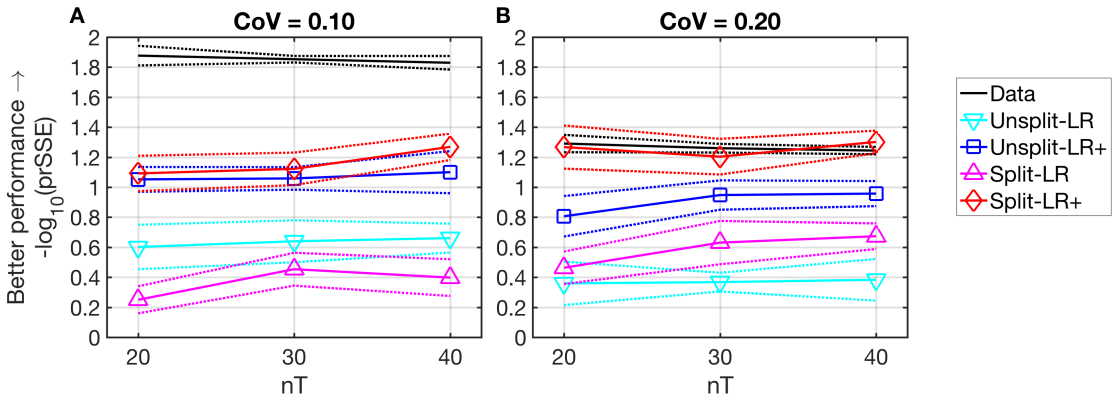

Figure 3.14. Results of fitting the Unsplit and Split LK-DFBA models to the *E. coli* data
A. CoV = 0.10.
B. CoV = 0.20.

We note several trends in Figure 3.14. First, the unsplit model behaves with the same general trends we observed in the Branched Pathway model, in which increasing nT and decreasing CoV consistently lead to improved prSSE, and the LR+ method outperforms the LR method. Second, the split model generally outperforms the unsplit model, with the exception of using the LR method with

CoV = 0.10 data. Third, the split model performs better on the CoV = 0.20 data than on the CoV = 0.10 data, for both the LR and LR+ methods. On average, for the LR+ method the unsplit model (46 parameters) took ~30 minutes to fit for each noisy dataset, while the split model (110 parameters) took ~50 minutes (For both split and unsplit models, the LR method took fractions of a second).

3.4 Discussion

In this work, we devised and implemented LK-DFBA, a fully linear modification of DFBA that allows us to capture metabolite dynamics and metabolite-dependent kinetic and regulatory interactions while retaining the linearity of regular FBA. Given the same information necessary for FBA, initial conditions for metabolites, and a suitable description of the connectivity and parameterization of the kinetics interactions, we showed that LK-DFBA successfully reproduces biologically relevant model dynamics.

Further, we demonstrated using a DFE approach to fit two models (Branched Pathway, *E. coli*) to synthetic noisy data to recapitulate the correct underlying model behavior. Our method performed competitively against ODE models using Michaelis-Menten and GMA kinetic rate laws, can generally handle cases where metabolite time courses are missing from the data. It is more robust than other methods under the most realistic cases, such as in the presence of noise or when the numbers of time points available for fitting are relatively scarce.

We also demonstrated the viability of LK-DFBA in a biologically relevant model, and examined some of the additional challenges inherent to more realistic models. These include different scaling for variables, the presence of reversible reactions, more complicated model topology, and procedures for fitting larger numbers of model parameters. We explored the impact of different modeling options on the model performance, finding that a more heavily parameterized model may still be beneficial for better capturing the correct behaviors, if sufficient data is available to justify their addition.

In the work discussed in this chapter, we modeled regulatory kinetics constraints that correspond to regulation of fluxes via rapid, direct mechanisms such as allostery. However, LK-DFBA is not inherently restricted to modeling this type of regulation. By choosing a simulation interval over which transcriptional changes are relevant, changes in enzyme levels could easily be modeled as well. Capturing these other types of regulation may require modifications to reflect differences in the underlying mechanisms. For example, the implementation described in this chapter assumed that the constraints on targeted fluxes are dependent on the concentration of the controller metabolites at the immediately preceding time point. However, changes in target fluxes associated with transcriptional regulation may be subject to a time delay due to the intermediate biochemical steps necessary to produce the relevant changes in enzyme levels. Such a time delay could be introduced by shifting the linkage between controller

metabolites and target fluxes from the adjacent time interval to instead a later time interval, with the exact offset specified using parameters set by the user or determined through parameter fitting.

By retaining the LP structure and the original stoichiometry of the FBA problem, we have created a problem that can integrate metabolite dynamics and regulation into the many strain design tools created around FBA. For example, we envision that a tool such as OptKnock could be used on a model represented in LK-DFBA. The mappings between knockout genes and fluxes in the FBA model could be applied to the flux values over the whole time course, producing predictions that now take into account metabolic dynamics and regulation. Metabolism is heavily regulated, and metabolic engineering efforts that ignore that will inevitably come up short. Development of genome-scale models with regulation will enable more accurate prediction and more effective metabolic engineering that could have a drastic impact on titers and productivity. Our work here establishes a basis for working towards that goal, and merits further investigation to see such applications to fruition.

3.5 References

1 Canelas, A. B. *et al.* Integrated multilaboratory systems biology reveals differences in protein metabolism between two reference yeast strains. *Nat Commun* **1**, 145, doi:10.1038/ncomms1150 (2010).

2 Dunn, W. B. *et al.* Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature protocols* **6**, 1060-1083, doi:10.1038/nprot.2011.335 (2011).

3 Link, H., Fuhrer, T., Gerosa, L., Zamboni, N. & Sauer, U. Real-time metabolome profiling of the metabolic switch between starvation and growth. *Nat Methods* **12**, 1091-1097, doi:10.1038/nmeth.3584 (2015).

4 Sevin, D. C., Stahlin, J. N., Pollak, G. R., Kuehne, A. & Sauer, U. Global Metabolic Responses to Salt Stress in Fifteen Species. *PLoS One* **11**, e0148888, doi:10.1371/journal.pone.0148888 (2016).

5 McKee, A. E. *et al.* Manipulation of the carbon storage regulator system for metabolite remodeling and biofuel production in Escherichia coli. *Microb Cell Fact* **11**, 79, doi:10.1186/1475-2859-11-79 (2012).

6 Nakagawa, A. *et al.* Total biosynthesis of opiates by stepwise fermentation using engineered Escherichia coli. *Nat Commun* **7**, 10390, doi:10.1038/ncomms10390 (2016).

7 Zampar, G. G. *et al.* Temporal system-level organization of the switch from glycolytic to gluconeogenic operation in yeast. *Mol Syst Biol* **9**, 651, doi:10.1038/msb.2013.11 (2013).

8 Chubukov, V. *et al.* Transcriptional regulation is insufficient to explain substrate-induced flux changes in Bacillus subtilis. *Mol Syst Biol* **9**, 709, doi:10.1038/msb.2013.66 (2013).

9 Goncalves, E. *et al.* Systematic Analysis of Transcriptional and Post-transcriptional Regulation of Metabolism in Yeast. *PLoS Comput Biol* **13**, e1005297, doi:10.1371/journal.pcbi.1005297 (2017).

10 Burgard, A. P., Pharkya, P. & Maranas, C. D. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng* **84**, 647-657, doi:10.1002/bit.10803 (2003).

11 Varma, A. & Palsson, B. O. Metabolic capabilities of Escherichia coli: I. synthesis of biosynthetic precursors and cofactors. *J Theor Biol* **165**, 477-502, doi:10.1006/jtbi.1993.1202 (1993).

12 Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst Biol* **7**, 74, doi:10.1186/1752-0509-7-74 (2013).

13 Segre, D., Vitkup, D. & Church, G. M. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 15112-15117, doi:10.1073/pnas.232349399 (2002).

14 Covert, M. W. & Palsson, B. O. Transcriptional regulation in constraints-based metabolic models of Escherichia coli. *The Journal of biological chemistry* **277**, 28058-28064, doi:10.1074/jbc.M201691200 (2002).

15 Cotten, C. & Reed, J. L. Mechanistic analysis of multi-omics datasets to generate kinetic parameters for constraint-based metabolic models. *BMC Bioinformatics* **14**, 32, doi:10.1186/1471-2105-14-32 (2013).

16 Henry, C. S., Broadbelt, L. J. & Hatzimanikatis, V. Thermodynamics-based metabolic flux analysis. *Biophys J* **92**, 1792-1805, doi:10.1529/biophysj.106.093138 (2007).

17 Hamilton, J. J., Dwivedi, V. & Reed, J. L. Quantitative assessment of thermodynamic constraints on the solution space of genome-scale metabolic models. *Biophys J* **105**, 512-522, doi:10.1016/j.bpj.2013.06.011 (2013).

18 Chowdhury, A., Zomorrodi, A. R. & Maranas, C. D. k-OptForce: integrating kinetics with flux balance analysis for strain design. *PLoS Comput Biol* **10**, e1003487, doi:10.1371/journal.pcbi.1003487 (2014).

19 Kumar, V. S. & Maranas, C. D. GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. *PLoS Comput Biol* **5**, e1000308, doi:10.1371/journal.pcbi.1000308 (2009).

20 Ranganathan, S., Suthers, P. F. & Maranas, C. D. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput Biol* **6**, e1000744, doi:10.1371/journal.pcbi.1000744 (2010).

21 Zomorrodi, A. R., Islam, M. M. & Maranas, C. D. d-OptCom: Dynamic multi-level and multi-objective metabolic modeling of microbial communities. *ACS Synth Biol* **3**, 247-257, doi:10.1021/sb4001307 (2014).

22 Zomorrodi, A. R. & Maranas, C. D. OptCom: a multi-level optimization framework for the metabolic modeling and analysis of microbial communities. *PLoS Comput Biol* **8**, e1002363, doi:10.1371/journal.pcbi.1002363 (2012).

23 Wu, L. *et al.* Short-term metabolome dynamics and carbon, electron, and ATP balances in chemostat-grown Saccharomyces cerevisiae CEN.PK 113-7D following a glucose pulse. *Appl Environ Microbiol* **72**, 3566-3577, doi:10.1128/AEM.72.5.3566-3577.2006 (2006).

24 Kromer, J. O., Sorgenfrei, O., Klopprogge, K., Heinzle, E. & Wittmann, C. In-depth profiling of lysine-producing Corynebacterium glutamicum by combined analysis of the transcriptome, metabolome, and fluxome. *J Bacteriol* **186**, 1769-1784, doi:10.1128/jb.186.6.1769-1784.2004 (2004).

25 Covert, M. W., Schilling, C. H. & Palsson, B. Regulation of gene expression in flux balance models of metabolism. *J Theor Biol* **213**, 73-88, doi:10.1006/jtbi.2001.2405 (2001).

26 Cotten, C. & Reed, J. L. Constraint-based strain design using continuous modifications (CosMos) of flux bounds finds new strategies for metabolic engineering. *Biotechnology Journal* **8**, 595-604, doi:10.1002/biot.201200316 (2013).

27 Tran, L. M., Rizk, M. L. & Liao, J. C. Ensemble modeling of metabolic networks. *Biophys J* **95**, 5606-5617, doi:10.1529/biophysj.108.135442 (2008).

28 Costa, R. S., Machado, D., Rocha, I. & Ferreira, E. C. Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis-Menten and approximate kinetic equations. *Biosystems* **100**, 150-157, doi:10.1016/j.biosystems.2010.03.001 (2010).

29 Usuda, Y. *et al.* Dynamic modeling of Escherichia coli metabolic and regulatory systems for amino-acid production. *J Biotechnol* **147**, 17-30, doi:10.1016/j.jbiotec.2010.02.018 (2010).

30 Covert, M. W., Xiao, N., Chen, T. J. & Karr, J. R. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics* **24**, 2044-2050, doi:10.1093/bioinformatics/btn352 (2008).

31 Gutenkunst, R. N. *et al.* Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol* **3**, 1871-1878, doi:10.1371/journal.pcbi.0030189 (2007).

32 Goel, G., Chou, I. C. & Voit, E. O. System estimation from metabolic time-series data. *Bioinformatics* **24**, 2505-2511, doi:10.1093/bioinformatics/btn470 (2008).

33 Jia, G., Stephanopoulos, G. N. & Gunawan, R. Parameter estimation of kinetic models from metabolic profiles: two-phase dynamic decoupling method. *Bioinformatics* **27**, 1964-1970, doi:10.1093/bioinformatics/btr293 (2011).

34 Chou, I. C. & Voit, E. O. Estimation of dynamic flux profiles from metabolic time series data. *BMC Syst Biol* **6**, 84, doi:10.1186/1752-0509-6-84 (2012).

35 Jia, G., Stephanopoulos, G. & Gunawan, R. Incremental parameter estimation of kinetic metabolic network models. *BMC Syst Biol* **6**, 142, doi:10.1186/1752-0509-6-142 (2012).

36 Zomorrodi, A. R., Lafontaine Rivera, J. G., Liao, J. C. & Maranas, C. D. Optimization-driven identification of genetic perturbations accelerates the convergence of model parameters in ensemble modeling of metabolic networks. *Biotechnol J* **8**, 1090-1104, doi:10.1002/biot.201200270 (2013).

37 Mahadevan, R., Edwards, J. S. & Doyle, F. J., 3rd. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophys J* **83**, 1331-1340, doi:10.1016/S0006-3495(02)73903-9 (2002).

38 Karr, J. R. *et al.* A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389-401, doi:10.1016/j.cell.2012.05.044 (2012).

39 Knies, D. *et al.* Modeling and Simulation of Optimal Resource Management during the Diurnal Cycle in Emiliania huxleyi by Genome-Scale Reconstruction and an Extended Flux Balance Analysis Approach. *Metabolites* **5**, 659-676, doi:10.3390/metabo5040659 (2015).

40 Vardi, L., Ruppin, E. & Sharan, R. A linearized constraint-based approach for modeling signaling networks. *Journal of computational biology : a journal of computational molecular cell biology* **19**, 232-240, doi:10.1089/cmb.2011.0277 (2012).

41 Voit, E. O. & Almeida, J. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics* **20**, 1670-1681, doi:10.1093/bioinformatics/bth140 (2004).

42 Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. & Reuss, M. Dynamic modeling of the central carbon metabolism of Escherichia coli. *Biotechnol Bioeng* **79**, 53-73, doi:10.1002/bit.10288 (2002).

43 Kauffman, K. J., Prakash, P. & Edwards, J. S. Advances in flux balance analysis. *Current Opinion in Biotechnology* **14**, 491-496, doi:10.1016/j.copbio.2003.08.001 (2003).

44 Lewis, N. E. *et al.* Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models. *Mol Syst Biol* **6**, 390, doi:10.1038/msb.2010.47 (2010).

45 Dromms, R. A. & Styczynski, M. P. Improved metabolite profile smoothing for flux estimation. *Mol Biosyst* **11**, 2394-2405, doi:10.1039/c5mb00165j (2015).

46 Hoops, S. *et al.* COPASI--a COmplex PAthway SImulator. *Bioinformatics* **22**, 3067-3074, doi:10.1093/bioinformatics/btl485 (2006).

47 Ishii, N., Nakayama, Y. & Tomita, M. Distinguishing enzymes using metabolome data for the hybrid dynamic/static method. *Theor Biol Med Model* **4**, 19, doi:10.1186/1742-4682-4-19 (2007).

48 Le Novere, N. *et al.* BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res* **34**, D689-691, doi:10.1093/nar/gkj092 (2006).

49 Strelow J, D. W., Iversen PW, et al. *Mechanism of Action Assays for Enzymes.*, <https://www.ncbi.nlm.nih.gov/books/NBK92001/> (2012).

50 Gurobi Optimization Inc. (2016).

51 Dromms, R. A. & Styczynski, M. P. Systematic applications of metabolomics in metabolic engineering. *Metabolites* **2**, 1090-1122, doi:10.3390/metabo2041090 (2012).

Chapter 4: Identifying Non-Stoichiometric Metabolite-Flux Interactions from Data and LK-DBFA

4.1 Background

In the previous chapter, we established and validated a constraint-based modeling framework called Linear Kinetics-Dynamic Flux Balance Analysis (LK-DFBA) designed to modify dynamic flux balance analysis (DFBA) to instead retain the linear program structure of classic FBA[1]. In order to induce dynamic metabolite behavior, we introduced constraints that curtailed system flux in accordance with metabolite concentrations. By implementing these kinetics constraints as linear equations, we could successfully induce metabolite dynamics, incorporate regulatory interactions into the model, and retain the linear structure of the overall optimization.

LK-DFBA kinetics constraints reflect two basic classes of interactions. The first class comprises cases in which the controller metabolite participates as a substrate for the enzymatic reaction. The constraint then represents a sort of mass action effect on enzyme kinetics, dependent on the concentration of metabolite available. Because these metabolites are substrates, the resulting reaction consumes the metabolite, and the metabolite mass balance reflects this effect by including the relevant flux in the resulting stoichiometric matrix. For this class, identifying a controller-target pair is trivial: this can be read directly from

the stoichiometric matrix by identifying the list of fluxes that consume as a substrate the metabolite in question.

The second class comprises interactions in which the controller metabolite does not participate as a substrate in the target flux. In these cases, the metabolite instead modulates the reaction rate via some mechanism other than mass action. As discussed in the previous chapter, there are multiple mechanisms by which this may occur, with distinct properties and time-scales. An allosteric binding interaction may change protein conformation; by binding the enzyme non-competitively, the enzyme conformation may change to an active state (in the case of an allosteric activator), or to an inactive state (in the case of an allosteric inhibitor)[2]. The time-scale for these changes is rapid, on the orders of seconds or less[2]. At longer time scales (on the order of hours), metabolite concentrations may induce transcriptional changes, leading to changes in enzyme concentrations[3]. The linear constraints in LK-DFBA are gross linear approximations and are fairly agnostic as to the specific mechanism at play. But because these mechanisms do not directly participate in reaction stoichiometry, they cannot be directly deduced from it. If they are to be included in the model, this fact must come from additional information—whether biological knowledge or experimental data.

These kinetics constraints are critical to the performance and behavior of the model, and much of the work of the previous chapter explored the impact of the parameterization of the constraints on model behavior and performance. However, that analysis assumed that the true structure of these constraints were known, in that the complete and correct list of mappings between metabolite controllers and flux targets was available for the parameter identification and model simulation steps. In practice, we may not be able to assume that this is the case, and we need to have some sense for the consequences of this knowledge gap. How severely does an error or an omission hamstring model performance? Can we use the data and the model to postulate corrections, or to produce a ranked list of putative missing interactions? In this chapter, we explore these questions.

This information can come from existing biochemical knowledge, or must be inferred from available data. In the former case, this can occur via a preliminary literature search and will be supported by previous experimental results. In the latter case, such experimental validation is often a necessary follow-up step to validate an inferred and therefore putative interaction. Computational tools are often useful means of identifying candidates for experimental validation, and here we are interested in exploring if LK-DFBA may have a potential application in generating such candidates.

Previous work has sought to combine computational modeling with data to construct ranked lists of putative regulatory interactions. To perform this analysis with LK-DFBA, we work specifically with the procedure established by Link *et al.*[4], in which they investigated the influence of various regulatory connections on an ordinary differential equations (ODE) model of *E. coli* metabolism. In their work, they tested a set of candidate models, determined the Akaike Information Criterion (AIC) value for each model given the parameterization required to construct the model and the resulting fit, and compared those candidates against a nominal, regulation-free model[4,5]. By examining the trends in the candidate models, they were able to assess the influence of correct model connections on the model performance. We use this analysis approach with synthetic data from the modified Branched Pathway model to determine the impact of a small list of elementary regulatory connections on the performance of our fitting procedure[6]. In particular, we are interested in recapitulating the two correct regulatory interactions from the data used to generate the model.

4.2 Methods

The work in this chapter uses the approach of Link *et al.* with synthetic noisy data generated using the procedures described in previous chapters[4]. Specifically, the modified Branched Pathway model of the previous chapter is used to initially generate 50 noisy datasets with number of sampling data points nT = 50 and Coefficient of Variation CoV = 0.05 for the analysis described below[7,8]. We later

repeat this analysis with two additional sets of 50 noisy replicates: one at nT = 20 and CoV = 0.05, and the other at nT = 50 and CoV = 0.25.

The candidate models are constructed by combining elementary regulatory connections together. In a model with m metabolites and n fluxes, there are (m × n) potential elementary controller-target pairings or connections. These pairs can be grouped into three categories. First, some controller-metabolite pairs represent the mass action kinetics resulting from the metabolite participating in the reaction flux as a substrate. Another set of controller-metabolite pairs represent regulatory interactions, such as from allostery or transcriptional mechanisms. The third set comprises the remaining possible connections, and represents those connections not present in the system. In this analysis, we assume the mass action connections are known, and are interested in searching over the remaining two categories to identify the true regulatory connections.

In the modified Branched Pathway model, there are 5 metabolites and 5 system fluxes, allowing for a total of 25 potential elementary controller-flux pairings[6]. However, X_5 represents biomass rather than a true metabolite that we expect would act as a potential regulator, and we set v_1 to be a constant influx in the modeling assumptions for the modified Branched Pathway model. The result is instead a space of 16 potential pairings, five of which are represented already by the model's substrate-activity mass action relationships**.** We avoid re-using these

pairings as regulatory interactions due to the high risk that these interactions are not identifiable; in LK-DFBA, there is no clear way to distinguish between effects due to participation in mass action kinetics as a substrate vs. participation in a regulatory mechanism such as allostery. This leaves 11 potential regulatory interactions, 2 of which represent the correct regulatory interactions in the underlying model. This list of elementary regulatory interactions is shown in Table 4.1.

Table 4.1. The 11 elementary regulatory connections investigated in this analysis
The true connections, {3;4} and {4;3}, are shaded.

Connection Name	Controller Metabolite	Target Flux
{1;3}	1	3
{1;5}	1	5
{2;2}	2	2
{2;4]	2	4
{2;5}	2	5
{3;2}	3	2
{3;3}	3	3
{3;4}	3	4
{4;2}	4	2
{4;3}	4	3
{4;4}	4	4

These elementary connections can be combined to produce additional models, but the size of the resulting space is combinatoric; there are 55 models with two elementary connections, 165 models with 3 elementary connections, and 330 models with 4 elementary connections. To focus the analysis and reduce the model space in a systematic manner, we introduce criteria for combining

elementary connections. First, elementary connections with the same controller cannot be combined. Second, elementary connections with the same target cannot be combined. These criteria allowed us to considerably prune the combinatoric space while still retaining a wide sampling of the possible combinations (though, they were selected primarily for convenience rather than on some biological basis). The result is a total of 87 possible models (including the unregulated model), shown in Table 4.2.

Using the approach of Link *et al.*, we tested this set of candidate models[4]. Each candidate is fitted to the 50 noisy data sets using the procedure from the previous chapter. As per Link *et al.*, we also calculated the AIC for a given model m and noisy dataset n as

$$AIC_{m,n} = N \log \frac{SSR_{m,n}}{N} + 2p_m$$

where the sum-of-squares residual

$$SSR_{m,n} = \sum_{i,j} \frac{\left(x_{i,j,n} - \tilde{x}_{i,j,m,n}\right)^2}{\max\left(x_{i,n}\right) - \min\left(x_{i,n}\right)},$$

$x_{i,j}$ is the concentration of metabolite i at timepoint j in noisy dataset n, $\tilde{x}_{i,j,m,n}$ is the concentration of metabolite i at timepoint j predicted by model m using parameters fitted to data from noisy replicate n, N = 250 is the number of fitted data points (nT = 50 x 5 metabolites), and p_m is the number of model parameters in model m (8 for the mass action kinetics constraints in the unregulated model, plus 2 for every elementary regulatory connection included in model m)[4].

Table 4.2. A list of the candidate regulatory models in Chapter 4
The true model with connections {3;4} and {4;3} is shaded.

Model #	Controller	Target	Model #	Controller	Target	Model #	Controller	Target	Model #	Controller	Target
1	[]	[]	34	2	4	56	1	3	73	2	2
2	1	3		4	2		3	2		3	4
3	1	5	35	2	4		4	4		4	3
4	2	2		4	3	57	1	3	74	2	4
5	2	4	36	2	5		3	4		3	2
6	2	5		3	2		4	2		4	3
7	3	2	37	2	5	58	1	5	75	2	4
8	3	3		3	3		2	2		3	3
9	3	4	38	2	5		3	3		4	2
10	4	2		3	4	59	1	5	76	2	5
11	4	3	39	2	5		2	2		3	2
12	4	4		4	2		3	4		4	3
13	1	3	40	2	5	60	1	5	77	2	5
	2	2		4	3		2	2		3	2
14	1	3	41	2	5		4	3		4	4
	2	4		4	4	61	1	5	78	2	5
15	1	3	42	3	2		2	2		3	3
	2	5		4	3		4	4		4	2
16	1	3	43	3	2	62	1	5	79	2	5
	3	2		4	4		2	4		3	3
17	1	3	44	3	3		3	2		4	4
	3	4		4	2	63	1	5	80	2	5
18	1	3	45	3	3		2	4		3	4
	4	2		4	4		3	3		4	2
19	1	3	46	3	4	64	1	5	81	2	5
	4	4		4	2		2	4		3	4
20	1	5	47	3	4		4	2		4	3
	2	2		4	3	65	1	5	82	1	3
21	1	5	48	1	3		2	4		2	5
	2	4		2	2		4	3		3	2
22	1	5		3	4	66	1	5		4	4
	3	2	49	1	3		3	2	83	1	3
23	1	5		2	2		4	3		2	5
	3	3		4	4	67	1	5		3	4
24	1	5	50	1	3		3	2		4	2
	3	4		2	4		4	4	84	1	5
25	1	5		3	2	68	1	5		2	2
	4	2	51	1	3		3	3		3	3
26	1	5		2	4		4	2		4	4
	4	3		4	2	69	1	5	85	1	5
27	1	5	52	1	3		3	3		2	2
	4	4		2	5		4	4		3	4
28	2	2		3	2	70	1	5		4	3
	3	3	53	1	3		3	4	86	1	5
29	2	2		2	5		4	2		2	4
	3	4		3	4	71	1	5		3	2
30	2	2	54	1	3		3	4		4	3
	4	3		2	5		4	3	87	1	5
31	2	2		4	2	72	2	2		2	4
	4	4	55	1	3		3	3		3	3
32	2	4		2	5		4	4		4	2
	3	2		4	4						

The individual $AIC_{m,n}$ values were compared against the unregulated model case, M-01 (i.e. m=1). First, we calculated

$$\Delta AIC_{m,n} = AIC_{m=1,n} - AIC_{m,n}$$

and

$$ave\Delta AIC_m = \frac{1}{50}\sum_{n=1}^{50} \Delta AIC_{m,n}$$

to determine $ave\Delta AIC_m$, the average ΔAIC for each model[5].

From here, we explore the $ave\Delta AIC_m$ values for the models listed in Table 4.2. However, many of these models are constructed from multiple elementary connections. To more directly assess the impact of the elementary connections themselves, we used the $\Delta AIC_{m,n}$ values to perform a linear regression as follows. Each model can be mapped to a binary vector indicating the presence or absence of a particular elementary connection in Table 4.1. For example, the unregulated model M-01 (m=1) maps to a vector of zeros, whereas model M-02 (m=2) contains a vector of zeros, except for the element corresponding to connection {1;3}. This set of vectors can be collated into a design matrix $\boldsymbol{A}$, in which the number of rows is the total number of fitted models (87 models fitted each to 50 noisy datasets, or 4350 total), and the number of columns is the number of elementary connections (12 total). The list of $\Delta AIC_{m,n}$ values comprise the right-hand side of the regression equation

$$\boldsymbol{A}\vec{x} = \overrightarrow{\Delta AIC}$$

which can be solved for regression coefficient vector $\vec{x}$. These coefficients represent the sensitivity of ΔAIC to each of the elementary connections; a larger value of x_k indicates that ΔAIC is more sensitive to elementary connection k.

4.3 Results and Discussion

4.3.1 High-level trends in the low noise, high sampling frequency synthetic dataset

We performed the fitting and analysis as described in the previous section, using first the Branched Pathway model with CoV = 0.05, nT = 50, and 50 noisy replicates. We fitted each of the 87 regulatory models described in Table 4.2 to each of the 50 noisy replicates, producing 4350 fitted models. For each fitted model, we calculated the ΔAIC as described previously. We then calculated the metrics described in Section 4.2.

We first looked at aveΔAIC_m with the intent of examining some of the trends in model structure across candidate models. The values of this metric was ranked and ordered, yielding the results shown in Table 4.3 and Figure 4.1.

Table 4.3. Ranked aveΔAIC_m for 50 noisy datasets (CoV = 0.05, nT = 50)
The true model with connections {3;4} and {4;3} is shaded.

Rank	Model	Score	Rank	Model	Score	Rank	Model	Score	Rank	Model	Score
1	M-39	91.03	23	M-11	77.85	45	M-55	38.29	67	M-13	29.07
2	M-10	88.69	24	M-35	77.81	46	M-12	37.33	68	M-32	28.66
3	M-18	87.93	25	M-47	77.80	47	M-16	37.27	69	M-04	27.92
4	M-54	87.46	26	M-71	77.00	48	M-27	37.16	70	M-77	27.72
5	M-34	85.92	27	M-26	75.70	49	M-15	37.00	71	M-67	27.53
6	M-40	85.53	28	M-74	75.45	50	M-33	36.70	72	M-63	26.07
7	M-78	85.53	29	M-65	74.61	51	M-05	36.67	73	M-62	25.80
8	M-25	85.25	30	M-60	74.32	52	M-45	35.69	74	M-84	25.74
9	M-80	85.24	31	M-42	74.14	53	M-14	34.82	75	M-72	25.02
10	M-44	84.70	32	M-73	73.99	54	M-23	34.81	76	M-21	24.76
11	M-51	82.65	33	M-85	73.31	55	M-61	34.19	77	M-09	24.01
12	M-75	82.52	34	M-86	70.66	56	M-28	34.14	78	M-53	22.83
13	M-57	82.25	35	M-66	66.79	57	M-69	33.81	79	M-17	21.68
14	M-64	82.19	36	M-37	52.78	58	M-58	33.70	80	M-20	21.67
15	M-46	81.91	37	M-52	45.43	59	M-06	33.17	81	M-29	20.62
16	M-68	81.12	38	M-36	45.18	60	M-19	32.86	82	M-48	19.68
17	M-83	81.06	39	M-41	40.12	61	M-82	31.11	83	M-24	12.79
18	M-70	79.88	40	M-08	39.89	62	M-56	31.00	84	M-59	12.33
19	M-30	78.53	41	M-22	39.82	63	M-50	30.96	85	M-02	8.82
20	M-87	78.46	42	M-79	39.54	64	M-43	30.52	86	M-03	4.64
21	M-81	78.14	43	M-07	39.35	65	M-38	29.93	87	M-01	0.00
22	M-76	77.96	44	M-31	38.46	66	M-49	29.54			

We note a few characteristics the graph in Figure 4.1. First, the vast majority of the models produce a value of aveΔAIC_m indicating that the model dynamics are clearly distinguishable from the unregulated model, as can be inferred from the consistent lack of overlap between the distributions for each model and the reference value, ave$\Delta AIC_{m=1} = 0$. Second, we note the rapid drop in aveΔAIC_m for models ranked 35 or below. This suggests that certain regulatory connections may be acting as a strong driver of model performance, and without them, the model performs noticeably worse.

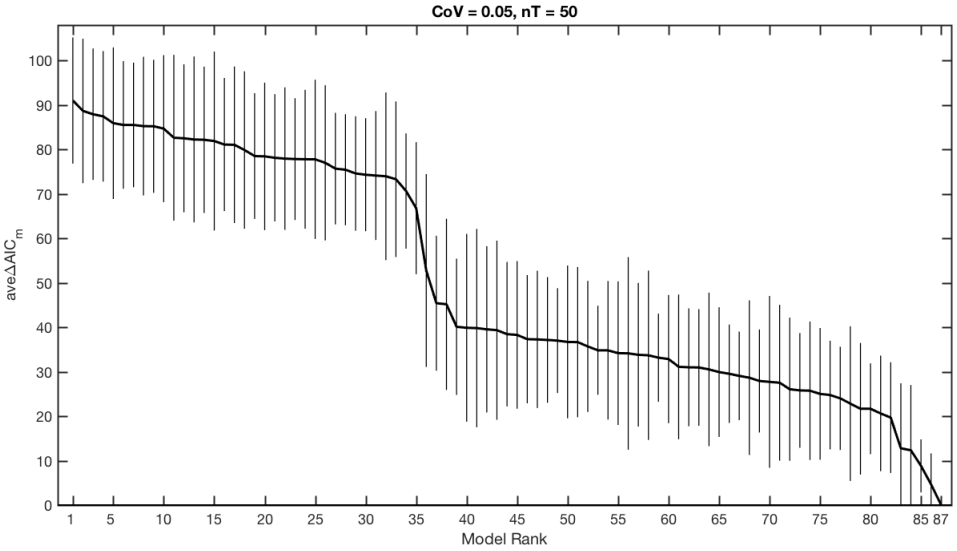

Figure 4.1. Quantitative model performance for CoV = 0.05 and nT = 50 datasets
The thick solid line represents the average model ΔAIC across replicates, and the thin error bars show the corresponding sample standard deviation. Model results are plotted by decreasing $ave\Delta AIC_m$, corresponding to the order shown in Table 4.3

4.3.2 Sensitivity of model performance to elementary regulatory connections in the low noise, high sampling frequency dataset

As described in the Section 4.2, we performed regressions to determine the sensitivity of $\Delta AIC_{m,n}$ to each of the elementary regulatory connections. The results of these regressions are shown in Table 4.4.

Table 4.4. Regression against the participation of elementary regulatory connections
Models fitted to data of CoV = 0.05 and nT = 50 to determine sensitivity of ΔAIC to elementary connections. The true regulatory connections are shaded.

CoV = 0.05, nT = 50

Connection	{1; 3}	{1; 5}	{2; 2}	{2; 4}	{2; 5}	{3; 2}	{3; 3}	{3; 4}	{4; 2}	{4; 3}	{4; 4}
Coefficient	9.84	5.12	11.01	11.52	16.42	12.31	14.99	6.13	66.32	60.13	13.74

One notable result observed in Table 4.4 is that while the true connection {4;3} has a high coefficient, it is actually the second highest, behind the connection

{4;2}. Further, the true connection {3;4} has one of the lowest sensitivities in the list, indicating that detecting this interaction is difficult to capture using $\Delta AIC_{m,n}$. We note that in the underlying model, the connection represented by {4;3} is an activation interaction, whereas {3;4} represents an inhibition. What is not discernable from this model and analysis is if inhibitors are inherently difficult to capture robustly with LK-DFBA, or if this is a byproduct primarily of the stoichiometry and regulatory network in this particular model.

4.3.3 Performance of a greedy search over model space for replicates of the low noise, high sampling frequency dataset

We further considered the case of a greedy model selection search, in which ΔAIC is optimized by selectively adding a single elementary connection at each round of optimization. For a given round in the search, the current model is compared against a pool of models constructed by adding to each a single eligible elementary connection. Whichever model in this pool most improves the ΔAIC from the current model is selected as the reference for the next round, and the procedure repeated. If the current model outperforms all the candidate models, then the search terminates. Using the $\Delta AIC_{m,n}$ data used to construct Table 4.3 and Figure 4.1, we can determine the behavior we would have observed for this search procedure for each of the noisy datasets. Figure 4.2 depicts what the search procedure results would look like for a single noisy dataset using hypothetical ΔAIC_m values as an example.

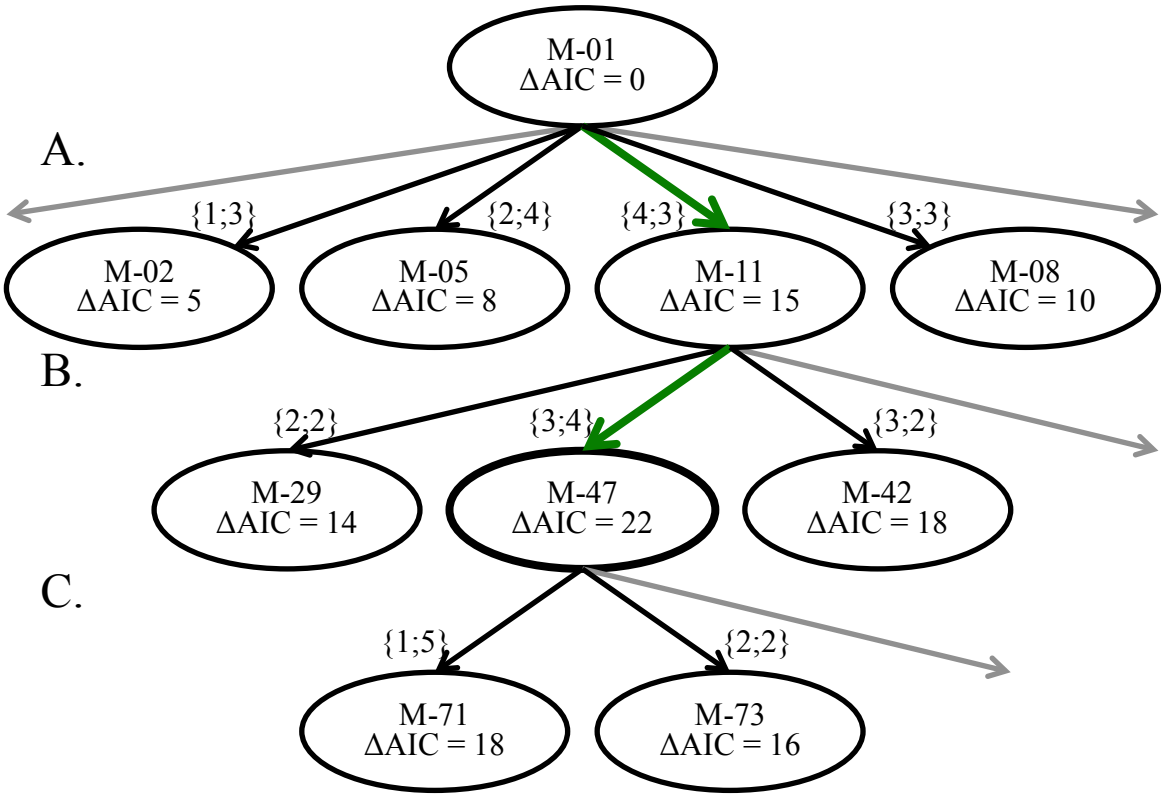

Figure 4.2 A graphical depiction of model selection using greedy search for a single noisy dataset with ΔAIC

Nodes indicate model structures, and edges indicate candidate models constructed by modifying the parent model. Green edges indicate steps taken by the search procedure in this example. Grey edges indicate other models not shown in the diagram for clarity.

A. The ΔAIC of the unregulated model is compared against the ΔAIC of models constructed by adding a single elementary connection. In this case, the model with {4;3} most improves ΔAIC, and model M-11 is chosen as parent for the next round.

B. From model M-11, new candidates are generated by adding an additional eligible regulatory connection. A smaller subset of elementary connections is eligible for consideration at this stage. In this case, the greatest increase in ΔAIC is found by adding connection {3,4}.

C. The algorithm terminates when either no new candidate models can be constructed, or the current model outperforms all the candidate models. In this case, ΔAIC of M-47 exceeds that of M-71 and M-73, and the final result from the search is M-47.

In the example provided in Figure 4.2, a single execution of the search method led to a model with connections {3;4} and {4;3}. We are interested in the range and distribution of behaviors in practice. We simulated this search procedure independently over each of the 50 replicates with CoV = 0.05 and nT = 50 using

the observed $\Delta AIC_{m,n}$ values to identify the model that locally maximized $\Delta AIC_{m,n}$. The distribution of results we observed is shown in Figure 4.3.

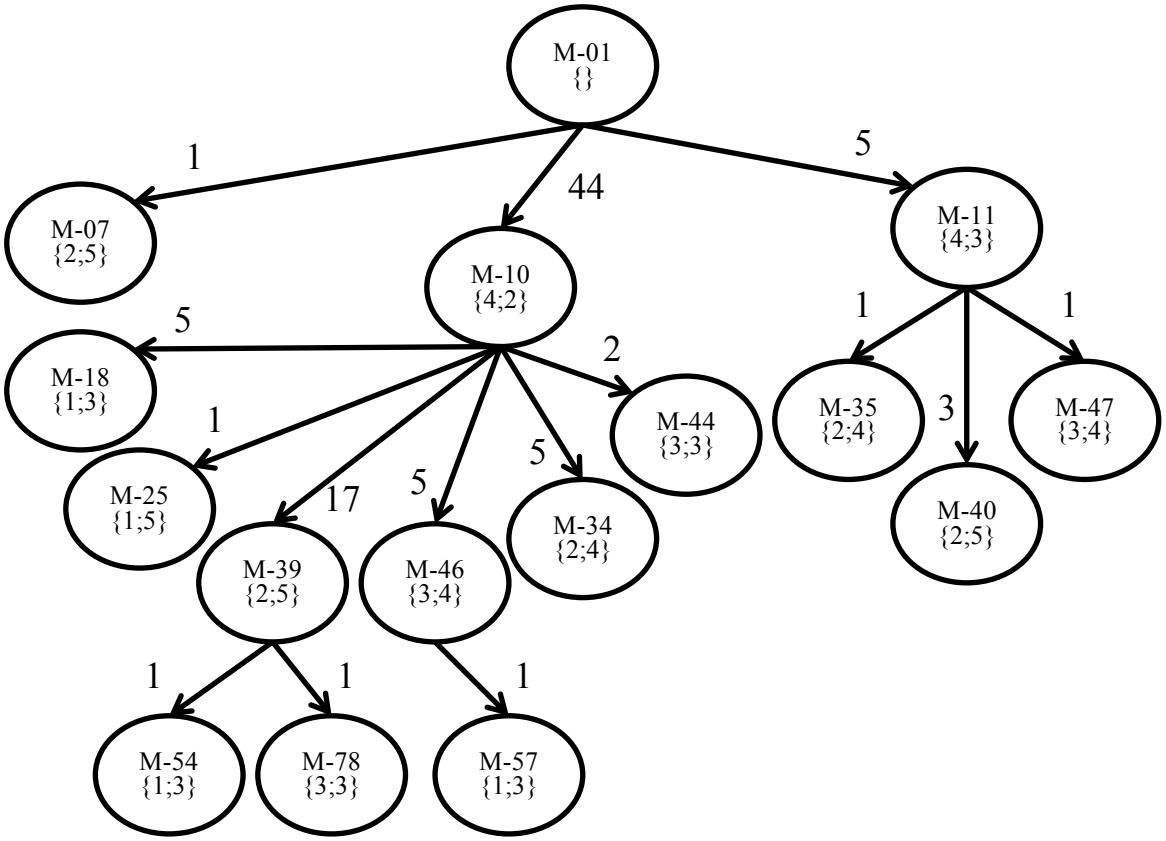

Figure 4.3. The distribution of results for a greedy model selection search using $\Delta AIC_{m,n}$ for 50 noisy replicates (at CoV = 0.05 and nT = 50)
Nodes represent models, and edges indicate instances in which the search selected that model over the parent node. At each stage in the search, an additional elementary connection has been added to the model to improve the ΔAIC. The number next to each edge indicates the frequency with which that edge was added to the parent model. For some replicates, the search terminated at an intermediate node in the graph. The label within each node indicates the model number and the regulatory connection added relative to the parent node.

We observe a few major trends in the greedy search results shown in Figure 4.3. First, by far the most frequent first step is to add {4;2} to the model. This happens in 44 of the 50 cases, compared to the 5 cases in which the correct connection of {4;3} is added. In one case, {2;5} was added in the first step. However, this

connection was added on the second step in 20 of the other cases, which made it the most common connection added at that stage. Looking at the sensitivities shown in Table 4.4, we observe that these tendencies match the sensitivity of $\Delta AIC_{m,n}$ to each connection: {4;2} had by far the highest sensitivity, and {2;5} the next highest, after {4;3} (which by our criteria is incompatible with {4;2}). A few other common occurrences were the addition of connections where x_1 is the controller, or v_5 the target.

4.3.4 High-level trends and sensitivity to elementary regulatory connections in the low sampling frequency datasets and in the high noise datasets.

For our initial assessment described this far, we used low noise (CoV = 0.05) and high sample frequency (nT = 50), leading to conditions under which we expected more ideal behavior. We repeated our assessment using two additional conditions, again with 50 noisy replicates. The first additional data set reduced sampling to nT = 20 (with CoV = 0.05). The second additional data set explored the effect increased noise of CoV = 0.25 (with nT = 50). The results of these analyses are shown in Table 4.5, Table 4.6, Figure 4.4, Table 4.7, Figure 4.5, and Figure 4.6.

Table 4.5. Ranked aveΔAIC_m for 50 noisy datasets (CoV = 0.05 and nT = 20)
The true model with connections {3;4} and {4;3} is shaded.

Rank	Model	Score	Rank	Model	Score	Rank	Model	Score	Rank	Model	Score
1	M-10	35.79	23	M-42	25.59	45	M-41	8.96	67	M-33	2.91
2	M-39	33.35	24	M-81	24.90	46	M-04	7.66	68	M-20	2.54
3	M-18	31.67	25	M-76	24.44	47	M-22	7.06	69	M-02	2.36
4	M-44	31.61	26	M-65	24.31	48	M-16	7.05	70	M-63	2.03
5	M-34	31.47	27	M-83	23.98	49	M-27	6.92	71	M-24	1.88
6	M-25	31.13	28	M-87	23.34	50	M-19	6.91	72	M-29	1.39
7	M-11	30.98	29	M-60	23.26	51	M-31	6.52	73	M-58	0.90
8	M-40	30.25	30	M-71	23.17	52	M-52	6.51	74	M-69	0.73
9	M-46	29.36	31	M-74	23.06	53	M-09	6.23	75	M-56	0.70
10	M-80	29.01	32	M-73	22.33	54	M-45	5.82	76	M-03	0.54
11	M-78	28.91	33	M-66	22.00	55	M-38	5.81	77	M-62	0.31
12	M-54	28.60	34	M-86	19.72	56	M-55	5.53	78	M-77	0.16
13	M-75	28.57	35	M-85	19.27	57	M-21	5.10	79	M-01	0.00
14	M-68	28.36	36	M-08	14.93	58	M-14	5.04	80	M-49	-0.05
15	M-51	28.19	37	M-37	14.68	59	M-13	4.99	81	M-50	-0.06
16	M-30	28.14	38	M-12	10.78	60	M-43	4.54	82	M-67	-0.53
17	M-64	27.96	39	M-06	10.04	61	M-79	4.53	83	M-48	-1.57
18	M-26	27.03	40	M-15	9.67	62	M-32	3.69	84	M-82	-1.72
19	M-35	26.56	41	M-07	9.62	63	M-28	3.66	85	M-72	-2.08
20	M-57	26.17	42	M-05	9.53	64	M-17	3.52	86	M-59	-2.42
21	M-47	25.97	43	M-23	9.42	65	M-61	3.16	87	M-84	-3.77
22	M-70	25.81	44	M-36	9.09	66	M-53	2.92			

Keeping in mind our observations in Figure 4.1, we note several features of Figure 4.4A and 4.4B. In the low sampling frequency case, we observe that the graph retains the marked shift around models ranked 35 and worse, but the overall aveΔAIC_m values are lower. The result of this is that the lower aveΔAIC_m models may not be consistently distinguishable from the unregulated model. However, we can still clearly distinguish certain models from the base model and expect similar trends in the regulatory interactions to those observed previously.

Table 4.6. Ranked aveΔAIC_m for 50 noisy datasets (CoV = 0.25 and nT = 50)
The true model with connections {3;4} and {4;3} is shaded.

Rank	Model	Score	Rank	Model	Score	Rank	Model	Score	Rank	Model	Score
1	M-10	0.44	23	M-47	-6.60	45	M-78	-8.72	67	M-50	-11.0
2	M-01	0.00	24	M-19	-7.04	46	M-54	-8.80	68	M-63	-11.1
3	M-11	-0.72	25	M-02	-7.09	47	M-13	-8.86	69	M-53	-11.2
4	M-06	-1.87	26	M-46	-7.47	48	M-65	-8.87	70	M-52	-11.2
5	M-40	-2.27	27	M-80	-7.59	49	M-17	-8.88	71	M-64	-11.4
6	M-05	-3.79	28	M-14	-7.61	50	M-45	-9.05	72	M-48	-11.8
7	M-39	-3.94	29	M-38	-7.76	51	M-43	-9.10	73	M-75	-12.1
8	M-03	-4.24	30	M-22	-7.96	52	M-76	-9.10	74	M-69	-12.4
9	M-25	-4.38	31	M-27	-8.07	53	M-44	-9.26	75	M-83	-12.6
10	M-12	-4.57	32	M-31	-8.14	54	M-74	-9.30	76	M-67	-13.0
11	M-34	-4.91	33	M-23	-8.22	55	M-24	-9.40	77	M-86	-13.1
12	M-30	-4.96	34	M-21	-8.23	56	M-66	-9.64	78	M-85	-13.1
13	M-26	-4.97	35	M-36	-8.23	57	M-71	-10.2	79	M-79	-13.3
14	M-08	-5.03	36	M-57	-8.39	58	M-55	-10.4	80	M-77	-13.5
15	M-35	-5.19	37	M-16	-8.45	59	M-62	-10.5	81	M-61	-13.7
16	M-04	-5.32	38	M-33	-8.46	60	M-28	-10.6	82	M-58	-13.8
17	M-07	-5.33	39	M-32	-8.49	61	M-49	-10.6	83	M-72	-14.2
18	M-42	-5.99	40	M-41	-8.50	62	M-70	-10.8	84	M-87	-14.5
19	M-18	-6.07	41	M-29	-8.54	63	M-56	-11.0	85	M-59	-14.6
20	M-15	-6.40	42	M-51	-8.57	64	M-81	-11.0	86	M-82	-15.7
21	M-09	-6.56	43	M-60	-8.64	65	M-73	-11.0	87	M-84	-16.2
22	M-37	-6.59	44	M-20	-8.65	66	M-68	-11.0			

However, in the high noise case, the graph has lost the marked shift around Rank 35. Further, one of the highest ranked models is the unregulated model, M-01, and the majority of models have lower aveΔAIC_m. More notably, nearly the entire range of models falls within a standard deviation of each other, indicating an expected general difficulty in differentiating between models with any reliability.

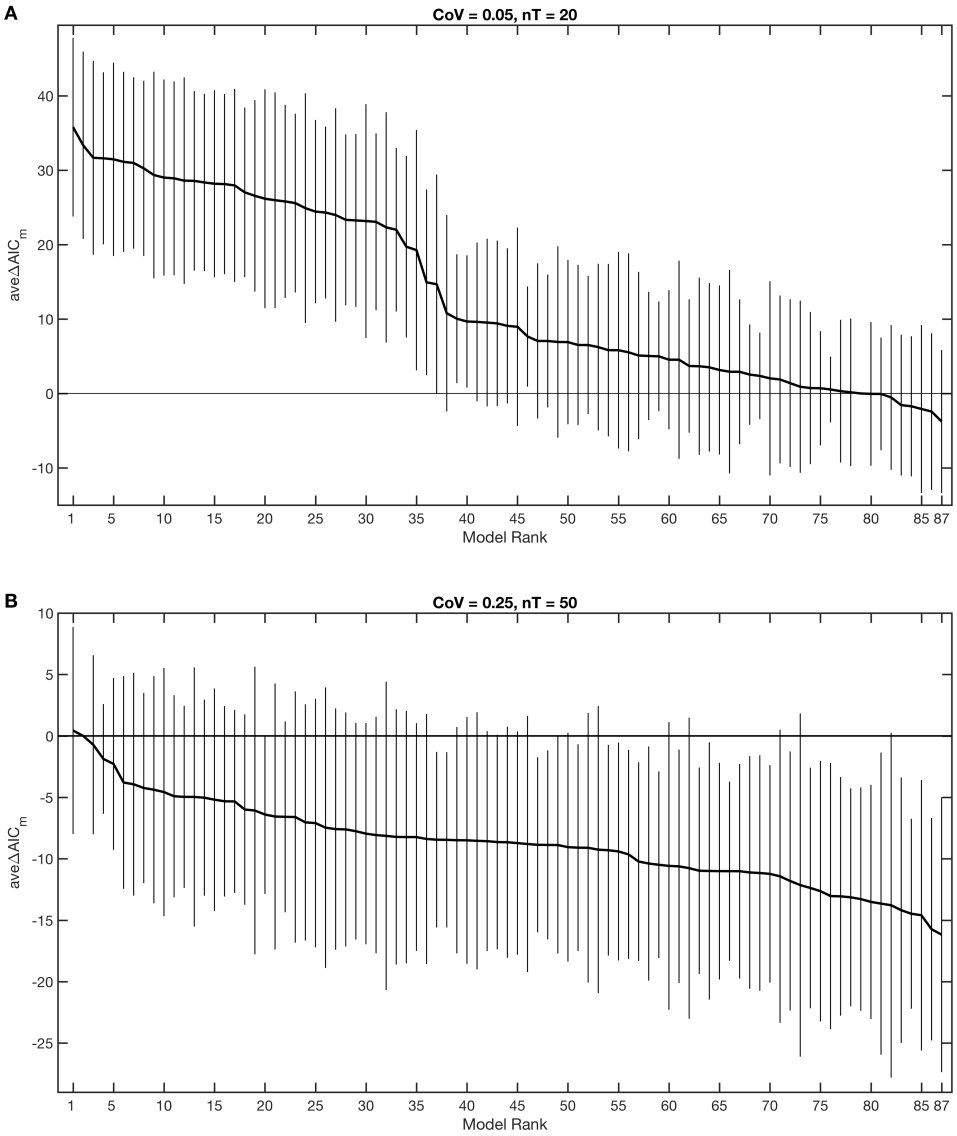

Figure 4.4. Quantitative model performance for aveΔAIC_k for low sampling frequency and high noise datasets

The thick solid line represents the average model ΔAIC across replicates, and the thin error bars show the corresponding sample standard deviation.

A. Average and standard deviation of $\Delta AIC_{m,n}$ at low sampling (CoV = 0.05 & nT = 20). Model results are plotted by decreasing aveΔAIC_m, corresponding to the order shown in Table 4.6.

B. Average and standard deviation of $\Delta AIC_{m,n}$ at high noise (CoV = 0.25 & nT = 50). Model results are plotted by decreasing aveΔAIC_m, corresponding to the order shown in Table 4.7.

We compare aveΔAIC_m trends across the nominal, low frequency, and high noise data sets by calculating Pearson (P) and Spearman (S) correlations for each pair. Comparing the nominal and low frequency cases, we observe P = 0.9635 and

S = 0.9254. Comparing the nominal and high noise cases, we observe P = 0.1567 and S = 0.2088. Finally, comparing the low frequency and high noise cases, we observe P = 0.3564 and S = 0.4361. These indicate that the low frequency case in Figure 4.4A mostly matches the trends in the nominal case, whereas the high noise case is producing divergent results and that producing robust trends from this data for Table 4.6 and Figure 4.4B is difficult. However, despite the high noise, there is still detectable information in the dataset: if the noise completely dominated the information in the high noise cases, we would expect comparably low correlations with both nominal and low frequency.

Table 4.7. Regression against the participation of elementary regulatory connections for low frequency or high noise
The true regulatory connections are shaded.

CoV = 0.05, nT = 20

Connection	{1; 3}	{1; 5}	{2; 2}	{2; 4}	{2; 5}	{3; 2}	{3; 3}	{3; 4}	{4; 2}	{4; 3}	{4; 4}
Coefficient	1.059	-0.083	0.280	1.003	3.720	0.446	2.261	-0.659	27.043	23.903	1.039

CoV = 0.25, nT = 50

Connection	{1; 3}	{1; 5}	{2; 2}	{2; 4}	{2; 5}	{3; 2}	{3; 3}	{3; 4}	{4; 2}	{4; 3}	{4; 4}
Coefficient	-3.425	-3.759	-4.375	-3.756	-2.868	-4.714	-5.136	-5.096	-1.662	-1.071	-4.278

Looking at Table 4.7, we observe that for both cases, the basic trends observed in Table 4.4 are recapitulated. We observe a Pearson correlation coefficient of 0.9963 between the nominal and low sampling coefficients, 0.8435 between the nominal and high noise coefficients, and 0.8641 between the low sampling and high noise coefficients, indicating strong overall agreement. The negative sign on the coefficients for the high noise data is in agreement with the tendency of

adding parameters to the model to lower $\Delta AIC_{m,n}$, seen in Figure 4.4B, but the largest (least negative) values again are seen for {4;2} and {4;3}, followed by {2;5}.

4.3.5 Performance of a greedy search over model space for replicates of the low sampling frequency synthetic dataset.

Since repeating the analysis with $ave\Delta AIC_m$ and sensitivity provided us with some insight into the effects of noise and sampling on overall behavior, we also repeat the greedy search analysis to explore the impact of data quality in individual noisy datasets. We first explore greedy search in the low sampling case, the results of which are shown in Figure 4.5. First, for an overwhelming fraction of the datasets (46/50), the search found that adding the connection {4;2} was the most effective first step. In fact, this single step was often enough to terminate the search, leading to the most common outcome (34/50 cases): the only regulatory interaction added to the model was {4;2}. Another 12 cases led to a second interaction being added, most frequently {2;5} (in 7 cases). The only other interaction to get added in the first round was {4;3}, accounting for only 4 cases. Interestingly, in only two cases (both resulting in M-44 as the final model) was X_3 identified as a controller metabolite.

This analysis is consistent with what we observed previously from the regression coefficients in Table 4.8. While both {4;3} and {4;2} had strong influences on

ΔAIC, of the two, {4;2} was more dominant. In addition, the elementary connections with X_3 as controller had some of the lowest ΔAIC sensitivities. As before in Table 4.4 and Figure 4.3, the results of the sensitivity analysis are reflected in the resulting behavior of the greedy search method.

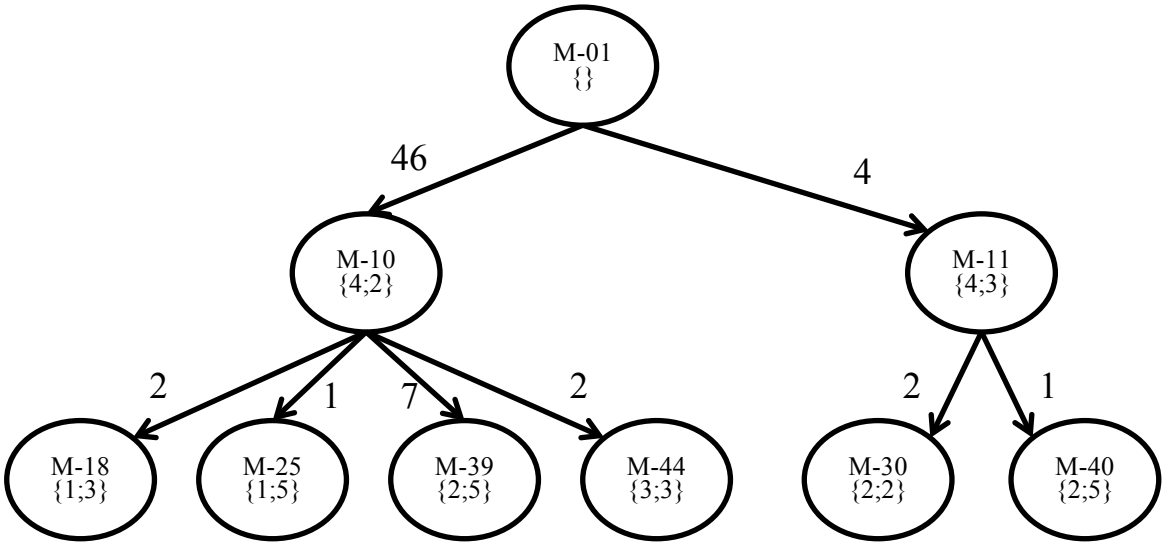

Figure 4.5. The distribution of results for a greedy model selection search using $\Delta AIC_{m,n}$ for 50 replicates at low frequency (CoV = 0.05 and nT = 20)
Nodes represent models, and edges indicate instances in which the search selected that model over the parent node. At each stage in the search, an additional elementary connection has been added to the model to improve the ΔAIC. The number next to each edge indicates the frequency with which that edge was added to the parent model. For some replicates, the search terminated at an intermediate node in the graph. The label within each node indicates the model number and the regulatory connection added relative to the parent node.

4.3.6 Performance of a greedy search over model space for replicates of the high noise synthetic dataset.

We explore next the greedy search under conditions of high noise, the results of which are shown in Figure 4.6. Two features of Figure 4.6 are particularly noteworthy. First, in 13 cases, the search terminated with Model M-01, indicating that none of the elementary interactions produced a sufficient reduction in residuals to offset adding additional model parameters and structure for that

dataset. Second, there was much greater variety in the initial steps taken by the search. While in 18 cases, the usual choices of {4;2} and {4;3} were added, in 7 cases {4;4} was added, and in 12 cases, one of the other elementary connections was added. Only in 4 cases were models with two regulatory interactions added. This much flatter, broader tree structure is a reflection of the high degree of noise in the data, resulting in greater difficulty justifying model complexity with AIC, and the higher probability of ending up with a spurious structure due to over-fitting.

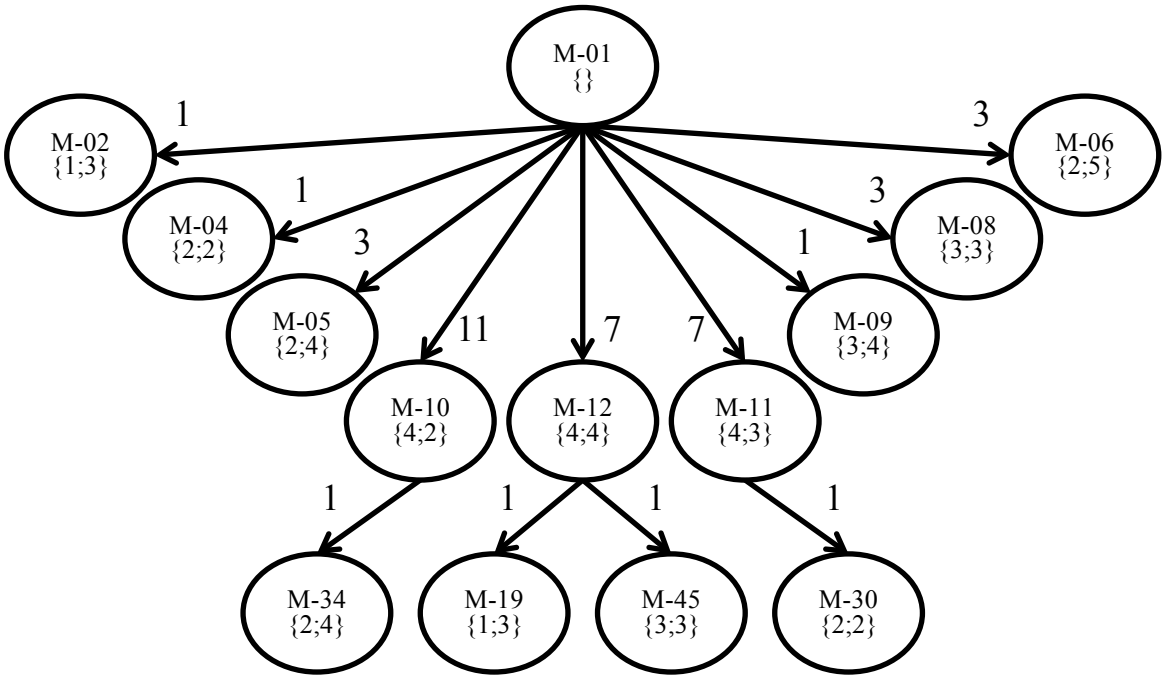

Figure 4.6. The distribution of results for a greedy model selection search using $\Delta AIC_{m,n}$ for 50 replicates at high noise (CoV = 0.25 and nT = 50)
Nodes represent models, and edges indicate instances in which the search selected that model over the parent node. At each stage in the search, an additional elementary connection has been added to the model to improve the ΔAIC. The number next to each edge indicates the frequency with which that edge was added to the parent model. For some replicates, the search terminated at an intermediate node in the graph. The label within each node indicates the model number and the regulatory connection added relative to the parent node.

While the precise details vary between the 3 conditions for CoV and nT, we observe similar and consistent trends between all 3 cases. Greedy search results closely matched the trends observed in the ΔAIC sensitivity analysis via regression. Further, in all three cases, we identified extremely similar trends concerning the importance of various regulatory connections. X_4 was a strongly important controller metabolite, and X_3 was very difficult to reliably detect as a regulator.

We note that in the underlying model, X_4 is an activator and X_3 is an inhibitor. The models we are fitting have biomass generation as their objective, but this objective is hampered by X_3 inhibition. As a result, the optimization problem we solve to simulate the model time course has an incentive to route fluxes and accumulation to minimize the impact of this inhibition on the resulting biomass generation. Coupled with the parameter optimization of the LR+ methods, this may bias parameter optimization to better fit the overall concentration behaviors at the expense of accurately modeling the effect of this single regulatory interaction. Such an error compensation effect would lead to a more spurious relationship between simulation output and inhibitory regulatory parameters, leading to the inability to reliably detect this regulatory interaction.

In the case of Figure 4.6, we observed much less consistent behavior in the greedy search. This is to be expected when the noise is greatly increased: higher

noise increases the likelihood that a spurious regulatory connection may appear more favorable at a given stage, leading to a more diverse distribution of resulting models, depending on the specific noisy dataset. Given the behavior observed in Figure 4.4B, this perhaps is to be expected: across all models, the degree of variability is such that it is not clear that ΔAIC can be improved by moving to another model, and as a result it is more difficult to reliably distinguish between them.

4.4 Conclusions

In this chapter, we explored the impact of regulatory connections on LK-DFBA. We identified which connections have a greater impact on model performance and explored whether or not we can use the LK-DFBA to identify the correct regulatory connections. We assessed the performance of a greedy search using ΔAIC for model identification in the context of our modeling approach, looked at the overall trends across noisy synthetic datasets, and at the behavior for individual data sets. This analysis gives us a context for interpreting the results of our methods on experimental data, which are too limited in availability and quality to perform a robust validation.

While we were able to consistently detect one of the true regulatory interactions from the data, we had a difficult time capturing the other. In addition, the interaction we could capture had a tendency of getting confused with a similar

regulatory interaction, which uses the same controller metabolite to instead target a flux one step upstream from the correct target. A possible consequence of this is that by targeting a flux upstream of the correct target (v_2 instead of v_3), the resulting model is able to use that regulatory connection to influence the dynamics of both X_2 and X_3, instead of just X_3. An additional mass action constraint tying v_3 to X_2 allows these concentrations to be partially decoupled, negating the risk during parameter fitting of losing the ability to capture X_3 by targeting the wrong flux. The interaction we had difficulty capturing is an inhibitor reaction, but it is unclear from the analysis performed if this was difficult because of this inhibition role, structural and stoichiometric limitations of the model used to generate synthetic data, or the particular set of model parameters and initial conditions used to generate the underlying time course.

We explored performing model identification using a greedy search method to optimize ΔAIC, and compared that with the trends observed looking at the exhaustive analysis. We observed that the connections most frequently added by greedy search were those that displayed the highest effect in the sensitivity analysis, and that in general the greedy search exhibited trends consistent with the more exhaustive analyses.

As might be expected, model identification performed less consistently when the sampling rate was reduced or noise increased, but even in these cases, the

analysis still yielded similar trends to those we observed when using higher quality data. For example, the two regulatory connections we consistently identified in the nominal conditions were also strong drivers in the low sampling frequency and high noise cases. Of the two cases, higher noise produced a much more severe detrimental effect on model identification, making it difficult to justify adding the additional parameters necessary to add regulatory connections to the model in many cases. This highlights how it is important to be mindful of the degree of noise in the data, to be wary of over-fitting, and to adjust confidence in the learned model structure as appropriate.

For larger models, we note that additional issues may make this task more challenging. These include limitations of data availability and quality, the combinatoric increase in candidate model search space as model size increases, and bias in model dynamics due to the assumptions of LK-DFBA. Depending on the specific circumstances, other approaches may be helpful, appropriate, or complementary for further tackling this challenge. Further work along those lines may be merited, but is outside the scope of the current chapter.

Based on the analysis performed in this chapter, we conclude that a greedy search method to optimize ΔAIC with LK-DFBA appears to be a reasonable approach for identifying unknown regulatory interactions in the model from available data, though it comes with limitations. It may be reliable for producing

models with a good ability to simulate the underlying data, but the means by which this accomplished may not quite accurately represent the underlying regulatory structure. Competing candidates for regulatory connections with similar performance should be noted, and additional targeted experimental validation may be able to discern which of the candidates is most correct. Additional care is warranted when a higher degree of noise is present in the data, and methods such as bootstrapping or cross-validation would likely be appropriate. However, as long as these considerations have been accounted for, LK-DFBA may be a useful and valid way of generating these putative interactions and for guiding experimental design.

4.5 References

1 Mahadevan, R., Edwards, J. S. & Doyle, F. J., 3rd. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophys J* **83**, 1331-1340, doi:10.1016/S0006-3495(02)73903-9 (2002).

2 Nikolaev, Y. V., Kochanowski, K., Link, H., Sauer, U. & Allain, F. H. Systematic Identification of Protein-Metabolite Interactions in Complex Metabolite Mixtures by Ligand-Detected Nuclear Magnetic Resonance Spectroscopy. *Biochemistry* **55**, 2590-2600, doi:10.1021/acs.biochem.5b01291 (2016).

3 Goncalves, E. *et al.* Systematic Analysis of Transcriptional and Post-transcriptional Regulation of Metabolism in Yeast. *PLoS Comput Biol* **13**, e1005297, doi:10.1371/journal.pcbi.1005297 (2017).

4 Link, H., Kochanowski, K. & Sauer, U. Systematic identification of allosteric protein-metabolite interactions that control enzyme activity in vivo. *Nat Biotechnol* **31**, 357-361, doi:10.1038/nbt.2489 (2013).

5 Akaike, H. in *Second International Symposium on Information Theory.* (eds B. N. Petrov & F. Csaki) 267-281 (Akadémiai Kiado).

6 Voit, E. O. & Almeida, J. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics* **20**, 1670-1681, doi:10.1093/bioinformatics/bth140 (2004).

7 Ishii, N., Nakayama, Y. & Tomita, M. Distinguishing enzymes using metabolome data for the hybrid dynamic/static method. *Theor Biol Med Model* **4**, 19, doi:10.1186/1742-4682-4-19 (2007).

8 Dromms, R. A. & Styczynski, M. P. Improved metabolite profile smoothing for flux estimation. *Mol Biosyst* **11**, 2394-2405, doi:10.1039/c5mb00165j (2015).

Chapter 5. Future Directions

5.1 Introduction

The goal of this thesis was to develop computational tools that can improve the quality and feasibility of metabolic engineering efforts. By improving the accuracy of metabolic models, we can more reliably and consistently produce accurate strain design predictions. Improving these predictions in turn reduces the time, effort, and materials spent in the lab by allowing metabolic engineers to more consistently produce their desired outcomes, rather than relying as heavily on trial and error. Producing a more accurate model comes with its own time investment and difficulties, and there is value in being able to produce a model with comparable accuracy in less time, or with less data.

As genome-scale snapshots of metabolism, metabolomics data are well-suited for aiding both types of improvements. Metabolomics data reflect the dynamics and regulation that determine the real behavior of metabolism under the conditions present when the data were collected, which may not be apparent in assays measuring a single, pre-selected target[1]. Additionally, there is a much greater amount of data collected in an individual metabolomics assay than a targeted assay; collecting the same amount of data separately from targeted assays would take orders of magnitude longer, and this metabolomics is an

appealing approach to generate the data needed to train genome-scale metabolic models[2].

The work presented in this thesis addresses both model accuracy and feasibility. In the impulse smoothing work, I demonstrated that the impulse function from the gene transcription literature could also be used to smooth metabolite data, and that doing so allows for more accurate and interpretable results than the commonly used polynomial and rational functions.

In the chapter on Linear Kinetics-Dynamic Flux Balance Analysis (LK-DFBA), I introduced a new modeling framework that incorporates attractive computational aspects of FBA[3], DFBA[4], and iFBA[5] while producing results that are competitive with popular ODE-based frameworks, such as BST-style models with generalized mass action kinetic rate laws[6]. With it, I was able to model metabolite dynamics, incorporate regulatory interactions, and (as a result) produce interesting model behavior that reflected expected dynamics. LK-DFBA therefore captures detail outside the scope of the original FBA, and unlike ODE models, DFBA, or iFBA, retains one of FBA's most attractive features: its membership in the subclass of convex optimization problems designated as Linear Programs.

I further explored some basic properties of LK-DFBA and the influence of regulatory interactions on model performance. Representing these interactions is

important for capturing the behavior of real metabolic networks. This analysis gives us a sense for the impact these interactions have on the model behavior and on the ability of the correct model to make distinguishably correct predictions of metabolic behaviors.

While I have laid the groundwork for a promising modeling framework, and I demonstrated its reasonableness and its viability, my work as it stands is still far from producing strain predictions. Next steps need to show that LK-DFBA really can be used for metabolic strain design. Specifically, these next steps need to build on foundation established in this thesis by taking real metabolomics data, incorporating it into a genome-scale metabolic model implemented in LK-DFBA, and using it to produce actual strain predictions. These predictions then need to be implemented in the lab, and compared against the model predictions and the original metabolomics data. This will allow us to definitively answer two questions. First: how well does LK-DFBA work in practice? Second: can we use the results of this analysis to systematically improve our model and strain design efforts, closing the gap between predictions and experimental results? In this chapter, I describe what these efforts entail and what we might expect for their outcomes.

It is worth noting that while the specific trajectory described in the previous paragraph is necessary to see the goals behind LK-DFBA brought to fruition,

there are additional side paths that may provide useful further improvements. That is, their pursuit may improve the quality or the feasibility of producing a specific metabolic model, but doing so is likely not necessary for the overall process. Such efforts may also be academically interesting or have applications outside the scope of LK-DFBA or metabolic engineering, and as such may be worth pursuing for those reasons as well. I explore these ideas in this chapter as well.

5.2 Towards Strain Design and Experimental Validation

In this first section, I describe in detail the series of steps necessary to experimentally validate LK-DFBA by using it to produce real, testable predictions. I discuss some of the main considerations relevant to each step of this procedure, and provide recommendations on how best to proceed. These steps are as follows: first, using in-house metabolomics data and information from the literature, a genome-scale model for an organism of interest must be constructed in LK-DFBA. The resulting model must then be used with an appropriate strain-design tool to identify a set of genetic engineering interventions for the modeled organism, to satisfy some metabolic engineering objective. The resulting predictions must then be experimentally tested by genetically engineering the modeled organism as dictated. Follow-up metabolomics assays and appropriate targeted assays can then be used to assess and confirm the model predictions. From here, the model can be further refined and tested as necessary.

5.2.1 Expanding to a genome-scale model

In Chapter 3, where I discussed the development and validation of LK-DFBA, I worked with two models. The first was a modified version of a simple model from Biochemical Systems Theory, the Branched Pathway Model. The second was a model of glycolysis and the pentose phosphate pathway in *Escherichia coli*. The first of these was a toy model, and has little real biological relevance. While the second was originally constructed from experimental data to simulate real metabolic pathways, the scope of this model is small compared to the entirety of *E. coli* metabolism. A model of this scope is insufficient both for effective strain design and for effectively leveraging the breadth of data available from metabolomics analyses. Instead, a model that accounts for metabolism at the genome-scale is necessary.

The construction of genome-scale metabolic models is an active area of research, with a key goal of enabling the use of constraint-based models for metabolic engineering strain design. I reviewed these efforts earlier in the document, during my discussion of metabolic reconstruction in Section 1.4.1.2. Two particular aspects of this discussion bear repeating here: the range of organisms relevant to metabolic engineering, and the challenges inherent to working with models at this scale, regardless of the choice of constraint-based model.

One critical question moving forward is the choice of organism for metabolic engineering projects. There are a number of considerations that go into this selection. The first is model availability and quality. Databases of genome-scale models, such as the BiGG Database[7] provide ready access to models of the most popular and well-studied organisms. These include organisms relevant to metabolic engineering such as *E. coli*[8,9], *Saccharomyces cerevisiae*[10], *Chlamydomonas reinhardtii*[11], and *Bacillus subtilis*[12], as well as models for *Mycoplasma genitalium*[13], *Homo sapiens*[14,15], and *Mus musculus*[16] that are less relevant to metabolic engineering specifically.

Of these, the two most relevant organisms are *E. coli* and *S. Cerevisiae*. These two organisms are some of the most well-studied in biology, and many of our insights into metabolism and regulation come from extensive study of these organisms. They are also two of the most popular organisms in the bioprocessing industry.

Recent research has begun to explore algae such as *C. reinhardtii*[11] and *Emiliania huxleyi*[17] for metabolic engineering, due to a few attractive features of these organisms: algae are free-floating single-cell photosynthetic organisms that are known to accumulate and store lipids, making them an ideal candidates for renewably and sustainably producing lipophilic products such as biofuels. However, these organisms aren't as well-studied or as easy to work with as *E.*

coli or *S. cerevisiae*[17]. Modeling and engineering algae using LK-FBA would be a substantial contribution, but would best be approached by demonstrating the metabolic modeling and engineering pipeline first in another organism, then working with collaborators with experience in algae to expand into that application after[18,19].

Of *E. coli* and *S. cerevisiae*, I consider here the case of moving into engineering *S. cerevisiae* largely due to the wide range of existing biochemical knowledge[20] and characterization of yeast metabolism[2]. *S. cerevisiae* has multiple models available at the genome scale[21] that may be appropriate for adaptation into LK-DFBA, such as YeastNet[10], iMM904[22], and iIN800[23]. Implementing one of these genome-scale models into LK-DFBA will come with certain challenges, some of which I encountered when implementing the Chassagnole *E. coli* model[24]. In that chapter, I sidestepped the matter of finding an accurate dynamic flux distribution from the dynamic flux estimation (DFE) procedure in favor of using noise-added flux values[25]. This data will not normally be available, and a better choice of methods for addressing this issue is called for. One possible way to address this is to implement the procedure described by Chou *et al.* for estimating the relationship between metabolite and flux pairs[26]. A weakness of this method is the potential for uncertainty in the offset of the flux values when data points at low metabolite concentration values aren't available to ground the flux values to a specific scale via considerations such as mass action kinetics. However, the

linear regression parameter fitting method may be able to address this with little extra effort by selecting the offset such that b = 0 in the resulting linear regression.

Another challenge for implementing a genome-scale LK-DFBA model is closely tied to estimating dynamic flux distributions, and that is the problem of determining the ratio of forward and reverse reaction rates of reversible reactions. At any given point in time, the net flux is equal to the forward rate of reaction minus the reverse rate of reaction; adding a constant to both rates has no effect on the net flux. While this doesn't matter for calculating an overall flux distribution, it does affect efforts to model kinetics of the flux as metabolite concentrations shift due to accumulation, depletion, or regulation. This is fundamentally an issue of model identifiabilty, and is not at all specific to LK-DFBA. Handling these cases accurately will likely require additional data, potentially extracted from the literature. For example, existing kinetic studies of an enzyme in question may provide the necessary dynamics for fitting these individual parameters. Searching for these data would be called for when model performance can be shown to be sensitive to the parameters describing a particular flux.

One feature of LK-DFBA, which is shared with iFBA, is the ability to easily handle cases in which we have no data about whole sections of metabolism[5]. In this

instance, these metabolites can be treated as static or unknown quantities, and the overall model simplified to remove pooling fluxes and kinetic interaction parameters. That part of the metabolic network would revert to the behavior observed under regular FBA, subject to the dynamic fluxes at the borders of these network segments. The important kinetic interaction parameters for the overall model would then be those controlling the flux into the "static" network segments. While reverting sections of metabolism back to standard FBA may be less informative, without data with which to train the model for these segments, there is not much justification for including the extra detail. However, the option to include dynamics or regulation is still available, and may be justified if other sources of data are available, such as kinetic rate law data for individual fluxes.

For an *S. cerevisiae* model in particular, existing publications and databases will be important sources of information for training accurate models. In addition to the genome-scale stoichiometric models mentioned earlier in this section, databases such as KEGG[27], BRENDA[28], and SGD[29] are rich sources of information on *S. cerevisiae* biology. Existing models of *S. cerevisiae* metabolic dynamics are also useful sources of information; for example, the model of Hynne *et al.*[30] pulls data from multiple sources that could be used for reference as well[31-36].

The primary source of data for experiments will need to come from metabolomics assays. These chemical analyses will provide the widest cross-section of metabolism for data, but a few considerations for using this data should be highlighted.

First, experimental design should take into account a variety of conditions in order to provide a robust and representative picture of how yeast metabolism functions. Without this wide cross section, any model trained to the data will lack information that may be necessary to capture key regulatory or dynamic aspects of the underlying biology. Some appropriate experiments include growth on glucose, diauxic shift between carbon sources, heat shock, and salt shock. Experiments such as these capture a number of regulatory programs that influence metabolism in response to common environmental stresses, which may be relevant to a genetically modified strain of yeast.

However, using metabolomics data will in itself present a few difficult challenges; these challenges may prove to be the primary source of difficulty in training a reliable model. The first results from a fundamental limitation in metabolomics assays due to chemistry. While metabolomics seeks to be comprehensive and genome-scale, this is incredibly challenging in practice due to the sheer chemical diversity of metabolites compared to transcript (nucleic acids) and protein (polypeptides). As a result, different analytical methods are best-suited for

capturing different chemistries. LC-MS is preferential for large non-polar metabolites, while GC-MS lends itself to small polar metabolites. The choice of engineering objective should reflect these limitations; our lab uses GC-MS, and as a result engineering carbon and amino acid metabolism is more likely to be effective than, for example, lipid metabolism. A target such as succinate would be appropriate, given its chemistry; this choice has also been previously demonstrated and would provide ample opportunity for comparison against previously reported methods and results.

The second challenge with using metabolomics data stems from the difficulties inherent to quantifying untargeted data. As discussed in the Chapter 1, metabolomics data processing requires multiple steps before a final data matrix of metabolite abundances can be produced. Steps such as peak detection, peak alignment, and corrections for quality control all make this challenging, and will influence the resulting data quality. In addition, individual peak heights cannot be compared across metabolites to determine relative concentrations, and absolute quantification (i.e. of metabolite concentrations) cannot be performed without use of calibration curves from serial dilution experiments. This presents a difficult problem for model identification: how does one use semi-quantitative data to train and assess model performance? Questions of metabolite concentration scales will have a major impact on model parameterization and final performance, and this difficult problem will need to be addressed.

5.2.2 Strain Design using LK-DFBA

One of the main considerations when I designed LK-DFBA was retaining a linear structure for the constraints, while still adding the features necessary for incorporating metabolite dynamics. This is a feature I have repeatedly highlighted, for important reason. In this section, I describe in more detail the main way in which I envision taking advantage of this decision; later, I will discuss several other possibilities.

The parameterized genome-scale model produced by the process described in the previous section can then be used to engineering a strain that can subsequently be produced in the lab. In Chapter 1, I discussed one of the most notable cases of performing this task using constraint-based models, OptKnock[37,38]. For an initial experimental validaiton of LK-DFBA, OptKnock is both an appropriate and (I expect) effective tool for strain design.

As a Linear Program, FBA satisfies the criteria for Strong Duality, which provides guarantees on the optimality of the solution identified by the solver. OptKnock takes advantage of this by identifying the complementary Dual Linear Program, and constructing a constraint from it to enforce FBA optimality[37]. This allows the OptKnock algorithm to search over the design space while ensuring the inner problem remains optimized.

It is this Strong Duality condition that motivated the choice to implement strictly linear equations in LK-DFBA. By constructing the model equations such that the resulting system remains a linear program, we guarantee Strong Duality and therefore compatibility with approaches that take advantage of it, such as OptKnock. Further, LK-FBA uses the same stoichiometry as an FBA problem; the map OptKnock uses to tie design variables to system fluxes can be re-used in an LK-FBA model by expanding it across time points. No additional control variables or dummy variables must be introduced, which means that the problem difficulty only scales due to increase in size of the inner problem LP (which is far less detrimental than an increase in the number of integer design variables).

On this basis, implementation of a genome-scale LK-DFBA model in OptKnock is an appropriate step. While subsequent strain design tools[39-42] explore more sophisticated changes than gene knockouts, the simpler design space for knockouts and the ease of constructing knockouts strains experimentally makes OptKnock implementation an appropriate proof-of-principle.

5.2.3 Experimental Validation

The goal of the strain design step is to identify a set of genetic engineering interventions that will lead to a maximal increase in the production of a target compound. Once these interventions have been identified, they must be carried out to experimentally validate the design predictions. The resulting strains can be

assessed to determine the quality of the predictions, and to then improve the model by retraining it with the new data. This cycle of data acquisition, model training, strain design, and genetic engineering can be iteratively conducted to repeatedly refine the model until the desired performance has been attained.

Protocols for cell culture and gene knock-outs are well established in the literature[2,43,44], and our lab has previously reported on our metabolomics data acquisition and processing pipeline[45-48].

As noted above, an ideal target to use for this validation would be succinate. This was the objective in the original OptKnock publications[37,41], and has been described in other sources as well[44,49]. The existing experimental data for this choice is a useful point of comparison. Even outside of this literature context, it is an appropriate objective for a metabolic engineering project: succinate is commercially valuable, with uses in the production of polyethers and polyurethanes[50]. Engineering an *S. cerevisiae* strain to optimize succinate production is a reasonable objective that should provide ample opportunity for validating the strain design method and comparing it against the existing alternatives.

Performing this sort of experimental validation and assessment is a critical goal for next steps on this project, and is necessary to justify implementing additional models in LK-DFBA.

5.3 Model Improvements

In the second half of this chapter, I discuss a number of possible changes, refinements, and improvements that are relevant to the performance of LK-DFBA described in earlier chapters, but are not strictly necessary to achieving the ultimate goal of experimentally producing and validating this approach. Rather, these modifications are complementary, and are posited as potential means of improving model accuracy, taking advantage of the model structure to solve one of the model identification steps in a novel manner, or of relaxing assumptions inherent to the current implementation and exploring the consequences.

5.3.1 Accounting for biomass accumulation

An explicit goal when I developed LK-DFBA was to retain as much of the simplicity of FBA as reasonably possible, and consequently to preserve its attractive Linear Program structure. One of the earliest assumptions I invoked to achieve this end was an assumption that over the simulation interval, the biomass would not change sufficiently to influence the scaling of the variables represented in the model. The doubling time for organisms in culture can range from 30 minutes to 24 hours or more, whereas many of the fast dynamics in

metabolism are on the order of 30 seconds or less. For a model at this relatively short time scale a constant biomass assumption is reasonable. However, when the simulation interval becomes long enough that biomass changes by 10% or more, this assumption may begin to introduce unacceptable levels of inaccuracy into the results.

In the original non-linear program formulation of DFBA, the system equations were scaled by multiplying them by the current biomass as appropriate to reflect the change in compartment volume between the intracellular and extracellular environments. This accounts for several effects. First, enzyme kinetics and regulation are affected by local (intracellular) concentrations, not absolute metabolite mass. Second, as the amount of biomass increases, the relative rate of substrate uptake increases relative to the pool of substrate available in the extracellular media. Third, the concentration of metabolites that are not being constantly produced (do not have a source term) will be diluted as biomass volume increases; the total mass of metabolite remains constant, while the intracellular volume grows. When conditions are appropriate for exponential growth (or during chemostat culture), the specific growth rate predictions of FBA are often sufficient. But outside these conditions, scaling reactor volume against cellular volumes must be accounted for.

To account for these effects, biomass must be reintroduced into the LK-DFBA model as a scaling factor when the desired simulation interval is long enough to call for it. Since our model explicitly models biomass concentration when a biomass generation flux is available to facilitate this, the open question focuses rather on how to use this quantity to adjust the other quantities modeled. There are several ways of accomplishing this, each with advantages and drawbacks.

One method is to convert some linear constraints to bi-linear constraints by re-introducing current biomass as a multiplier on model quantities. For the LK-DFBA kinetics constraints, the flux distribution over a given simulation time step is constrained by the metabolite concentration at the beginning of the time step. Using the DFBA equations as a reference, the model quantities can be scaled by a factor of biomass concentration. Because both the biomass and fluxes or concentrations in these equations are also system variables, the result is that the LP becomes a Bi-linear Program (BLP). Previously, the conversion of the problem from an LP to a QP by penalizing the L_2-norm only affected the objective function (a modification which preserved the convexity of the solution space). The constraints remained linear equations. This is not the case in the BLP. The upside of this BLP conversion is that the only bi-linear terms over each time step are a result of a common quantity, the biomass concentration. The downside is that because this biomass value has a separate value over each interval, an additional bi-linear variable is introduced for each time step—quickly

compounding the complexity of the resulting optimization problem. For a simulation with nT = 100 or 200 time steps—and therefore 100 or 200 separate variables driving bi-linear constraints—this may prove to make the problem intractable. And at values of nT for which the problem becomes tractable, the value of nT may be too low to produce numerically reliable or consistent results. Depending on the specific model, this tradeoff between poor simulation accuracy and the tractability of the optimization may be irreconcilable. This concern led me to side step issues of biomass accumulation for my initial assessment of LK-DFBA.

An alternate method is to split the overall simulation interval into a series of sub-intervals, and to sequentially solve a separate simulation problem for this interval. For each sub-interval, the biomass concentration at the beginning of the interval can be treated as a constant multiplier on the appropriate quantities over that time step. Because this multiplier is a constant, rather than a variable subject to optimization, this removes the bi-linear optimization problem described in the previous chapter, and retains the LP structure. The disadvantage of this approach is that it requires the user to solve multiple independent optimization problems to cover the overall simulation interval. However, this problem scales up linearly with the number of sub-intervals specified, and should be as tractable as solving the individual optimization problems as a result.

An approach similar to the one just described might be to explore integrating the FBA problem into an ODE framework, using the DFBAlab approach of Barton *et al.*[18,51] Their research has focused on using a classic FBA problem to identify a flux distribution over the subintervals of an ODE simulation. Direct integration of FBA into the standard ODE solver framework available in e.g. MATLAB's `ode45()` function fails due to frequent attempts to evaluate the FBA model under conditions that produce an infeasible LP, which in turns disrupts calculating the right-hand-side of the differential equation[52]. The DFBAlab procedure modifies the FBA LP to handle these situations, returning values that appropriately penalize these cases in a way that allows the ODE solver to adjust accordingly. Incorporating a model represented in LK-DFBA instead of classic FBA may help improve the quality DFBAlab simulations.

5.3.2 Novel methods for parameter estimation

One of the key steps for generating a usable model in LK-DFBA is the identification of parameters representing the kinetics constraints. These constraints represent elements such as mass action kinetics and metabolite-dependent regulation, and including these constraints is critical for driving the metabolite dynamics in the resulting model. Once they are identified, the resulting model can be used for the downstream analysis and design steps, but parameter estimation is an important and non-trivial hurdle to reaching that point.

In the earlier chapter, I focused on using two common methods for parameter estimation to produce fitted models. The first method was the procedure described in dynamic flux estimation, which ultimately led to independent regression problems for each of the model's metabolite-flux mappings. The second was a more generic global optimization strategy, for which I tested employing a genetic algorithm and the Nelder-Mead simplex algorithm[53]. I further modified this problem by splitting it into a series of sequential optimizations on subsets of the parameter space to more quickly search for optima, at the expense of increasing the probability that the solution was only a local minimum.

An advantage of both the genetic algorithm and the Nelder-Mead simplex is that neither approach requires much information about the problem, making them versatile. As long as a fitness function mapping parameter values to an objective value can be specified, either method can be applied. However, in my experience, factors such as poor choice of initial parameters may lead these methods to perform poorly and produce unsatisfactory results. The regression procedure turned out to be critical for providing a suitable initial seed for the global optimization, allowing me to actually produce reasonable models from the optimization.

However, a feature of the LK-DFBA modeling approach is that it comprises a linear system of constraints, and that because of this, we can take advantage of

certain properties that result from Duality Theory. For strain design, this means that we can guarantee an optimum objective for the FBA problem, while searching over the design space for the engineering problem. Can we similarly use this structure to instead search over the parameter space, using parameter estimation as our outer problem instead? A proposed bi-level optimization problem is presented in Figure 5.1.

$$\vec{\omega} = [\vec{w}^T(t_1), \vec{w}^T(t_2), \ldots \vec{w}^T(t_{nT-1}), \vec{w}^T(t_{nT}), \vec{x}^T(t_0), \vec{x}^T(t_1), \ldots \vec{x}^T(t_{nT-1}), \vec{x}^T(t_{nT})]^T$$

$$\min_{(\boldsymbol{\theta})} f = \sum_{\ell} \phi_\ell(\omega_\ell - y_\ell)^2$$

s.t. $\max_{\vec{\omega}} z = \vec{c}^T\vec{\omega} - \lambda\vec{\omega}^T\vec{\omega}$

s.t. $0 = \boldsymbol{A}\vec{w}(t_k) \qquad \forall k \in [1, nT]$

$\vec{w}_{LB} \le \vec{w}(t_k) \le \vec{w}_{UB} \qquad \forall k \in [1, nT]$

$\vec{x}_{LB} \le \vec{x}(t_k) \le \vec{x}_{UB} \qquad \forall k \in [1, nT]$

$\vec{x}(t_0) = \vec{x}_0$

$x_i(t_k) = x_i(t_{k-1}) + \Delta t \cdot v_{p,i}(t_k) \qquad \forall k \in [1, nT]$

$$\sum_i v_{i,n}(t_{k+1}) \le b_n + a_n \sum_j x_{j,n}(t_k)$$

$$\forall k \in (1, nT), \forall i \in \{v\}_n, \forall j \in \{x\}_n, \forall n \in (1, n_r)$$

Figure 5.1. A bi-level optimization problem for parameter estimation in LK-DFBA
The outer problem seeks to find the parameters $\boldsymbol{\theta}$ (The set of all (a_n, b_n)) that minimize the value of the fitness function f, which is a weighted sum-of-squares-error between the solution vector $\vec{\omega}$ and the corresponding data, represented in vector form as $\vec{y}$. The weights are specified by variable ϕ_ℓ and can be set as appropriate (*c.f.* Section 3.2.5.3). The inner problem is specified in detail in Section 3.2.1.9.

The major challenge in using this approach stems from the problem described in the previous section on biomass accumulation: the conversion of the Linear Program into a Bi-linear Program. This introduces three main challenges.

The first is the inherent increase in difficulty moving from linear to bi-linear constraints. While an LP can be solved very easily with the Simplex algorithm, more sophisticated approaches are required to solve a BLP. For smaller problems, this may not be prohibitive.

Second is the issue of scale-up. In the biomass BLP of the previous section, a new bi-linear variable was introduced for each time step in the simulation, making time resolution the primary barrier to scale up. For the parameter estimation problem, the bi-linear variables are instead the slope terms of the kinetics constraints. As a result, the major barrier to scale-up is model size and the resulting parameterization. The tradeoff to be considered here is instead between the tractability of the parameter optimization and the structural accuracy of the resulting model; removing kinetics constraints to simplify the model may be necessary. This may be more acceptable however, depending on whether or not including the additional parameters was justified relative to the improvement in model performance (*c.f.* use of AIC in Chapter 4).

The third issue is the most challenging one, and concerns the convexity of the resulting inner problem. When the kinetics constraints parameters are fixed quantities, the resulting system of equations is an LP, and therefore a convex optimization problem. However, the bi-linear program specified in Figure 5.1 requires further scrutiny. Is this problem still convex? Does the resulting BLP satisfy the conditions for Strong Duality? Can we actually render this bi-level optimization into a single level, as is done in OptKnock[37]? It can be shown that by restricting all $a > 0$, the BLP solution space is convex. However, if any $a < 0$, then the resulting space becomes concave. As a result, we lose the a priori guarantee that the BLP is strongly dual. However, convexity is a sufficient, but not necessary, condition for strong duality.

In order to implement the parameter optimization proposed in this suggestion, two steps are necessary. First, it must be determined whether or not the concave optimization problem created when the solution space allows $a < 0$ for one or more kinetics parameters still satisfies the necessary conditions to guarantee strong duality. If this cannot be demonstrated, then no further effort is called for: there is no way to add a constraint to the outer problem to guarantee the inner problem is optimal (at least as is done in OptKnock). Second, if Strong Duality can be proven, there is a practical question of implementation. Bi-linear optimization problems are an active area of research, and the method proposed

here may provide an additional application case to further motivate this research[54-56].

5.3.3 Structural Learning Methods for identifying regulatory interactions

In the previous chapter, I explored the impact of model structure on the quality of the resulting fits by testing different combinations of putative regulatory interactions. I assumed the model stoichiometry and mass balances, added the appropriate mass action based kinetic constraints, and then investigated the influence of metabolites acting as non-stoichiometric regulators. At short time scales, such interactions may be allosteric interactions, in which a metabolite binds to an enzyme non-competitively to modulate its activity. At longer time scales, a metabolite may induce a transcriptional response, leading to an increase or decrease in concentration of the enzyme itself. In either situation, the net effect is that enzyme flux is influenced by a metabolite concentration despite that enzyme not directly participating as substrate or product.

5.3.3.1 Broader Context

As was demonstrated in the previous chapter, the performance of the fitted model is sensitive to the presence or absence of these interactions. Adding the correct interactions can greatly improve model performance, and even modestly incorrect connections may be better than nothing. The approach taken in that analysis is based off of work originally intended to identify those interactions in

the context of ODE models[57], and resorted to brute-force exploration of a subset of the combinatoric space. For a small model such as the Branched Pathway, this is feasible. But for larger models where it is infeasible or when the data is ill suited for LK-DFBA, other approaches may be complementary, providing useful biological insight.

This broader problem is of general interest, and has been explored in multiple contexts, both computational and experimental[58-61]. While high-throughput experimental techniques have exponentially increased our knowledge of metabolic network structure and its associated transcriptional regulation, knowledge of allosteric interactions has not accumulated nearly as quickly; it is likely that only a fraction of these interactions have been reported in the literature. While effort is being made to develop high-throughput assays for detecting allosteric interactions[62,63], these assays are not yet common.

5.3.3.2 Bayesian Networks

One tool that may be useful for identifying these interactions are Bayesian Networks (BNs). BNs are a graphical representation of the multivariate probability distribution that describes the relationship between a set of variables. These directed acyclic graphs (DAGs) consist of a set of variables (nodes) and directed edges that compactly and intuitively describe the relationship between the variables. Edges represent the conditional dependence of one variable on

another in the underlying probability distribution; two variables are conditionally independent (i.e., given the value of all other variables, knowledge of one variable provides no additional information about the value of the other) if there is no edge connecting them. An example of a BN derived from the modified Branched Pathway model is shown in Figure 5.2. By inferring a BN from metabolomics data and model predictions, we may be able to identify important metabolite-dependent regulatory interactions and improve the model's performance.

While BNs are efficient tools for inference, my primary interest here is not in using BNs to directly calculate the probable system state from existing data—LK-DFBA was designed to serve that purpose. Rather, I am interested in taking advantage of the structural learning algorithms that have been developed to produce BNs when expert sources are unavailable[64 59]. Using structure learning, we can perhaps identify potential metabolite-dependent regulatory interactions and incorporate them into our LK-DFBA models.

Two general classes of structure-learning algorithms exist. Search-and-score methods use search algorithms to systematically explore the space of potential DAGs and find a structure that best describes the data, as determined by some scoring criterion (such as the Bayesian Information Criterion or Bayesian Dirichlet scoring metric)[65]. These scoring criteria measure the relative probability

that the network structure describes the data, while penalizing additional model complexity (i.e., the number of edges). Constraint-based methods begin with a completely connected network and remove edges using tests for statistical independence. The Peter-Clark (PC) algorithm is an example of such[66]. Other specific algorithms that relevant to these methods include Sparse Candidate[67], Max-Min Hill-Climbing[68], and Three Phase Dependency Analysis[69], and relevant software for performing calculations includes the Causal Explorer package[70].

As previously mentioned, a key challenge for learning BNs from metabolomics data stems from a requirement typically for large quantitates of data to produce robust results[60]. A key element of the approach proposed here is to focus on learning regulatory interactions by incorporating existing knowledge of the stoichiometry into the initial BN. The narrow focus and use of pre-existing knowledge may reduce the amount of data necessary to robustly infer regulatory interactions.

5.3.3.3 The proposed BN structure learning approach

To allow metabolite-flux interactions to be captured, both metabolite concentrations and fluxes will need to be represented as nodes in the BN. BNs are DAGs, and in light of this restriction, the metabolic network can be rendered with a bipartite structure in which metabolites are strictly parent nodes, fluxes are strictly child nodes, metabolites may only share edges with fluxes, and fluxes

may only share edges with metabolites. An example of this structure is shown in Figure 5.2, in which the modified Branched Pathway model has been rendered as a BN according to this convention.

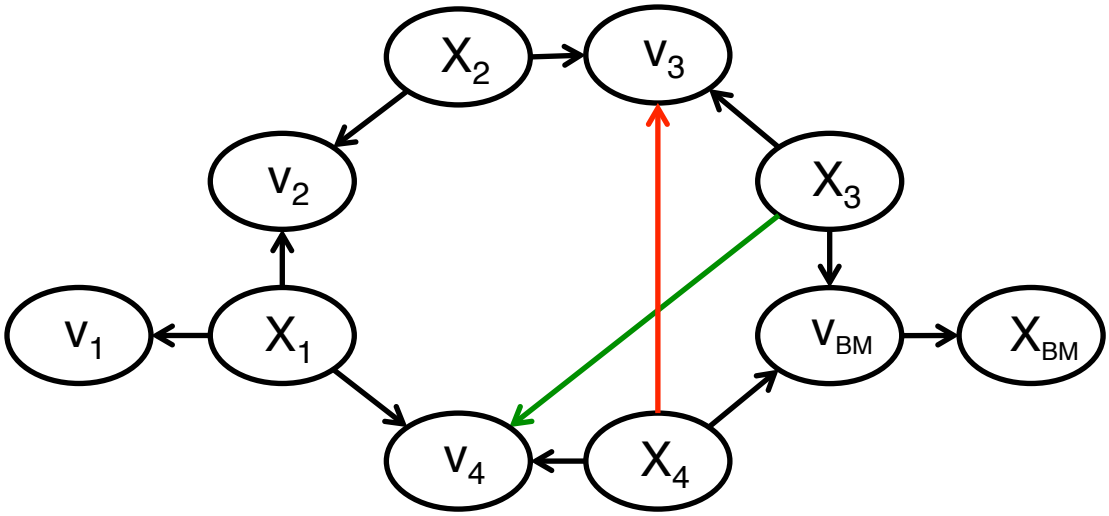

Figure 5.2. The modified branched pathway model represented as a Bayesian Network
Black arrows indicate connections derived from the network stoichiometry. The green arrow indicates allosteric activation, and the red arrow allosteric inhibition.

The other key element is the use of the existing stoichiometric matrix to better inform the structure identification problem. First, a procedure such as DFE may be used to infer flux time course values. Providing this data, even if only inferred, may be necessary for performing the structure learning algorithms. Care should be taken to avoid or reduce bias introduced by this step. Second, the stoichiometry can be used to initialize the BN structure. Additional steps should be taken to preserve these edges during structure learning by enforcing that only putative regulatory interactions may be added or removed from the graph. Alternatively, metabolite-metabolite and flux-flux edges might be removed from the resulting structure manually.

Once an optimal structure has been identified, regulatory connections (those note specified by model stoichiometry) can be sorted by their likelihood in to a list of metabolite-flux interactions. This list represents putative regulatory connections that must be further validated. If multiple algorithms are tested, common elements across lists may be compared to identify higher priority targets. Incorporating these putative connections into a model to test their impact (as was done in Chapter 4) may also be used as a screening step, but ultimately validation will need to come from experimental work.

5.3.3.4 Methods for evaluating the BN learning procedure

Previous efforts in our lab with Bayesian Networks has highlighted the difficulty of working with these structure learning methods in the absence of sufficient data[60]. For an initial assessment of the soundness of the described method, I propose the use of synthetic data from a small model to sidestep these issues, and to assess the impact of data availability on the method's performance. Given the previous content presented in this thesis, the modified Branched Pathway model seems an obvious and ideal candidate. Data from those chapters can be re-used as is, allowing for rapid implementation and assessment of a data processing pipeline for the proposed method.

Characterization of this method on the Branched Pathway model can allow for several tests to determine the method's viability and performance. First, it is

critical to assess the impact of data sampling on the resulting learned structure. At high data availability and quality, one would expect to be able to consistently learn the correct model structure. Further, for a given noise level, one can decrease the number of samples available to the structure-learning algorithm to determine when its accuracy starts to degrade, and at what rate. This performance curve would give us a sense for how much data the proposed method requires before we trust its results to be robust.

A second analysis should compare the method against a full structure-learning algorithm, in which the search is “naïve”, i.e., model stoichiometry is not used to initialize the network structure. While a version of this structure in which edges may be drawn in any order or orientation may be interesting as an extreme base case, the more important comparator is one in which the resulting graph obeys the same bipartite structural limitations on edge directions between metabolites and fluxes as shown in Figure 5.2. From this analysis, a performance curve depicting saturation accuracy at high sample availability and decreasing accuracy at low sample availability can be generated for the naïve method. This curve should then be compared against the curve for the proposed method generated as described in the previous paragraph (i.e. with known stoichiometry supplied and enforced). How many samples are required to guarantee high accuracy in each model? How does this change as noise is increased?

5.3.3.5 Potential difficulties

A practical limitation of this approach is the availability of flux data for the structure leaning algorithms. For an initial analysis, it may be acceptable to use noise-added flux data, in much the same way that I did for the *E. coli* model when assessing LK-DFBA. Such data is available in the Branched Pathway model datasets used in the analysis in Chapters 3 and 4. A reasonable next step would be to repeat the analysis of the structure learning methods using flux values generated from the DFE procedure. What influence does this have on structure accuracy? On sample quantity requirements? Does this introduce a bias issue? Does it recapitulate the results observed before?

On the smaller Branched Pathway model, it may be realistic to expect that the structure-learning algorithm is able to produce the correct structure consistently, given enough data. However, for larger models, this may not be the case. It will be worth investigating the behavior of the proposed method as data quantity decreases, to explore if there is a consistent pattern that emerges. Can the results of structure-learning be used to generate a triage list of putative interactions that still provides a relatively reasonable starting point for more targeted investigation? How does this list compare against the results from Chapter 4, in which a similar list was generated by performing brute-force analysis of AIC between different structural models?

Once these cases have been addressed in the small-scale model, moving to larger models can be explored. Natural options include the *E. coli* model of Chassagnole *et al.*[24] used in previous chapters, and any model of *S. cerevisiae* used for strain design efforts as described in the first half of this chapter. In these larger models, the goal may be merely to recapitulate known allosteric interactions from the underlying model, or from the literature, respectively. By first characterizing the proposed structure-learning approach on the toy model, one would have a reliable sense for the influence of data availability and noise on the reliability and usefulness of the method.

5.4 Closing Remarks

In this thesis, I have discussed a number of contributions to metabolic engineering and metabolomics. The previous chapters described these contributions in detail, and discussed their impact and limitations. But these efforts are only a partial realization of a bigger vision. This chapter described in some detail the necessary steps left to fully realize this vision, established expectations for what these steps are to accomplish, and provided recommendations for how best to pursue them. In addition, I highlighted some interesting side problems that may prove useful in the context of the previous and proposed projects, or perhaps might represent opportunities to have an impact on a broader scale. It is my hope that these next steps will continue to contribute to research in metabolomics, metabolic engineering, and biochemical modeling.

5.5 References

1 Toya, Y., Nakahigashi, K., Tomita, M. & Shimizu, K. Metabolic regulation analysis of wild-type and arcA mutant Escherichia coli under nitrate conditions using different levels of omics data. *Mol Biosyst* **8**, 2593-2604, doi:10.1039/c2mb25069a (2012).

2 Sevin, D. C., Stahlin, J. N., Pollak, G. R., Kuehne, A. & Sauer, U. Global Metabolic Responses to Salt Stress in Fifteen Species. *PLoS One* **11**, e0148888, doi:10.1371/journal.pone.0148888 (2016).

3 Varma, A. & Palsson, B. O. Metabolic capabilities of Escherichia coli: I. synthesis of biosynthetic precursors and cofactors. *J Theor Biol* **165**, 477-502, doi:10.1006/jtbi.1993.1202 (1993).

4 Mahadevan, R., Edwards, J. S. & Doyle, F. J., 3rd. Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophys J* **83**, 1331-1340, doi:10.1016/S0006-3495(02)73903-9 (2002).

5 Covert, M. W., Xiao, N., Chen, T. J. & Karr, J. R. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics* **24**, 2044-2050, doi:10.1093/bioinformatics/btn352 (2008).

6 Voit, E. O. Modelling metabolic networks using power-laws and S-systems. *Essays in biochemistry* **45**, 29-40, doi:10.1042/BSE0450029 (2008).

7 King, Z. A. *et al.* BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res* **44**, D515-522, doi:10.1093/nar/gkv1049 (2016).

8 Orth, J. D. *et al.* A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. *Mol Syst Biol* **7**, 535, doi:10.1038/msb.2011.65 (2011).

9 Feist, A. M. *et al.* A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol* **3**, 121, doi:10.1038/msb4100155 (2007).

10 Kim, H. *et al.* YeastNet v3: a public database of data-specific and integrated functional gene networks for Saccharomyces cerevisiae. *Nucleic Acids Res* **42**, D731-736, doi:10.1093/nar/gkt981 (2014).

11 Dal'Molin, C. G., Quek, L. E., Palfreyman, R. W. & Nielsen, L. K. AlgaGEM--a genome-scale metabolic reconstruction of algae based on

the Chlamydomonas reinhardtii genome. *BMC Genomics* **12 Suppl 4**, S5, doi:10.1186/1471-2164-12-S4-S5 (2011).

12 Oh, Y. K., Palsson, B. O., Park, S. M., Schilling, C. H. & Mahadevan, R. Genome-scale reconstruction of metabolic network in Bacillus subtilis based on high-throughput phenotyping and gene essentiality data. *The Journal of biological chemistry* **282**, 28791-28799, doi:10.1074/jbc.M703759200 (2007).

13 Suthers, P. F. *et al.* A genome-scale metabolic reconstruction of Mycoplasma genitalium, iPS189. *PLoS Comput Biol* **5**, e1000285, doi:10.1371/journal.pcbi.1000285 (2009).

14 Thiele, I. *et al.* A community-driven global reconstruction of human metabolism. *Nat Biotechnol* **31**, 419-425, doi:10.1038/nbt.2488 (2013).

15 Swainston, N. *et al.* Recon 2.2: from reconstruction to model of human metabolism. *Metabolomics* **12**, 109, doi:10.1007/s11306-016-1051-4 (2016).

16 Selvarasu, S., Karimi, I. A., Ghim, G. H. & Lee, D. Y. Genome-scale modeling and in silico analysis of mouse cell metabolic network. *Mol Biosyst* **6**, 152-161, doi:10.1039/b912865d (2010).

17 Knies, D. *et al.* Modeling and Simulation of Optimal Resource Management during the Diurnal Cycle in Emiliania huxleyi by Genome-Scale Reconstruction and an Extended Flux Balance Analysis Approach. *Metabolites* **5**, 659-676, doi:10.3390/metabo5040659 (2015).

18 Flassig, R. J., Fachet, M., Hoffner, K., Barton, P. I. & Sundmacher, K. Dynamic flux balance modeling to increase the production of high-value compounds in green microalgae. *Biotechnol Biofuels* **9**, 165, doi:10.1186/s13068-016-0556-4 (2016).

19 Li, X. *et al.* An Indexed, Mapped Mutant Library Enables Reverse Genetics Studies of Biological Processes in Chlamydomonas reinhardtii. *Plant Cell* **28**, 367-387, doi:10.1105/tpc.15.00465 (2016).

20 Goncalves, E. *et al.* Systematic Analysis of Transcriptional and Post-transcriptional Regulation of Metabolism in Yeast. *PLoS Comput Biol* **13**, e1005297, doi:10.1371/journal.pcbi.1005297 (2017).

21 Heavner, B. D. & Price, N. D. Comparative Analysis of Yeast Metabolic Network Models Highlights Progress, Opportunities for Metabolic Reconstruction. *PLoS Comput Biol* **11**, e1004530, doi:10.1371/journal.pcbi.1004530 (2015).

22 Mo, M. L., Palsson, B. O. & Herrgard, M. J. Connecting extracellular metabolomic measurements to intracellular flux states in yeast. *BMC Syst Biol* **3**, 37, doi:10.1186/1752-0509-3-37 (2009).

23 Nookaew, I. *et al.* The genome-scale metabolic model iIN800 of Saccharomyces cerevisiae and its validation: a scaffold to query lipid metabolism. *BMC Syst Biol* **2**, 71, doi:10.1186/1752-0509-2-71 (2008).

24 Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. & Reuss, M. Dynamic modeling of the central carbon metabolism of Escherichia coli. *Biotechnol Bioeng* **79**, 53-73, doi:10.1002/bit.10288 (2002).

25 Goel, G., Chou, I. C. & Voit, E. O. System estimation from metabolic time-series data. *Bioinformatics* **24**, 2505-2511, doi:10.1093/bioinformatics/btn470 (2008).

26 Chou, I. C. & Voit, E. O. Estimation of dynamic flux profiles from metabolic time series data. *BMC Syst Biol* **6**, 84, doi:10.1186/1752-0509-6-84 (2012).

27 Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* **45**, D353-D361, doi:10.1093/nar/gkw1092 (2017).

28 Scheer, M. *et al.* BRENDA, the enzyme information system in 2011. *Nucleic Acids Res* **39**, D670-676, doi:10.1093/nar/gkq1089 (2011).

29 Cherry, J. M. *et al.* Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res* **40**, D700-705, doi:10.1093/nar/gkr1029 (2012).

30 Hynne, F., Dano, S. & Sorensen, P. G. Full-scale model of glycolysis in Saccharomyces cerevisiae. *Biophys Chem* **94**, 121-163, doi:http://dx.doi.org/10.1016/S0301-4622(01)00229-0 (2001).

31 Dano, S., Sorensen, P. G. & Hynne, F. Sustained oscillations in living cells. *Nature* **402**, 320-322, doi:10.1038/46329 (1999).

32 Richard, P., Bakker, B. M., Teusink, B., Dam, K. & Westerhoff, H. V. Acetaldehyde Mediates the Synchronization of Sustained Glycolytic Oscillations in Populations of Yeast Cells. *European Journal of Biochemistry* **235**, 238-241, doi:10.1111/j.1432-1033.1996.00238.x (1996).

33 Teusink, B. *et al.* Synchronized Heat Flux Oscillations in Yeast Cell Populations. *Journal of Biological Chemistry* **271**, 24442-24448, doi:10.1074/jbc.271.40.24442 (1996).

34 Teusink, B., Diderich, J. A., Westerhoff, H. V., van Dam, K. & Walsh, M. C. Intracellular Glucose Concentration in Derepressed Yeast Cells Consuming Glucose Is High Enough To Reduce the Glucose Transport Rate by 50%. *Journal of Bacteriology* **180**, 556-562 (1998).

35 Kreuzberg, K. H. & Betz, A. Amplitude and period length of yeast NADH oscillations fermenting on different sugars in dependence of growth phase, starvation and hexose concentration. *Journal of Interdisiplinary Cycle Research* **10**, 41-50, doi:10.1080/09291017909359650 (1979).

36 Betz, A. & Hinrichs, R. Incorporation of Glucose into an Insoluble Polyglycoside during Oscillatory Controlled Glycolysis in Yeast Cells. *European Journal of Biochemistry* **5**, 154-157, doi:10.1111/j.1432-1033.1968.tb00351.x (1968).

37 Burgard, A. P., Pharkya, P. & Maranas, C. D. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng* **84**, 647-657, doi:10.1002/bit.10803 (2003).

38 Pharkya, P., Burgard, A. P. & Maranas, C. D. Exploring the overproduction of amino acids using the bilevel optimization framework OptKnock. *Biotechnol Bioeng* **84**, 887-899, doi:10.1002/bit.10857 (2003).

39 Vital-Lopez, F. G., Armaou, A., Nikolaev, E. V. & Maranas, C. D. A computational procedure for optimal engineering interventions using kinetic models of metabolism. *Biotechnol Prog* **22**, 1507-1517, doi:10.1021/bp060156o (2006).

40 Kumar, V. S. & Maranas, C. D. GrowMatch: an automated method for reconciling in silico/in vivo growth predictions. *PLoS Comput Biol* **5**, e1000308, doi:10.1371/journal.pcbi.1000308 (2009).

41 Ranganathan, S., Suthers, P. F. & Maranas, C. D. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput Biol* **6**, e1000744, doi:10.1371/journal.pcbi.1000744 (2010).

42 Chowdhury, A., Zomorrodi, A. R. & Maranas, C. D. k-OptForce: integrating kinetics with flux balance analysis for strain design. *PLoS Comput Biol* **10**, e1003487, doi:10.1371/journal.pcbi.1003487 (2014).

43 Nakagawa, A. *et al.* Total biosynthesis of opiates by stepwise fermentation using engineered Escherichia coli. *Nat Commun* **7**, 10390, doi:10.1038/ncomms10390 (2016).

44 Stols, L. & Donnelly, M. I. Production of succinic acid through overexpression of NAD(+)-dependent malic enzyme in an Escherichia coli mutant. *Applied and Environmental Microbiology* **63**, 2695-2701 (1997).

45 Dunn, W. B., Broadhurst, D. I., Atherton, H. J., Goodacre, R. & Griffin, J. L. Systems level studies of mammalian metabolomes: the roles of mass spectrometry and nuclear magnetic resonance spectroscopy. *Chem Soc Rev* **40**, 387-426, doi:10.1039/b906712b (2011).

46 Dunn, W. B. *et al.* Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. *Nature protocols* **6**, 1060-1083, doi:10.1038/nprot.2011.335 (2011).

47 Dhakshinamoorthy, S., Dinh, N. T., Skolnick, J. & Styczynski, M. P. Metabolomics identifies the intersection of phosphoethanolamine with menaquinone-triggered apoptosis in an in vitro model of leukemia. *Mol Biosyst* **11**, 2406-2416, doi:10.1039/c5mb00237k (2015).

48 Cipriano, R. C., Smith, M. L., Vermeersch, K. A., Dove, A. D. & Styczynski, M. P. Differential metabolite levels in response to spawning-induced inappetence in Atlantic salmon Salmo salar. *Comp Biochem Physiol Part D Genomics Proteomics* **13**, 52-59, doi:10.1016/j.cbd.2015.01.001 (2015).

49 Cotten, C. & Reed, J. L. Constraint-based strain design using continuous modifications (CosMos) of flux bounds finds new strategies for metabolic engineering. *Biotechnol J* **8**, 595-604, doi:10.1002/biot.201200316 (2013).

50 Pacific Northwest National Laboratory & National Renewable Energy Laboratory. Top Value Added Chemicals from Biomass, Volume 1: Results of Screening for Potential Candidates from Sugars and Synthesis Gas. (U.S. Department of Energy., 2004).

51 Gomez, J. A., Hoffner, K. & Barton, P. I. DFBAlab: a fast and reliable MATLAB code for dynamic flux balance analysis. *BMC Bioinformatics* **15**, 409, doi:10.1186/s12859-014-0409-8 (2014).

52 Hoffner, K., Harwood, S. M. & Barton, P. I. A reliable simulator for dynamic flux balance analysis. *Biotechnol Bioeng* **110**, 792-802, doi:10.1002/bit.24748 (2013).

53 Nelder, J. A. & Mead, R. A Simplex Method for Function Minimization. *The Computer Journal* **7**, 308-313, doi:10.1093/comjnl/7.4.308 (1965).

54 Kleniati, P.-M. & Adjiman, C. S. Branch-and-Sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. Part I: Theoretical development. *Journal of Global Optimization* **60**, 425-458, doi:10.1007/s10898-013-0121-7 (2014).

55 Kleniati, P.-M. & Adjiman, C. S. Branch-and-Sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. Part II: Convergence analysis and numerical results. *Journal of Global Optimization* **60**, 459-481, doi:10.1007/s10898-013-0120-8 (2014).

56 Kleniati, P.-M. & Adjiman, C. S. A generalization of the Branch-and-Sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. *Computers & Chemical Engineering* **72**, 373-386, doi:10.1016/j.compchemeng.2014.06.004 (2015).

57 Link, H., Kochanowski, K. & Sauer, U. Systematic identification of allosteric protein-metabolite interactions that control enzyme activity in vivo. *Nat Biotechnol* **31**, 357-361, doi:10.1038/nbt.2489 (2013).

58 Vilela, M. *et al.* Parameter optimization in S-system models. *BMC Syst Biol* **2**, 35, doi:10.1186/1752-0509-2-35 (2008).

59 Gavai, A. K. *et al.* Constraint-based probabilistic learning of metabolic pathways from tomato volatiles. *Metabolomics* **5**, 419-428 (2009).

60 Yin, W., Garimalla, S., Moreno, A., Galinski, M. R. & Styczynski, M. P. A tree-like Bayesian structure learning algorithm for small-sample datasets from complex biological model systems. *BMC Syst Biol* **9**, 49, doi:10.1186/s12918-015-0194-7 (2015).

61 Nikolaev, Y. V., Kochanowski, K., Link, H., Sauer, U. & Allain, F. H. Systematic Identification of Protein-Metabolite Interactions in Complex Metabolite Mixtures by Ligand-Detected Nuclear Magnetic Resonance Spectroscopy. *Biochemistry* **55**, 2590-2600, doi:10.1021/acs.biochem.5b01291 (2016).

62 Tagore, R., Thomas, H. R., Homan, E. A., Munawar, A. & Saghatelian, A. A Global Metabolite Profiling Approach to Identify Protein–Metabolite Interactions. *Journal of the American Chemical Society* **130**, 14111-14113, doi:10.1021/ja806463c (2008).

63 Li, X., Gianoulis, T. A., Yip, K. Y., Gerstein, M. & Snyder, M. Extensive In Vivo Metabolite-Protein Interactions Revealed by Large-Scale Systematic

Analyses. *Cell* **143**, 639-650, doi:http://dx.doi.org/10.1016/j.cell.2010.09.048 (2010).

64 Hartemink, A., Gifford, D., Young, R. & Jaakkola, T. Using graphical models and genomic expression data to statistically validate models of genetic regulatory network. *Pac Symp Biocomput.*, 422-433 (2001).

65 Heckerman, D. A tutorial on learning with Bayesian networks. *Learning in Graphical Models* **89**, 301-354 (1996).

66 Spirtes, P., Glymor, C. & Scheines, R. Causation, Prediction, and Search, Edn 2. *MIT Press* (2000).

67 Friedman, N., Nachman, I. & Peér, D. in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* 206-215 (Morgan Kaufmann Publishers Inc., Stockholm, Sweden, 1999).

68 Tsamardinos, I., Brown, L. E. & Aliferis, C. F. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach Learn* **65**, 31-78, doi:10.1007/s10994-006-6889-7 (2006).

69 Cheng, J., Greiner, R., Kelly, J., Bell, D. & Liu, W. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence* **137**, 43-90, doi:http://dx.doi.org/10.1016/S0004-3702(02)00191-1 (2002).

70 Aliferis, C., Tsamardinos, I., Statnikov, A. & Brown, L. in *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Scienes, METMBS '03, June 23 - 26, 2003, Las Vegas, Nevada, USA.* (eds Faramarz Valafar, Homayoun Valafar, Faramarz Valafar, & Homayoun Valafar) 371-376 (CSREA Press).

APPENDIX A
Letter of permission for use of copyrighted material

From: MOLBIOSYST (shared) <MOLBIOSYST@rsc.org>
Received: Fri, Apr 28, 2017 at 12:36 PM

Dear Mr Dromms

Thank you for your e-mail. Permission is granted to reproduce your article in your thesis as long as it is fully acknowledged and includes a link back to the article on our website. Please ensure that all authors are aware that it is being included.

With thanks,

Molecular BioSystems Editorial Office
Royal Society of Chemistry, Thomas Graham House Science Park, Milton Road, Cambridge CB4 0WF

To: MOLBIOSYST (shared) <MOLBIOSYST@rsc.org>
Cc: Mark Styczynski <mark.styczynski@chbe.gatech.edu>
Sent: 18 April 2017 10:42 PM

Dear Dr. Darby,

I am the first author of the publication Improved metabolite profile smoothing for flux estimation in Mol. BioSyst., 2015, 11, 2394. I am writing to you to seek permission to reuse the text and figures from the publication for a portion of a chapter of my PhD thesis. My adviser, Dr. Styczynski, is the other author on the paper, and is copied on this email.

The thesis document will include both appropriate references to the work as having been previously published in Molecular BioSystems, as well as a copy of your response granting permission for this use. Please let me know if there is anything else needed or expected.

Thank you,
Robert Dromms
PhD Candidate
Styczynski Lab
Chemical & Biomolecular Engineering
Georgia Institute of Technology