NORTHWESTERN UNIVERSITY

Expanding Computational Metabolic Modeling Methods for Novel Metabolic Engineering Applications

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Chemical Engineering

By

Jennifer L. Greene

EVANSTON, ILLINOIS

March 2018

ABSTRACT

Expanding Computational Metabolic Modeling Methods for Novel Metabolic Engineering Applications

Jennifer Greene

Computational models greatly benefit metabolic engineering efforts by helping to elucidate experimental observations and predict engineering targets for improved cellular performance. Additionally, supplementing experimental efforts with computational modeling can reduce the loss of time and resources in the lab by narrowing down testing conditions. In optimal cases, computational models can be continuously improved as more experimental feedback is applied, lending to the success of iterative experimental and computational testing cycles.

In this dissertation, existing metabolic modeling paradigms are expanded for use on previously untested systems. First, constraint-based modeling methods are used to predict essential gene knockouts leading to metabolically active, non-growth states in *Escherichia coli* cells. Thirty of our predicted candidates were screened in the lab and predicted metabolite auxotrophies were confirmed. The goal of this effort is to identify ways to turn off growth in cells without shutting down metabolic activity (i.e. carbon uptake). The initial modeling work described here provides a foundation for uncovering the governing objectives of cells during non-growth conditions. These factors are currently unknown as most constraint-based modeling methods have been developed exclusively to predict optimal growth conditions where maximizing flux toward biomass production is always assumed. In a non-growing cell, this assumption no longer holds and the best identifier of cell health to optimize under these conditions is unclear. However, being able to accurately predict the flux distribution of non-growth metabolism is an essential step toward enabling the development

on non-growth, high-carbon yield biosynthetic processes where cells will no longer divert fed substrates toward growth.

Second, we look toward expanding the application of kinetic models of metabolism to predict engineering targets for increased product formation. We specifically focus on improving the limitations of the computationally intensive kinetic ensemble modeling (EM) framework. EM is a Monte Carlo-based modeling method used to sample many, possible kinetic parameter sets of metabolism from a previously defined reference state and then screening them against additional phenotypic datasets. In its original form, the framework is prohibitively slow when applied to large metabolic networks and often results in non-stable solution sets. To alleviate these challenges, we implemented three acceleration strategies, each providing increased computational efficiency. Furthermore, by screening for locally stable parameter sets, we greatly reduce the sample space and generate more biologically representative solutions. Lastly, we applied our accelerated EM framework to develop a novel kinetic representation of *Clostridium autoethanogenum* which accurately predicts intracellular metabolite concentrations and engineering targets for increased ethanol production. Specifically, our average ensemble predictions fall within demonstrated experimental error ranges for sixty percent of observed metabolite species. Additionally, we were able to demonstrate the experimental observation of a limiting acetyl-CoA pool with increasing biomass concentration and confirm the production of ethanol from acetate to increase adenosine triphosphate (ATP) generation. Finally, through sensitivity analysis, we have identified several enzyme targets for improving ethanol production. Encouragingly, we show that two of the enzymes we have identified as potential down-regulation targets, phosphate acetyltransferase (PTA) and carbon monoxide dehydrogenase/acetyl-CoA synthase (CODH_ACS), have previously shown increased ethanol production when knocked out in similar clostridia strains.

Ultimately by demonstrating expanded applications for existing modeling methods, this dissertation highlights the expansive opportunities to improve metabolic engineering outcomes through creative computational design. These results will improve efforts to harness and optimize non-growth metabolism as well as increase access to kinetic exploration of metabolic pathways.

ACKNOWLEDGEMENTS

I would first like to thank my advisers, Professor Linda Broadbelt and Professor Keith Tyo. I have benefitted enormously as a student and scientist due to their guidance. I was incredibly fortunate to receive their undivided attention for 1-2 hours every three weeks for the past five years (that's over 5 straight days!) which does not even include the countless hours they have invested in editing my writing, attending my practice talks, and writing my letters of recommendation. I skew pessimistic and their constant support and encouragement navigated me through various research setbacks and successes.

Thanks also to Professor Andreas Wächter. His generous insights helped me redirect my stalled kinetic modeling efforts towards new and interesting research areas. Additionally, I would like to thank James Daniell and Michael Koepke from LanzaTech for providing a novel kinetic modeling application opportunity as well as significant support and feedback. Thank you as well to my committee members, Professor Neda Bagheri and Professor David Schwab, for your insights, encouragement, and time. I would also like to acknowledge the computational resources and staff contributions provided by the Quest high performance computing facility at Northwestern University and my funding sources, the NSF Graduate Fellowship and the Murphy Scholars programs, for enabling my research.

To the members of the Tyo and Broadbelt groups, I am very lucky to be amongst your impressive ranks. I especially appreciate my fellow "impractical scientist" computational subgroup members (James, J(D)ante, and Jon). May our legacy of mostly dominating the Tyo Lab Cup reign for years to come. Special thanks and paper guns to Will Bothfeld for being my enthusiastic, non-growth metabolism research collaborator. Thanks also to Namita Bhan, Mike Brotz, Matt Moura, Trang Vu, and Karthik Sekar for their constant research help and advice. To my Broadbelt labmates specifically, thank you for lively lunch conversations and indulging my disrupting visits. Finally, I want to thank Bola Adeniran and Alexandra de Paz for being my heroes. Their daily support made failure feel impossible and their intelligence and kindheartedness are ever inspiring.

Special thanks as well to Professor Renee Engeln for having me as her graduate assistant in Allison Hall for four years and for being a great mentor. Lastly and most importantly, I want to thank my family and friends. My mom and sister have read my endless applications, abstracts, emails, and proposal drafts on subjects they considerately deny are boring. They have also always answered my weirdly-timed phone calls, even when I just wanted company during my late-night walks home from Tech. They are my personal and professional role models and I owe all my great opportunities to them. Finally, I want to thank my wonderful friends from high school, college, and my time at Northwestern for traveling through this process with me and greatly enriching my life. I have learned so much from so many individuals over the course of this experience and I am very thankful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Research Motivation

Efforts to engineer microorganisms to produce commodity chemicals, pharmaceuticals, and biofuels from cheap, readily-available substrates span multiple research fields. In metabolic engineering specifically, we focus on mapping and altering metabolic pathways. Stephanopoulos describes metabolic engineering as "the directed improvement of cellular properties through the modification of specific biochemical reactions or the introduction of new ones."[1] This description includes not only experimental efforts to improve microbe strain design but computational efforts, too. Despite their small size, single-cell organisms are complicated systems. While a lot is known about some of these organisms (especially model organisms *Escherichia coli* and *Saccharomyces cerevisiae*), global understanding of metabolic factors governing their behavior under various conditions is lacking. To start to resolve gaps in knowledge and culminate known system properties, computational metabolic engineering efforts are being increasingly adopted and improved.[2,3] Local changes to a cell caused by changes in media, enzyme expression, or growth state often result in sometimes non-intuitive global changes across the metabolic network. As our knowledge of cellular genomics, fluxomics, and metabolomics increases, the opportunities to improve our computational models and their predictive power increase as well.[3]

This work highlights two areas where computational modeling would greatly benefit metabolic engineering goals. First, inducing and then mapping non-growth metabolism in cells is crucial to developing a quiescent, high-carbon yield bioproduction platform. Currently, many industrial biotechnology applications employing growing microbes to produce biofuels and other commodity chemicals are unable to maximize the conversion of feedstock to target products.[4] Cellular growth competes for nutrients that might otherwise be converted to the desired product

(e.g., biofuel). Furthermore for most non-growth states, metabolism is unacceptably slow making bioproduction platforms infeasible. If production and growth could be uncoupled, theoretical maximum yields would be feasible in stationary phase. In addition to improving industrial applications, our development of computational and experimental tools for investigating the induced stationary phase of microbes would benefit the fundamental microbiology and human health fields too. Most cells in the human body exist in a quiescent state. Also, diseases like tuberculosis and cystic fibrosis persist when microbes, Mycobacterium tuberculosis and Pseudomonas aeruginosa, respectively, take up extended residency in the human body by entering a quiescent persister phase.[5,6] Growing cells divert significant resources toward growth processes and to achieve efficient product yields, understanding how to turn growth down or off while maintaining high, non-dormant metabolic activity requires computational insight. While a few governing factors affecting cell growth are known, many remain unknown and the interactions between factors are not easily mapped intuitively. Instead we can use computational modeling techniques to interrogate these systems and begin to understand the possibilities and limitation of non-growth metabolism.

To fully understand metabolism (growth and non-growth) our second area of opportunity, kinetic modeling, must be strengthened. While constraint-based steady state flux modeling methods are well-characterized and have successfully aided in achieving numerous metabolic engineering successes they lack some necessary insights for improved pathway design.[2,3,7] Steady state models can predict flux distributions through pathways based on assumptions of growth or product maximization but their ability to capture regulatory behavior or identify rate-limiting steps is limited. However kinetic models, which do provide these insights, are very difficult to develop due to a lack of experimental data and an inability to observe individual enzymatic behavior *in vivo*. However, by improving the most promising kinetic modeling tools and adapting their framework to interrogate novel systems their utility and accuracy can be greatly improved.

In this work I will outline initial efforts to apply steady state, constraint-based models to non-growth metabolism. Then I will introduce work I have completed to advance kinetic modeling methods of microorganisms. Lastly, I will describe how we applied both steady state and kinetic modeling methods to develop a novel kinetic representation of the gas-fermenting acetogen, *Clostridia autoethanogenum*. This work demonstrates the power of computational metabolic modeling to provide insight into experimental observations and predict novel engineering targets. Additionally, these initial efforts open the door to broader applications and expanded capability to use these tools on existing and new metabolic engineering questions.

1.2 Research Outline

This work broadly aims to elucidate factors governing metabolic pathways in microorganisms specifically through developing and improving computational frameworks for modeling cellular metabolism.

In Chapter 2, the traditional constraint-based steady state modeling techniques used in Chapter 3 are introduced, and a case is made for developing improved kinetic models. Additionally, the traditional kinetic ensemble modeling framework which serves as the starting point for the work performed in Chapters 4 and 5 is described in detail.

In Chapter 3, initial work into predicting and inducing quiescence in *Escherichia coli* is described. Existing constraint-based modeling techniques are harnessed in a novel application to induce metabolically active, non-growing cells. This chapter also highlights how effective computational modeling can aid in experimental design and how interchange between computational and experimental insights benefits the overall investigative process. The inevitable goal of this work is to identify existing or develop new computational methods capable of predicting non-growth metabolism.

In Chapter 4, the focus shifts to improving the kinetic ensemble modeling framework specifically through reducing computation time and optimizing parameter sampling. We first reduce the structural complexity of the network by removing dependent species, and then we sample locally stable parameter sets to reflect realistic biological states of cells. Lastly, we presort the screening data to eliminate the most incorrect predictions in the earliest screening stages, saving further calculations in later stages. Our complementary improvements to the current EM framework are easily incorporated into concurrent EM efforts and broaden the application opportunities and accessibility of kinetic modeling across the field.

In Chapter 5, I describe using the methods developed in Chapter 4 to generate a novel kinetic representation of a gas-fermenting acetogen, *Clostridia autoethanogenum,* in collaboration with LanzaTech, a biotechnology company in Skokie, IL. Developing autotrophic, acetogenic bacteria strains as gas fermentation platforms is a promising avenue for converting industrial waste gas streams into valuable chemical products. One such strain, *Clostridium autoethanogenum*, naturally converts CO, CO_2, and H_2 gases into ethanol and acetate. Currently, lowering the acetate to ethanol production ratio is a key strategy for accomplishing large-scale industrial application of *C. autoethanogenum* gas fermentation. Unfortunately, the limited availability and time-intensive implementation of genetic engineering tools for clostridia strains greatly hinders metabolic engineering efforts toward this goal. To alleviate the lack of sufficient mutant phenotype data interrogating the pathways of interest, computational tools are needed to resolve experimental observations and predict engineering targets to help minimize experimental characterization in the lab. While stoichiometric models of *C. autoethanogenum* metabolism are available, they are unable to provide insight into regulatory relationships, rate-limiting steps, or the effects of altering enzyme expression. These limitations highlight the benefits of developing a robust, kinetic model of metabolism to supplement information unobtainable from traditional constraint-based models. In

this work, we offer the first kinetic representation of *C. autoethanogenum* developed using the Ensemble Modeling (EM) framework. We have adapted the existing method to enable the usage of non-genetic perturbation data, specifically the effects of changing biomass concentration, to sample and train our kinetic parameter sets. Our final kinetic parameter ensemble accurately predicts intracellular metabolite concentrations and engineering strategies for improved ethanol production.

In Chapter 6, the body of work presented is summarized and future areas of study are discussed. Elucidating metabolic factors governing non-growth metabolism will rely on modifying old and developing new computational tools. Additionally, the scope of kinetic ensemble modeling is vast and several avenues for expanding predictions and elucidating unknown metabolic features are possible. Lastly combining the two areas of this work by developing kinetic models of non-growth metabolism is a crucial step toward understanding and developing non-growth, metabolically active biosynthesis platforms.

Chapter 2: Background on Computational Metabolic Modeling

2.1 Constraint-Based Steady State Models

Metabolic models have been developed and refined for many species including the model organisms, E. coli and S. cerevisiae.[8,9] Metabolic models are developed from knowledge provided by gene sequencing and reported gene-protein-reaction relationships.[2,3,10] For E. coli in particular, various levels of metabolic models exist. Core models cover just the central carbon metabolism and basic energy generation processes while genome scale models aim to map the entire metabolic network of a cell. Specifically, the core model covers 56 genes and 95 reactions while genome scale models cover up to 1366 genes and up to 2251 reactions.[11–13] While the genomic models do provide more fine grain network data, there are still many applications necessitating the use of a simpler core model. Many metabolic processes are governed by central metabolism, and cells shuttle a majority of their flux through these main core pathways.

Developing and refining metabolic models is of great importance to the systems biology, metabolic engineering, and synthetic biology fields as they provide key insights and understanding to non-intuitive cellular pathway responses.[7,8,14] Cellular metabolic networks are robust and small perturbations to a cell's native state often result in global, rather than just local, changes across the network. The ability to take a small data set of experimentally measured extracellular concentrations and fluxes and then infer larger network insights underlines the importance of using highly curated metabolic models.

Moreover, well-developed models can help predict untested network perturbations. Specifically using metabolic models and constraint-based computational methods we can simulate, among others, gene knockouts, heterogeneous gene expression, and changing media conditions.

This predictive power can be utilized to engineer new production pathways, to redesign or optimize existing pathways, to understand the effects of changing media conditions, and to predict lethal cellular perturbations. Perhaps most importantly, by testing these experimental conditions in silico, we can run hundreds to thousands of potential simulations in relatively small computational time frames. These computational test cases help narrow down the best cases for in vivo experimental testing in the lab. In practice, metabolic modeling efforts minimize costly experiments and save countless person-hours at the lab bench.

Metabolic models allow us to model network wide flux distributions in a cell. Using a steady state approximation we can assume the rate of change in concentration of each metabolite is 0 or

$$\frac{dX}{dt} = 0 \tag{2.1}$$

where X is a vector of all metabolites in the defined metabolic system.[13] Building on this the sum of the stoichiometry-weighted fluxes of all reactions consuming and producing a given metabolite must also equal 0.

$$S_{m,n} \cdot v = 0 \tag{2.2}$$

where $S_{m,n}$ is the m x n stoichiometric matrix containing the stoichiometric coefficients of each metabolite (m) for every reaction (n) in the defined metabolic network. Constraint-based models solve for v, the n x 1 vector of reaction fluxes, satisfying the steady state assumption.[13] Additionally, reaction bounds can be set to limit the maximum forward and reverse rates of reactions and to limit nutrient uptake:

$$lb \leq v \leq ub \tag{2.3}$$

where lb and ub are n x 1 vectors containing the lower and upper bounds on reaction fluxes respectively.[13] These reaction bounds can incorporate thermodynamic constraints by setting the directionality for reversible and irreversible reactions.[11,15] Reversible reactions have unconstrained upper and lower flux bounds, while irreversible reactions have their lower flux bound set to 0.

Therefore, if we know the uptake and production reaction fluxes of a few key reactions (i.e. carbon source uptake), we can model all possible intracellular flux sets that meet the net flux constraints described above. Metabolic models are often underdetermined because there are far more reactions with calculated flux values than metabolites leading to multiple feasible flux solution sets. To identify the best solution set to describe the cell's true metabolic state, several methods have been developed. This work will focus and draw most from the flux balance analysis (FBA), flux variability analysis (FVA), and minimization of metabolic adjustments (MOMA) methods.

FBA methods select a network flux solution set by defining an objective function, z:

$$z = c^T v \tag{2.4}$$

where c is an n x 1 vector of weights used to identify which reactions should contribute to the objective function. Then by choosing to maximize or minimize the objective function, the FBA method will select the solution set with the largest or smallest objective value respectively.[13] Most applications select for maximized growth rate, and most metabolic models include finely curated biomass production reactions.[16,17] While maximizing growth is the most used objective function, one can also select an objective function to calculate theoretical energy and product yields or to minimize substrate consumption. ATP generation, production formation, or oxygen consumption are other frequently used objective function selections.[17,18] In some cases multiple objectives can be selected

by either successively optimizing hierarchical objective functions or co-optimizing dual selected objective functions.[18–21]

It is important to note that multiple distinct solution sets may still equally optimize the selected objective function as the system is underdetermined. This redundancy underlies the robust metabolisms of cells as they can quickly adapt to survive in varying environmental conditions.[22] When multiple solutions exist, flux variability analysis (FVA) methods can provide additional insight into which areas of the metabolism have flexible flux values and which areas are more stringently controlled for a given set of solutions.[23] FVA is a supplement to FBA and uses the same constraints-based solution strategies. Using FVA, a more computationally intensive method, one can identify all possible optimal solutions sets and generate the maximum and minimum flux values for each reaction in the network. By running FVA methods, we can ascertain a confidence level for our flux predictions and understand how certain perturbations lead to larger uncertainty than others in network simulations.[23]

When one or more reaction fluxes may not be the best objective function when predicting metabolic flux distribution, other methods have been developed using different hypotheses. For example, the Minimization of Metabolic Adjustments method (MOMA) selects a flux solution within all specified constraints most similar to a reference state.[24] The idea behind MOMA is that cells undergoing perturbation likely adjust their metabolic flux profile just enough to overcome the perturbation while still staying as close as possible to the initial, wild type (WT) state initially. However over longer time-scales, cells would eventually achieve the new optimal solution through evolutionary pressure. In some cases, the closest possible solution and the new optimal solution are the same, but often they are not. Using the MOMA method, flux predictions can be made even under

conditions when optimizing growth or ATP production cannot be assumed. As you will see in Chapter 3, we harness the MOMA method to predict the flux behavior of non-growing cells.

The Regulatory On/Off Minimization method (ROOM) is another constraint-based steady state modeling alternative to FBA.[25] ROOM hypothesizes a cell's metabolism will adjust to make the fewest significant changes to its flux distribution after a perturbation. Therefore, ROOM predicts the flux distribution state within the prescribed constraints with the smallest number of significant flux changes to mirror possible genetic regulatory control decisions following a gene expression change. Another approach called Relative Change (RELATCH) combines insights from the FBA, MOMA, and ROOM methods to predict flux distributions both immediately after and over longer periods following a reaction knockout.[26] The continuous development and refinement of these constraint-based steady-state modeling methods highlights the importance of accurately predicting a cell's response to an enzyme knockout. Similarly, the vast field of constraint-based modeling work implies the importance of aiding experimental characterization through computational modeling and predicting engineering targets for improved chassis strain design. These modeling techniques provide a great avenue for predicting gene knockouts, media changes, and the effects of expressing heterologous metabolic pathways, but are less suited for modeling regulatory effects and changes in enzyme expression. As highlighted in the next section, developing kinetic modeling methods building on the expansive constraint-based modeling efforts is the next key step in fully understanding cellular metabolism.

2.2 Kinetic Modeling

Kinetic and regulatory modeling of cellular metabolism is a major challenge in metabolic engineering and systems biology. Specifically, it is non-trivial to access true enzyme kinetic data. *In vitro* derived kinetic parameters and rate laws for enzymes do not necessarily reflect true *in vivo*

behavior and are often determined under varying experimental conditions. Additionally, these *in vitro* studies often use purified enzymes which prevents observation of any potential *in vivo* activation or inhibition by other cellular molecules. Therefore, to use these *in vitro* derived parameters together in a single metabolic kinetic model would be unrepresentative. As discussed above, constraint-based stoichiometric modeling has made great strides in improving and characterizing strain designs but without kinetic information it is difficult to identify rate limiting steps and system bottlenecks. This last point emphasizes the importance of developing kinetic regulatory models of non-growth metabolism.

Initial kinetic modeling efforts used simplified approximations of kinetic formulas to interrogate kinetic limitations and identify design parameters for improved metabolic engineering efforts.[27] Approximate methods including lin-log kinetics are able can be useful tools but only within prediction limits close to the reference state at which they were developed.[28–30] An alternative to approximation methods, Monte Carlo-based sampling and screening methods have emerged as strong tools for developing kinetic representations of metabolic models.[27,31–35] The ensemble modeling framework is one such method, first demonstrated by the Liao Lab and later built upon by the Maranas Lab.[33,34,36–44] The kinetic modeling work detailed in this work builds off of this existing ensemble modeling framework described below. However, it is important to note that despite the demonstrated benefits of both approximate and Monte Carlo kinetic modeling techniques, kinetic metabolic modeling as a whole is still greatly unqualified to truly fit accurate system-wide kinetic parameters due to limitations in data, understanding of regulatory interactions, and computational efficiency. However as reviewed by Saa and Nielsen, the work published by various groups across the kinetic modeling field overwhelmingly demonstrates that accurate kinetic predictions are possible and effective without object determination of individual kinetic parameter values.[27] The kinetic modeling work described here reflects these findings and accurate, qualitative predictions

are verified by the ability of these models to predict observed cellular behavior not used during the development of the models.

2.3 Ensemble Modeling

As discussed above, several methods have been developed to generate kinetic models from a single set of network-wide fluxomic data.[34,45–47] Unfortunately, the feasible solution space of larger networks quickly grows too large to sample fully as the number of kinetic parameters and unique solution sets increase. To address these problems, the ensemble modeling (EM) method developed by the Liao lab uses thermodynamic, stoichiometric, and steady state assumptions. These assumptions coupled with phenotypic perturbation data greatly reduce the feasible solution space and allow for more accurate kinetic parameter determination.[36,40,41,43,48]

Using the methods described by Tran et al., the only inputs required for performing EM simulations are the network stoichiometry, the steady state network fluxes, and the standard Gibbs' free energy values for each network reaction.[33] Metabolic models provide the required network stoichiometry. Well curated models for most widely-used microorganism are available and techniques for generating, novel accurate models are readily available and user friendly.[49,50] C-13 flux data sets available in the literature and constraint-based stoichiometric modeling provide the reference steady state fluxes. Lastly, values for the Gibb's free energy of each reaction, calculated using group contribution methods, are available in the literature as well as through online webtools.[15,50–53]

The traditional EM framework breaks all network reactions down into elementary reaction steps. An elementary reaction formalism is the most fundamental way to describe enzymatic kinetics and serves as the basis for other lumped kinetic approximations.[34] As detailed below, EM formulates each elementary reaction rate using mass action kinetics so they are proportional to the

concentration of all involved substrates and enzyme complexes. This aspect of elementary reaction formalism allows for easy incorporation of regulation interactions.[33] If a particular enzyme's kinetic rate law form is well-characterized it can be used instead of the elementary reaction formalism. However, the EM method does not require individual curation of each enzyme's kinetic rate law form. This alleviates any potential bias introduced through inconsistent experimental conditions.

As described by Tran et al., each reaction in the specified metabolic network is broken down into elementary reaction steps as follows[33]:

$$X_i \overset{E_i}{\leftrightarrow} X_{i+1} \tag{2.5}$$

$$\begin{array}{ccccc} & v_{i,1} & & v_{i,3} & & v_{i,5} & \\ X_i + E_i & \rightleftarrows & X_i E_i & \rightleftarrows & X_{i+1} E_i & \rightleftarrows & X_{i+1} + E_i \\ & v_{i,2} & & v_{i,4} & & v_{i,6} & \\ & j = 1 & & j = 2 & & j = 3 & \end{array} \tag{2.6}$$

so that each elementary reaction rate $v_{i,k}$ is described using mass action kinetics like $v_{i,1}$ below[33]:

$$v_{i,1} = k_{i,1}[X_i][E_i] \tag{2.7}$$

where j is the elementary step of the reaction catalyzed by enzyme i, $[X_i]$ is the concentration of metabolite X_i, $[E_i]$ is the concentration of the free enzyme E_i, and $k_{i,k}$ is the rate constant of the corresponding elementary reaction. By weighting metabolite concentrations and free enzyme and enzyme complex concentrations in the elementary reaction rate equations by their reference steady

state concentrations, $\left[X_i^{ref,SS}\right]$ and total enzyme concentrations, $\left[E_{i,total}^{ref}\right]$, respectively, Eq. (2.7) becomes[33]:

$$v_{i,1} = k_{i,1}\left[X_i^{ref,SS}\right]\left[E_{i,tot}^{ref}\right] \times \frac{[X_i]}{\left[X_i^{ref,SS}\right]} \times \frac{[E_i]}{\left[E_{i,tot}^{ref}\right]} = \widetilde{K}_{i,1}^{ref}\tilde{X}_i\tilde{e}_{i,1} \qquad (2.8)$$

Taking the log of equation (2.8), we get a linear representation[33]:

$$\ln v_{i,1} = \ln \widetilde{K}_{i,1}^{ref} + \ln \tilde{X}_i + \ln \tilde{e}_{i,1} \qquad (2.9)$$

where $\widetilde{K}_{i,k}^{ref}$ is a lumped kinetic parameter and $\tilde{e}_{i,j}$ is the fraction of enzyme i in each enzyme complex involved in each elementary step j. This log-linear form is more easily handled by computational solvers and allows for more-efficient scale up of EM methods on larger metabolic networks.[65] Now, using the reference steady state flux values provided from C-13 studies or constraint-based flux calculation simulations, we can sample the enzyme fractions and kinetic parameters for the entire network. At the reference steady state, $\tilde{X}_i$ is equal to $\frac{\left[X_i^{ref,SS}\right]}{\left[X_i^{ref,SS}\right]}$ i.e. 1. Therefore, $\ln \tilde{X}_i$, equals zero. So for Eq. (2.9) at the reference steady state we have[33]:

$$\ln v_{i,1}^{ref} = \ln \widetilde{K}_{i,1}^{ref} + \ln \tilde{e}_{i,1}^{ref} \qquad (2.10)$$

For every enzyme i, the total fraction of free enzyme i and all possible enzyme complexes between enzyme i and different metabolites participating in catalytic or regulatory elementary reaction steps must equal one[33]:

$$\sum_{j=1}^{n_i} \tilde{e}_{i,j}^{ref} = 1 \qquad (2.11)$$

where n_i represents the number of elementary steps in which enzyme i participates either in standard enzymatic catalysis or through regulation with inhibiting or activating metabolites. Additionally the reference elementary reaction rates are constrained by the provided net reference steady state fluxes, $V_{i,net}^{ref}$, through each reaction[33]:

$$V_{i,net}^{ref} = v_{i,2j-1}^{ref} - v_{i,2j}^{ref} \tag{2.12}$$

As many possible combination of the elementary reaction rates, $v_{i,2j-1}^{ref}$and $v_{i,2j}^{ref}$ could provide the required net flux rate, $V_{i,net}^{ref}$, Tran et al. sample reaction reversibilities constrained by thermodynamic principles to calculate these values instead to narrow the solution space. The reaction reversibility is defined as[33]:

$$R_{i,j} = \frac{\min\left(v_{i,2j-1}^{ref}, v_{i,2j}^{ref}\right)}{\max\left(v_{i,2j-1}^{ref}, v_{i,2j}^{ref}\right)} \tag{2.13}$$

This definition constrains reversibility values between 0 and 1. A reversibility of 0 indicates an irreversible reaction step while a reversibility of 1 indicates a reaction step in equilibrium. Using the Gibbs free energy values for each reaction, ΔG_i, which can be obtained from group contribution method calculations performed by Henry et al., the sum of all elementary step reversibilities for reaction i equals[15,33]:

$$\sum_{j=1}^{n_i} \ln(R_{i,j}) = sign\left(V_{i,net}\right) \times \frac{\Delta G_i}{RT} \tag{2.14}$$

where R is the gas constant and T is the temperature at which the reaction occurs. However as the Gibbs free energy values are dependent on the possible metabolite concentration ranges, Eq. (2.14) is reworked to reflect these possible ranges[33]:

$$\left(\frac{\Delta G_i}{RT}\right)_{lower\ bound} \leq sign\left(V_{i,net}^{ref}\right) \times \sum_{j=1}^{n_i} \ln R_{i,j}^{ref} \leq \left(\frac{\Delta G_i}{RT}\right)_{upper\ bound} \tag{2.15}$$

The derivations of Eqs. (2.14) and (2.15) are more explicitly detailed in the appendix of Tran et al.[33]

In summary, to generate a single solution set of kinetic parameters for the entire network, the following workflow is completed[33]:

1. Sample reaction reversibilities for all elementary steps j of each enzyme catalyzed reaction i. If the sampled reversibilities satisfy Eq. (2.15) for each reaction i, proceed to step 2. If Eq. (2.15) is not satisfied, then reversibilities are resampled.
2. Elementary reaction rates $v_{i,2j-1}^{ref}$ and $v_{i,2j}^{ref}$ are calculated from Eqs. (2.12) and (2.13) using the provided steady state net reaction fluxes, $V_{i,net}^{ref}$, and sampled reversibilities, $R_{i,j}^{ref}$, from step 1.

$$v_{i,2j-1}^{ref} = \frac{V_{i,net}^{ref}}{1 - R_{i,j}^{sign\left(V_{i,net}^{ref}\right)}} \tag{2.16}$$

$$v_{i,2j}^{ref} = \frac{V_{i,net}^{ref} R_{i,j}^{sign\left(V_{i,net}^{ref}\right)}}{1 - R_{i,j}^{sign\left(V_{i,net}^{ref}\right)}} \tag{2.17}$$

3. If enzyme fractions, $\tilde{e}_{i,j}^{ref}$, are unknown, sample $n_i - 1$ enzyme fractions for each enzyme i to satisfy Eq. (2.11).
4. Calculate kinetic parameters, $\tilde{K}_{i,j}^{ref}$, from Eq. (2.10) using elementary reaction rates calculated in step 2 and enzyme fractions sampled in step 3.

So for each unique set of sampled reaction reversibilities and enzyme fractions, a unique set of possible kinetic parameters to define the system can be generated. Each solution set of kinetic parameters is considered to be one potential model describing the kinetics of the entire metabolic system.[64]

$$Model_n = f\left(\boldsymbol{R}_{i,j}^{ref}, \tilde{\boldsymbol{e}}_{i,j}^{ref}\right) \tag{2.18}$$

Using the models, we can calculate the change in metabolite concentrations over time by solving the following ODE for each model[64]:

$$\frac{d\tilde{X}_i}{dt} = \frac{1}{\tilde{X}_i^{ref,SS}}\left(\sum v_{generation} - \sum v_{consumption}\right) \tag{2.19}$$

Completing the above workflow iteratively, an initial ensemble of models can be generated, all of which will contain different feasible kinetic parameter sets to describe the reference steady state flux behavior. To narrow down the true kinetic parameter solution, each model can be perturbed. These perturbations can include an enzyme knockout or overexpression. Data sets containing resulting fluxes from *E. coli* and other organism genetic perturbations are available in the literature for a variety of conditions. Therefore, after perturbing the models, new steady state fluxes can be calculated using the kinetic parameters from the initial ensemble. Then models whose resulting fluxes best match the behavior of the experimental data being considered are kept for additional screening while models with incorrect behavior are removed from the set. In this manner iterative rounds are run to continuously narrow down the models that correctly describe all tested perturbation conditions. Once a robust sampling of models remains, they can be used to predict the behavior of future genetic perturbations to aid in strain design. As described by Tran et al., to perturb

individual models by changing the enzyme concentration (i.e. through a knockout or overexpression), the elementary reaction rates are recalculated as[33]:

$$v_{i,1} = k_{i,1}\left[X_i^{ref,SS}\right]\left[E_{i,tot}^{ref}\right] \times \frac{\left[E_{i,total}\right]}{\left[E_{i,tot}^{ref}\right]} \times \frac{[X_i]}{\left[X_i^{ref,SS}\right]} \times \frac{[E_i]}{\left[E_{i,tot}^{ref}\right]}$$
$$= \tilde{K}_{i,1}^{ref} E_{i,r} \tilde{X}_i \tilde{e}_{i,1} \quad (2.20)$$

where $\left[E_{i,total}\right]$ is the new total concentration of enzyme i after the perturbation and $E_{i,r}$ is the fold change of enzyme i. The traditional ensemble modeling framework is best depicted pictorially in Fig. 2.1, from Contador et al.[41]

Other groups have improved on the EM method and have utilized it successfully to guide metabolic engineering efforts in traditional growth coupled microbe systems.[41,43,48] One such improvement by the Maranas lab involves a less limiting phenotype data screening process.[39,54] Their method involves using a genetic algorithm method to screen through grouped elementary reaction kinetic parameter sets. It mixes and matches elementary parameter sets clustered by enzyme from different models generated in the initial ensemble seeding. It then looks to see if different combinations of these theoretically accurate parameters better fit the provided data than the original

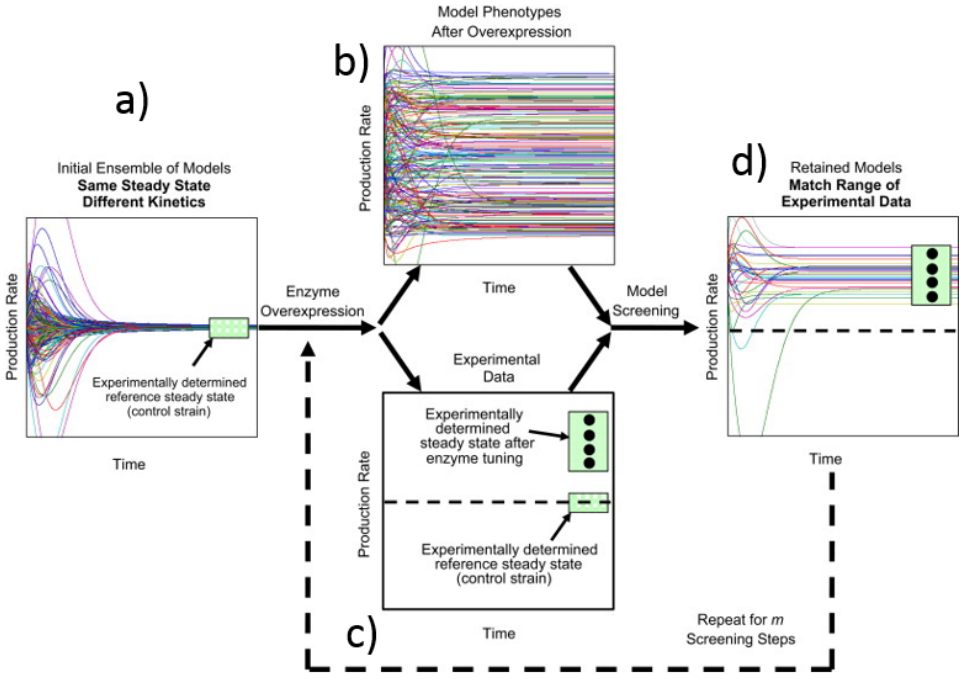

Figure 2.1: Metabolic Ensemble Modeling Framework. a) Given a metabolic network, a set of dynamic models are sampled that reach the same steady-state flux at a given condition. b) After changing an enzyme concentration, the different dynamic models predict different flux solutions. c) The flux at new enzyme concentration is measured experimentally. d) Dynamic models unable to predict the new perturbed steady state flux are discarded, and the cycle is iterated. *Figure from Contador et al.*[41]

individual model sets. This method keeps all originally generated parameters in contention for the solution rather than completely throwing out models that do not accurately predict the data gathered for a number of cellular perturbations. Additionally, ensemble modeling has been used to interrogate the robustness and stability of metabolic pathways.[38,42] These efforts highlight the expansive applications for kinetic ensemble modeling and the potential new areas for using this tool to better understand cellular metabolism.

Despite its many benefits, kinetic ensemble modeling still requires improvements and further characterization to understand its limits. In its traditional form, ensemble modeling attempts to perform the difficult task of sampling and fitting many (anywhere from 10^2 to 10^3) kinetic parameters using a small amount of experimental data (typically less than ten). To best perform this task, a large

number of parameter sets need to be sampled and screened. However, integrating the parameter sets to solve for the new steady state metabolite concentrations is time intensive so large implementations of the method are computationally limited. The work detailed in Chapter 4 features the improvements we have made to the traditional ensemble modeling framework that now allow us to complete ensemble modeling faster and generate more robust, biologically relevant solutions.

Chapter 3: Constraint-Based Modeling of Non-Growth Metabolism

3.1 Introduction

3.1.1 Decoupling Growth and Production

Most commercially developed applications utilizing microbial cells to generate products at an industrial scale employ growing cells. This is not ideal as the primary objective of most cell types is to grow and reproduce when in a resource rich environment. Therefore, microbial production is inherently limited by substrate loss to the cell's native competitive growth pathways. Some groups have harnessed this survival-based reproduction drive of cells by computationally designing growth-coupled production pathways that require the production of their target product to achieve growth.[55,56] For example, to increase production of lactic acid, knocking out phosphofructokinase, a non-intuitive choice, can help improve production. This particular knockout forces more of the cell's carbon flux down the pentose phosphate and Entner-Doudoroff pathways leading to increased NADPH and pyruvate production. The lactate production pathway consumes pyruvate and NADPH. Then to restore redox balance and maximize growth potential, the cell will also reroute more carbon flux toward producing lactic acid.[20] While this clever workaround does increase overall titers by coupling the cell's growth and production objectives, it is limited in the type of product pathways one can target. Most importantly, this method still suffers from substrate loss to growth.

The above scenario exemplifies how microbial production in growing cells does not allow achievement of maximum theoretical product yields. Regardless of how many other side reactions and carbon sinks are eliminated, a sizeable portion of fed carbon sources still must go toward growth-related processes and essential metabolite production. In some systems, more than fifty percent of fed carbon substrate goes toward biomass production and maintenance requirements.[57] As biomass is often not the desired product, its formation should really be re-envisioned as an unwanted side

reaction to the desired chemical synthesis. In this view, eliminating biomass production is an obvious area to target for optimizing cellular production processes moving forward.

Some groups, recognizing this potential, are working to eliminate substrate loss to growth processes by developing cell free systems. In these systems, purified enzyme mixtures or crude cell lysates are used to perform metabolic chemistry without using living, growing cells.[58–60] In this way, cell free systems do not lose carbon and energy substrates to biomass production. However, cell free systems are still subject to limiting regulatory effects.

The alternative to growing cells or using a cell free system would be to harness non-growing cells. To do this we would need to 1) halt the cell's native desire and ability to grow and 2) remove or inhibit the cell's regulatory mechanisms preventing metabolic activity in the absence of growth. This second point is the idea of inducing cellular quiescence, a non-growth state in which cells maintain metabolic activity. Instead of quiescence, *E. coli* tend to enter dormant or non-growth states characterized by low metabolic activity and increased resistance to environmental stresses.[61]

Microbes enter a stationary phase under conditions of environmental stress or nutrient starvation. Historically, stationary phase metabolism studies are performed under carbon-starvation conditions while carbon-rich stationary phase behavior is not well characterized. Without carbon, many microbes are unable to produce sufficient energy to be metabolically active. However, non-carbon nutrient starvation and genetic perturbations can also induce stationary phase even in carbon-rich environments. In this chapter, we have identified non-carbon starvation strategies using computational metabolic modeling methods.

3.1.2 Modeling Non-Growth Metabolism

We specifically use constraint-based stoichiometric steady state models to predict engineering strategies for turning off cellular growth without abolishing metabolic activity. We hypothesize that identifying strategies leading to a net flux distribution throughout metabolism most similar to the native wild type (WT) flux distribution are the best targets. Unfortunately, the traditional objective functions utilized in the flux balance analysis (FBA) and flux variability analysis (FVA) methods are less suited for modeling non-growth metabolism. Specifically maximizing biomass production cannot be used because the cells are not growing. As far as using other objective functions, it is not completely understood what a cell's specific metabolic objective is while under growth arrest. Survival is the believed objective, but the primary mechanism for that metabolically is not known; it may not be a single objective. We do know resting cells must generate maintenance energy requirements, but additional requirements for survival are not well-characterized.

Instead we can look to the Minimization of Metabolic Adjustment (MOMA) method for answers. MOMA is a quadratic programming method designed to identify the flux solution set with the smallest overall difference from a reference flux solution set.[24] This idea is based on the solution that cells, in small time frames following a genetic or environmental perturbation in which full cellular evolution would not have time to transpire, will reroute fluxes in the smallest way possible to survive. Therefore, the optimal solution following a perturbation is the solution closest to the reference solution. If a reaction is knocked out, then MOMA will provide a solution set closest to the reference solution set that still meets all previously defined mass balance constraints (Fig. 3.1). The creators of MOMA argue that while FBA may provide the best solution for an evolved strain after a perturbation, MOMA provides a more accurate representation of cellular network flux immediately after an environmental or genetic perturbation.[24,62,63]

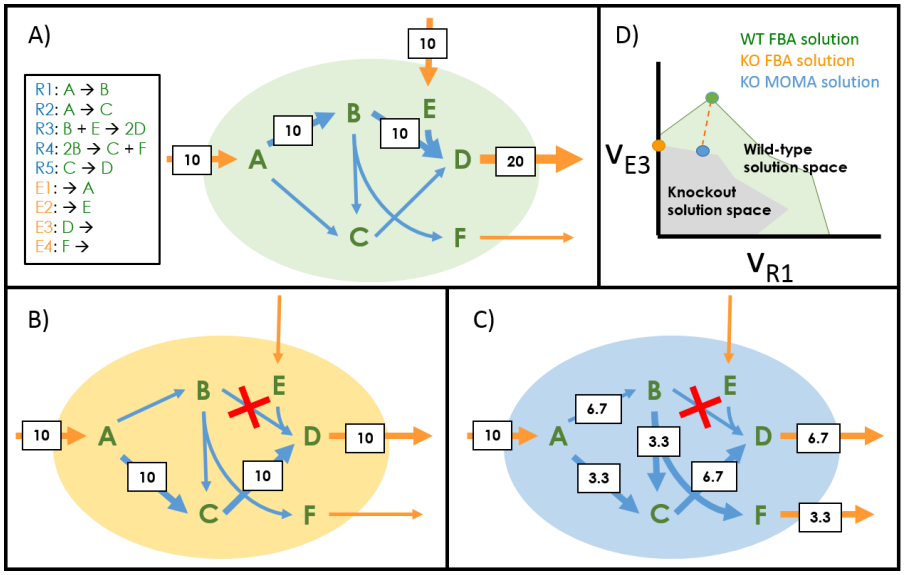

Figure 3.1: Example of difference between MOMA and FBA predicted solutions. A) WT network flux solution to maximize production of metabolite D. B) FBA solution to maximize production of D if reaction R3 is knocked out. C) MOMA solution if reaction R3 is knocked out. D) Example of changes in solution space after knockout. FBA solution select optimal path for production of D. MOMA instead predicts solution closest to original network flux solution. *Figure adapted from Segrè et al: OMICS: A Journal of Integrative Biology 7(3): 301-316 (2003).*

Ultimately, MOMA provides two key benefits toward our objective of modeling non-growth cellular metabolism. First, it can generate non-growth solution sets. Second, it better predicts immediate network flux changes in cells. This is more appropriate in non-growing cells with limited capacity to undertake the protein network overhaul required to achieve the new optimal flux solution. To best design non-growing, metabolically active cells we need to harness the power of metabolic modeling to fully understand the current limitations and bottlenecks preventing a full metabolic program. While many metabolic methods have been developed, they have not yet been applied to non-growth metabolism. Using metabolic modeling, we can begin to understand differences in experimental observations for cells entering a non-growth state under different environmental and genetic conditions. Different perturbation types lead to different levels of metabolic activity. Through modeling, we can begin to understand what metabolic features lead to

"healthier" non-growing cells and what limitations are hindering catabolic activity through production pathways the most.

In this work we identify a set of thirty essential reaction knockouts causing auxotrophy in cells for one or more non-carbon metabolites. We also predicted the metabolites rescuing the auxotrophy to aid in experimental design. These candidates are now being characterized experimentally and the results of those tests will help us ascertain the ability of MOMA to predict non-growth metabolic pathway flux distributions.

3.2 Materials and Methods

3.2.1 Metabolic Model

In this work we used the genome-scale metabolic model of *Escherichia coli*, iJO1366.[12] This model includes 2251 metabolic reactions and 1136 metabolites. It includes all core reactions of *E. coli* metabolism and extensively covers secondary metabolism as well.

3.2.2 Minimization of Metabolic Adjustments (MOMA) Simulations

To identify all single enzyme knockouts leading to non-growth, a MOMA calculation was performed to test elimination of each reaction in the network. All MOMA simulations were performed using the COBRA Toolbox.[64] First a reference wild type (WT) flux distribution was determined by fitting experimental fluxomic data published by Haverkorn et al. to the iJO1366 model.[65] Thirty-three model fluxes were constrained to match the published experimental values within reported confidence intervals. The flux constraints imposed are listed in Appendix A. Then, flux balance analysis was performed to determine the flux values for the remaining reactions in the network. The solution maximizing flux through the biomass reaction was selected as the reference. The MOMA framework uses the reference flux distribution when calculating the perturbed state flux distributions (Fig. 3.1). The solver identifies the solution closest to the reference state that does not

violate the added constraints (i.e. no flux through the knocked-out reaction). To simulate knocking out reactions, the constraints were set to match the reference state. However, the minimum glucose uptake flux was set to -1000 (i.e. allowed to be any value) and the flux through the knocked out was set to zero. Additionally, the lower bound of the non-growth associated ATP maintenance reaction was set to 0 rather than 3.15 mmol ATP $gDCW^{-1}$ h^{-1}, the requirement determined by Orth et al. for maximum growth on glucose.[12] This change was made as we assume ATP requirements for maximal growth will not necessarily hold for a non-growth metabolic state.

The MOMA solutions for each reaction knockout were compared to the reference flux. Knocked out reactions resulting in solutions predicting zero flux through the biomass reaction were marked as essential.

3.2.3 Predicting Auxotrophies

MOMA simulations of each essential knockout candidate on glucose minimal media were used to identify if one or more metabolites could rescue the growth of each essential knockout candidate. These simulations allowed us to design a rich media containing all the supplements required for each knockout to achieve growth. Similar to determining essential reactions above, we used the same reference WT flux state. However, when simulating the MOMA results for a reaction knockout we also allowed uptake of a single media supplement to test if it would rescue growth. Each essential knockout was simulated across each of the 324 exchange metabolites. When no one supplemental metabolite could rescue growth, the knockout pathway was examined individually, and new simulations were run where combinatorial supplement metabolites were allowed to enter the cell. Essential knockouts able to predict growth when one or more additional metabolites was available for uptake by the cell were labeled as auxotrophic essential knockouts.

3.2.4 Computational Resources

All calculations were written and executed in MATLAB (MATLAB and SimBiology Toolbox 2015/2016a, The MathWorks, Inc., Natick Massachusetts, United States). Scripts and results for each reaction tested and the resulting auxotrophy predictions are available in the Tyo lab Box repository.

3.3 Results

3.3.1 Identifying Essential Reactions

After completing single knockout MOMA simulations, we identified 279 internal and 21 external reactions essential for biomass production (Appendix E). The essential knockouts span 28 of the 37 subsystems identified in the iJO1366 model (Fig. 3.2). However most of the knockouts fall into cofactor and amino acid synthesis pathways indicating the inability to produce essential biomass precursors prevents growth most often.

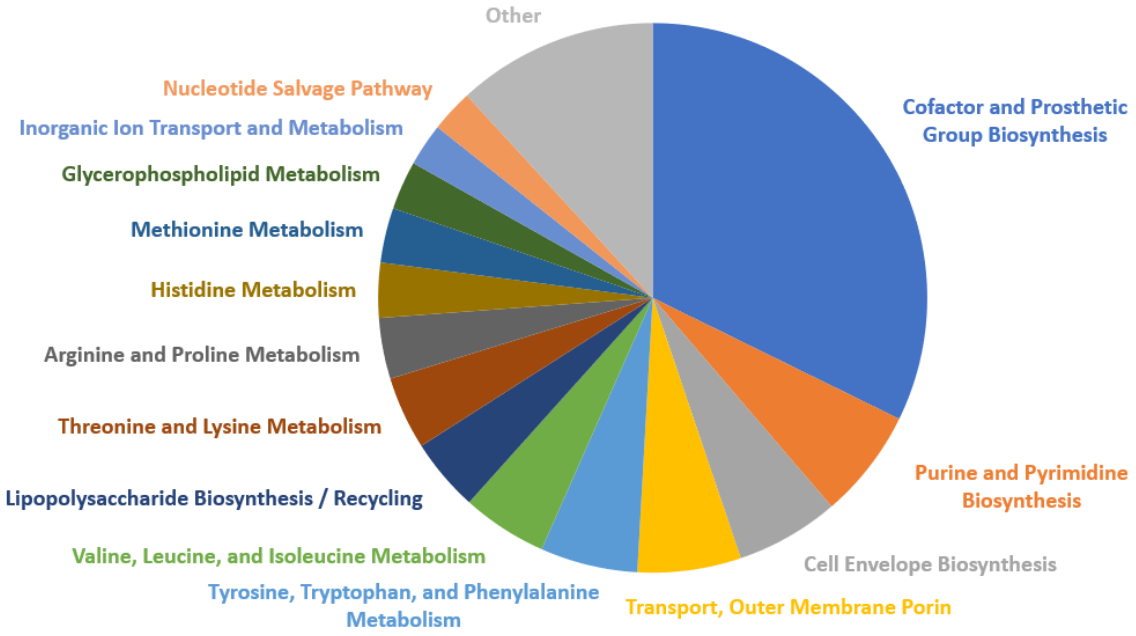

Figure 3.2: Essential reaction span most metabolic network subsystems. The 'Other' category includes the following subsystems: Cysteine Metabolism; Transport, Inner Membrane; Citric Acid Cycle; Glycolysis/Gluconeogenesis; Alanine and Aspartate Metabolism; Alternate Carbon Metabolism; Folate Metabolism; Membrane Lipid Metabolism; Murein Biosynthesis; Glutamate Metabolism; Oxidative Phosphorylation

3.3.2 Designing Rich Media to Rescue Essential Auxotrophies

Of the approximately 300 target essential reactions, we identified that 153 of these reactions could achieve growth through supplementation of a single metabolite. Furthermore, 95 of these reactions were able to grow by supplementation of several different individual metabolites. See Appendix D for a list of auxotrophic reactions and metabolites predicted to recover growth.

Thirty of these reactions inducing cellular autotrophy were chosen as a representative subset of the E. coli core metabolism (Fig. 3.3) for experimental characterization (see Appendices B-C for details). These thirty reactions span *E. coli* metabolism and include amino acid, nucleotide, and vitamin auxotrophies.

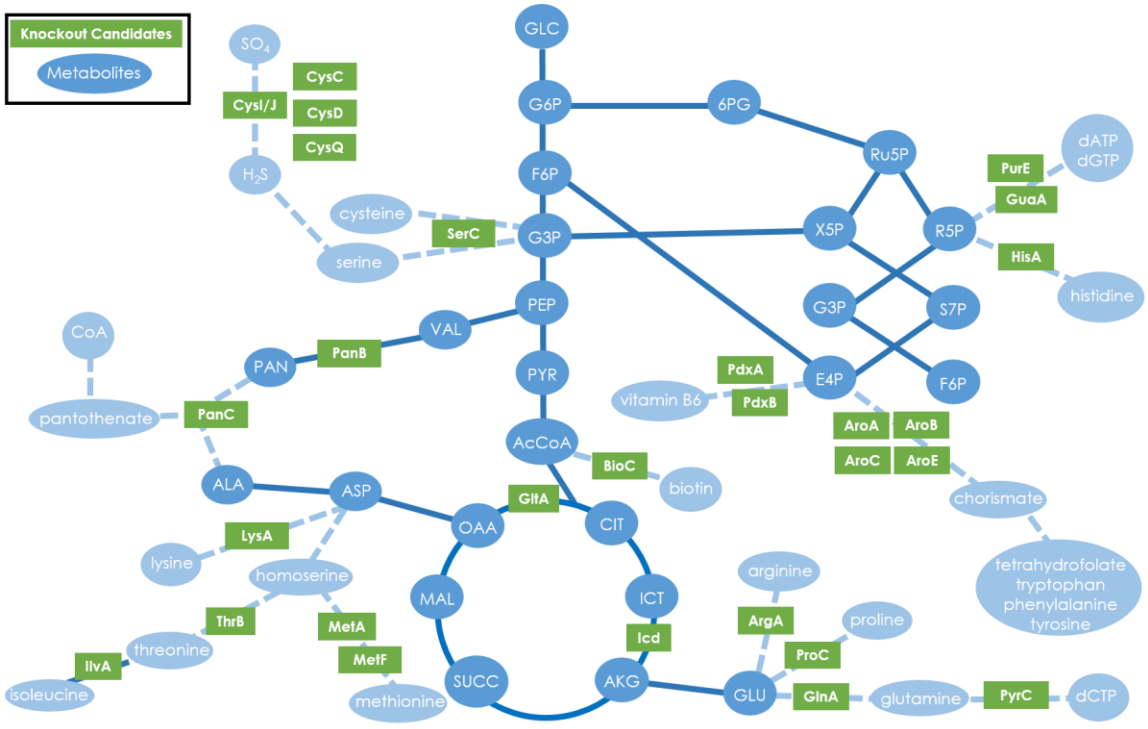

Figure 3.3: Essential reaction knockout candidates selected for experimental characterization. Blue circles indicate metabolites. Green rectangles indicate knockout candidates. Solid lines indicate direct enzymatic reactions. Dashed lines indicate a pathway between reaction includes two or more individual enzymatic reactions. Abbreviations for metabolite and enzyme names are taken directly from original iJO1366 publication and Ecocyc database.[12,51]

3.4 Discussion

In our design scheme we need to be able to rapidly shut off the essential reaction we aim to study after the cells have grown up to a desired level. As the reactions are essential, we cannot generate a starting culture if they are initially inactive. Collaborators in the lab are working on different experimental methods for rapid induction of protein degradation and nutrient-dependent toggles for gene expression. However, for this work, we wanted to identify a quicker, experimentally easier method for analyzing and modeling the metabolic activity of non-growth metabolism. To do this we chose thirty unique reaction knockouts inducing cellular auxotrophy from the 205 auxotrophic candidates. Auxotrophic cells are unable to catabolize one or more essential metabolite required for biomass production. Therefore, if the auxotrophic cell is not grown in media supplemented with its missing essential metabolite(s), it is unable to grow. Usefully, we can grow these auxotrophic cells up in rich media containing the essential metabolite they cannot make. Then we can switch them to a minimal media that does not contain their needed essential metabolite to halt their growth. The minimal media does contain glucose allowing us to monitor the metabolic activity, if any, these non-growing cells display. By rapidly inducing a non-growth state, we can better monitor and capture the catabolic activity of non-growing cells in a carbon rich environment.

Experimental collaborators in the lab, led by fellow PhD student Will Bothfeld, are currently collecting data including the glucose uptake and product secretion rates for these thirty auxotrophic cells. The thirty reactions map to thirty different enzyme knockouts available in the Keio collection.[66] The data from these lab studies will be used to calculate the strength of MOMA as a modeling method for non-growth metabolism using the methods described above. Initially studies were done using MOPS EZ rich media containing supplemental metabolites covering the auxotrophies of a majority of the candidate knockouts.[67,68] The results of this preliminary study are provided in Appendix B. These

initial observations confirm we have accurately predicted the auxotrophic metabolites needed for each reaction knockout. Additional experiments were then performed using minimal media only containing the predicted additional metabolite and then switching to minimal media without the supplement to further confirm our auxotrophy predictions. These experiments are referred to as shock switch experiments and the initial results have been presented in the master's thesis of Michael Brotz (2015) and summarized in Appendix C.

Ideally once further experimental characterization is completed, the glucose uptake and product fluxes measured can be compared to the MOMA flux distribution predictions for each candidate. Should the MOMA predictions not accurately predict the observed results, additional constraint-based methods and different non-growth hypotheses could be tested. However, this work does display the utility of using MOMA to predict essential single enzyme knockouts and to design experimental conditions needed to test auxotrophic behavior.

Chapter 4: Acceleration strategies to enhance metabolic ensemble modeling performance

4.1 Introduction

Enabling kinetic and regulatory modeling of cellular metabolism is a major challenge in metabolic engineering and systems biology.[35,69–73] Constraint-based stoichiometric modeling greatly aids in characterizing and improving strain designs, but without kinetic information, it is difficult to identify rate limiting steps and interrogate regulatory behavior. Some studies using kinetic models for metabolic applications do exist, but they are limited. For example, individual kinetic models were developed for the red blood cells of 24 different patients to interrogate differences in metabolite levels and enzyme activities which are difficult to capture with constraint-based models alone.[74] Another example is in strain design efforts where the cellular objective of maximizing growth cannot be assumed (i.e. studying stationary, non-growth phase metabolism), the constraint-based stoichiometric methods are harder to utilize effectively. The ability to incorporate extensive regulatory behavior in kinetic models is also useful when studying systems where regulation heavily governs a cell's metabolism and even prevents cells from operating at maximum metabolic capacity.[47,75,76] Constraints-based models can not explicitly track metabolite concentrations, making regulation based on metabolites difficult. Ultimately, generating quality kinetic models of cellular metabolism will allow us to better resolve and interrogate cellular metabolism for strain design and biological discovery applications.

To build a kinetic model of metabolism the rate laws and parameters are needed for each enzyme in the network. Some kinetic modeling methods combine rate laws and kinetic parameters from public databases or literature and combine them into a single metabolic model.[77,78] Unfortunately, the *in vitro* derived kinetic parameters for enzymes most often reported in literature

do not necessarily reflect true *in vivo* behavior and are often determined under varying experimental conditions without accounting for local concentration effects.[78] Moreover, for some enzymes these *in vitro* parameters and rate laws have not been determined, and exhaustive regulatory relationship studies have not been completed. Consequently, a single kinetic model combining these *in vitro* derived parameters or several smaller models built on these parameters are often unable to resolve experimentally observed *in vivo* data or describe metabolic states outside the immediate realm of the reference state.[29,79,80]

The ensemble modeling (EM) framework was previously developed to address these hurdles by sampling kinetic parameters for the entire metabolic network simultaneously and screening them against an experimental dataset collected under consistent conditions.[33,34,36] During the screening step, predictions from the sampled kinetic parameters are compared to experimental results, and kinetic parameters that predict the results poorly are rejected. While estimates of individual kinetic parameters may not be strictly accurate, EM seeks to develop a network model that explains system behavior. Furthermore, the EM method constrains the large kinetic parameter sample space using readily available thermodynamic, stoichiometric, and steady state flux data. The EM method has been successfully employed to model and resolve the kinetics of various metabolic pathway designs for bioproducts and to interrogate cancer metabolism.[40,41,43,48] Additionally, improvements made to the screening methods used in EM led to the development of a kinetic core model of *E. coli* metabolism and more recently a genome-scale model, k-ecoli457.[39,54] Lastly, the way reactions are defined in ensemble modeling allows for sampling different network structures to interrogate different potential governing or previously unknown regulation reactions as well as predict unresolvable flux distributions previously unconstrainted in traditional constraint-based methods.[43,81]

However, despite its numerous advantages, EM rapidly becomes computationally limiting with increasing network size and complexity. As our field increases the availability and uses of larger, genome-scale metabolic models for various organisms, it is imperative to improve our ability to generate larger kinetic models of these systems as well.[8,35,71,82,83]

To enhance the computational efficiency of EM, it is critical to focus efforts on the step in which predictions created by the generated kinetic parameter sets are compared to the available experimental perturbation data, as it is the major rate limiting step of the EM process. To compare to a single data set, each parameter set in the ensemble is perturbed by solving a system of ordinary differential equations (ODE) with different concentrations in one or more proteins (i.e., knockout or overexpression). Solving a single ODE calculation is time intensive and screening parameter ensembles across the available perturbation data sets requires many iterations of these calculations. In total, these ODE-based screening steps account for over 99% of the required time to complete the traditional EM method, and we observe that screening times scale non-linearly with increasing metabolic network size.

Another prevalent challenge with the EM method is trying to resolve a large number of kinetic parameters using a limited amount of experimental observations. The predictive power of the resulting kinetic model can be limited by the quality of the data used to train it. Generating more experimental data at different states is costly, time intensive, and, depending on the type of experiment, sometimes physically infeasible. Kinetic model systems are traditionally undetermined as there are far more parameters to fit than training data sets available. Therefore, the problem is ill-posed and several, unique kinetic parameter sets can be found with equal ability to fit any one flux distribution. At the same time, unavoidable experimental error or noise means it is likely there will be no one set that can describe all the flux distributions at the different cellular states. This challenge

intensifies as the need to develop larger models with increasing numbers of kinetic parameters to identify grows while the availability of experimental observations to screen against does not increase at the same rate. To alleviate this widening gap, additional constraints to narrow the kinetic parameter sample space and select more accurate kinetic parameters are warranted.

In this work, we addressed the computational limitations and parameter sampling difficulties inherent in the EM method. We have greatly reduced overall EM run times by utilizing additional parameter screening techniques and introducing previously developed methods to reduce structural model complexity during ODE integration (Fig. 4.1). Specifically, we have implemented a conservation analysis step to eliminate linear dependency present in the EM models used to date. We found eliminating linear dependency reduces the stiffness and screening time of the kinetic parameter sets. We have also elucidated a preferred method for selecting the order of screening data sets to reduce the number of incorrect parameter sets carried forward in each screening iteration. To improve kinetic parameter sampling, we have also further characterized and implemented a parameter screening step capitalizing on the known stability of wild-type cellular metabolism. This screening method was previously incorporated for robustness analysis of metabolic pathways designs and metabolic control analysis.[38,42,46,84] It addresses observations of local instability in traditional EM solutions and reduces the kinetic parameter sampling space by removing parameter sets not locally stable at the initial wild-type (WT) steady state condition. By incorporating this screen into the existing EM framework (Fig 4.1), our final parameter set solutions more accurately reflect true biological behavior. In this work we specifically demonstrate how local instability increases with network size and slows down parameter screening making the case for incorporating this additional screen in all ensemble modeling efforts outside of its original use in ensemble modeling robustness analysis (EMRA) studies. Through our efforts to speed up and optimize the method, we are overcoming some of ensemble modeling's inherent limitations and moving toward

enabling the method for more widely-accessible application. Furthermore, our work can benefit concurrent EM efforts to develop genome-scale kinetic models across our field.[8,82]

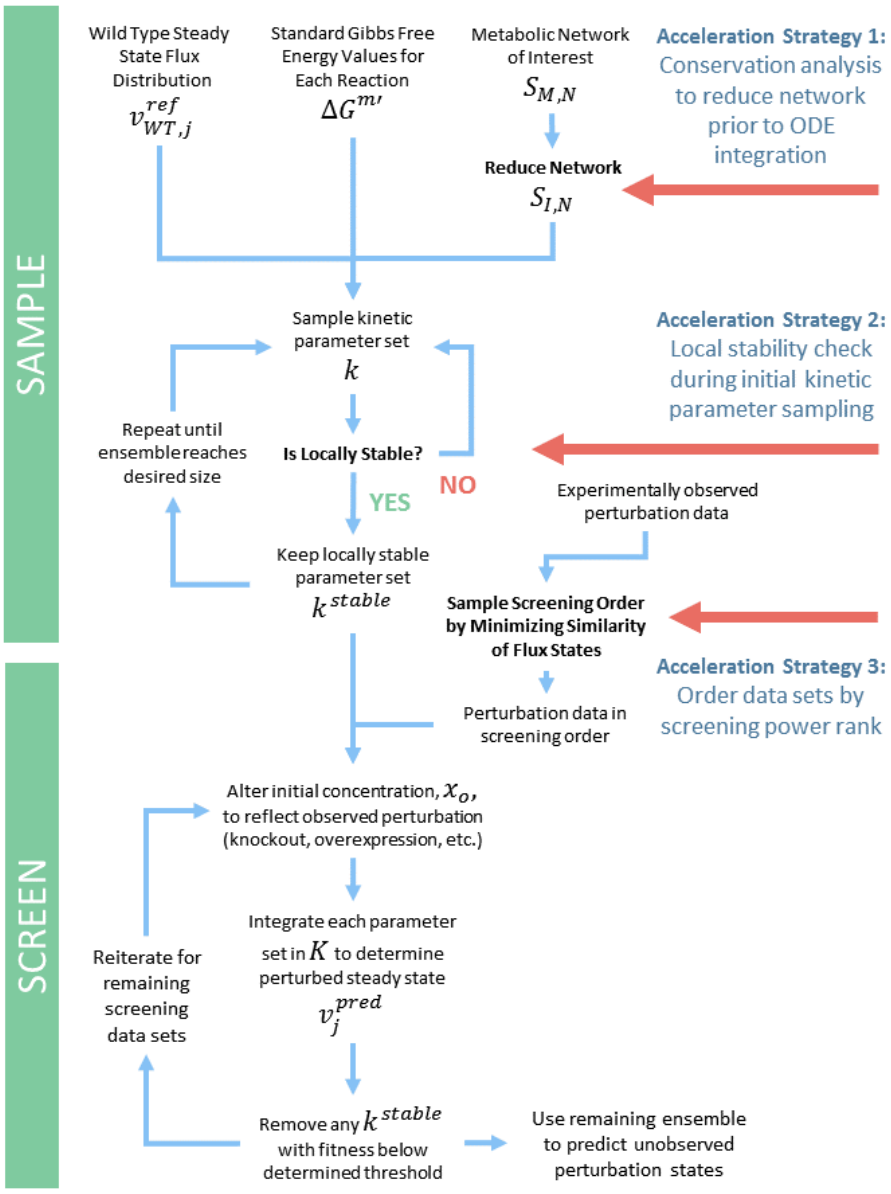

Figure 4.1: Modifications to Base Ensemble Modeling Framework. General overview of base ensemble modeling framework highlighting where to apply acceleration strategies. In this work, we have implemented conservation analysis to reduce our metabolic network and only track independent metabolite and enzyme fraction species (Acceleration Strategy 1). We have also characterized the local stability check after implementing it after the initial kinetic parameter sampling process (Acceleration Strategy 2). Lastly, we have described the benefits of pre-ranking perturbation data sets prior to screening the ensemble (Acceleration Strategy 3).

4.2 Methods

4.2.1 Metabolic Models

Three metabolic models were used for this study: a small toy model with six reactions adapted from a preexisting toy model[34]; a medium model with 34 reactions created to resemble simplified central carbon metabolism; and a large model with 138 reactions adapted from the core *E. coli* model provided by Khodayari et al. (Fig 4.2).[39] Minor changes were made to the large model to remove duplicate metabolite entries and correct reaction stoichiometry (Table S1 in the Supplemental Material of Greene et al.).[85] MATLAB (The MathWorks, Inc. Natick Massachusetts, United States) model files are available in the supplement and reaction stoichiometries are detailed in Supplemental Tables S2-S11 of Greene et al.[85] The previously published k-ecoli457 model was also used to demonstrate the opportunity to use conservation analysis on genome-scale models.[54] To use this model, duplicate reactions and metabolites were removed as with the large model prior to applying conservation analysis calculations (Supplmental Table S1).[85]

4.2.2 Obtaining WT steady state flux distribution

The experimental fluxomic data for the wild-type and enzyme knockout conditions used to screen the large model were taken from the Ishii study.[86] Forty-two of the flux values reported in this data were mapped to our large model. The method previously described by Khodayeri et al. was adapted to determine the full WT flux distribution of our large metabolic model.[39] First to determine the flux values of unmapped reactions in the large model under growth-optimized conditions, 31 reactions in the genome scale *E. coli* model, iAF1260, were mapped to the experimental fluxes, and the reported 90% confidence interval values were used to constrain the bounds of these reactions.[86,87] Next the Flux Variability Analysis (FVA) method provided in the COBRA toolbox was

used to determine the flux bounds for the remaining iAF1260 reactions given the experimental flux constraints for the 31 reactions and the objective of maximizing growth.[88,89] Then the literature data

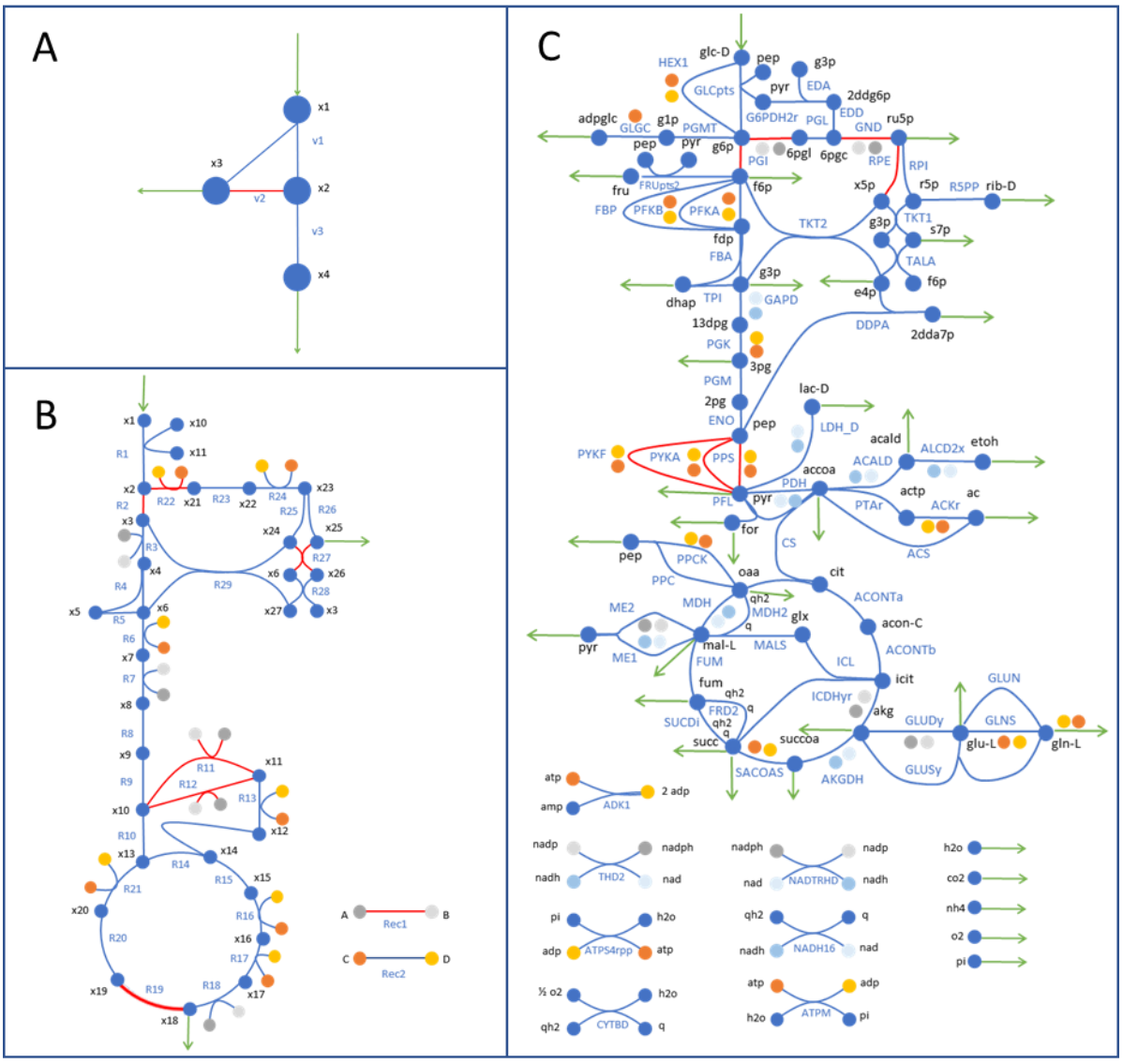

Figure 4.2: Test Metabolic Networks used Throughout Study. A) Small model with six enzymatic reactions adapted from a preexisting toy model.[34] B) Medium model with 34 enzymatic reactions and 3 regulation reactions created to resemble a simplified central carbon metabolism. C) Large model with 138 enzymatic reactions and 60 regulation reactions adapted from the core *E. coli* model provided by Khodayari et al.[39] The line color denotes the reaction type: blue (internal reaction); green (exchange reaction); red (reaction knocked out during screening and time trial testing). Metabolites are represented by blue circles, and common cofactors present in the medium and large models are uniquely colored to distinguish their repetitive occurrence throughout the models.

and the iAF1260 FVA results were used to determine the WT flux distribution of our 138 reaction large model. First, the 90% confidence interval flux values reported for the 42 reactions reported in the Ishii et al. dataset were used to constrain their corresponding large model reactions. The bounds for the remaining unmapped reaction for the large model were then set using the iAF1260 growth-optimized FVA results as the upper and lower bounds. Next, the COBRA toolbox Flux Balance Analysis (FBA) function was used to determine a feasible WT steady state flux distribution constrained by both the experimental values and the previously generated growth-objective driven FVA results in the large model.[90] Detailed information on reaction mappings used to constrain iAF1260 and the large model in these calculations is provided in the published supplemental materials (Supplemental Tables S12-S15).[85]

For the small and medium models, a random flux distribution satisfying the mass-balance constraints of the model was generated and consistently used as the reference WT flux distribution throughout all simulations.

4.2.3 Sampling Kinetic Parameter sets

This work builds directly on the previously developed ensemble modeling method which uses steady state metabolic flux data and known thermodynamic constraints to generate and screen kinetic models.[33,34] Each reaction in our defined metabolic networks was broken down into elementary steps defined by elementary kinetic rate laws. Elementary kinetic rate laws are most commonly used in ensemble modeling publications as they are generalizable to all reactions and allow for easy incorporation of regulatory interactions.[33,39–41,43,44,48,54,91,92] For example, the overall reaction of enzyme E converting reactant A to product B would be written as

$$A + E \underset{v_2}{\overset{v_1}{\rightleftarrows}} AE \underset{v_4}{\overset{v_3}{\rightleftarrows}} BE \underset{v_6}{\overset{v_5}{\rightleftarrows}} B + E \tag{4.1}$$

The changes in metabolite concentrations in our network over time were defined as

$$\frac{dx}{dt} = S_{M,N} v(x,t); \;\; x = x_o \; at \; t = 0 \tag{4.2}$$

where x is the m x 1 vector of metabolite concentrations and enzyme fractions and v is the n x 1 vector of elementary reaction fluxes. The initial condition x_o contains the metabolite concentrations and enzyme fractions at timepoint 0. $S_{M,N}$ is the M x N elementary stoichiometric matrix where each column refers to the elementary reactions and each row refers to a metabolite or enzyme species fraction. The values in the matrix refer to the reactant or product stoichiometry for each elementary reaction. Each elementary reaction flux is defined as

$$v_n = k_n \prod_{\substack{m \\ S_{m,n}>0}} x_m^{S_{m,n}} \tag{4.3}$$

where k_n is the elementary kinetic parameter for the corresponding elementary reaction. All the elementary kinetic parameters for the entire metabolic model were combined into one set and stored in the n x 1 vector k.

After obtaining the WT steady state flux through each net model reaction as described above, several individual kinetic parameter sets were sampled following the procedure previously developed by the Liao group.[33,34] The parameter sampling was constrained by the WT flux values as well as the Gibbs free energy ranges possible based on the standard Gibbs free energy of the overall reaction and selected metabolite concentration ranges. Because the kinetic parameter estimation problem is ill-posed, i.e. we are trying to determine many unknown parameters from a relatively small set of experimental data, there are several unique parameter sets that correctly predict the

wild-type steady state metabolism. Therefore, several, unique k sets were sampled that, despite having different kinetic profiles, all reach the same steady state.

4.2.4 Screening Kinetic Parameter Sets

To further parse the sampled kinetic parameter sets, perturbed state experimental data sets were compared to parameter set predictions. For example, using flux or concentration data for a cell after a reaction knockout, that same knockout perturbation can be simulated with the sampled parameter sets by solving a system of ordinary differential equations with the protein of the reaction knockout set to zero, as we describe below. Then parameter sets unable to predict this new perturbed state can be screened out and only the parameter sets accurately predicting the wild type and knockout phenotypes are kept. This screening step can be rerun iteratively until a small subset of kinetic parameter sets able to accurately predict all available screening phenotypes is left. Furthermore, the resulting post-screen kinetic parameter set ensemble then provides predictive power as a metabolic engineering design tool to interrogate untested phenotypes.

To screen parameter sets sampled for the large model using knockout fluxomic data, the WT initial condition concentration values were used but the enzyme fractions corresponding to the reaction being knocked out were set to 0. The system of equations defined by Eq. (4.2) is then integrated for each parameter set using the altered initial condition to find the new steady state post enzyme knockout. To determine the predictive power of each parameter set, a scoring function to calculate the fitness (y) was devised:

$$y = \frac{1}{n_{KOs}} \sum_{k=1}^{n_{KOs}} \frac{1}{n_{fluxes}} \sum_{j=1}^{n_{fluxes}} \frac{1}{CV} \left| \frac{v_{k,j}^{pred} - v_{k,j}^{ref}}{v_{k,glucose\ uptake}^{ref}} \right| \tag{4.4}$$

where the absolute difference between the predicted flux ($v_{k,j}^{pred}$) and experimentally observed flux ($v_{k,j}^{ref}$) weighted by the experimentally-observed uptake flux of glucose ($v_{k,glucose\ uptake}^{ref}$) for the given perturbation was averaged across all available (1 to n_{fluxes}) observed reaction fluxes for all included (1 to n_{KOs}) perturbed knockout data sets. The weighted differences were also multiplied by the coefficient of variation (CV) for each reaction with available experimental measurements from the literature data in the reference strain to favor fitting reaction fluxes with more consistent experimental values. Smaller fitness values indicate kinetic parameter sets that better predict the data. When screening the parameter sets, a fitness threshold was used to determine which parameter sets are removed from the ensemble after each perturbation test. For the large model a fitness threshold of 0.1 was selected after testing threshold values ranging from 0.05 to 0.3 on a subset of 1000 parameter sets. The 0.1 threshold was the smallest threshold tested that was not so stringent that no parameter set passed all screening tests (i.e. an empty ensemble). The 0.1 threshold value determined using the smaller initial 1000 parameter set ensemble was then used to screen the larger ensemble (10000 parameter sets). The threshold value selected is specific to each EM application and data set.

4.2.5 Conservation Analysis

We adopted the conservation analysis protocol developed by Vallabhajosyula et al to reduce the system stiffness of our metabolic networks.[93] Eq. (4.2) is rewritten to separate the metabolite and enzyme fraction species into vectors of independent (x_I) and dependent (x_D) species:

$$\frac{dx}{dt} = \begin{bmatrix} \frac{dx_I}{dt} \\ \frac{dx_D}{dt} \end{bmatrix} = \begin{bmatrix} S_{I,N} \\ S_{D,N} \end{bmatrix} v(x,t) \tag{4.5}$$

where $S_{I,N}$ is the reduced, full rank stoichiometric matrix with I independent species rows and N elementary reaction columns. $S_{D,N}$ is the dependent species stoichiometric matrix of dependent species with D dependent species rows and N columns of elementary reactions. The D x I matrix ($L_{D,I}$) relates the reduced and dependent stoichiometric matrices:

$$S_{D,N} = L_{D,I} S_{I,N} \tag{4.6}$$

Following the derivation by Vallabhajosyula et al., the $L_{D,I}$ matrix allowed us to calculate the conserved moieties vector T from the initial metabolite concentrations and enzyme fractions[93]:

$$T = x_D(0) - L_{D,I} x_I(0) \tag{4.7}$$

Using COPASI, an existing software toolset, the reduced stoichiometric matrix ($S_{I,N}$) of our metabolic network was calculated.[94] COPASI also generated the $L_{M,I}$ and $L_{D,I}$ matrices needed to calculate the reduced Jacobian as well as back calculate the conserved moieties vector and the dependent metabolite concentrations. The MATLAB SimBiology toolbox can also calculate these matrices. Scripts to perform this in MATLAB are provided in the supplemental code.

The system of differential equations for only the independent metabolite and enzyme fraction species was then integrated using this reduced network format:

$$\frac{dx_I}{dt} = S_{I,N} v(x_I, x_D, t) \tag{4.8}$$

where the dependent metabolite and enzyme fraction species concentrations are a function of the independent concentrations and the T and $L_{D,I}$ matrices:

$$x_D = f\left(x_I, T, L_{D,I}\right) \tag{4.9}$$

While solving the system of equations defined in Eq. (4.8) for the evolution of the independent concentrations over time, the dependent concentrations were also calculated using Eq. (4.7). The dependent concentrations were calculated at each time step in order to solve for the elementary reaction fluxes v needed in Eq. (4.8).

4.2.6 Local Stability Check

Kinetic parameter sets generated using traditional ensemble modeling sampling were further screened for local stability at the wild type steady state equilibrium point. To check for the local stability of each kinetic parameter set, k, the eigenvalues of the reduced Jacobian (J_R) at the initial metabolite concentration ($x_I(0)$) were calculated:

$$\lambda = eig(\, J_R(x_I(0), k)\,) \tag{4.10}$$

If the real parts of the eigenvalues, λ_{Re}, were negative, the parameter set, k, was marked as locally stable. If one or more λ_{Re} were positive, the parameter set was marked as locally unstable and was rejected. The maximum λ_{Re} represents the slowest timescale of the system.[95] To select for the most stable parameter sets and avoid computational error with eigenvalue calculations, λ_{Re} values less than or equal to -0.001 were treated as negative and λ_{Re} greater than -0.001 were treated as positive. The MATLAB eig function used to calculate the eigenvalues is subject to numerical round off error so -0.001 was used as the zero-threshold value to avoid carrying over false positive parameter sets. When selecting for 100 locally stable parameter sets for the large model, less than 0.2% of the parameter sets thrown out had a maximum λ_{Re} between -0.001 and 0.

4.2.7 Knockout Order Validation

The knockout order test was performed on the large model using the seven available knockout data sets (Δpgi, ΔpykA, ΔpykF, ΔppsA, Δzwf, Δgnd, and Δrpe).[86] To rank the knockouts by

screening power, the cosine similarity cos_{sim} was first calculated between the WT and knockout fluxomic data:

$$cos_{sim}\left(v_{WT,j}^{ref}, v_{k,j}^{ref}\right) = \frac{\sum_{j=1}^{n_{fluxes}} v_{WT,j}^{ref} v_{k,j}^{ref}}{\sqrt{\sum_{j=1}^{n_{fluxes}} \left(v_{WT,j}^{ref}\right)^2} \sqrt{\sum_{j=1}^{n_{fluxes}} \left(v_{k,j}^{ref}\right)^2}} \tag{4.11}$$

The knockouts were ranked from lowest similarity to highest similarity, and the knockout with the lowest cosine similarity was selected as the first for screening. The remaining knockouts were sampled exhaustively in order of which one would result in the smallest average cluster similarity score when added to the group of previously selected knockouts. The distance from the selected cluster to a given knockout was calculated as the average cosine similarity value between the knockout in question and each knockout previously added to the cluster. For example, our first pick would be the data set whose metabolic state is most different from the WT state. Then, our second pick would be the data set whose metabolic state is most different from the cluster combining the WT's and the first pick's metabolic states. We continue to sample the remaining data sets from our available pool in this way until all are selected.

After sampling 10000 kinetic parameter sets using the local stability test described above, the sets were screened against one knockout at a time in order of selection. The fitness value for the new predicted perturbed steady state after each knockout was scored using Eq. (4.4). Parameter sets with a fitness value falling below the threshold of 0.1 were discarded. The rationale for selecting 0.1 as the fitness threshold is described in the "Screening Kinetic Parameter Sets" section above. The same initial 10000 sets were then screened in the reverse selection order. The fitness of the remaining ensemble for all seven knockouts was calculated after each screen by substituting the average predicted flux values for a given knockout, k, across all the parameter sets remaining in the ensemble ($\bar{v}_{k,j}^{pred}$) into our original fitness equation:

$$y = \frac{1}{n_{KOs}} \sum_{k=1}^{n_{KOs}} \frac{1}{n_{fluxes}} \sum_{j=1}^{n_{fluxes}} \frac{1}{CV} \left| \frac{\bar{v}_{k,j}^{pred} - v_{k,j}^{ref}}{v_{k,glucose\ uptake}^{ref}} \right| \quad (4.12)$$

The CPU time and the number of kinetic parameter sets remaining were recorded after each screening step.

During screening for the knockout order validation test, an event function was used with the MATLAB ode15s solver. If the ODE calculation took longer than 80 seconds (i.e. the parameter set was not able to converge to the new steady state in 80 seconds) the ODE was stopped and the parameter set did not pass the screen as a new steady state was not reached. To select the time cut off, 100 parameter sets were sampled for the large model and screened against all 7 knockouts in triplicate. Then the time required to include the average calculation times of 99% of all ODE calculations was selected as the cutoff. The lowest fitness scores from the time trial correlated with the fastest total ODE solve times for a given parameter set; therefore, stopping the ODEs after 80 seconds did not eliminate valid candidate parameter sets. Additionally, the ODE calculation was stopped if the mass balance error tolerance fell below 0.0001. This prevents the need to complete the remainder of the integration if the concentration profile has already reached a reasonable, new mass-balanced steady state. The event function code is included in the provided MATLAB scripts.[85] The logic for our event function was derived from previous EM work but a different time cut-off and error tolerance was selected.[39]

4.2.8 Computational Resources

All tests were performed using MATLAB (MATLAB and SimBiology Toolbox 2015/2016a, The MathWorks, Inc., Natick Massachusetts, United States) on the Northwestern Quest High Performance Computing Cluster. Ensemble modeling code was adapted from MATLAB scripts provided by the Liao

and Maranas groups.[34,36,39] The parameter screening step was coded to run in parallel and was run across 12 nodes on the Quest Computing Cluster. Sample MATLAB code is available for download.[85]

4.3 Results

4.3.1 Conservation analysis reduces parameter screening time

We used conservation analysis to identify and remove dependent metabolite and enzyme fraction species from our three defined metabolic networks. The number of dependent species identified for each model is detailed in Table 4.1. Furthermore, as a proof of concept we also applied the conservation analysis method to the recently published genome-scale kinetic model of *E. coli*, k-ecoli457 (Table 4.1).[54] As model size increased, we identified larger numbers of dependent species present in the network.

Table 4.1: Breakdown of independent and dependent metabolite and enzyme fraction species identified after conservation analysis.

Model	Net Reactions	Elementary Reactions	Species of Metabolites and Enzyme Fractions	Independent Species of Metabolite and Enzyme Fractions	Dependent Species of Metabolite and Enzyme Fractions
Small	6	26	16	12	4
Medium	34	268	164	130	34
Large	138	1474	828	687	141
k-ecoli457	455	5209	2987	2529	458

Integrating ODEs including implicitly dependent variables is known to create very stiff integration conditions. We hypothesized removing the dependent variables may remove stiffness, and decrease integration time. To investigate the computational speed improvement provided after using conservation analysis to reduce the large metabolic network to only independent species, 100 kinetic parameter sets were sampled using traditional ensemble modeling methods. For the large model, each of the parameter sets was perturbed seven times by individually knocking out the seven enzymes tested when developing the core *E. coli* kinetic model (Δpgi, ΔpykA, ΔpykF, ΔppsA, Δzwf, Δgnd, and Δrpe).[86] The ODE calculations were performed using the original formulation where all species concentrations were tracked. Then the same test was performed using the conservation analysis formulation where only the independent species were tracked. In total 700 calculations were performed for each ODE formulation in triplicate, and the total CPU time was recorded. As Fig. 4.3 illustrates, conservation analysis halved the total CPU time for our ODE time test. We ran similar time trial tests on the medium and small models and observed this beneficial decrease in solve time scales with model size (Fig 4.4). This trend implies this method of model reduction may provide even greater benefit as we move toward generating kinetic models for networks approaching the genome scale. To investigate this claim, we performed a small ODE time trial test on the genome-scale model to observe the effect of the conservation analysis method on the ODE integration speed. We sampled 100 kinetic parameter sets for the corrected k-ecoli457 genome-scale model using the traditional sampling method. We then perturbed six of the enzymes used to screen the large model by over and under expressing the enzyme values by different fractions (pgi: 0.95, pykF: 1.2, ppsA: 1.3, zwf: 0.9, rpe: 0.8, gnd: 1.1 and 0.95). We integrated each of these cases over twenty time steps for a total of 700 calculations using the base ODE framework as well as the conservation analysis framework. We observed a 96% improvement in solve time reducing the total CPU time from 382 ± 49 hours down to 15 ± 1.8 hours after running this comparison in triplicate. These perturbations lead to steady states

close to the wildtype conditions, but this test provides a strong first demonstration of the positive benefits conservation analysis would provide in training and screening genome-scale models.

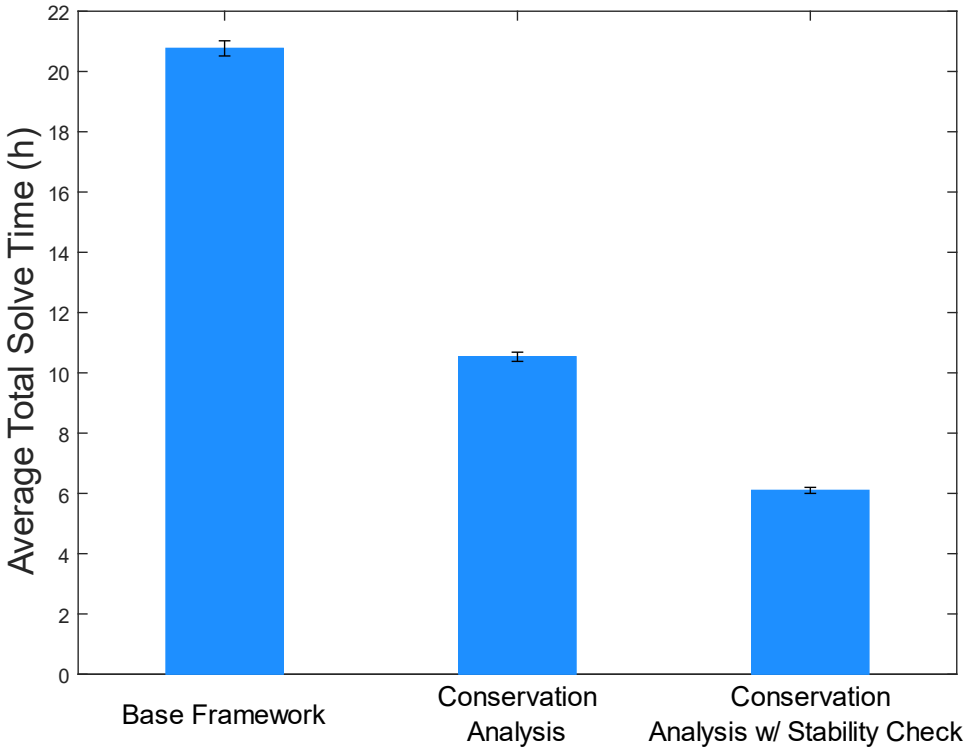

Figure 4.3: Conservation Analysis and Local Stability Check Improve Solve Time. The total CPU time for integration of 700 ODE calculations (100 kinetic parameter sets for large model perturbed for seven unique enzyme knockouts) using different ensemble modeling frameworks: base framework, reduced network after conversation analysis, and reduced network after conservation analysis with local stability check. Solve times compared using student t-test indicated $p<0.00001$ between all condition pairs. Error bars are standard deviation (n=3).

4.3.2 Local stability check selects for biologically representative parameter sets

After sampling each individual kinetic parameter set, the reduced Jacobian was calculated for the parameter set at the initial independent metabolite concentrations. As shown in Table 4.2, when parameter sets are sampled for the small model, all of them pass the local stability test. However, as

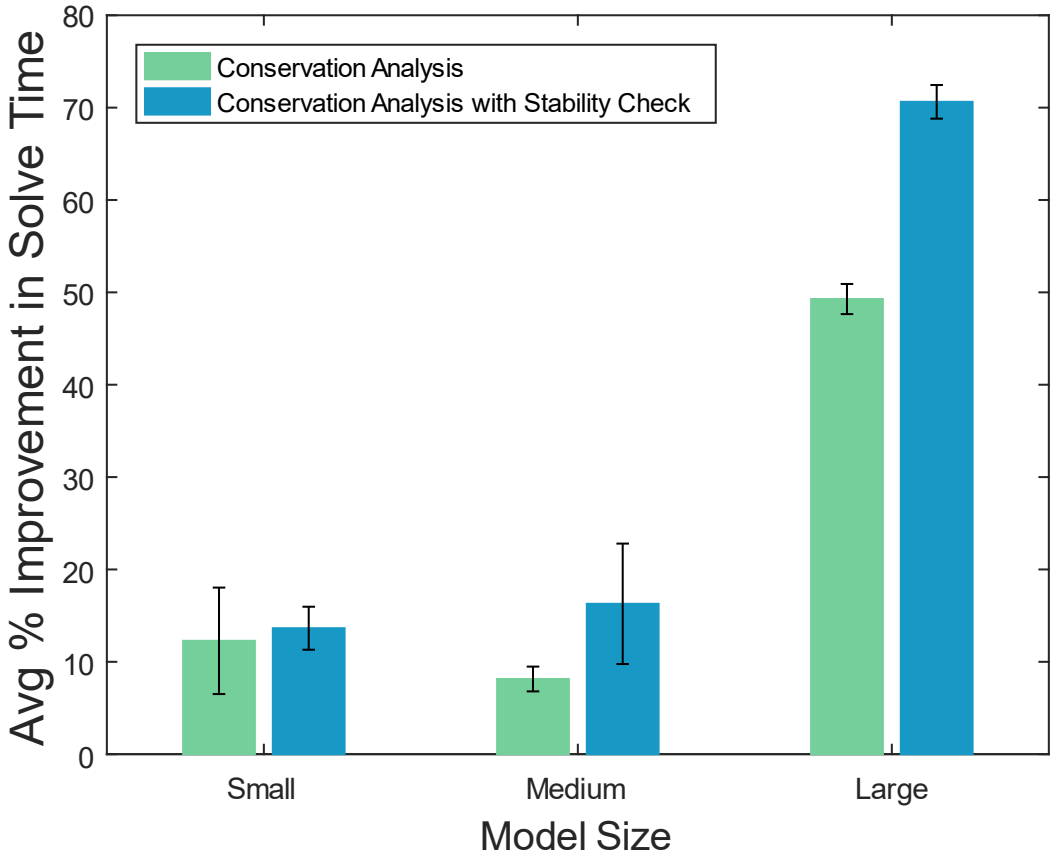

Figure 4.4: Solve Time Improvements Conferred by Conservation Analysis and Local Stability Check Scale with Model Size. The percent improvement in solve time from the base EM framework for the small, medium, and large models after implementing conservation analysis and the local stability check. Small model: 100 kinetic parameter sets screened against one knockout. Medium model: 100 kinetic parameter sets screened against seven knockouts. Large model: 100 kinetic parameter sets screened against seven knockouts. Student's t-test performed on average solve times across three trials. Average solve times between implementing conservation analysis treatment and implementing the conservation analysis and stability test for the medium model ($p<0.05$) and the large model ($p<0.00001$) were significant. The small model did not see significant improvement when adding the stability test. Error bars are standard deviations using propagation of error for percentage change calculations (n=3).

we move up to the medium model, less than half of the sampled parameter sets are locally stable. This trend continues as we sample parameter sets for the large model where less than one percent of the parameter sets are locally stable at the WT steady state equilibrium point. To the best of our knowledge, other ensemble modeling efforts would likely have similar rates of instability in their models.

Table 4.2: Number of total parameter sets sampled to acquire 100 locally stable kinetic parameter sets.

Model	No. Parameter Sets Sampled	% Locally Stable
Small	100	100%
Medium	217	46%
Large	13953	<1%

Eigenvalues with negative real parts for the reduced Jacobian imply that small perturbations from the equilibrium point will return to the equilibrium point (local stability). Alternatively, parameters sets leading to a Jacobian with one or more eigenvalues with positive real parts point to local instability.[95] We observe an increase in ODE integration speed when selecting for locally stable kinetic parameter sets over locally unstable kinetic parameter sets after a perturbation to the initial condition. As a result, we see an additional decrease in compute time when solving the 700 ODE calculations (7 conditions) during the time trial test when we select 100 locally stable kinetic parameter sets (Fig. 4.3). This increase in speed also trends with model size (Fig. 4.4) with the exception of the small model. As noted in Table 4.2, all the parameter sets sampled for the small model are locally stable, so no additional benefit is seen in solve time when adding the stability check. Instead, in some runs a small decrease is shown compared to the conservation analysis step alone because of the time needed to check for the local stability of each parameter set, although none were rejected. The time to perform the local stability calculation is negligible for the large and medium models compared to the improvements in overall solve time. A Student t-test (n=3) was performed to compare the total CPU run times for each model size (small, medium, and large) at each treatment (original ODE formulation, reduced network with conservation analysis, and reduced network with conservation analysis and local stability check). Through these tests we confirmed the run time

improvement after adding the local stability check to the small model is not significant whereas it does provide significant run time improvement for the other model sizes (Fig. 4.3 & Fig. 4.4).

4.3.3 PRE-RANKING SCREENING DATA IMPROVES SOLVE TIME

We ordered the experimental data we would compare to our models based on the cosine similarity algorithm. The final screening order of the available perturbed steady state data after selection in order of decreasing average similarity was Δpgi, Δgnd, Δzwf, ΔpykF, Δrpe, ΔppsA, and ΔpykA. As shown in Fig 4.5a, when screening kinetic parameter sets for the large model, the Δpgi knockout screen provides the majority of the predictive power to our final ensemble, leading to the largest drop in the fitness score of our overall ensemble. This result is confirmed when we feed the knockout data sets in the reverse selection order and do not see a significant improvement in overall solution fitness until the Δpgi knockout is included.

Helpfully, more parameter sets are removed early in the process when knockout data sets are fed in order of screening rank (Fig 4.5b). This means fewer ODE calculations are performed during the remaining screening steps resulting in a significant decrease in overall solve time (Fig 4.5c). Specifically, for the large model, the screening time improves 48% when the knockout data sets are fed in rank order instead of reverse rank order. These solution times are reported in total CPU hours.

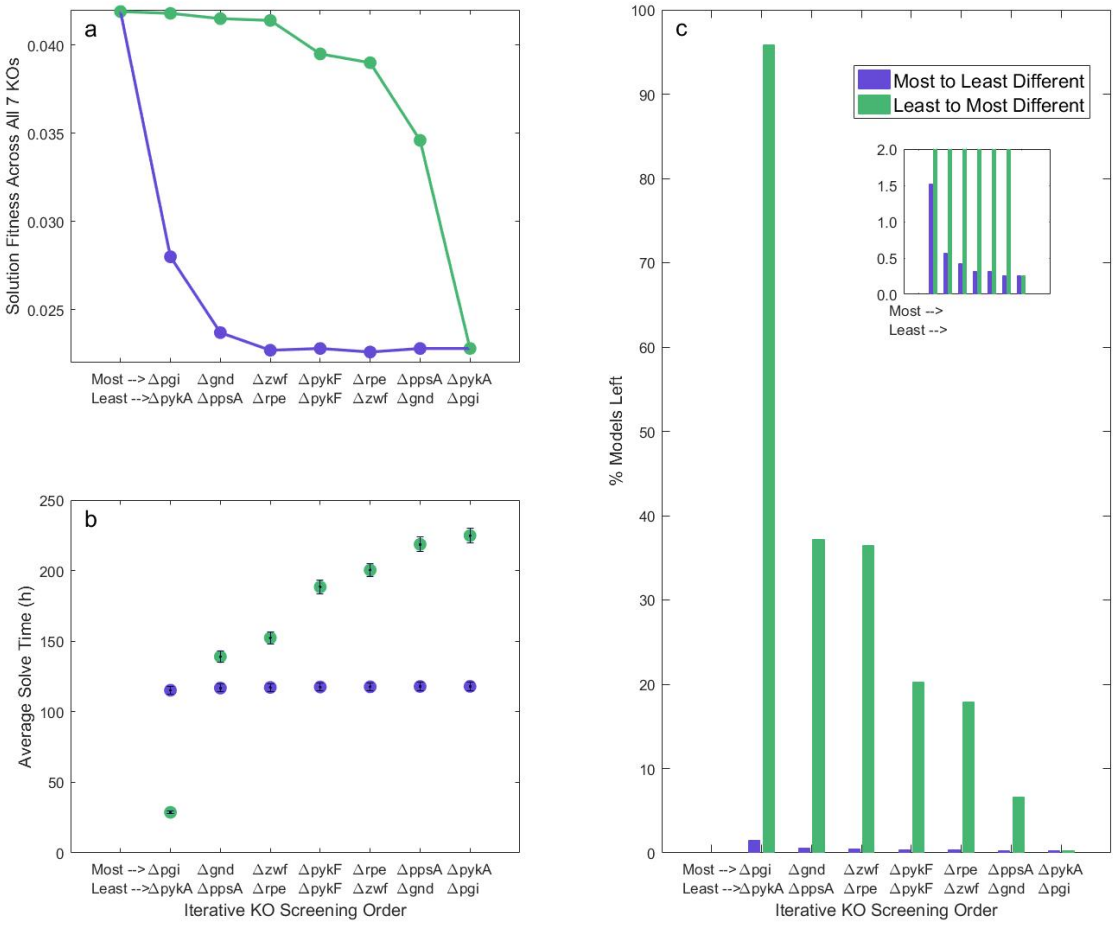

Figure 4.5: Screening KOs in Rank Order Eliminates Unpredictive Models Earlier. A) The solution fitness averaged for large model across predictions for all seven knockouts of the initial ensemble of 10000 locally stable models is 0.042. Moving across the figure from left to right, the initial ensemble is screened against knockouts one at a time ranked from most to least different (blue) and least to most different (orange) flux distribution determined by clustering. The average solution fitness for all seven knockouts is plotted after each additional knockout screen. B) Total additive solve time after each screening step is recorded in total CPU hours. When the knockouts are fed from least to most different from the WT for the large model, the total time to screen all 7 knockouts is 118 total CPU hours. The time to screen all 7 knockouts when fed from most to least different is 225 total CPU hours. The simulations were run in parallel across 12 nodes. Error bars are standard deviation (n=3). C) Regardless of knockout screening order, the initial ensemble of 10000 kinetic parameter sets for the large model is reduced to 26 kinetic parameter sets. However, when knockouts are fed from highest screening power to lowest (blue) the ensemble size is reduced by >98% to 152 kinetic parameter sets after the first screening step. Alternatively, when the knockouts are fed from lowest to highest screening power (orange) the ensemble size is only reduced 4% to 9580 parameter sets. Inset plot zooms in on bottom 2% of remaining models.

4.4 Discussion and Conclusion

4.4.1 Reducing model screening time

The traditional elementary stoichiometric matrix $S_{m,n}$ describing the reactions in a metabolic network is often not full rank due to the presence of conserved metabolite relationships. For example, the pool of cofactors like the adenine moieties (ATP, ADP, and AMP) maintains a constant total level in the metabolic model. For every ATP consumed, a stoichiometric amount of ADP or AMP is produced and vice versa. Therefore, the concentrations of the adenine moieties are linearly dependent. Across large metabolic reaction networks, several of these conserved relationships are present, and some are nonobvious. The presence of these conserved groups in turn leads the Jacobian of the system to be singular. When we solve the system of differential equations to determine the new steady state of the network after a perturbation, the singularity of the Jacobian makes it difficult to integrate the stiff system, and the solve times are computationally limiting.

To generate a non-singular Jacobian, we used the conservation analysis method described by Sauro et al. to calculate a full rank stoichiometric matrix.[93,96] Conservation analysis identifies the conserved metabolite and enzyme complex species and generates a reduced stoichiometric matrix excluding all dependent metabolite entries. We were then able to calculate the non-singular reduced Jacobians for our reduced metabolic networks which resulted in faster ODE calculations. Furthermore, despite only solving the concentration time profiles for independent metabolite and enzyme fraction species, we do not observe a loss in predictive power of the final overall steady state metabolite profiles.

Conservation analysis was previously developed for use in large biochemical networks.[93,96] However, its effects on ODE integration time in EM has not yet been fully characterized. As we have shown, conservation analysis provides significant computational speed improvements to the current

EM process without a loss in solution accuracy. It requires minimal effort to implement and can be plugged in to benefit many concurrent EM applications. Additionally, by applying conservation analysis and calculating the reduced Jacobian, we are able to employ the local stability check discussed below.

4.4.2 IMPROVING LOCAL STABILITY

Wild-type cells at steady state biologically maintain a robust metabolic concentration profile, meaning they can quickly recover from small, local perturbations to their environment.[86,97] Similarly, the kinetic parameter sets we are generating to define cellular metabolism should be robust to small deviations from the initial WT metabolite concentrations. Mathematically, the WT metabolite concentration profile of a cell during steady state growth should be a stable fixed point. Therefore, we want the steady state defined by our WT steady state conditions and sampled kinetic parameter sets to be locally stable. Specifically, for a given kinetic parameter set, we expect the ODE integration solution to converge to the WT steady state after a small perturbation to the initial condition. Consequently, by employing the local stability check, we are further minimizing the available parameter search space. To ensure we are generating locally stable kinetic parameter sets, we further characterized a local stability test previously implemented by the Liao group as the first step in their robustness analysis work.[38,42,84] In this analysis we wanted to specifically characterize the local stability screen across a range of network sizes. Using the reduced metabolic network after conservation analysis, we calculated the eigenvalues of the reduced Jacobian to determine the stability of the WT stationary state using fixed point stability theory.

For larger models, we have shown most parameter sets sampled are not able to return to their steady state metabolite concentrations after small perturbations to their initial conditions. By selecting for local stability, we are generating more biologically-reflective kinetic parameter sets for

screening. We have demonstrated for smaller models, most parameter sets sampled traditionally are locally stable at the reference steady state. Therefore, the unstable behavior around the stationary point we observe for the larger model likely did not emerge in previous ensemble modeling efforts where smaller models were used to address metabolic engineering challenges.[33,34,36,38] This work emphasizes that in addition to the robustness analysis of pathways for which the test was first implemented by the Liao group[38,42,84], the local stability test should be incorporated during the parameters sampling step in all ensemble modeling efforts to reduce the screening time and parameter sampling space. The local stability test is an imperfect screen in that some of the parameter sets that pass may still exhibit unstable behavior near the stationary state. However, the local stability test does help us identify more-stable parameter set candidates. Overall we believe our observations on instability in larger metabolic models makes the case for continued study in this area of kinetic modeling. Specifically, as we move toward larger kinetic model structures, understanding model stability and screening for it become more crucial to generating reliable, predictive tools. Fortunately, like with the conservation analysis step, the local stability test is easy to plug into the existing EM framework and can be used in parallel with other EM efforts with great benefit.

4.4.3 Evaluating strength of screening data sets

Previous work demonstrates the importance of having screening data available with broad spatial coverage of the metabolic network.[39,44] Some experimental states provide more screening value than others which has been demonstrated in a previous leave-one-out analysis.[39] For example, a knockout near a central pathway branch would provide more resolution than a knockout down a peripheral linear pathway.[44] However, visual inspection is not enough to identify the best knockout phenotypes to test when developing a strong screening data set. In some cases, a set of perturbation data may be available and being able to assess its screening strength prior to running any EM

calculations is valuable. We have found that simply comparing the difference between the WT and perturbed metabolic states provides a coarse rank of screening power. Specifically, perturbations leading to a new steady state similar to the WT flux distribution do not provide additional fitness resolution. On the other hand, perturbations leading to large deviations from the WT flux profile provide more insight. Additionally, perturbations covering more disparate portions of the metabolism provide better screening resolution than several perturbations clustering around the same area.

Our findings emphasize the importance of selecting a strong screening data set when using EM. When designing experiments to capture kinetic information, perturbations can be selected that lead to the largest variations using a similar ensemble modeling approach.[44] However, when using EM to resolve existing perturbation data, it is important to evaluate the best screening order of the data to ensure the most incorrect parameter sets are removed first. This allows our screening step to converge to the solution much faster by reducing the number of ODE calculations. We hypothesized and our results confirmed that knockouts leading to metabolic states furthest from the WT state provide the most screening power. To determine the order of the knockouts encompassing the largest spread in metabolic states, we derived a sampling method to ensure the knockouts picked formed the most varied set of states possible. For our clustering calculations, we chose cosine similarity as our distance metric and minimum average distance (low cosine similarity equates to bigger differences) as our linkage method to determine our knockout order as these methods work well with sparse, high-dimensional data sets.[98,99] Regardless of method, the Δpgi knockout, which overwhelmingly provides the most screening resolution, was always selected first. Our results confirm that selecting perturbations in this biased way provides a significant improvement to overall run time and does not affect the overall fitness result. In this work, we investigated enzyme knockout perturbation sets to maintain consistency with the original data set used to generate the published

core *E. coli* kinetic model.[39,86] However this method of ranking datasets collected under consistent conditions should hold for other perturbations commonly used in metabolic engineering including enzyme overexpression and knockdown cases.

Additionally, we chose to employ a specific fitness value to consistently measure how well the remaining ensemble described the experimental observations across our various simulations. For example, screening can be performed based on phenotypic response alone where the ability to predict the relative increase or decrease of a product of interest is the only requirement for keeping a parameter set after a given perturbation.[41] Selecting looser screening criteria decreases the likelihood of a null set and does not negate the positive benefits of presorting the perturbation data. However, we would also like to note that the observance of a null set (i.e. no parameter sets can predict a specific perturbation) may also indicate inconsistencies or errors in the assumed model structure. This can be a useful result when the exact network structure is unknown. Previous work shows generating ensemble predictions for various defined network structures can help elucidate unknown regulation relationships and pathway split ratios specifically by identifying cases where a given model structure is better able to resolve observed phenotypes.[40,43,81] Other EM efforts incorporate different screening paradigms when training their sampled ensemble modeling parameter sets. For example the Maranas group regularly employs a genetic algorithm framework to identify a single, best-case parameter set which can be beneficial over the traditional ensemble solutions for certain applications.[39,44,54] These alternative screening methods would still benefit from employing the conservation analysis, local stability test, and perturbation data ranking acceleration strategies described in this work, specifically by speeding up their integration calculations, minimizing their parameter solution space during sampling, and achieving faster convergence to the best solution faster by fitting against the perturbations that are hardest to predict first.

Lastly, we would like to note the sampled kinetic parameter values are heavily influenced by the chosen reference WT flux distribution because they are selected to reach the provided WT steady state over time. Previous work demonstrates performing EM on different WT flux distributions can even help resolve unknown branch point flux distributions.[40] Unfortunately, in most flux data sets, there will be reaction fluxes that can still not be resolved, yet a value for that flux must be imposed to follow the EM algorithm. For unresolvable reaction fluxes in the reference flux distribution, i.e. reactions where an infinite number of flux values can satisfy the mass balance constraints, the kinetic parameters are fit to a flux that might not accurately reflect true metabolic behavior. While we did not specifically investigate this possible limitation in this work, the EM method could be altered in future efforts to avoid fitting kinetic parameters to reactions where we do not have high confidence in their flux values.

Chapter 5: Kinetic Ensemble Model of Gas Fermenting *Clostridia Ethanogenum* Core Metabolism for Improved Ethanol Production

5.1 Introduction

Acetogenic bacteria can fix one-carbon substrates (i.e. CO and CO_2) and convert them to acetyl-CoA through the Acetyl-CoA Wood/Ljungdahl pathway.[100–102] One such acetogen, *Clostridium autoethanogenum*, can autotrophically produce acetate, ethanol, and 2,3-butanediol through consumption of CO, CO_2, and H_2 gases.[103,104] With these capabilities, *C. autoethanogenum* is a strong candidate to enable biofuel production from syngas, a natural waste product of biomass and industrial gasification processes rich in CO, CO_2, and H_2.[105–107] *C. autoethanogenum* was first isolated from rabbit feces and reported to produce ethanol and acetate from carbon monoxide in 1994 and has since been the subject of several studies aiming to optimize its ability to produce ethanol over acetate under industrially-relevant conditions.[103,108–111] However complete understanding of forces governing the ethanol to acetate production ratio is absent. Unlike more bio-industrially prevalent strains (i.e. *Saccharomyces cerevisiae* and *Escherichia coli*), the genetic engineering tools needed to interrogate the clostridia metabolic pathways are not fully developed or easily implemented.[105,112,113] While rapid advancement of these tools is underway, in many cases the methods developed are strain-dependent and not easily transferable across different clostridia.[114–116] This lack of easily accessible engineering tools prohibits in-depth studies of metabolic pathways because genetic modification (i.e. enzyme knockout, knockdown, or overexpression) is expensive, time intensive, and often not executable especially when attempting combinatorial changes.[117] The potential of *C. autoethanogenum* as a waste-gas fermenting biofuel platform coupled with these genetic engineering limitations highlights a pressing need to develop robust, predictive computational models of its metabolism. As the number of genetic changes that are able to be implemented to this species is

currently limiting, *in silico* analysis will help confidently elucidate the most informative metabolic network features to interrogate experimentally prior to exhausting resources in the lab.[113,118]

Previously, a genome-scale metabolic model (GEM) for *C. autoethanogenum* (iCLAU786) was developed and refined to resolve several unknown aspects of the organism's metabolism including cofactor-dependencies of certain reactions and methods for improving ATP production of native pathways.[110,111,119] However, steady state constraint-based modeling cannot predict some regulatory behavior, rate-limiting steps, or changes in metabolite pool concentrations. To bridge this gap, the development of kinetic models is crucial to fully understanding and predicting metabolic behavior. Kinetic models also build upon the predictive engineering power of traditional steady-state modeling methods as they can predict phenotypic responses to enzymatic expression changes. Typically, constraint-based models can only predict binary on or off enzymatic states. Additionally, previous studies imply thermodynamics play a major role in governing product distribution in clostridia.[120,121] Therefore, a more detailed free energy analysis of major enzymatic reactions for incorporation into a kinetic model is an increasingly valuable next step.

Building representative kinetic models with coverage of all high flux metabolic pathways is challenging as reliable kinetic parameter estimates are not available for most enzymatic reactions, especially in the less characterized clostridia strains. When they are available, they are typically defined under *in vitro* conditions where local concentrations, temperature, and pH dependencies present under *in vivo* conditions are not captured. Furthermore, even in cases where parameters are available for reactions, the varying conditions under which they are typically characterized hinders combining them into a single model that can accurately predict cellular behavior under typical growth conditions. The ensemble modeling framework was previously developed to circumvent these traditional kinetic modeling issues by sampling multiple kinetic parameter sets constrained by

a single, experimental reference state.[33,34,36] This initial parameter set ensemble is then screened against additional perturbation datasets typically generated in metabolic engineering studies to prune down the ensemble iteratively. The ensemble modeling framework does not require knowledge of internal metabolite concentrations or enzyme levels. The combined predictions of the final ensemble provide a confidence landscape for how tightly we can resolve various aspects of cellular metabolism. Ensemble modeling efforts have been successfully used to improve strain designs by predicting productive enzyme knockout or overexpression targets and estimating the robustness of competing pathway designs.[38–43,48,54,92]

In this work, we generate a novel kinetic ensemble representation of *C. autoethanogenum* metabolism that accurately predicts the intracellular concentrations of key metabolites and reproduces experimentally observed trends not used to train the model. Typically, ensemble modeling efforts use enzyme perturbation data to prune down the initial ensemble of sampled kinetic parameters.[33,34,37,39,41,54] However as discussed above, few studies demonstrating phenotypic outcomes from genetic modifications to *C. autoethanogenum* are available. Valgepea et al. have demonstrated that the ratio of acetate to ethanol produced by wild type (WT) *C.* autoethanogenum decreases with increasing biomass concentration but has an inherent lower limit of one.[111] Maximizing ethanol production over acetate is a critical design challenge for enabling this bioproduction platform. Therefore, in this study, we demonstrate a novel perturbation strategy in which we use the Valgepea et al. dataset, where substrate uptake is modified by increasing biomass concentration over three conditions, to screen our parameter sets.[111] The traditional ensemble modeling framework assumes the concentration of substrates outside the cell and predicts the flux distribution after a change in enzyme expression. In the Valgepea et al. dataset, low (0.5 gDCW/L), medium (1.1 gDCW/L), and high (1.3 gDCW/L) biomass concentration states were achieved by altering the gas-liquid mass transfer rates and therefore the extracellular concentration of gases in

the cultures.[111] However, dissolved CO, CO_2, and H_2 concentrations are difficult to measure in a bioreactor so instead the uptake flux was monitored by analyzing the composition of the feed gas and off gas streams into and out of the reactor. As the gas uptake fluxes are known, we wanted to constrain our models to these observed values and predict resulting changes to the remaining metabolic pathway fluxes. This type of non-genetic perturbation has not been previously demonstrated with the current ensemble modeling framework. So, to accommodate this available dataset, we redefine the governing equations of the method as a set of differential-algebraic equations (DAE). Then, rather than using genetic perturbation data, we train our model by screening against changes in gas uptake fluxes. Most powerfully, our final kinetic ensemble model predicts new metabolic engineering strategies to increase production of ethanol. This work is an initial effort to understand the kinetic and thermodynamic limitations of *C. autoethanogenum* metabolism and makes a strong case for the value of further analysis and development of these methods.

5.2. Methods

5.2.1 Metabolic Network

A model representing the core metabolism of *C. autoethanogenum* was used in this study. The model accounts for 70 reactions and 62 metabolites (Fig. 5.1). The model was curated as a subset of the published genome scale model, iCLAU786, to include all reactions responsible for carrying greater than 97% of the flux at growth optimized conditions.[110,111] The leftover reactions accounting for the remaining flux were reduced and incorporated into the biomass equation, which was modified to reflect these changes. Four additional reactions were added to the model to reflect findings in the literature.[122,123] Specifically, NADPH-dependent analogs of acetaldehyde dehydrogenase (ACALDy), butanediol dehydrogenase (BTDDy), and alcohol dehydrogenase (ALCDy) were added. Similarly, an NADH-dependent electron-bifurcating [FeFe]-hydrogenase reaction (HYDFDNx) was added to

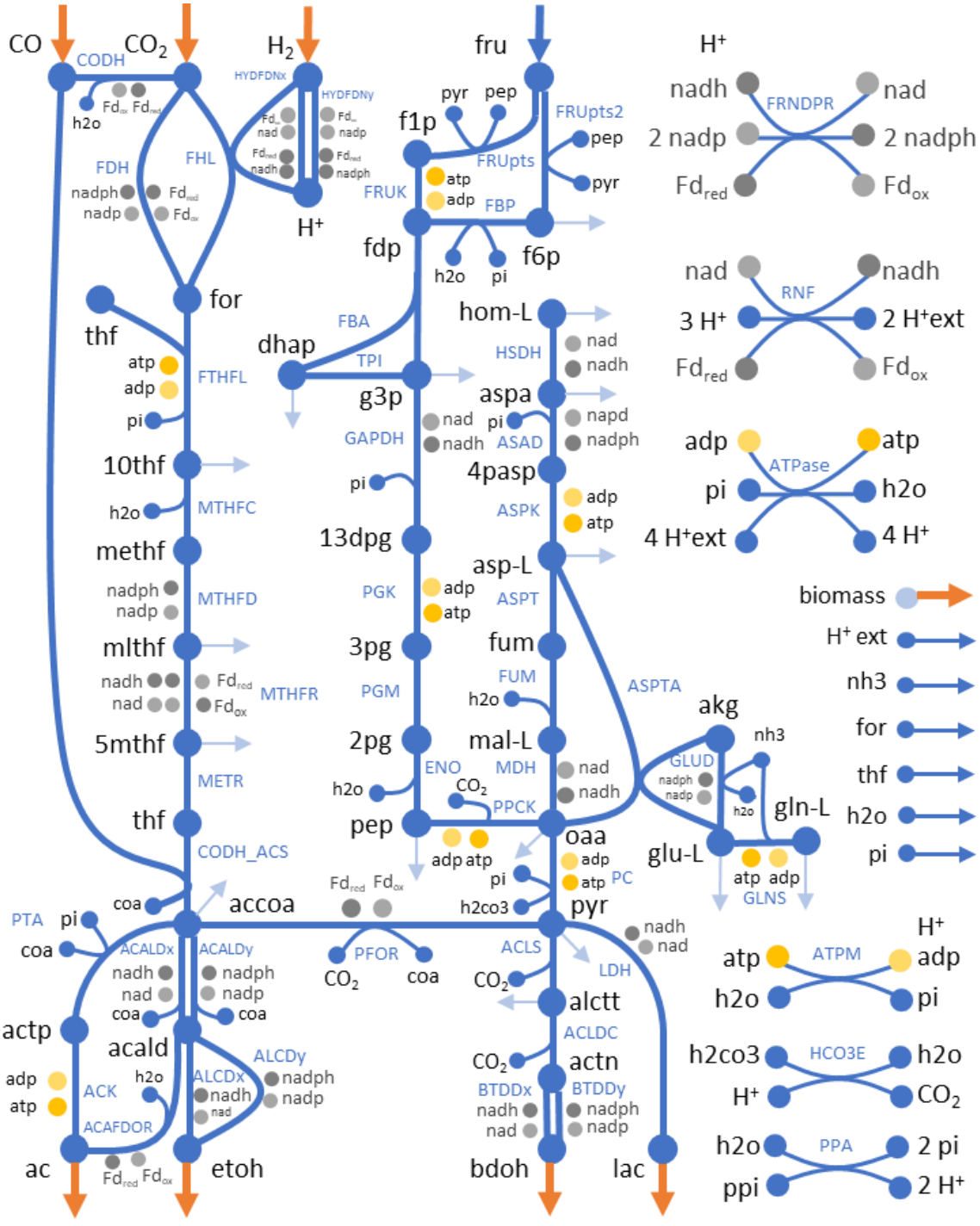

Figure 5.1: Metabolic Network of C. autoethanogenum Core Metabolism. Circles indicate metabolites. Cofactors and energy equivalents are colored in grey and yellow, respectively, to highlight their prevalence throughout the network. Lines indicate enzymatic reactions. The orange arrows indicate exchange fluxes in the screening dataset, and the light blue arrows indicate metabolites consumed in the biomass equation. Metabolite and reaction abbreviations provided in Supplemental Information Tables S1-S2.

supplement the predominantly active NADPH-dependent hydrogenase.[122] The final core model, corresponding iCLAU786 reactions, and added reactions are provided in the Supplemental Information file located in the Tyo Lab Box repository (Tables S1-S3).

5.2.2 Defining Reaction Thermodynamics

Per the previously published ensemble modeling method, Gibbs free energy ranges ($\Delta_r G'_{min}, \Delta_r G'_{max}$) were determined for each net metabolic reaction to constrain our kinetic parameter space within thermodynamically relevant bounds.[33,36] The Gibbs free energy ranges were calculated from the standard Gibbs free energy values ($\Delta_r G'^{\circ}$) and corrected to account for physiological metabolite activity ranges, pH differences from standard state, and energy requirements associated with ion transport for exchange reactions as previously described for assigning reversibility to genome-scale metabolic models[11,15]:

$$\Delta_r G'_{min} = \Delta_r G'^{\circ} + RT \sum_{i=1}^{Products} n_i \ln(x_{min}) + RT \sum_{i=1}^{Reactants} n_i \ln(x_{max}) + \Delta G_{transport} + \Delta G_{pH} - U_{r,est} \quad (5.1)$$

$$\Delta_r G'_{max} = \Delta_r G'^{\circ} + RT \sum_{i=1}^{Products} n_i \ln(x_{max}) + RT \sum_{i=1}^{Reactants} n_i \ln(x_{min}) + \Delta G_{transport} + \Delta G_{pH} + U_{r,est} \quad (5.2)$$

where n_i is the stoichiometric coefficient of metabolite i, R is the universal gas constant, and T is the temperature (298 K). The standard Gibbs free energy values, $\Delta_r G'^{\circ}$, and the calculated uncertainty in these values, $U_{r,est}$, were predominately provided by the online eQuilibrator, ModelSEED, and

MetaCyc tools which both use group contribution methods to estimate the free energy changes associated with enzymatic reactions.[50,52,124] For less common, organism-specific reactions not included in these repositories, free energies reported in the literature were used.[121,122] Corrections for ion transport ($\Delta G_{transport}$) and pH (ΔG_{pH}) changes were calculated as described previously.[15,125] The internal pH and external pH were assumed to be 6 and 5.3 for *C. autoethanogenum* to mirror experimental conditions.[111,123] The minimum and maximum metabolite activities (x_{min}, x_{max}) were set to 0.01 mM and 20 mM, respectively, for all non-gaseous metabolites. The minimum activities for dissolved gases (H_2, CO_2, and CO) were set to 0.00001 mM, and the maximum activities were set to the saturation constants at standard conditions.[11,15] Please see the supplemental files for the complete list of reaction free energy ranges and standard value references (Supplemental Information Table S4).

5.2.3 Determining Reference Flux Distribution

Valgepea et al. report gas uptake and production secretion rates for each biomass concentration state (low, medium, and high) used in this study.[111] The authors also performed flux balance analysis (FBA) using their experimental observations and the genome scale model iCLAU786 to determine carbon-balanced, network-wide flux distribution predictions for each state.[13,110,111] Their FBA method was adapted in this study to determine the flux distributions for each biomass concentration state for our smaller, core model. First, using the COBRA Toolbox in MATLAB, the lower and upper flux bounds for each reaction were set to reflect the reversibility of each reaction as determined by the Gibbs free energy ranges ($\Delta_r G'_{min}, \Delta_r G'_{max}$) calculated above.[64] Specifically, the minimum lower flux bound was set to 0 for reactions whose entire Gibbs free energy range is negative (e.g. the ATP synthase and RNF complex reactions). Next, the CO and H_2 uptake fluxes and the acetate, ethanol, lactate, butanediol, and biomass production fluxes were constrained to match the genome-scale flux values predicted by

Valgepea et al.[111] Flux through the hydrogenase reaction was also constrained to just the NADPH-dependent enzyme (HYDFDNy) to reflect previously published observations of its prominence in CO and H_2 feed gas fermentations.[123] However, flux was allowed through the NADH-dependent counterpart hydrogenase (HYDFDNx) during screening of other biomass concentration states. Despite setting these constraints, Flux Variability Analysis (FVA) reveals there are still multiple, valid flux distribution solutions remaining, so a distribution maximizing CO_2 and ATP production was selected to best match the Valgepea et al. analysis.[23,111]

5.2.4 Sampling Kinetic Parameter Sets

In this work, kinetic parameter sets were selected from a sample space constrained by steady state reaction fluxes and Gibbs free energy ranges. We use the ensemble modeling framework previously developed by the Liao group.[33,36] In our previous ensemble modeling efforts, we described several acceleration strategies for improving the traditional framework which we have incorporated into this work as well.[85]

First, each net reaction in our network was defined as a set of elementary steps defined by elementary kinetic rate laws. For example, a simple reaction where enzyme E converts a substrate, A, to a product, B, would consist of the following elementary steps:

$$A + E \underset{v_2}{\overset{v_1}{\rightleftarrows}} AE \underset{v_4}{\overset{v_3}{\rightleftarrows}} BE \underset{v_6}{\overset{v_5}{\rightleftarrows}} B + E \tag{5.3}$$

The change in each metabolite concentrations x over time t was defined by:

$$\frac{dx}{dt} = S_{m,n} v(x,t); \;\; x = x_o \; at \; t = 0 \tag{5.4}$$

where x is the 1 x M vector of all species, either metabolites or enzyme fractions, in our network and v is the 1 x N vector of elementary fluxes. The initial conditions x_o is a 1 x M vector of metabolite concentrations and enzyme fractions at the steady state reference condition. $S_{m,n}$ is the M x N elementary stoichiometric matrix which contains stoichiometric coefficients for each metabolite in each elementary reaction. The elementary flux through each reaction v_n was defined by an elementary kinetic rate law:

$$v_n = k_n \prod_{\substack{m \\ S_{m,n}<0}} x_m^{|S_{m,n}|} \tag{5.5}$$

Here k_n is the reaction specific elementary kinetic parameter, and x_m is the metabolite concentration or enzyme fraction of each reactant species in the elementary reaction. In this work, all elementary kinetic parameters are combined into a single 1 x N kinetic parameter set K. As our system is ill-posed, there are several kinetic parameter sets within the constraints imposed that accurately predict our reference state flux distribution. For this reason, numerous, unique kinetic parameter sets were sampled that all accurately describe the reference state.

Conservation analysis was applied to our reaction network to identify conserved metabolite and enzyme fraction species.[93] For our elementary network, 15% of the 458 metabolite and enzyme fraction species were identified as linearly dependent on the concentrations and fractions of the remaining species. By removing these conserved moieties, a reduced metabolic network was generated to represent the core metabolism of *C. autoethanogenum*. As shown previously, applying conservation analysis to elementary metabolic networks reduced computational speed and enabled local stability analysis of the reduced network.[85]

Kinetic parameter sets were screened for local stability at the reference state equilibrium point.[85] The eigenvalues (λ) of the reduced Jacobian (J_r) of our system at the initial independent metabolite concentrations and enzyme fractions ($x_{i,o}$) were calculated for each parameter value:

$$\lambda = eig\big(J_R\big(x_{i,o}, k\big) \big) \tag{5.6}$$

If the real parts of all eigenvalues (λ_{Re}) were negative, the parameter set was designated as locally stable and retained. Locally unstable parameter sets were discarded. To avoid computational error associated with the MATLAB eig function, values less than -1x10^{-7} were considered negative and values above -1x10^{-7} were considered positive. When selecting locally stable parameter sets using the high biomass concentration reference state, only 1.2% of parameter sets sampled passed the stability screen.

5.2.5 Differential Algebraic Equation (DAE) Implementation and Screening Kinetic Parameter Sets

To perturb each kinetic parameter set sampled at the reference state, typically enzyme knockouts or overexpressions are simulated by changing the initial conditions (x_o) to reflect the loss of an enzyme species in the system or an increase in enzyme levels, respectively.[40,41] Using these perturbed initial conditions, the system of ordinary differential equations defined in Eq. (5.4) is solved to calculate the new steady state for each parameter set. With this data set, no specific genetically engineered enzyme perturbations were tested.[111] Instead, the tested states are distinguished by changing the biomass concentration in the experiment, which results in changes in CO and H_2 gas as uptake flux. Therefore, the traditional EM framework could not be directly implemented, and a new approach was required.

To interrogate how well each kinetic parameter set predicted these changes in gas uptake rates, the system of ordinary differential equations describing the change in metabolite concentrations and enzyme fractions over time in Eq. (5.4) was restructured as a differential algebraic equation (DAE):

$$\frac{dx}{dt} = S_{m,n}v(x,t);\ \ x = x_o\ at\ t = 0$$
$$v(r_{uptake}) - U = 0 \tag{5.7}$$

where r_{uptake} is a vector containing the indices of the elementary CO and H_2 uptake reactions, and U is a constant flux vector containing the experimental gas uptake fluxes. By adding this additional algebraic constraint, gas uptake into the model is forced to reflect experimentally observed values.

After solving the DAE to determine the flux distribution at the new gas uptake flux distribution state, a fitness value (y) is calculated to score how well the parameter set predicts the new observed state:

$$y = \frac{1}{n_{fluxes}} \sum_{j=1}^{n_{fluxes}} \left| \frac{v_j^{pred} - v_j^{ref}}{v_{CO\ uptake}^{ref}} \right| \tag{5.8}$$

where the absolute difference between the predicted flux (v_j^{pred}) and experimentally observed flux (v_j^{ref}) is weighted by the experimentally-observed uptake flux of CO ($v_{CO\ uptake}^{ref}$) for the given observed state.

To calculate the DAE solution for each parameter set at a given gas uptake flux profile, the MATLAB ode15s solver was used. An event function was also incorporated to halt the integration if a mass balance was achieved prior to finishing the specified time interval or if the overall

computation time exceeded a predetermined limit. A time trial determined the computation time limit where solve time and parameter fitness were compared for 100 parameter sets to understand the average time to solve. In extreme scenarios, certain parameter sets would require tens of minutes to solve. Some kinetic parameter sets are unable to resolve the required gas uptake fluxes and result in DAE solutions where intermediate metabolites accumulate indefinitely, and a new steady state prediction is not achieved. To ensure a valid, mass-balanced, steady state solution was met, the values of the dx/dt vector calculated using Eq. (5.7) at the final concentration profile were checked. If any individual metabolite balance deviated from 0 by more than 0.1, the fitness score for that parameter set was penalized and set to 100, effectively eliminating the parameter set. Our time trial indicated that a solve time cut off of 300 seconds captured all parameter sets with fitness values below 100. Parameter sets requiring greater than 300 seconds to solve typically did not reach a new mass-balanced steady state so halting their integration did not negatively reduce the predictive power of the final ensemble. To check for completed mass balance, the norm of the dx/dt vector calculated using Eq. (5.7) at the final concentration profile was determined. If the norm was close to 0 (i.e. less than $1x10^{-6}$), the integration was considered complete, and the solver was halted.

5.2.6 Estimating Change in Ethanol Production

To identify potential engineering targets for increasing ethanol production in the final kinetic parameter ensemble, the change in ethanol production with respect to the change in enzyme level for each enzymatic reaction in the network was calculated:

$$\frac{dV_{etoh}}{dEnz_r^{tot}} = \lim_{h \to 0} \frac{V_{etoh}(K_i, Enz_r^{tot} + h) - V_{etoh}(K_i, Enz_r^{tot})}{h} \tag{5.9}$$

where V_{etoh} is the predicted absolute flux through the net ethanol exchange reaction as a function of changing the level of an enzymatic reaction (r) for a given kinetic parameter set K_i in the final

ensemble of size I. Enz_r^{tot} is the sum of all enzyme species fractions (free and metabolite-bound), $e_{r,l}$, associated with a given enzymatic reaction (r) made up of elementary steps (l). For the reaction described in Eq. (5.3), these species would be E, AE, and BE. Enz_r^{tot} is always constrained to equal the initial total value to not violate mass conservation laws. The initial Enz_r^{tot} for all enzymes at the reference state is 1. To simulate over or under expression conditions, this value would go up or down respectively and each initial enzyme fraction associated with the enzyme would be multiplied by the new value of Enz_r^{tot}. To simulate over or under expression Eq. (5.5) is redefined to reflect the changes in initial enzyme fraction values:

$$v_n = k_n \prod_{\substack{m \\ S_{m,n}<0}} x_{m,perturbed}^{|S_{m,n}|}$$
$$where\ x_{m,perturbed}(e_{r,l}) = Enz_r^{tot} \tag{5.10}$$

The slopes defined in Eq. (5.9) were calculated using the forward finite difference approximation of derivatives with the step size h specified as 1 x 10^{-6}. These slope estimates indicate how altering the expression of a given enzyme affects the rate of ethanol production.

5.2.7 Computational Resources

All code was written and executed in MATLAB (MATLAB and SimBiology Toolbox 2015/2016a, The MathWorks, Inc., Natick Massachusetts, United States) on the Northwestern Quest High Performance Computing Cluster. The ensemble modeling framework was adapted from previous work by the Liao and Maranas groups, and the code was adapted from our previously published study.[34,36,39,85] The parameter screening tests were run in parallel across 12 nodes on the

Quest Computing Cluster. Sample MATLAB code is provided in the Tyo Lab Box repository for all steps described above.

5.3. Results

5.3.1 Exhaustive Cross-validation to Select Reference State from Available Dataset

In the traditional ensemble modeling framework, kinetic parameter sets are sampled using a reference steady state network flux distribution.[33] Typically, the flux distribution describing a growth-optimized, wildtype cell is used as the reference state from which the initial ensemble of kinetic parameter sets is sampled. Then, the perturbed enzyme states (i.e. knockout or overexpression phenotypes) are used to screen the ensemble.[33,34,37,41] In this study published flux states for wildtype *C. autoethanogenum* cells at low (0.5 gDCW/L), medium (1.1 gDCW/L), and high (1.3 gDCW/L) biomass concentrations were used.[111] There was not an obvious reference state, as there were no genetic perturbations.

To determine which of these three states would best serve as the reference state, 1000 kinetic parameter sets were sampled at each biomass concentration state and screened using the remaining two states (Table 5.1). This combinatorial, pair-wise pilot test revealed that only parameter sets sampled from the high biomass concentration state were able to resolve the remaining biomass concentration states. Given this observation, the high biomass concentration flux state was used as the reference flux state for the remaining, larger-scale ensemble modeling efforts. This dataset shows that carbon uptake increases with increasing biomass concentration resulting in larger flux values for most reactions in the network. In turn, the cross-validation test illustrates kinetic parameters sets that were sampled from lower absolute reaction fluxes are less able to resolve the larger flux

values observed at higher biomass concentrations. This trend has not been shown previously and highlights a limitation of this sampling method useful to consider in future modeling efforts.

Table 5.1: Pair-Wise Pilot Test Results as Percentage of Total Parameter Sets Sampled at Each Reference State Able to Resolve the Remaining Screening Conditions

Reference State	Biomass Concentration (gDCW/L)	CO Uptake Flux (mmol/gDCW/h)	H_2 Uptake Flux (mmol/gDCW/h)	**Screening Condition**		
				Low BC	Med BC	High BC
Low BC	0.5	-21.4	-13.1		0%	0%
Med BC	1.1	-25.6	-13.6	3.7%		0%
High BC	1.3	-31.6	-12.9	2.4%	6.2%	

5.3.2 Ensemble Generation and Screening

To generate a kinetic ensemble model of *C. autoethanogenum*, an initial ensemble of 359,000 locally stable kinetic parameter sets were sampled at the high biomass concentration reference flux state (Fig. 5.2). Our previous study demonstrated increased computational efficiency when parameter sets were screened against flux states in order of greatest distance from the reference state.[85] For this dataset, the low biomass state is the furthest from the high biomass state when the cosine similarity of fluxes through measured uptake and export reactions are compared between states. Therefore, the initial ensemble was screened against the low biomass state first using the DAE method described above. The first screening step took approximately 10,000 CPU hours to complete.

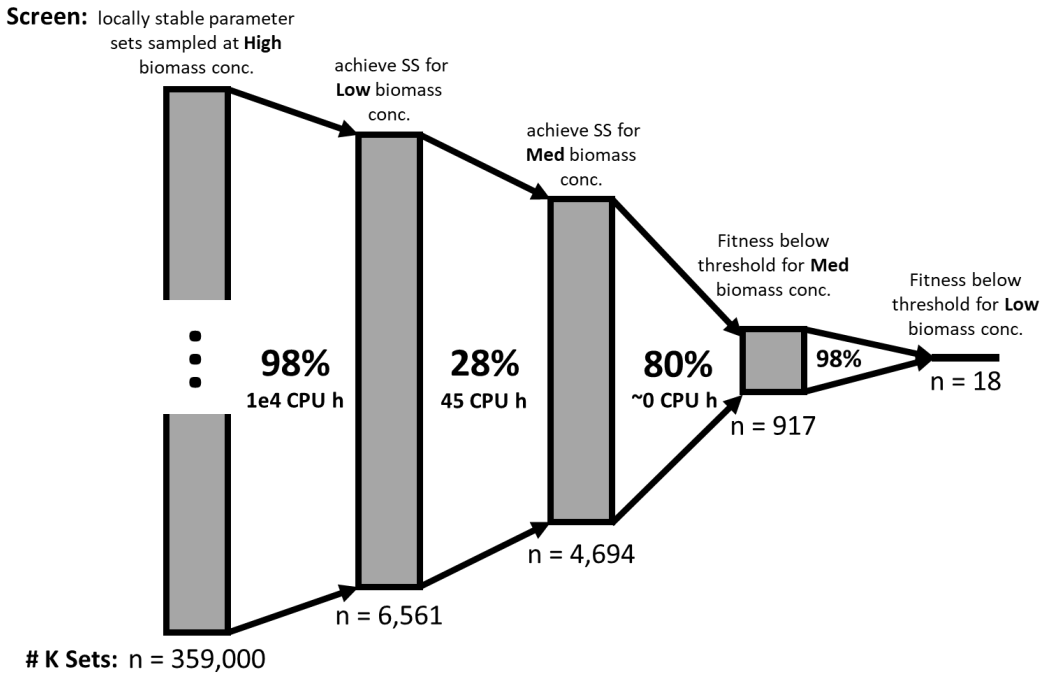

Figure 5.2: Screening Steps Used to Reduce Initial Ensemble Down to Final Representative Kinetic Parameter Sets. General overview of ensemble modeling framework. Each screening step is denoted by a column, and the percent of models removed after each step is displayed in between. The number of models remaining after each step (n) and the compute time (CPU hours) drops significantly following each screening step. After the four screening steps are completed, the initial ensemble of 359,000 locally stable kinetic parameter sets is reduced by 99.995%.

As the DAE formulation imposes specific CO and H_2 gas uptake rate constraints during integration, not all unique kinetic parameter sets in the initial ensemble are able to mathematically converge to a new, mass-balanced steady state, but result in the indefinite accumulation of intermediate(s). In the first screening step, 98% of parameter sets were unable to achieve steady states at the lower gas uptake condition, leaving only 6,561 parameter sets in the ensemble. These sets were then tested for steady states at the medium biomass condition, and 28% were removed, leaving 4,694 members in the ensemble. After removing parameter sets unable to predict the low and medium biomass concentration steady states, the fitness of each remaining parameter set was calculated, and a fitness threshold was applied (Fig. 5.3). As shown, the remaining parameter sets are overwhelmingly better at predicting the medium biomass state over the low biomass state. This

trend is expected, as the product fluxes of the medium biomass state are closer to the high biomass values as discussed above. Parameter sets with fitness scores below 0.05 for both the low and medium biomass states were retained, leaving a final ensemble consisting of 18 unique kinetic parameter sets.

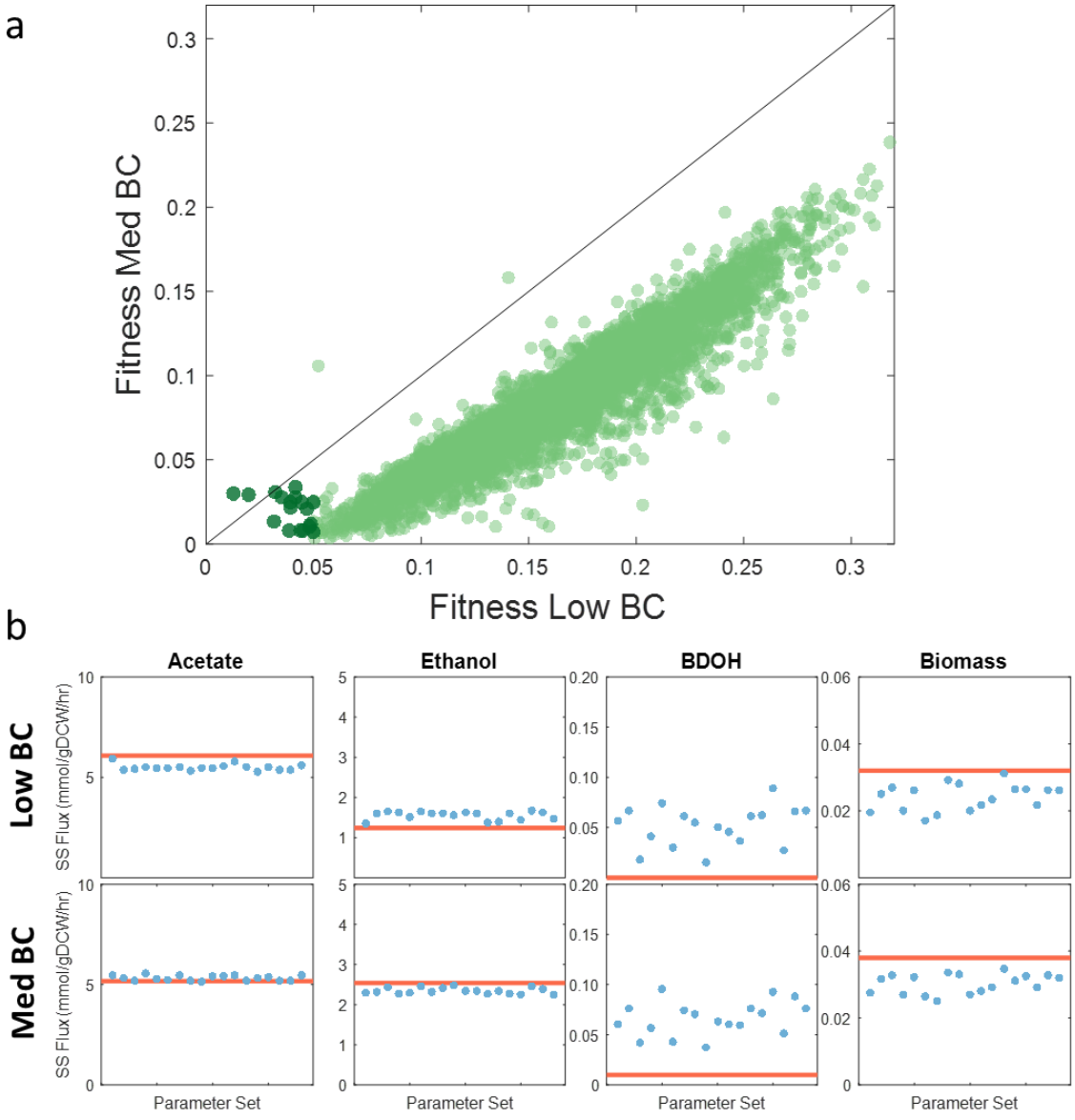

Figure 5.3: Final Kinetic Ensemble Accurately Predicts Product Distributions of Low and Medium Biomass States. a) Fitness values calculated for low and medium biomass concentrations using Eq. 8 for each parameter set remaining after the first two screening steps (n = 4,694). Dark green markers represent the final

ensemble parameter sets with fitness scores below 0.05 for both conditions (n = 18). b) Product flux predictions for each remaining parameter set (n=18) when screened at the low and medium biomass gas uptake fluxes. Red lines indicate the experimentally observed values reported in Valgepea et al.[111] 2,3-butanediol (BDOH), Biomass Concentration (BC), Steady-state Flux (SS Flux).

The remaining 18 kinetic parameter sets accurately predict the experimental flux values reported for gas uptake and product secretion for all three biomass concentration levels despite the differences in individual kinetic parameter values across each set (Fig. 5.3). This final ensemble was used for all remaining model predictions to describe the kinetic behavior of *C. autoethanogenum* metabolism.

5.3.3 Predicting Intracellular Metabolite Concentrations

The Valgepea et al. publication included intracellular metabolite concentrations for 20 metabolites included in the scope of our model.[111] Metabolite concentrations were not used to generate the kinetic parameter sets as the ensemble modeling framework tracks metabolite concentrations normalized by the reference state concentrations as opposed to absolute values.[33] However, the model can be used to predict the concentration of these 20 metabolites at both the low and medium biomass states by multiplying the predicted relative changes in metabolite concentrations by the measured metabolite concentrations at the high biomass reference state.

The final kinetic ensemble accurately predicts the metabolite concentration of both the low and medium biomass states, despite no metabolic concentration information used to train the model (Fig. 5.4). For metabolites in our model where an average measurement and standard deviation were reported by Valgepea et al., 78% and 79% of the reported metabolite concentrations fell within the range covered by the parameter set predictions for the low and medium biomass concentrations, respectively.[111] When comparing the mean of the ensemble, 59% of the predicted metabolite concentrations averaged across all 18 parameter sets in the model fell within one standard deviation of the measured average.

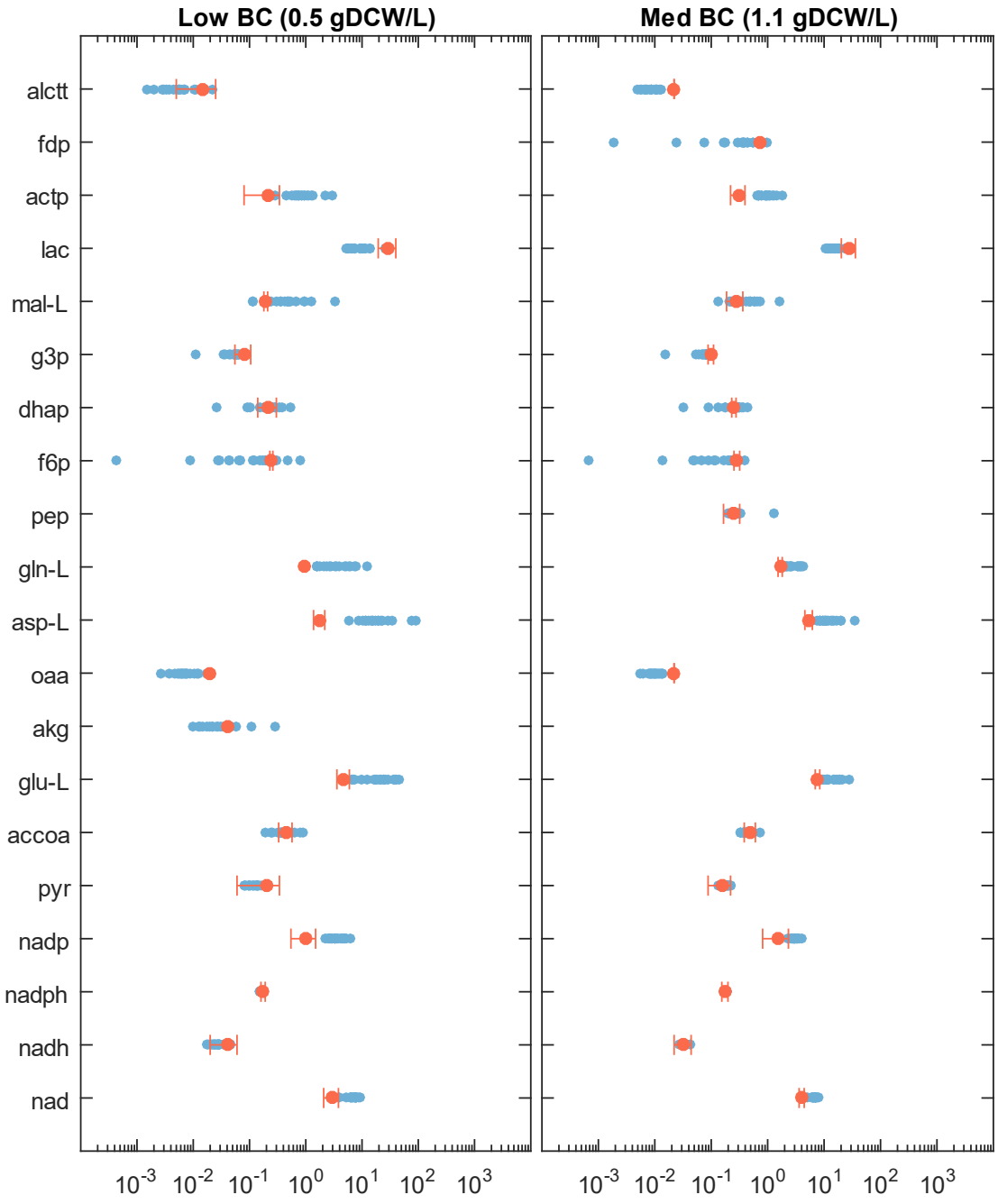

Figure 5.4: Kinetic Ensemble Accurately Predicts Intracellular Metabolite Concentrations. Average metabolite concentrations reported in Valgepea et al. are plotted in red for both low biomass concentration and medium biomass concentration states.[111] Blue markers represent individual predictions of each remaining kinetic parameter set in the ensmble (n = 18). Concentrations reported on x-axis in μM. Error bars indicate standard deviation of experimental measurements reported by Valgepea et al.[111]

5.3.4 Reproducing Experimental Observations

To further test the predictive power of the final kinetic ensemble model, we wanted to determine how well the parameter set could recreate additional experimental observations. Valgepea et al. (2017a) observed a diminishing acetyl-CoA pool as biomass concentration increased despite an accompanying increased flux through the Wood/Ljungdahl pathway which converts CO to the acetyl-CoA intermediate. Our kinetic ensemble model demonstrates this same behavior (Fig. 5a-b) and captures the limiting acetyl-CoA pool trend previously identified. Additionally, the flux balance analysis performed by Valgepea et al. (2017a) indicates that most ethanol produced by the cell likely results from acetate conversion to acetaldehyde through the acetaldehyde oxidoreductase reaction (ACAFDOR) rather than from acetyl-coA through either of the acetaldehyde dehydrogenase reactions (ACALDx/y) (Fig. 5.5c). The authors hypothesized this route was preferred as it produces an additional ATP energy molecule. Liew et al. also showed knocking out the aldehyde:ferredoxin oxidoreductase (AOR) enzyme responsible for the ACAFDOR reaction greatly reduced ethanol production during growth on CO and almost completely abolished ethanol production during growth on CO_2 and H_2.[117] Our final kinetic ensemble model predicts this same behavior and further demonstrates that most individual parameter sets predict excessive acetaldehyde is likely produced and converted back to acetyl-CoA, forming an ATP-generating loop in the network (Fig. 5.5d). This requires the ACALDx/y reactions to run in reverse, a possibility validated by our Gibbs free energy estimations. Another study indicated *Clostridium coskatii*'s lack of the ACAFDOR reaction is responsible for its lack of ethanol production [126]**.** Our model predictions reiterate the importance of this reaction in ethanol production and imply further characterization of this pathway loop might benefit ethanol production.

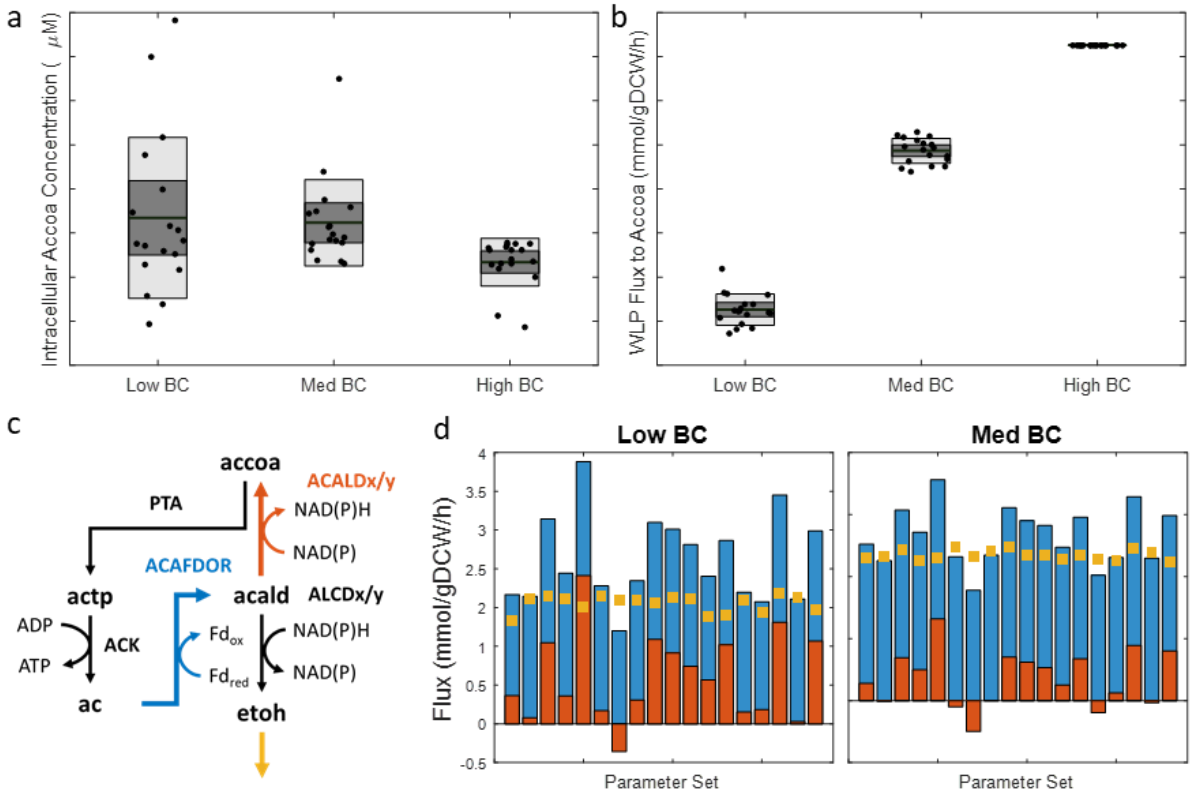

Figure 5.5: Kinetic Ensemble Model Validates Previously Observed Phenotypes. a) Intracellular acetyl-CoA concentration (μM) predicted for each identified kinetic parameter (n=18) set at low (0.5 gDCW/L), medium (1.1 gDCW/L), and high (1.3 gDCW/L) biomass concentrations. Individual predictions are plotted over box plots representing prediction consistency across the entire ensemble. Light grey boxes indicate 95% confidence intervals. Dark grey boxes indicate 1 S.D. Center lines indicate the prediction mean. b) Flux through Wood/Ljungdahl pathway toward acetyl-CoA (mmol/gDCW/L) at low, medium, and high biomass concentrations. Please note that all parameter sets predict the same flux at the high biomass state because all initial parameter sets are sampled as a function of the high biomass flux distribution. c) Metabolic pathway loop governing ethanol production flux. Metabolite and enzyme abbreviations provided in Supplemental Information Tables S1-S2. d) Flux predicted through ACALDx/y, ACAFDOR, and ethanol production reactions for each parameter set in the final ensemble. Colors in bar graph correspond with arrow colors in metabolic map displayed in part c).

5.3.5 Identifying Engineering Targets for Increased Ethanol Production

To determine the best enzyme candidates to target for improved ethanol production, we analyzed the change in ethanol production as a function of change in enzyme expression. By calculating the forward finite difference slopes for each enzymatic reaction across each parameter set we identified reactions where over or under expression may increase ethanol production (Fig.

5.6). As shown in the violin plot, the effect of changing enzyme concentration is tightly predicted by the entire ensemble for some reactions (e.g. FTHFL and HYDFDNy) but shows more variation for others (e.g. RNF, CODH_ACS, and METR).

Overall, the predictions are consistent with intuition from the metabolic network. For example, overexpressing alcohol dehydrogenase (ALCDx), which is the final step in ethanol synthesis, is predicted to improve productivity. Furthermore, in some cases, our target enzyme predictions correlate with previous experiments regarding ethanol production. For example, a Δpta mutant of *C. ljungdahlii* revealed a lower acetate to ethanol ratio than the WT and reduced acetate production.[127] Similarly, when PTA was inactivated or knocked out in an undisclosed strain, *Clostridium* sp. MT112, acetate production was abolished, and ethanol production increased.[128] Berzin et al. do not report the effects these changes have on growth rate.[128] While the PTA knockouts were performed in a different strain, the results do reconfirm our prediction that reducing the PTA enzyme activity might increase ethanol production over acetate in *C. autoethanogenum*. Initially, knocking down PTA expression is non-intuitive. Our model predictions indicate most ethanol flux is generated through this reaction's role in acetate production (Fig. 5.5 c-d). However, reducing PTA expression and forcing acetyl-CoA flux to acetaldehyde could improve ethanol production to an extent. Unfortunately, knocking down PTA expression might not be a sustainable engineering solution as the loss of ATP generation from acetate formation is the suspected cause of decreased cell growth for this *C. ljungdahlii* mutant.[127] However, this prediction may indicate an opportunity to tune PTA expression to an optimum point where all acetate produced is converted to ethanol.

A similar study in *C. autoethanogenum* showed knocking out the carbon monoxide dehydrogenase and acetyl-CoA synthase enzyme complex (CODH_ACS) increased ethanol production and reduced biomass production during heterotrophic (consuming CO and fructose) growth.

However, the authors also found that the CODH_ACS enzyme complex is essential during auxotrophic (CO only) growth, and the mutant cannot grow.[129] Again, our model predictions imply knocking down (rather than completely knocking out) the CODH_ACS reaction may lead to high ethanol production without fully compromising cell growth. Hypothetically, an optimized CODH_ACS expression could alleviate this essentiality limitation while still improving ethanol production.

In some cases, different parameter sets in the ensemble predict under expression would increase ethanol production while others predict over expression would be better (e.g. ATPase and MTHD). In these cases, while the type of enzyme engineering to best increase ethanol production may not be clear, the variation in ensemble prediction implies ethanol production is sensitive to these reactions, suggesting they are potentially additional candidates for further experimental characterization.

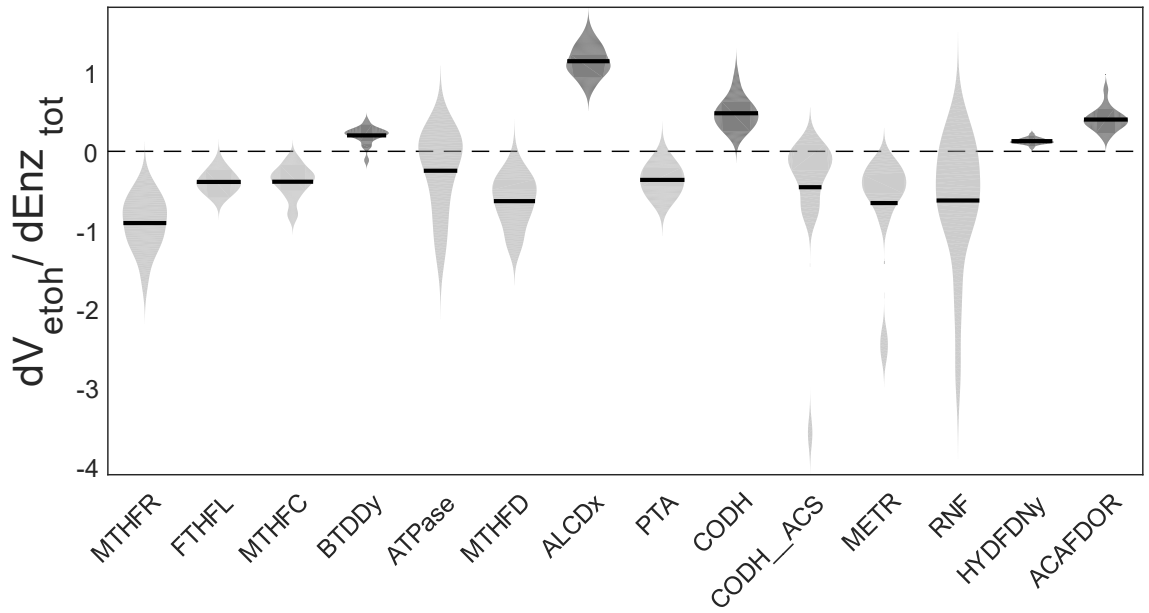

Figure 5.6: Evaluating the Change in Ethanol Production as a Function of Enzyme Expression Elucidates Potential Engineering Targets. Violin plots show distribution of finite difference estimates of entire kinetic ensemble model for multiple enzymatic reactions. Shorter, wider violins indicate more tightly predicted changes while longer, narrower violins indicate a larger variation in ensemble predictions. The midlines indicate the population mean. All enzymes were evaluated, but only enzymes with absolute slope means above 0.1 are displayed.

5.4 Discussion

Acetogenic clostridia including *C. autoethanogenum* provide a promising avenue for gas phase fermentation of industrial waste products into valuable chemicals. Current engineering efforts are focused on ethanol and 2,3-butanediol production, but the metabolic engineering potential of the strain could likely be harnessed to develop even higher complexity products.[112,115] To best activate the full potential of these strains, accurate computational models are needed to identify the governing forces of metabolism including but not limited to cofactor and energy limitations, regulation pathways, and rate-limiting steps. While progress has been made to develop detailed stoichiometric constraint-based models, our work demonstrates the first kinetic representation of *C. autoethanogenum.*[110,111,119] Specifically, we have shown that our final ensemble of kinetic parameters accurately predicts intracellular metabolite concentrations and reconfirms previously observed trends across varying biomass concentrations.

Traditionally in ensemble modeling, enzyme perturbation data sets are used to iteratively screen the kinetic parameter sets.[33,34,36,39,41] However, as discussed previously, the genetic engineering toolbox for *C. autoethanogenum* is limited. Therefore, we needed to utilize datasets differentiated by environmental conditions rather than changes in enzyme expression as they are more readily available. To take advantage of an existing published dataset generated at different biomass concentration states, we reframed the ensemble modeling governing equations as a DAE problem. Instead of simulating enzyme level changes, specific, constant gas uptake fluxes are enforced, resulting in changes to the CO and H_2 substrate pools from the initial reference state. By first screening out parameter sets unable to resolve these changes in gas uptake, we drastically prune

down our initial kinetic ensembles prior to even comparing their prediction fitness. In reframing our problem as a DAE where we can screen for specific flux values rather than genetic changes, we have broadened the application of the method. For example, studies comparing a cell's ability to uptake different substrates could equally benefit from a kinetic analysis utilizing this expanded framework.

The dataset showed increases in biomass concentration led to increased CO gas uptake by cells.[111] As the gas uptake distribution changed, the ratio of ethanol to acetate production also changed. Our final kinetic ensemble model predicts this shift in product flux distribution across all three measured biomass concentration states. Furthermore, our ensemble confirms a limiting acetyl-CoA pool and the importance of the ACAFDOR reaction in ethanol production.[111] Additionally, as the absolute fluxes increased with increasing biomass concentration, we observed a limitation in selecting low flux distributions as the reference state used to sample the initial kinetic parameter sets. Parameters screened at the low and medium biomass concentrations were not able to predict the higher carbon uptake flux solutions. This observation would likely not surface using traditional data sets for ensemble modeling because the wild-type flux distribution under optimal growth conditions is typically selected as the reference state. In contrast, knockout or overexpression datasets typically do not result in flux values higher than the reference values across the network as is the case with the increasing biomass concentration dataset used in this study.

Importantly, we were able to construct a kinetic representation of *C. autoethanogenum* core metabolism using a small dataset consisting of only three experimental conditions over a range of biomass concentrations. Quality experimental data is difficult to collect, and often larger datasets are unavailable in the literature. But in the case of our model, fewer observations were needed than expected to generate an accurate model with validated predictive power. Interestingly, the inflexible topology of the *C*. *autoethanogenum* metabolism may be the reason we can successfully capture the

cell's kinetic behavior with fewer screening conditions. The core metabolic network covered in this study has few branch points or cycles and few alternative pathways for producing energy equivalents. Therefore, given the constrained substrate uptake, the network has only so many options for resolving the flux through the system. More complex metabolic network representations (i.e. *E. coli* core metabolism) would require more screening observations to achieve the same resolution.

In addition to resolving existing experimental observations, our kinetic model of *C. autoethanolgenum* also informs future experimental efforts. In this work, we have shown the model's ability to predict enzyme targets for improved ethanol production. However, the scope of the model enables additional predictive capabilities as well. Lastly, as the ensemble modeling method heavily relies on the reference flux state used during parameter sampling as the predominant input, this method could be expanded to resolve flux distribution uncertainties.[33] When predicting the reference flux state, FVA results indicate multiple flux distributions meet the known experimentally observed constraints. For this analysis, we chose a solution maximizing CO_2 and ATP production. However, as demonstrated previously, the procedure outlined in this study could be performed for multiple reference states within the allowable FVA solution, and split ratios between parallel reactions using different cofactors (e.g. ACALDx/y or HYDFDNx/y) could be elucidated.[34,43] In all, our DAE-based ensemble modeling framework appears to be a useful strategy for building a kinetic metabolic model using non-genetic changes with the capability to make engineering predictions.

Chapter 6: Conclusions and Future Directions

6.1 Conclusions

This dissertation presents computational modeling solutions to metabolic engineering challenges. Through iterative efforts between computational and experimental efforts, this work demonstrates the importance of model-guided engineering design in tackling complicated biological questions. By adapting existing constraint-based and kinetic modeling methods for novel metabolic engineering uses, we have broadened the application of these methods and in some cases improved their performance ability.

In Chapter 2, I have made a case for the importance of metabolic modeling and provided a general overview of major constraint-based modeling methods and their successful implementations. I also highlight the limitations of constraint-based steady models and the inherent need for robust kinetic modeling methods. Lastly, I introduce the specific Monte Carlo sampling-based Ensemble Modeling method. By describing the simple, step-by-step implementation of the base framework, I provide a foundation for the expansive work described in Chapters 4 and 5 which builds on this method.

In Chapter 3, I discussed the importance of inducing and harnessing non-growth metabolism for the development of an improved carbon-yield bioproduction platform. Current understanding of non-growth metabolism is limited but knowledge of what governs cellular metabolism under

quiescence has broad potential impacts in the metabolic engineering and human health fields. Using the Minimization of Metabolic Adjustment (MOMA) constraint-based modeling method, essential enzyme knockouts resulting in metabolite auxtotrophies were identified as potential targets for inducing quiescence in *E. coli* cells. Thirty knockouts spanning *E. coli* metabolism were selected for experimental characterization and initial tests comparing growth in rich and minimal media confirmed the predicted auxotrophies. Further validating its potential to model non-growth metabolism, we also used MOMA to identify the individual media supplements needed to recover growth in each knockout candidate for shock switch experimental tests and were able to confirm at least one correct supplement for each candidate. This work indicates the promise of coupling experimental and computational efforts to enable the study of non-growth metabolism. Following completion of additional experimental characterization, the MOMA method predictions for non-growth flux distributions for each auxotrophic candidate will be analyzed and hypothesizes of metabolic objectives for non-growing cells can be validated.

In Chapter 4, we turn to kinetic ensemble modeling and our efforts to alleviate computational limitations in the existing framework. By identifying and removing linearly dependent metabolite and enzyme fraction species and screening for locally stable parameter sets, computational time requirements for representative unit tests on a large model were reduced 70%. We also demonstrated a way to rank screening data *a priori* to ensure the most difficult to predict perturbations are screened first. This simple pre-calculation step is quick and resulted in a computational time improvement of 48% over screening data ranked in reverse order for our large model unit testing. These improvements, which can be incorporated into all ensemble modeling efforts, provide computational benefits but also ensure selection of locally stable and thus more biologically representative kinetic descriptions of cellular metabolism.

In Chapter 5, the accelerated ensemble modeling framework described in Chapter 4 was used to develop the first kinetic representation of *C. autoethanogenum*. By broadening the method to screen against non-genetic perturbations of metabolism, we used ensemble modeling to accurately predict internal metabolite concentrations and potential engineering targets for improving ethanol production on syngas. This work opens avenues for further kinetic modeling efforts to benefit commercial strain development and highlights the benefits of supplementing standard constraint-based modeling with kinetic information on a less-characterized microorganism where a lack of genetic engineering tools is limiting.

6.2 Future Directions

6.2.1 Improving Model Accuracy

The Monte Carlo-based ensemble modeling method takes a continuous kinetic parameter space and samples several possible kinetic parameter sets that each accurately represent the provided reference state flux condition.[33] The Maranas Lab adapted the traditional EM method by incorporating a global optimization step, specifically implementing a genetic algorithm framework into their workflow.[39,54] Rather than discarding an entire kinetic parameter set for being unable to predict a single enzyme perturbation, as a few bad parameters in a set of several hundred could be the culprit, the genetic algorithm allows you to mix and match elementary parameter sets clustered by enzyme from different models generated in the initial ensemble seeding. This is possible as parameters are sampled at the reaction level to ensure the net flux through each reaction matches the specified reference flux distribution. The genetic algorithm framework then looks to see if different combinations of these theoretically accurate parameters better fit the provided data than the original parent parameter sets. This method keeps all originally generated parameters in

contention for the solution rather than completely throwing out models that do not accurately predict the data gathered for a number of cellular perturbations.

While the genetic algorithm potentially helps explore the sample space more thoroughly there is no specific evidence it is the best global optimization technique for this type of problem. Other non-metabolic kinetic modeling efforts have successfully utilized different metaheuristic techniques including particle swarm optimization which may be better suited.[130] The genetic algorithm method requires setting several parameters in advance, including but not limited to the number of mutations per iteration, the number of generations per test case, and the size of population screened in each generation. There are not specific guidelines for pre-determining these parameters and therefore extensive characterization must be completed to define the optimal choices for each unique application. Extensive work was completed to determine optimal genetic algorithm parameter values for the core E. coli model during the course of this dissertation research. However, the results indicated that each GA design parameter is not independent of the other so clear governing factors were not identifiable. Instead solutions generally improved with increasing parameter set sample size regardless of the genetic algorithm design parameter tested. To see the detailed results of these tests, please refer to the Tyo Lab Box Repository file "Genetic Algorithm Implementation Notes" in my designated dissertation folder. While the results of genetic algorithm investigation were not conclusive regarding the effectiveness of implementing this method in coordination with the traditional ensemble modeling framework, we do hypothesize that identifying and implementing other meta-heuristic global optimization methods in tandem with the initial parameter sampling could be very valuable toward defining more accurate kinetic models of metabolism.[131,132] In any case, we know that there are a range of parameter sets that can equally predict all experimental conditions because of the presence of many local optima in the solution space. Therefore, if we later choose to employ an additional optimization step like a genetic

algorithm in tandem with the traditional EM framework, we would want to run the analysis multiple times to generate several (rather than just one), optimized kinetic parameter solutions. This could provide a more robust way to generate our final, representative kinetic parameter ensemble, with the hope being these parameter sets would have better prediction ability than those generated without the additional global optimization step. In addition to expanding the EM-GA idea implemented by Khodayari et al. to generate multiple, optimized parameter sets, we would also like to explore the potential loss of local stability that might occur when mixing and matching kinetic parameter sets for a given reaction across the ensemble.[39] Our hypothesis is that by swapping out kinetic parameters for a given reaction, the new child parameter set may no longer meet our local stability criteria. However additional analysis of local stability for these child parameter sets could also potentially point to net reactions most likely leading to over local instability.

As the acceleration techniques described in Chapter 4 allow us to sample and screen more parameter sets, expanding the method to include a follow-up global optimization step is no longer computationally intractable. For example, the traditional framework could be used to generate a final ensemble of parameter sets able to predict the screening data and then these sets could be used to seed a global search metaheuristic method. A more thorough review of the available methods and their strengths may also indicate a better metaheuristic algorithm than the genetic algorithm for searching the kinetic parameter sample spaces. For example other evolutionary population-based metaheuristic algorithms, including scatter search, particle swarm optimization, or differential evolution, may be better choices.[133–135] Alternatively, a trajectory-based algorithm could be applied but due to the multi-model solution space typically seen in systems biology models, local search methods would likely not perform well unless our initial parameter set guesses were each tested individually or a uniform initial population was selected covering the sample space.[136,137] The final *Clostrdium autoethanogenum* kinetic representation developed in Chapter 5 would be an optimal

test case for investigating the potential benefits of coupling a metaheuristic algorithm to the ensemble modeling framework. The final 18 unique parameter sets could be used as starting points for further optimization and improved prediction fitness.

6.2.2 Structural Ensemble Modeling

Using EM, we can also interrogate possible regulation effects by creating individual ensembles of separate structural models that include different unconfirmed regulatory relationships.[47] For example the base model network would be defined and then a second model structure would be created that incorporated a single, unique regulatory interaction (i.e. metabolite X_m allosterically inhibits enzyme E_n). Then an initial ensemble of parameter sets would be sampled for each model structure (the base case and the plus-one regulation case) and screened. If the model structure including the regulation reaction's final ensemble of parameters had the better group fitness scores than the base model, the regulation reaction incorporated would be marked as a potential candidate for model inclusion. After testing several individual regulatory relationships, combinatorial model structures would also be interrogated. In this iterative method, a final list of highly-probable regulatory reactions would be compiled for better informed experimental characterization.

It is important to note that these simulations quickly expand the number of computations required as a traditional ensemble modeling effort would be applied to each model structure tested. However, our efforts to accelerate the calculations in the screening step greatly relieve this computational strain and allow for more exhaustive, previously intractable, structural interrogation.

Exploring regulatory control of cellular metabolism is crucial to successfully predicting phenotypic outcomes. In some cases, kinetic models lacking allosteric effects were unable to accurately predict experimental data, highlighting the importance of including regulation in kinetic

models.[36,81] By exploiting these structural EM methods, a novel application, we can identify the major regulatory effects limiting metabolic activity in quiescent cells as well as uncover previously unidentified regulatory relationships in the *C. autoethanogenum* model.

6.2.3 Interrogating Parallel Reactions

As mentioned in Chapters 4 and 5, ensemble modeling can be used to flush out unresolved flux values. Previous studies have demonstrated the successful use of ensemble modeling to resolve unknown split ratios between unresolved branched pathways.[40,43] For example, Rizk et al. performed ensemble modeling using four unique reference states which varied in the split ratio between the glycolysis and pentose phosphate pathways.[40] The remaining parameter set ensembles after screening were then compared and split ratios resulting in poor fitness results were ruled out. Through this analysis they were able to constrain the previously unresolved split ratio more accurately. I believe this technique could be adapted to begin to understand flux splits between parallel enzymatic reactions using different cofactors.

For *C. autoethanogenum*, flux variability when fitting the reference flux state is predominantly governed by the need to balance redox cofactors (i.e. NAD(P)H and ferredoxin) system-wide. Several reactions throughout *C. autoethanogenum* can be performed by parallel enzymes with different cofactor requirements. For our analysis in Chapter 5, the reference flux distribution was selected by assuming the majority of flux through a given reaction would be utilized by the enzyme with the highest measured activity.[122,123] However by pre-assuming different flux ratio splits between these parallel reactions and comparing ensemble modeling results from different reference flux distributions, we could gain insight in to which co-factor specific enzyme dominates flux through its respective pathway under varying metabolic conditions. The most interesting case would be pinning down the flux split between the NADP- and NAD-dependent hydrogenases (HYDFDNy and HYDFDNx

respectively in our model). Previous experimental characterization implies the NADP-dependent enzyme is predominantly expressed but a NAD-dependent analog is present at significantly lower levels.[122] Constraint-based modeling results indicate a significant impact of both enzymes being present in the system. Therefore, using our kinetic modeling technique to determine a reasonable flux split between these two enzymes would provide valuable insight into understanding the redox balance of the entire metabolism. Thoroughly understanding energy balances for clostridia strains is particularly important as previous work indicates central pathways are governed by thermodynamics rather than substrate-level regulation.[111,119,120]

6.2.4 Exhaustive Sensitivity Analysis

In Chapter 5, a slope-based sensitivity analysis was performed to determine reactions throughout the *C. autoethanogenum* network potentially governing ethanol production. This analysis could be scaled across other areas of metabolism as well. For example, a similar enzyme expression analysis could be performed to increase other products of interest including 2,3-butanediol or pathway intermediates like acetyl-CoA or pyruvate as starting points for heterologous pathway designs. Combinatorial enzyme expression and knockout changes could be interrogated as well. We could also build upon the framework and expand the scope of the model (i.e. include more reactions) as additional screening dataset become available from our experimental collaborators.

In addition to an enzyme-specific sensitivity analysis, performing a sensitivity analysis on the kinetic parameter set solutions would help elucidate rate-limiting reactions in the network. Specifically the rate-determining step of each reaction can be determined by computing the degree of rate control for each step l, $X_{rc,l}$[138,139]:

$$X_{rc,l} = \left(\frac{k_l}{R}\right)\left(\frac{\delta R}{\delta k_l}\right) \quad \#(6.1)$$

where R is the net rate of reaction, k_l is the rate constant for step l, and the partial derivative $\delta R / \delta k_l$ is taken while holding the equilibrium constant for step l, $k_{l,forward}/k_{l,reverse}$ constant. If the degree of rate control is above 0.95, the step is considered rate-determining.[138,139] In our networks, linear pathway reactions can be lumped to determine the most rate-determining step across all included elementary steps. We can also compare the ratio of the forward rate of each step to the net reaction rate to identify rate-determining steps. An elementary step is considered quasi-equilibrated if this ratio is close to 0 and rate-determining if this ratio is close to 1.[138]

6.2.5 Kinetic Models of Non-Growth Metabolism

Ultimately combining the two efforts of this dissertation, predicting non-growth metabolism outcomes and building kinetic ensemble models, is a necessary goal in achieving quiescent cell biosynthesis platforms.

Understanding the steady state fluxes of non-growing cells is a necessary first step in characterizing stationary metabolism. However, previous work indicates microbes likely employ extensive regulation during stationary phase to reduce metabolic activity and conserve their limited resources.[140] Specifically, allosteric regulation likely governs the cell's global response, as opposed to transcriptional or translational regulation, because the cells are too resource limited to induce a full scale genetic overhaul.[86,140] E. coli in particular are known to maintain their genetic and metabolic machinery in a way that allows them to quickly ramp back up to a full growth phenotype as soon as resources become available again.[76] Therefore, once we confidently identify the governing kinetics and regulatory behavior of stationary phase metabolism, we can work towards overcoming the current limitations to metabolic activity and fully realize uncoupled growth and production. The FBA and MOMA methods do not explicitly model regulation, but allow us to make assumptions about global optima and indirectly infer regulation.

While the ensemble methods for system-wide kinetic parameter determination are well developed and have undergone numerous improvements, I propose this method has further unexplored application in modeling the kinetic and regulatory behavior of non-growing cells. This work would point to the rate-determining steps in central carbon metabolism and the regulatory control parameters governing cellular decisions.

Unfortunately, while there are several genetic and environmental options for inducing an E. coli cell to enter a non-growth phase, as mentioned in Chapter 3, these cells often become metabolically dormant. In essence, the cells enter a lock-down mode to attempt to survive but in doing so also shut down large portions of their metabolic activity.

Despite some consistent mechanisms, the intensity of the cellular response program varies significantly in regard to the type of perturbation it experiences.[76,141] Some perturbations are less intense to the cells and allow them to still maintain some of their metabolic activity.[142] For example if cells are still able to synthesize energy, cells may not completely reduce their nutrient uptake.[140] To date, most studies investigating cells entering a non-growth phase focus on carbon starvation transitions.[140] Our work will instead focus on understanding metabolic limitations in carbon-rich environments which are not well understood or characterized. If the limitations can be identified through a combination of experimental and computational efforts, we can then work toward alleviating them.

We know complicated regulation networks inside the cell govern the perturbation responses leading to non-growth phenotypes. These regulation mechanisms assess the risk of a given perturbation to the cell and then initiate the appropriate response cell-wide. For this project it is important to characterize what regulatory mechanisms govern metabolic activity 1) during the

transition from growth to non-growth and 2) during the sustained non-growth phase following a growth inhibiting perturbation.

The exact mechanisms of regulation during and after the transition to non-growth stages are not well understood and govern cellular processes at multiple levels including transcription, translation, and post-translation. Many studies conclude that transcriptional regulation may be over-emphasized due to ease of screening methods.[143] For example, in a study of the small, genome reduced bacteria, Mycoplasma pneumoniae, the authors observed similar complex regulatory responses to environmental and genetic perturbations as seen in more complex bacteria. However, M. pneumonia only have about eight transcription and sigma factors available for genetic regulation which is far fewer than these more complex bacterial species.[144] This disparity indicates that transcriptional regulation may play a less significant role in overall metabolic regulation than previously assumed. Ultimately, this smaller bacterium is able to induce equally effective responses to perturbations despite far fewer tools for regulating gene expression.[143]

Furthermore, other studies indicate post-translational and allosteric regulation of metabolic enzymes and other metabolite-protein interactions may play the most prominent roles in stress responses.[145,146] These studies are especially significant to our work as non-growing cells may not have the capacity for a proteome overhaul which further points to the importance of understanding post-translational regulation in our systems.

Similarly the short response times required for cells to quickly recover growth rates when nutrients are reintroduced into their environments point to regulation mechanisms occurring on faster time scales than gene expression can resolve.[143,147] Studies monitoring protein synthesis during stationary phase also echo this sentiment. While protein synthesis still occurs during stationary phase, the rates are as low as 0.05% of exponential growth protein synthesis.[76] Also while

protein synthesis is essential during a cell's transition from growth to non-growth phases, it can be completely knocked out during stationary phase without killing the cells.[76] As cells do not seem to rely on synthesis of new proteins to survive in stationary phase, it could imply that modification or inhibition of existing protein activity, rather than transcriptional or translational regulation, prevents metabolic activity in non-growing cells.

The above work indicates that cells, through years of evolution, have developed robust metabolisms with the ability to respond to and survive the dangers of numerous environmental perturbations.[22] However, cells entering non-growth stages from genetic perturbations may not be subject to all of the same extensive levels of regulation. Recently, Slavov et al. studied auxotrophic yeast in a glucose-rich environment. They observed expression of stationary phase proteins at levels 40-50% below what is typically seen in naturally starved strains indicating the auxotrophic strain was slightly deregulated.[148] This research hints that genetically induced auxotrophs, which enter a non-growth phase through non-native scenarios, may bypass certain regulation signaling pathways normally triggered in natural starvation scenarios.

The above points lead us to believe that substrate-level regulation of existing metabolic proteins in stationary phase govern metabolic activity rather than the production of new gene products regulated at the transcriptional and translational levels. This hypothesis signals to us the importance of understanding and adapting protein-level regulation of non-growth cells to improve their metabolic activity. We are interested in genetic manipulation of cellular growth processes by inducing non-growth phase in nutrient rich environments. However, we need to achieve better understanding of non-growth metabolism and regulation first. Through robust computational models we can better resolve experimental data to point towards mechanisms of regulation and even make predictions for regulatory improvement. By better understanding the metabolism of non-

growing cells with the aim of improving their metabolic activity, we will be better equipped to decouple microbial production from growth in industrial strains and achieve higher product yields. Most importantly the implied prominence of substrate-level regulation indicates our auxotrophic studies are well-suited for ensemble modeling characterization as the framework is already well suited to capture these effects.[39,47,149]

REFERENCES

1. Stephanopoulos, G. Metabolic Fluxes and Metabolic Engineering. *Metab. Eng.* (1999). doi:10.1006/mben.1998.0101
2. Covert, M. W. *et al.* Metabolic modeling of microbial strains in silico. *Trends in Biochemical Sciences* (2001). doi:10.1016/S0968-0004(00)01754-0
3. Edwards, J. S., Covert, M. & Palsson, B. Metabolic modelling of microbes: The flux-balance approach. *Environ. Microbiol.* (2002). doi:10.1046/j.1462-2920.2002.00282.x
4. Peralta-Yahya, P. P., Zhang, F., Del Cardayre, S. B. & Keasling, J. D. Microbial engineering for the production of advanced biofuels. *Nature* (2012). doi:10.1038/nature11478
5. Gomez, J. E. & McKinney, J. D. M. tuberculosis persistence, latency, and drug tolerance. in *Tuberculosis* (2004). doi:10.1016/j.tube.2003.08.003
6. Oliver, A. High Frequency of Hypermutable Pseudomonas aeruginosa in Cystic Fibrosis Lung Infection. *Science (80-.).* (2000). doi:10.1126/science.288.5469.1251
7. Bordbar, A., Monk, J. M., King, Z. A. & Palsson, B. O. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics* (2014). doi:10.1038/nrg3643
8. Feist, A. M. & Palsson, B. The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli. *Nature Biotechnology* (2008). doi:10.1038/nbt1401
9. Österlund, T., Nookaew, I. & Nielsen, J. Fifteen years of large scale metabolic modeling of yeast: Developments and impacts. *Biotechnology Advances* **30,** 979–988 (2012).
10. Price, N. D., Reed, J. L. & Palsson, B. Ø. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nat. Rev. Microbiol.* **2,** 886–97 (2004).
11. 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,** 1–18 (2007).
12. Orth, J. D. *et al.* A comprehensive genome-scale reconstruction of Escherichia coli metabolism-2011. *Mol. Syst. Biol.* (2011). doi:10.1038/msb.2011.65
13. Orth, J. D., Thiele, I. & Palsson, B. O. What is flux balance analysis? *Nat. Biotechnol.* **28,** 245–248 (2010).
14. Price, N. D., Papin, J. A., Schilling, C. H. & Palsson, B. O. Genome-scale microbial in silico models: The constraints-based approach. *Trends in Biotechnology* (2003). doi:10.1016/S0167-7799(03)00030-1
15. Henry, C. S., Jankowski, M. D., Broadbelt, L. J. & Hatzimanikatis, V. Genome-Scale Thermodynamic Analysis of Escherichia coli Metabolism. *Biophys. J.* **90,** 1453–1461 (2006).
16. Feist, A. M., Herrgård, M. J., Thiele, I., Reed, J. L. & Palsson, B. Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology* (2009). doi:10.1038/nrmicro1949
17. Raman, K. & Chandra, N. Flux balance analysis of biological systems: applications and challenges. *Brief. Bioinforma.* (2009). doi:10.1093/bib/bbp011

18. Schuetz, R., Kuepfer, L. & Sauer, U. Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli. *Mol. Syst. Biol.* (2007). doi:10.1038/msb4100162
19. Budinich, M., Bourdon, J., Larhlimi, A. & Eveillard, D. A multi-objective constraint-based approach for modeling genome-scale microbial ecosystems. *PLoS One* (2017). doi:10.1371/journal.pone.0171744
20. Fong, S. S. *et al.* In silico design and adaptive evolution of Escherichia coli for production of lactic acid. *Biotechnol. Bioeng.* (2005). doi:10.1002/bit.20542
21. Nagrath, D., Avila-Elchiver, M., Tilles, F. B. and A. W., Messac, A. & Yarmush, M. L. Integrated energy and flux balance based multiobjective framework for large-scale metabolic networks. *Ann Biomed Eng* (2007). doi:10.1007/s10439-007-9283-0
22. Stelling, J., Sauer, U., Szallasi, Z., Doyle, F. J. & Doyle, J. Robustness of cellular functions. *Cell* (2004). doi:10.1016/j.cell.2004.09.008
23. Mahadevan, R. & Schilling, C. H. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. Eng.* **5,** 264–276 (2003).
24. Segre, D., Vitkup, D. & Church, G. M. Analysis of optimality in natural and perturbed metabolic networks. *Proc. Natl. Acad. Sci.* (2002). doi:10.1073/pnas.232349399
25. Shlomi, T., Berkman, O. & Ruppin, E. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proc. Natl. Acad. Sci.* (2005). doi:10.1073/pnas.0406346102
26. Kim, J. & Reed, J. L. RELATCH: relative optimality in metabolic networks explains robust metabolic and regulatory responses to perturbations. *Genome Biol.* (2012). doi:10.1186/gb-2012-13-9-r78
27. Saa, P. A. & Nielsen, L. K. Formulation, construction and analysis of kinetic models of metabolism: A review of modelling frameworks. *Biotechnol. Adv.* **35,** 981–1003 (2017).
28. 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* (2006). doi:10.1186/1471-2105-7-540
29. Smallbone, K., Simeonidis, E., Swainston, N. & Mendes, P. Towards a genome-scale kinetic model of cellular metabolism. *BMC Syst. Biol.* **4,** 6 (2010).
30. Wang, F. S., Ko, C. L. & Voit, E. O. Kinetic modeling using S-systems and lin-log approaches. *Biochem. Eng. J.* (2007). doi:10.1016/j.bej.2006.11.002
31. Chakrabarti, A., Miskovic, L., Soh, K. C. & Hatzimanikatis, V. Towards kinetic modeling of genome-scale metabolic networks without sacrificing stoichiometric, thermodynamic and physiological constraints. *Biotechnol. J.* (2013). doi:10.1002/biot.201300091
32. 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,** (2006).
33. Tran, L. M., Rizk, M. L. & Liao, J. C. Ensemble modeling of metabolic networks. *Biophys. J.* **95,** 5606–17 (2008).
34. Rizk, M. L. & Liao, J. C. Ensemble modeling and related mathematical modeling of

metabolic networks. *J. Taiwan Inst. Chem. Eng.* **40,** 595–601 (2009).
35. Srinivasan, S., Cluett, W. R. & Mahadevan, R. Constructing kinetic models of metabolism at genome-scales: A review. *Biotechnology Journal* (2015). doi:10.1002/biot.201400522
36. Tan, Y., Rivera, J. G. L., Contador, C. a, Asenjo, J. a & Liao, J. C. Reducing the allowable kinetic space by constructing ensemble of dynamic models with the same steady-state flux. *Metab. Eng.* **13,** 60–75 (2011).
37. Tan, Y. & Liao, J. C. Metabolic ensemble modeling for strain engineers. *Biotechnol. J.* **7,** 343–53 (2012).
38. Lee, Y., Lafontaine Rivera, J. G. & Liao, J. C. Ensemble Modeling for Robustness Analysis in engineering non-native metabolic pathways. *Metab. Eng.* (2014). doi:10.1016/j.ymben.2014.06.006
39. 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.* (2014). doi:10.1016/j.ymben.2014.05.014
40. Rizk, M. L. & Liao, J. C. Ensemble modeling for aromatic production in Escherichia coli. *PLoS One* **4,** e6903 (2009).
41. 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,** 221–33 (2009).
42. Theisen, M. K., Lafontaine Rivera, J. G. & Liao, J. C. Stability of Ensemble Models Predicts Productivity of Enzymatic Systems. *PLoS Comput. Biol.* (2016). doi:10.1371/journal.pcbi.1004800
43. Dean, J. T., Rizk, M. L., Tan, Y., Dipple, K. M. & Liao, J. C. Ensemble modeling of hepatic fatty acid metabolism with a synthetic glyoxylate shunt. *Biophys. J.* **98,** 1385–95 (2010).
44. 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.* 1–15 (2013). doi:10.1002/biot.201200270
45. Wang, L. & Hatzimanikatis, V. Metabolic engineering under uncertainty. I: Framework development. *Metab. Eng.* (2006). doi:10.1016/j.ymben.2005.11.003
46. Wang, L., Birol, I. & Hatzimanikatis, V. Metabolic Control Analysis under Uncertainty: Framework Development and Case Studies. *Biophys. J.* (2004). doi:10.1529/biophysj.104.048090
47. Link, H., Christodoulou, D. & Sauer, U. Advancing metabolic models with kinetic information. *Current Opinion in Biotechnology* (2014). doi:10.1016/j.copbio.2014.01.015
48. Khazaei, T., McGuigan, A. & Mahadevan, R. Ensemble modeling of cancer metabolism. *Front. Physiol.* **3,** 135 (2012).
49. King, Z. A. *et al.* BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.* (2016). doi:10.1093/nar/gkv1049
50. Henry, C. S. *et al.* High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat. Biotechnol.* **28,** 977–982 (2010).
51. Keseler, I. M. *et al.* The EcoCyc database: Reflecting new knowledge about

Escherichia coli K-12. *Nucleic Acids Res.* (2017). doi:10.1093/nar/gkw1003
52. Flamholz, A., Noor, E., Bar-Even, A. & Milo, R. EQuilibrator - The biochemical thermodynamics calculator. *Nucleic Acids Res.* **40,** 770–775 (2012).
53. Noor, E. *et al.* An integrated open framework for thermodynamics of reactions that combines accuracy and coverage. *Bioinformatics* **28,** 2037–2044 (2012).
54. Khodayari, A. & Maranas, C. D. A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains. *Nat. Commun.* (2016). doi:10.1038/ncomms13806
55. Burgard, A. P., Pharkya, P. & Maranas, C. D. OptKnock: A Bilevel Programming Framework for Identifying Gene Knockout Strategies for Microbial Strain Optimization. *Biotechnol. Bioeng.* (2003). doi:10.1002/bit.10803
56. Feist, A. M. *et al.* Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli. *Metab. Eng.* (2010). doi:10.1016/j.ymben.2009.10.003
57. Chotani, G. *et al.* The commercial production of chemicals using pathway engineering. *Biochim. Biophys. Acta* (2000).
58. Carlson, E. D., Gan, R., Hodgman, C. E. & Jewett, M. C. Cell-free protein synthesis: Applications come of age. *Biotechnology Advances* (2012). doi:10.1016/j.biotechadv.2011.09.016
59. Swartz, J. Developing cell-free biology for industrial applications. in *Journal of Industrial Microbiology and Biotechnology* (2006). doi:10.1007/s10295-006-0127-y
60. Hodgman, C. E. & Jewett, M. C. Cell-free synthetic biology: Thinking outside the cell. *Metab. Eng.* (2012). doi:10.1016/j.ymben.2011.09.002
61. Chang, D. E., Smalley, D. J. & Conway, T. Gene expression profiling of Escherichia coli growth transitions: An expanded stringent response model. *Mol. Microbiol.* (2002). doi:10.1046/j.1365-2958.2002.03001.x
62. Segrè, D. *et al.* From Annotated Genomes to Metabolic Flux Models and Kinetic Parameter Fitting. *Omi. A J. Integr. Biol.* (2003). doi:10.1089/153623103322452413
63. Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J. & Jirstrand, M. Kinetic models in industrial biotechnology - Improving cell factory performance. *Metabolic Engineering* (2014). doi:10.1016/j.ymben.2014.03.007
64. Schellenberger, J. *et al.* Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox v2.0. *Nat. Protoc.* **6,** 1290–1307 (2011).
65. Haverkorn van Rijsewijk, B. R. B., Nanchen, A., Nallet, S., Kleijn, R. J. & Sauer, U. Large-scale 13C-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in Escherichia coli. *Mol. Syst. Biol.* **7,** 477 (2011).
66. Baba, T. *et al.* Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: The Keio collection. *Mol. Syst. Biol.* (2006). doi:10.1038/msb4100050
67. EZ Rich Defined Medium provided by the E. coli Genome Project at the University of Wisconsin. Available at: https://www.genome.wisc.edu/resources/protocols/ezmedium.htm.
68. Neidhardt, F. C., Bloch, P. L. & Smith, D. F. Culture medium for enterobacteria. *J. Bacteriol.* (1974).

69. Du, B. *et al.* Evaluation of rate law approximations in bottom-up kinetic models of metabolism. *BMC Syst. Biol.* (2016). doi:10.1186/s12918-016-0283-2
70. Chowdhury, A., Khodayari, A. & Maranas, C. D. Improving prediction fidelity of cellular metabolism with kinetic descriptions. *Current Opinion in Biotechnology* (2015). doi:10.1016/j.copbio.2015.08.011
71. Jamshidi, N. & Palsson, B. Ø. Formulating genome-scale kinetic models in the post-genome era. *Mol. Syst. Biol.* **4,** 171 (2008).
72. Rohwer, J. M. Kinetic modelling of plant metabolic pathways. *Journal of Experimental Botany* (2012). doi:10.1093/jxb/ers080
73. Costa, R. S., Hartmann, A. & Vinga, S. Kinetic modeling of cell metabolism for microbial production. *Journal of Biotechnology* (2016). doi:10.1016/j.jbiotec.2015.12.023
74. Bordbar, A. *et al.* Personalized Whole-Cell Kinetic Models of Metabolism for Discovery in Genomics and Pharmacodynamics. *Cell Syst.* (2015). doi:10.1016/j.cels.2015.10.003
75. Matsuoka, Y. & Shimizu, K. Current status and future perspectives of kinetic modeling for the cell metabolism with incorporation of the metabolic regulation mechanism. *Bioresour. Bioprocess.* (2015). doi:10.1186/s40643-014-0031-7
76. Kolter, R. The Stationary Phase of the Bacterial Life Cycle. *Annu. Rev. Microbiol.* (1993). doi:10.1146/annurev.micro.47.1.855
77. Snoep, J. L., Bruggeman, F., Olivier, B. G. & Westerhoff, H. V. Towards building the silicon cell: A modular approach. in *BioSystems* (2006). doi:10.1016/j.biosystems.2005.07.006
78. Bruggeman, F. J. & Westerhoff, H. V. The nature of systems biology. *Trends in Microbiology* (2007). doi:10.1016/j.tim.2006.11.003
79. Teusink, B. *et al.* Can yeast glycolysis be understood terms of vitro kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem.* (2000). doi:10.1046/j.1432-1327.2000.01527.x
80. Stanford, N. J. *et al.* Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One* (2013). doi:10.1371/journal.pone.0079195
81. Link, H., Kochanowski, K. & Sauer, U. Systematic identification of allosteric protein-metabolite interactions that control enzyme activity in vivo. *Nat. Biotechnol.* (2013). doi:10.1038/nbt.2489
82. Oberhardt, M. A., Palsson, B. & Papin, J. A. Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology* (2009). doi:10.1038/msb.2009.77
83. Terzer, M., Maynard, N. D., Covert, M. W. & Stelling, J. Genome-scale metabolic networks. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* (2009). doi:10.1002/wsbm.37
84. Lafontaine Rivera, J. G., Lee, Y. & Liao, J. C. An entropy-like index of bifurcational robustness for metabolic systems. *Integr. Biol.* (2015). doi:10.1039/C4IB00257A
85. Greene, J. L., Wäechter, A., Tyo, K. E. J. & Broadbelt, L. J. Acceleration Strategies to Enhance Metabolic Ensemble Modeling Performance. *Biophys. J.* **113,** 1150–1162 (2017).
86. Ishii, N. *et al.* Multiple High-Throughput Analyses Monitor the Response of E. coli to

Perturbations. *Science (80-.).* (2007). doi:10.1126/science.1132067
87. 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.* (2007). doi:10.1038/msb4100155
88. Schellenberger, J. *et al.* Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox v2.0. *Nat. Protoc.* (2011). doi:10.1038/nprot.2011.308
89. Mahadevan, R. & Schilling, C. H. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. Eng.* (2003). doi:10.1016/j.ymben.2003.09.002
90. Orth, J. D., Thiele, I. & Palsson, B. O. What is flux balance analysis? *Nature Biotechnology* (2010). doi:10.1038/nbt.1614
91. Mišković, L. & Hatzimanikatis, V. Modeling of uncertainties in biochemical reactions. *Biotechnol. Bioeng.* **108,** 413–23 (2011).
92. Dash, S. *et al.* Development of a core Clostridium thermocellum kinetic metabolic model consistent with multiple genetic perturbations. *Biotechnol. Biofuels* (2017). doi:10.1186/s13068-017-0792-2
93. Vallabhajosyula, R. R., Chickarmane, V. & Sauro, H. M. Conservation analysis of large biochemical networks. *Bioinformatics* (2006). doi:10.1093/bioinformatics/bti800
94. Hoops, S. *et al.* COPASI - A COmplex PAthway SImulator. *Bioinformatics* (2006). doi:10.1093/bioinformatics/btl485
95. Grimbs, S., Selbig, J., Bulik, S., Holzhütter, H. G. & Steuer, R. The stability and robustness of metabolic states: Identifying stabilizing sites in metabolic networks. *Mol. Syst. Biol.* (2007). doi:10.1038/msb4100186
96. Sauro, H. M. & Ingalls, B. Conservation analysis in biochemical networks: Computational issues for software writers. *Biophysical Chemistry* (2004). doi:10.1016/j.bpc.2003.08.009
97. Stelling, J., Sauer, U., Iii, F. J. D. & Doyle, J. Complexity and robustness of cellular systems. *Syst. Model. Cell. Biol.* 3–18 (2006).
98. Jaskowiak, P. A. *et al.* On the selection of appropriate distances for gene expression data clustering. *BMC Bioinformatics* (2014). doi:10.1186/1471-2105-15-S2-S2
99. Ertöz, L., Steinbach, M. & Kumar, V. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. *Proc. Second SIAM Int. Conf. Data Min.* (2003). doi:doi:10.1137/1.9781611972733.5
100. Drake, H. L., Gößner, A. S. & Daniel, S. L. Old acetogens, new light. *Ann. N. Y. Acad. Sci.* **1125,** 100–128 (2008).
101. Drake, H. L. Acetogenesis, acetogenic bacteria, and the acetyl-CoA 'Wood/Ljungdahl' pathway: past and current perspectives. in *Acetogenesis* 3–60 (Springer US, 1991).
102. Ljungdahl, L. G. The Autotrophic Pathway of Acetate Synthesis in Acetogenic Bacteria. *Annu. Rev. Microbiol.* 415–450 (1986). doi:10.1146/annurev.mi.40.100186.002215
103. Abrini, J., Naveau, H. & Nyns, E.-J. Clostridium autoethanogenumt, sp. nov., an anaerobic bacterium that produces ethanol from carbon monoxide. *Arch. Microbiol.* **161,** 345–351 (1994).

104. Kopke, M. *et al.* 2,3-Butanediol Production by Acetogenic Bacteria, an Alternative Route to Chemical Synthesis, Using Industrial Waste Gas. *Appl. Environ. Microbiol.* **77,** 5467–5475 (2011).
105. Kopke, M. *et al.* Clostridium ljungdahlii represents a microbial production platform based on syngas. *Proc. Natl. Acad. Sci.* (2010). doi:10.1073/pnas.1004716107
106. Schiel-Bengelsdorf, B. & Dürre, P. Pathway engineering and synthetic biology using acetogens. *FEBS Lett.* **586,** 2191–8 (2012).
107. Phillips, J. R., Clausen, E. C. & Gaddy, J. L. Synthesis gas as substrate for the biological production of fuels and chemicals. *Appl. Biochem. Biotechnol.* **45–46,** 145–157 (1994).
108. Cotter, J. L., Chinn, M. S. & Grunden, A. M. Influence of process parameters on growth of Clostridium ljungdahlii and Clostridium autoethanogenum on synthesis gas. *Enzyme Microb. Technol.* **44,** 281–288 (2009).
109. Cotter, J. L., Chinn, M. S. & Grunden, A. M. Ethanol and acetate production by Clostridium ljungdahlii and Clostridium autoethanogenum using resting cells. *Bioprocess Biosyst. Eng.* **32,** 369–380 (2009).
110. Valgepea, K. *et al.* Arginine deiminase pathway provides ATP and boosts growth of the gas-fermenting acetogen Clostridium autoethanogenum. *Metab. Eng.* **41,** 202–211 (2017).
111. Valgepea, K. *et al.* Maintenance of ATP Homeostasis Triggers Metabolic Shifts in Gas-Fermenting Acetogens. *Cell Syst.* **4,** 505–515.e5 (2017).
112. Tracy, B. P., Jones, S. W., Fast, A. G., Indurthi, D. C. & Papoutsakis, E. T. Clostridia: the importance of their exceptional substrate and metabolite diversity for biofuel and biorefinery applications. *Curr. Opin. Biotechnol.* **23,** 364–81 (2012).
113. Latif, H., Zeidan, A. A., Nielsen, A. T. & Zengler, K. Trash to treasure: Production of biofuels and commodity chemicals via syngas fermenting microorganisms. *Curr. Opin. Biotechnol.* **27,** 79–87 (2014).
114. Pyne, M. E., Bruder, M., Moo-Young, M., Chung, D. A. & Chou, C. P. Technical guide for genetic advancement of underdeveloped and intractable Clostridium. *Biotechnol. Adv.* **32,** 623–641 (2014).
115. Cho, C., Jang, Y. S., Moon, H. G., Lee, J. & Lee, S. Y. Metabolic engineering of clostridia for the production of chemicals. *Biofuels, Bioprod. Biorefining* (2015). doi:10.1002/bbb.1531
116. Nagaraju, S., Davies, N. K., Walker, D. J. F., Köpke, M. & Simpson, S. D. Genome editing of Clostridium autoethanogenum using CRISPR/Cas9. *Biotechnol. Biofuels* **9,** 219 (2016).
117. Liew, F. *et al.* Metabolic engineering of Clostridium autoethanogenum for selective alcohol production. *Metab. Eng.* **40,** 104–114 (2017).
118. Daniell, J., Köpke, M. & Simpson, S. D. *Commercial biomass syngas fermentation*. *Energies* **5,** (2012).
119. Marcellin, E. *et al.* Low carbon fuels and commodity chemicals from waste gases – systematic approach to understand energy metabolism in a model acetogen. *Green Chem.* **18,** 3020–3028 (2016).
120. Richter, H. *et al.* Ethanol production in syngas-fermenting Clostridium ljungdahlii is

controlled by thermodynamics rather than by enzyme expression. *Energy Environ. Sci.* **9,** 2392–2399 (2016).
121. Schuchmann, K. & Müller, V. Autotrophy at the thermodynamic limit of life: A model for energy conservation in acetogenic bacteria. *Nat. Rev. Microbiol.* **12,** 809–821 (2014).
122. Wang, S. *et al.* NADP-Specific electron-bifurcating [FeFe]-hydrogenase in a functional complex with formate dehydrogenase in clostridium autoethanogenum grown on CO. *J. Bacteriol.* **195,** 4373–4386 (2013).
123. Mock, J. *et al.* Energy conservation associated with ethanol formation from H2 and CO2 in Clostridium autoethanogenum involving electron bifurcation. *J. Bacteriol.* **197,** 2965–2980 (2015).
124. Caspi, R. *et al.* The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* **44,** D471–D480 (2016).
125. Alberty, R. A. *Thermodynamics of biochemical reactions*. (Jon Wiley & Sons, 2003).
126. Bengelsdorf, F. R. *et al.* Industrial acetogenic biocatalysts: A comparative metabolic and genomic analysis. *Front. Microbiol.* **7,** 1–15 (2016).
127. Huang, H. *et al.* CRISPR/Cas9-Based Efficient Genome Editing in Clostridium ljungdahlii, an Autotrophic Gas-Fermenting Bacterium. *ACS Synth. Biol.* **5,** 1355–1361 (2016).
128. Berzin, V., Kiriukhin, M. & Tyurin, M. Cre-lox66/lox71-based elimination of phosphotransacetylase or acetaldehyde dehydrogenase shifted carbon flux in acetogen rendering selective overproduction of ethanol or acetate. *Appl. Biochem. Biotechnol.* **168,** 1384–1393 (2012).
129. Liew, F. *et al.* Insights into CO $_2$ Fixation Pathway of *Clostridium autoethanogenum* by Targeted Mutagenesis. *MBio* **7,** e00427-16 (2016).
130. Roy, M. & Finley, S. D. Computational model predicts the effects of targeting cellular metabolism in pancreatic cancer. *Front. Physiol.* (2017). doi:10.3389/fphys.2017.00217
131. Perez, J. & Basterrechea, J. Comparison of different heuristic optimization methods for near-field antenna measurements. *Antennas Propagation, IEEE …* (2007). doi:10.1109/TAP.2007.891508
132. Gilli, M. & Winker, P. A review of heuristic optimization methods in econometrics. *Swiss Financ. Instit* (2008). doi:10.1002/9780470748916.ch3
133. Glover, F., Laguna, M. & Martí, R. Fundamentals of scatter search and path relinking. *Control Cybern.* (2000). doi:10.1007/0-306-48056-5_1
134. Storn, R. & Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* (1997). doi:10.1023/A:1008202821328
135. Eberhart, J. K. and R. Particle swarm optimization. in *Proceedings of the IEEE International Conference on Neural Networks* (1995). doi:10.1109/ICNN.1995.488968
136. Tashkova, K., Korošec, P., Šilc, J., Todorovski, L. & Džeroski, S. Parameter estimation with bio-inspired meta-heuristic optimization: modeling the dynamics of

endocytosis. *BMC Syst. Biol.* (2011). doi:10.1186/1752-0509-5-159
137. Lin, J.-H., Chou, C.-W., Yang, C.-H. & Tsai, H.-L. A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. *Comput. Inf. Technol.* (2012).
138. Bjorkman, K. R., Schoenfeldt, N. J., Notestein, J. M. & Broadbelt, L. J. Microkinetic modeling of cis-cyclooctene oxidation on heterogeneous Mn-tmtacn complexes. *J. Catal.* (2012). doi:10.1016/j.jcat.2012.03.026
139. Campbell, C. T. Finding the Rate-Determining Step in a Mechanism. *J. Catal.* (2001). doi:10.1006/jcat.2001.3396
140. Chubukov, V. & Sauer, U. Environmental dependence of stationary-phase metabolism in bacillus subtilis and escherichia coli. *Appl. Environ. Microbiol.* (2014). doi:10.1128/AEM.00061-14
141. Peterson, C. N., Mandel, M. J. & Silhavy, T. J. Escherichia coli starvation diets: Essential nutrients weigh in distinctly. *Journal of Bacteriology* (2005). doi:10.1128/JB.187.22.7549-7553.2005
142. Ballesteros, M., Fredriksson, Å., Henriksson, J. & Nyström, T. Bacterial senescence: Protein oxidation in non-proliferating cells is dictated by the accuracy of the ribosomes. *EMBO J.* (2001). doi:10.1093/emboj/20.18.5280
143. Heinemann, M. & Sauer, U. Systems biology of microbial metabolism. *Curr. Opin. Microbiol.* (2010). doi:10.1016/j.mib.2010.02.005
144. Yus, E. *et al.* Impact of Genome Reduction on Bacterial Metabolism and Its Regulation. *Science (80-.).* (2009). doi:10.1126/science.1177263
145. Chubukov, V. *et al.* Transcriptional regulation is insufficient to explain substrate-induced flux changes in Bacillus subtilis. *Mol. Syst. Biol.* (2013). doi:10.1038/msb.2013.66
146. Daran-Lapujade, P. *et al.* The fluxes through glycolytic enzymes in Saccharomyces cerevisiae are predominantly regulated at posttranscriptional levels. *Proc. Natl. Acad. Sci.* (2007). doi:10.1073/pnas.0707476104
147. Chubukov, V., Gerosa, L., Kochanowski, K. & Sauer, U. Coordination of microbial metabolism. *Nature Reviews Microbiology* (2014). doi:10.1038/nrmicro3238
148. Slavov, N. & Botstein, D. Decoupling nutrient signaling from growth rate causes aerobic glycolysis and deregulation of cell size and gene expression. *Mol. Biol. Cell* (2013). doi:10.1091/mbc.E12-09-0670
149. Kuepfer, L., Peter, M., Sauer, U. & Stelling, J. Ensemble modeling for analysis of cell signaling dynamics. *Nat. Biotechnol.* **25,** 1001–6 (2007).

APPENDICES

APPENDIX A

Table A.1: Reaction constraints imposed on iJO1366 when determining WT reference flux distribution required for MOMA predictions. Values taken from 13C fluxomic measurements performed by Haverkorn et al.[65]

	Reaction	Reaction Description	Flux Value	95% CI
1	EX_glc(e)	D-Glucose exchange	-8.26	0
2	EX_ac(e)	Acetate exchange	4.89	0
3	GLCptspp	D-glucose transport via PEP:Pyr PTS (periplasm)	8.13	0.34
4	G6PDH2r	glucose 6-phosphate dehydrogenase	2.39	0.3
5	PGL	6-phosphogluconolactonase	1.65	0.44
6	GND	phosphogluconate dehydrogenase	1.65	0.44
7	PGI	glucose-6-phosphate isomerase	5.71	0.39
8	EDD	6-phosphogluconate dehydratase	0.74	0.65
9	PFK	phosphofructokinase	6.46	0.62
10	FBA	fructose-bisphosphate aldolase	6.46	0.62
11	TPI	triose-phosphate isomerase	6.46	0.62
12	TKT1	transketolase	0.53	0.15
13	TKT2	transketolase	0.27	0.15
14	TALA	transaldolase	0.53	0.15
15	GAPD	glyceraldehyde-3-phosphate dehydrogenase	13.87	0.84
16	PGK	phosphoglycerate kinase	-13.87	0.84
17	PGM	phosphoglycerate mutase	-12.94	0.84
18	ENO	enolase	12.94	0.84
19	PYK	pyruvate kinase	9.99	0.98
20	PDH	pyruvate dehydrogenase	9.14	0.64
21	CS	citrate synthase	2.2	0.45
22	ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	2.2	0.45
23	ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	2.2	0.45
24	ICDHyr	isocitrate dehydrogenase (NADP)	2.2	0.45
25	AKGDH	2-Oxogluterate dehydrogenase	1.29	0.44
26	SUCOAS	succinyl-CoA synthetase (ADP-forming)	-1.29	0.44
27	FUM	fumarase	1.29	0.44
28	MDH	malate dehydrogenase	0.81	0.5
29	ICL	Isocitrate lyase	0	0
30	MALS	malate synthase	0	0
31	PPCK	phosphoenolpyruvate carboxykinase	0.23	0.26
32	PPC	phosphoenolpyruvate carboxylase	2.64	0.78
33	PTAr	phosphotransacetylase	5.48	0.57

APPENDIX B

Table B.1: Growth outcomes for candidate essential auxotroph knockout strains grown on EZ Rich vs Minimal Media.

			Experimental Results	
	Knockout	**Model Predicted Media Supplements**	**EZ Rich Media**	**Minimal Media**
1	**ArgA**	arginine	☑	☒
2	**AroA**	phenylalanine && tyrosine && tryptophan	☑	☒
3	**AroB**	quinate || shikimate	☑	☒
4	**AroC**	phenylalanine && tyrosine && tryptophan && histidine	☑	☒
5	**AroEec**	shikimate	☑	☒
6	**BioC**	biotin	☑	☒
7	**CysC**	cysteine	☑	☒
8	**CysD**	cysteine	☑	☒
9	**CysI or CysJ**	cysteine	☑	☒
10	**CysQ**	cysteine	☑	☒
11	**GlnA**	glutamine	☑	☒
12	**GltA**	arginine || glutamine || glutamate || proline	☑	☒
13	**GuaA**	guanine	☑	☒
14	**HisA**	histidine	☑	☒
15	**Icd**	arginine || glutamine || glutamate || proline	☑	☒
16	**IcdC**	Note: this knockout was used as non-essential control	☑	☑
17	**IlvA**	isoleucine	☑	☒
18	**LeuB**	leucine	☑	☒
19	**LysA**	lysine	☑	☒
20	**MetA**	methionine	☑	☒
21	**MetF**	S-Methyl-L-methionine	☑	☒
22	**PanB**	pantothenate	☑	☒
23	**PanCec**	pantothenate	☑	☒
24	**PdxAJ**	pyridoxine	☒	☒
25	**PdxB**	pyridoxine	☒	☒
26	**ProC**	proline	☑	☒
27	**PurE**	guanine || adenine	☑	☒
28	**PyrC**	cytidine	☑	☒
29	**SerC**	pyridoxine	☒	☒
30	**ThrB**	threonine	☑	☒

Note: Knockout strains requiring pyridoxine did not grow under EZ rich media[66,67] because the media did not contain pyridoxine. This result indicates a positive prediction of the appropriate auxotroph metabolite. See Appendix C for further confirmation.

APPENDIX C

Table C.1: Predicted and Confirmed Auxotrophic Supplements by Strain

	KO	Model Predicted Supplements	Confirmed Supplements*
1	ArgA	arginine || ornithine	arginine
2	AroA	phenylalanine && tyrosine && tryptophan	phe&&tyr&&trp
3	AroB	quinate || shikimate	shikimate
4	AroC	phenylalanine && tyrosine && tryptophan && histidine	Phe/Tyr/Trp/AA (not His)
5	AroEec	shikimate	shikimate
6	BioC	biotin	biotin
7	CysC	cysteine || taurine || glutathione	cysteine, glutathione, taurine
8	CysD	cysteine || taurine || glutathione	cysteine, glutathione, taurine
9	CysI or CysJ	cysteine || glutathione	cysteine, glutathione
10	CysQ	cysteine || taurine || glutathione	cysteine, glutathione, taurine
11	GlnA	glutamine	glutamine
12	GltA	arginine || glutamine || glutamate || proline || 2-oxoglutarate || citrate || glutathione || ornithine	glu, gln, pro, combination, glutathione
13	GuaA	guanine || guanosine	guanosine
14	HisA	histidine	histidine
15	Icd	arginine || glutamine || glutamate || proline || 2-oxoglutarate || glutathione || ornithine	arginine || glutamine || glutamate || proline || glutathione
16	IcdC	Note: incorrect Keio strain obtained so this knockout was used as non-essential control	n/a
17	IlvA	isoleucine	isoleucine
18	LeuB	leucine	leucine
19	LysA	lysine	lysine
20	MetA	methionine	methionine
21	MetF	S-Methyl-L-methionine	methionine
22	PanB	pantothenate	pantothenate
23	PanCec	pantothenate	pantothenate
24	PdxAJ	pyridoxine || pyridoxal || pyridoxamine	pyridoxine, pyridoxamine
25	PdxB	pyridoxine || pyridoxal || pyridoxamine	pyridoxine, pyridoxamine
26	ProC	proline	proline
27	PurE	adenosine || adenine || guanosine || guanine || inosine || xanthine	adenine, adenosine, guanosine
28	PyrC	cytidine || cytosine || orotate || uracil || uridine	uracil, uridine, cytidine
29	SerC	pyridoxine || pyridoxal || pyridoxamine	pyridoxine
30	ThrB	threonine	threonine

* Note: not all predicted supplements were tested for each knockout but at least one supplement was confirmed for each candidate.

APPENDIX D

Table D.1: Predicted Auxotrophic Knockouts and Metabolite(s) Needed to Recover Growth

#	Rxn #	Rxn	Rxn Description	MOMA Predicted Recovery Exchange Metabolites
1	460	5DOAN	5-deoxyadenosine nuclosidase	val-L[e]
2	509	ACGK	acetylglutamate kinase	arg-L[e], orn[e]
3	510	ACGS	N-acetylglutamate synthase	arg-L[e], orn[e]
4	511	ACHBS	2-aceto-2-hydroxybutanoate synthase	ile-L[e]
5	513	ACLS	acetolactate synthase	val-L[e]
6	532	ACODA	acetylornithine deacetylase	arg-L[e], orn[e]
7	536	ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23cgmp[e], 2ddglcn[e], 34dhpac[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], acac[e], acgal[e], acgam1p[e], acmum[e], acolipa[e], acser[e], adocbl[e], ag[e], akg[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], arbt[e], arbtn-fe3[e], arg-L[e], asn-L[e], aso3[e], butso3[e], ca2[e], cbi[e], chol[e], cl[e], co2[e], cobalt2[e], colipap[e], crn[e], cu[e], cu2[e], cyan[e], cys-D[e], damp[e], dgmp[e], dgsn[e], dms[e], duri[e], eca4colipa[e], enlipa[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dhbzs[e], fe3hox-un[e], fecrm-un[e], feoxam[e], fru[e], fum[e], g1p[e], g3pe[e], g3pg[e], g6p[e], galct-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], gln-L[e], glu-L[e], glyb[e], glyclt[e], gthox[e], gthrd[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], k[e], lac-D[e], leu-L[e], lipa[e], lipoate[e], mal-L[e], malttr[e], man6p[e], melib[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mmet[e], mn2[e], mnl[e], mobd[e], n2o[e], na1[e], nh4[e], ni2[e], no[e], o16a4colipa[e], o2[e], o2s[e], orn[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], pro-L[e], progly[e], pser-L[e], r5p[e], rib-D[e], sbt-D[e], sel[e], slnt[e], so2[e], so3[e], so4[e], succ[e], sucr[e], taur[e], tcynt[e], thrp[e], thym[e], thymd[e], tma[e], tsul[e], tungs[e], udpacgal[e], udpg[e], udpglcur[e], urea[e], uri[e], zn2[e]

8	537	ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23cgmp[e], 2ddglcn[e], 34dhpac[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], acac[e], acgal[e], acgam1p[e], acmum[e], acolipa[e], acser[e], adocbl[e], ag[e], akg[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], arbt[e], arbtn-fe3[e], arg-L[e], asn-L[e], aso3[e], butso3[e], ca2[e], cbi[e], chol[e], cl[e], co2[e], cobalt2[e], colipap[e], crn[e], cu[e], cu2[e], cyan[e], cys-D[e], damp[e], dgmp[e], dgsn[e], dms[e], duri[e], eca4colipa[e], enlipa[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dhbzs[e], fe3hox-un[e], fecrm-un[e], feoxam[e], fru[e], fum[e], g1p[e], g3pe[e], g3pg[e], g6p[e], galct-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], gln-L[e], glu-L[e], glyb[e], glyclt[e], gthox[e], gthrd[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], k[e], lac-D[e], leu-L[e], lipa[e], lipoate[e], mal-L[e], malttr[e], man6p[e], melib[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mmet[e], mn2[e], mnl[e], mobd[e], n2o[e], na1[e], nh4[e], ni2[e], no[e], o16a4colipa[e], o2[e], o2s[e], orn[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], pro-L[e], progly[e], pser-L[e], r5p[e], rib-D[e], sbt-D[e], sel[e], slnt[e], so2[e], so3[e], so4[e], succ[e], sucr[e], taur[e], tcynt[e], thrp[e], thym[e], thymd[e], tma[e], tsul[e], tungs[e], udpacgal[e], udpg[e], udpglcur[e], urea[e], uri[e], zn2[e]
9	538	ACOTA	acetylornithine transaminase	arg-L[e], orn[e]
10	554	ADCL	4-aminobenzoate synthase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
11	555	ADCS	4-amino-4-deoxychorismate synthase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]

12	575	ADSK	adenylyl-sulfate kinase	ac[e], alltn[e], arab-L[e], arbtn-fe3[e], butso3[e], cgly[e], cpgn[e], cu[e], cys-D[e], cys-L[e], damp[e], dcmp[e], dgsn[e], dopa[e], enter[e], ethso3[e], frulys[e], g3pi[e], gal-bD[e], glu-L[e], gthrd[e], gtp[e], hdcea[e], ile-L[e], isetac[e], lac-D[e], lcts[e], leu-L[e], lipoate[e], mal-D[e], mal-L[e], malttr[e], man[e], melib[e], mso3[e], ocdca[e], orot[e], peamn[e], phe-L[e], pro-L[e], succ[e], sulfac[e], taur[e], thm[e], tre[e], ttdcea[e], udpg[e], udpgal[e], xan[e]
13	576	ADSL1r	adenylsuccinate lyase	23camp[e], 3amp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e]
14	577	ADSL2r	adenylosuccinate lyase	23camp[e], 23cgmp[e], 3amp[e], 3gmp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e], dgmp[e], dgsn[e], dimp[e], din[e], gmp[e], gsn[e], gua[e], hxan[e], imp[e], ins[e], xan[e], xmp[e], xtsn[e]
15	578	ADSS	adenylosuccinate synthase	23camp[e], 3amp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e]
16	602	AGPR	N-acetyl-g-glutamyl-phosphate reductase	arg-L[e], orn[e]
17	606	AIRC2	phosphoribosylaminoimidazole carboxylase	23camp[e], 23cgmp[e], 3amp[e], 3gmp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e], dgmp[e], dgsn[e], dimp[e], din[e], gmp[e], gsn[e], gua[e], hxan[e], imp[e], ins[e], xan[e], xmp[e], xtsn[e]
18	607	AIRC3	phosphoribosylaminoimidazole carboxylase (mutase rxn)	23camp[e], 23cgmp[e], 3amp[e], 3gmp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e], dgmp[e], dgsn[e], dimp[e], din[e], gmp[e], gsn[e], gua[e], hxan[e], imp[e], ins[e], xan[e], xmp[e], xtsn[e]
19	616	ALAR	alanine racemase	ala-D[e], alaala[e]
20	656	AMAOTr	adenosylmethionine-8-amino-7-oxononanoate transaminase	btn[e]
21	658	AMPMS2	4-amino-2-methyl-5-phosphomethylpyrimidine synthetase	thm[e]
22	665	ANPRT	anthranilate phosphoribosyltransferase	indole[e], trp-L[e]
23	666	ANS	anthranilate synthase	indole[e], trp-L[e]
24	668	AOXSr2	8-amino-7-oxononanoate synthase	btn[e]

25	688	APRAUR	5-amino-6-(5-phosphoribosylamino)uracil reductase	12ppd-R[e], 12ppd-S[e], 15dap[e], 23camp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn-fe3[e], arg-L[e], ascb-L[e], aso3[e], asp-L[e], but[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], co2[e], cobalt2[e], colipap[e], crn[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], dca[e], dcmp[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dopa[e], doxrbcn[e], dtmp[e], duri[e], eca4colipa[e], enlipa[e], etha[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fecrm[e], feenter[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcr[e], glcur[e], glu-L[e], glyald[e], glyb[e], glyc3p[e], gmp[e], gsn[e], gthox[e], gtp[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], his-L[e], hom-L[e], hxa[e], imp[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lys-L[e], mal-L[e], malt[e], maltttr[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], mg2[e], mincyc[e], minohp[e], mn2[e], mobd[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orot[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppal[e], pppn[e], ppt[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], tartr-L[e], tcynt[e], thm[e], thym[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpgal[e], urea[e], uri[e], val-L[e], xan[e], xtsn[e], xylu-L[e], zn2[e]
26	708	ARGSL	argininosuccinate lyase	arg-L[e]
27	709	ARGSS	argininosuccinate synthase	arg-L[e]
28	714	ASAD	aspartate-semialdehyde dehydrogenase	3hpppn[e], but[e], uri[e]

29	728	ASP1DC	aspartate 1-decarboxylase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e],

				thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
30	729	ASPCT	aspartate carbamoyltransferase	23ccmp[e], 23cump[e], 3cmp[e], 3ump[e], cmp[e], csn[e], cytd[e], dcmp[e], dcyt[e], dump[e], duri[e], orot[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], uri[e]
31	730	ASPK	aspartate kinase	3hpppn[e], but[e], uri[e]
32	736	ASPTA	aspartate transaminase	23cump[e], acmana[e], asn-L[e], asp-L[e], glcur[e], h2[e], lyx-L[e], sulfac[e], tartr-D[e], uacgam[e]
33	749	ATPPRT	ATP phosphoribosyltransferase	his-L[e]
34	758	BPNT	3,5-bisphosphate nucleotidase	ac[e], alltn[e], arab-L[e], arbtn-fe3[e], butso3[e], cgly[e], cpgn[e], cu[e], cys-D[e], cys-L[e], damp[e], dcmp[e], dgsn[e], dopa[e], enter[e], ethso3[e], frulys[e], g3pi[e], gal-bD[e], glu-L[e], gthrd[e], gtp[e], hdcea[e], ile-L[e], isetac[e], lac-D[e], lcts[e], leu-L[e], lipoate[e], mal-D[e], mal-L[e], malttr[e], man[e], melib[e], mso3[e], ocdca[e], orot[e], peamn[e], phe-L[e], pro-L[e], succ[e],

				sulfac[e], taur[e], thm[e], tre[e], ttdcea[e], udpg[e], udpgal[e], xan[e]
35	763	BTS5	Biotin synthase	btn[e]
36	812	CHORS	chorismate synthase	ala-B[e]
37	813	CHRPL	Chorismate pyruvate lyase	23camp[e], 23cump[e], 26dap-M[e], 3ump[e], 5dglcn[e], ade[e], agm[e], ala-D[e], amp[e], butso3[e], csn[e], dcmp[e], ddca[e], dmso[e], fald[e], fe3dhbzs[e], fecrm[e], feoxam-un[e], fruur[e], galctn-L[e], glcur[e], gln-L[e], glyc3p[e], gmp[e], lcts[e], lyx-L[e], maltpt[e], maltttr[e], man6p[e], melib[e], pacald[e], pro-L[e], progly[e], sulfac[e], tartr-L[e], udpacgal[e], urea[e]
38	878	CS	citrate synthase	12ppd-R[e], 23ccmp[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3hcinnm[e], 3hpp[e], 3ump[e], 4hoxpacd[e], 5mtr[e], ac[e], acald[e], acgal[e], acmum[e], acolipa[e], acser[e], adocbl[e], ag[e], akg[e], alaala[e], all-D[e], alltn[e], arbt[e], arg-L[e], asn-L[e], aso3[e], ca2[e], cbi[e], chol[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipap[e], cpgn-un[e], crn[e], cu2[e], cyan[e], cynt[e], cys-D[e], damp[e], dcyt[e], dgmp[e], dms[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], eca4colipa[e], enlipa[e], enter[e], etoh[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], feoxam[e], feoxam-un[e], fruur[e], fum[e], fusa[e], g3pe[e], g3ps[e], gal-bD[e], gal1p[e], galctn-D[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], gln-L[e], glu-L[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc3p[e], gthox[e], gthrd[e], gtp[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], ile-L[e], inost[e], ins[e], k[e], kdo2lipid4[e], leu-L[e], lipa[e], lipa_cold[e], mal-D[e], mal-L[e], malthx[e], man[e], manglyc[e], melib[e], meoh[e], met-D[e], mg2[e], mincyc[e], mn2[e], mnl[e], mobd[e], n2o[e], na1[e], nh4[e], ni2[e], nmn[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], octa[e], orn[e], pheme[e], pi[e], ppa[e], pro-L[e], progly[e], pydx[e], r5p[e], rfamp[e], sbt-D[e], sel[e], ser-D[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], tartr-D[e], tartr-L[e], tcynt[e], thym[e], tma[e], tsul[e], ttdcea[e], ttrcyc[e], tungs[e], udpacgal[e], udpgal[e], urea[e], xtsn[e], xyl-D[e], zn2[e]
39	888	CTPS2	CTP synthase (glutamine)	23ccmp[e], 3cmp[e], cmp[e], cytd[e]

40	906	CYSS	cysteine synthase	12ppd-R[e], 23ccmp[e], 23cump[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpppn[e], 5dglcn[e], acald[e], acgal1p[e], ade[e], agm[e], ala-B[e], ala-L[e], asp-L[e], cgly[e], cm[e], cmp[e], crn-D[e], cys-D[e], cys-L[e], dca[e], dgsn[e], dha[e], dopa[e], doxrbcn[e], fe3hox[e], fecrm-un[e], fru[e], frulys[e], fusa[e], g1p[e], g3pc[e], g3pg[e], g3pi[e], gal1p[e], galt[e], galur[e], glcn[e], glcr[e], glcur[e], glu-L[e], gthrd[e], idon-L[e], ile-L[e], lac-D[e], mincyc[e], nmn[e], novbcn[e], ppt[e], pro-L[e], ptrc[e], rfamp[e], sbt-D[e], sulfac[e], tmao[e], ttdca[e], ttrcyc[e], val-L[e]
41	908	CYSTL	cystathionine b-lyase	23ccmp[e], 23dappa[e], 2ddglcn[e], 3hpppn[e], 5dglcn[e], ac[e], agm[e], arg-L[e], cmp[e], fe3dhbzs[e], glcur1p[e], glu-L[e], gly[e], hdca[e], his-L[e], leu-L[e], malthx[e], maltpt[e], man[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mmet[e], mnl[e], ptrc[e], pyr[e], ser-D[e], thm[e], tyrp[e], udpacgal[e], xylu-L[e]
42	943	DAPDC	diaminopimelate decarboxylase	frulys[e], lys-L[e], psclys[e]
43	955	DB4PS	3,4-Dihydroxy-2-butanone-4-phosphate synthase	12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 26dap-M[e], 34dhpac[e], 3amp[e], 3hpp[e], 3hpppn[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acmana[e], acmum[e], acnam[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arg-L[e], aso3[e], asp-L[e], btn[e], but[e], ca2[e], cbi[e], cgly[e], chol[e], chtbs[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], cu[e], cu2[e], cyan[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgsn[e], dha[e], dimp[e], dms[e], doxrbcn[e], dump[e], duri[e], eca4colipa[e], enlipa[e], etha[e], etoh[e], f6p[e], fe2[e], fe3[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], feoxam[e], fru[e], frulys[e], fusa[e], g3pc[e], g3pg[e], gal-bD[e], gal1p[e], galctn-L[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyb[e], glyc[e], glyc2p[e], gmp[e], gthox[e], gthrd[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hg2[e], his-L[e], hom-L[e], hxa[e], idon-L[e], inost[e], k[e], lac-L[e], lcts[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], man6p[e], manglyc[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], orot[e], pacald[e], peamn[e], pheme[e], pi[e], ppa[e], ppal[e], pppn[e], ppt[e], psclys[e], pydam[e], pyr[e], quin[e], r5p[e],

				rfamp[e], rmn[e], sbt-D[e], sel[e], ser-D[e], slnt[e], so2[e], so3[e], so4[e], succ[e], tartr-L[e], tcynt[e], thm[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], tsul[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], udpacgal[e], udpgal[e], ump[e], ura[e], urea[e], uri[e], xmp[e], xyl-D[e], zn2[e]
44	956	DBTS	dethiobiotin synthase	btn[e]
45	969	DDPA	3-deoxy-D-arabino-heptulosonate 7-phosphate synthetase	man[e], quin[e], skm[e]
46	976	DHAD1	dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylbutanoate)	val-L[e]
47	977	DHAD2	Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylpentanoate)	ile-L[e]
48	989	DHFS	dihydrofolate synthase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
49	994	DHNPA2r	dihydroneopterin aldolase reversible	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e],

				pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
50	999	DHORTS	dihydroorotase	23ccmp[e], 23cump[e], 3cmp[e], 3ump[e], cmp[e], csn[e], cytd[e], dcmp[e], dcyt[e], dump[e], duri[e], orot[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], uri[e]
51	1001	DHPPDA2	diaminohydroxyphosphoribosyl aminopyrimidine deaminase (25drapp)	12ppd-R[e], 12ppd-S[e], 15dap[e], 23camp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn-fe3[e], arg-L[e], ascb-L[e], aso3[e], asp-L[e], but[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], co2[e], cobalt2[e], colipap[e], crn[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], dca[e], dcmp[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dopa[e], doxrbcn[e], dtmp[e], duri[e], eca4colipa[e], enlipa[e], etha[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fecrm[e], feenter[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcr[e], glcur[e], glu-L[e], glyald[e], glyb[e], glyc3p[e], gmp[e], gsn[e], gthox[e], gtp[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], his-L[e], hom-L[e], hxa[e], imp[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lys-L[e], mal-L[e], malt[e], maltttr[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], mg2[e], mincyc[e], minohp[e], mn2[e], mobd[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orot[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppal[e], pppn[e], ppt[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], tartr-L[e], tcynt[e], thm[e], thym[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpgal[e],

				urea[e], uri[e], val-L[e], xan[e], xtsn[e], xylu-L[e], zn2[e]
52	1002	DHPS2	dihydropteroate synthase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
53	1007	DHQS	3-dehydroquinate synthase	man[e], quin[e], skm[e]
54	1008	DHQTi	3-dehydroquinate dehydratase, irreversible	skm[e]
55	1025	DNMPPA	Dihydroneopterin monophosphate dephosphorylase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]

56	1026	DNTPPA	Dihydroneopterin triphosphate pyrophosphatase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
57	1031	DPCOAK	dephospho-CoA kinase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e],

				o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

58	1032	DPR	2-dehydropantoate 2-reductase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e],

				thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
59	1055	E4PD	Erythrose 4-phosphate dehydrogenase	pydam[e], pydx[e], pydxn[e]
60	1100	EGMEACPR	Enoylglutaryl-[ACP] methyl ester reductase	btn[e]
61	1102	ENO	enolase	12ppd-R[e], 2ddglcn[e], 4abut[e], akg[e], ala-D[e], ala-L[e], alaala[e], arg-L[e], asn-L[e], asp-L[e], cbl1[e], chtbs[e], cit[e], dopa[e], fe3dcit[e], fum[e], gln-L[e], glu-L[e], glyc3p[e], glyclt[e], gthrd[e], ile-L[e], lac-L[e], mal-L[e], orn[e], pro-L[e], psclys[e], ptrc[e], succ[e], tartr-D[e], tartr-L[e], thr-L[e], val-L[e]
62	1106	EPMEACPR	Enoylpimeloyl-[ACP] methyl ester reductase	btn[e]

63	1220	FMNAT	FMN adenylyltransferase	23cump[e], 23dappa[e], 3gmp[e], acald[e], acgal1p[e], acmana[e], acnam[e], ade[e], adn[e], adocbl[e], ala-B[e], alaala[e], anhgm[e], arbtn-fe3[e], ascb-L[e], asn-L[e], asp-L[e], butso3[e], cd2[e], chtbs[e], csn[e], cys-L[e], cytd[e], damp[e], dimp[e], dtmp[e], duri[e], ethso3[e], etoh[e], fe3dcit[e], for[e], fruur[e], fum[e], g3pi[e], g3ps[e], g6p[e], galct-D[e], gam6p[e], glu-L[e], glyc[e], glyc-R[e], gmp[e], gtp[e], gua[e], his-L[e], imp[e], indole[e], ins[e], isetac[e], kdo2lipid4[e], lac-L[e], leu-L[e], lipoate[e], lys-L[e], lyx-L[e], mal-L[e], malthx[e], maltpt[e], man[e], man6p[e], mnl[e], mso3[e], nac[e], nmn[e], no[e], no3[e], ocdca[e], orot[e], pacald[e], peamn[e], ppal[e], pro-L[e], psclys[e], pser-L[e], pydam[e], pydx[e], r5p[e], rmn[e], spmd[e], taur[e], thr-L[e], thrp[e], thymd[e], ttdca[e], tym[e], uacgam[e], uri[e], xan[e], xtsn[e], xylu-L[e]
64	1267	G3PD2	glycerol-3-phosphate dehydrogenase (NADP)	g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], glyald[e], glyc[e], glyc2p[e], glyc3p[e]
65	1280	G5SADs	L-glutamate 5-semialdehyde dehydratase (spontaneous)	pro-L[e], progly[e]
66	1315	GAPD	glyceraldehyde-3-phosphate dehydrogenase	etoh[e]
67	1319	GCALDD	Glycolaldehyde dehydrogenase	23camp[e], 26dap-M[e], 3cmp[e], 3ump[e], adocbl[e], ala-L[e], anhgm[e], arbtn-fe3[e], asn-L[e], asp-L[e], crn-D[e], cytd[e], dad-2[e], dha[e], etha[e], fum[e], g3pg[e], galct-D[e], galt[e], gly[e], glyc[e], gua[e], his-L[e], ins[e], isetac[e], lac-L[e], lipoate[e], lyx-L[e], mal-L[e], malttr[e], maltttr[e], nmn[e], progly[e], pydx[e], r5p[e], spmd[e], succ[e], tre[e], trp-L[e], xan[e], xylu-L[e]
68	1326	GF6PTA	glutamine-fructose-6-phosphate transaminase	acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], anhgm[e], chtbs[e], gam[e], gam6p[e], uacgam[e]
69	1359	GLNS	glutamine synthetase	gln-L[e]
70	1421	GMPS2	GMP synthase	23cgmp[e], 3gmp[e], dgmp[e], dgsn[e], gmp[e], gsn[e], gua[e]
71	1452	GTPCI	GTP cyclohydrolase I	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]

72	1453	GTPCII2	GTP cyclohydrolase II (25drapp)	12ppd-R[e], 12ppd-S[e], 15dap[e], 23camp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn-fe3[e], arg-L[e], ascb-L[e], aso3[e], asp-L[e], but[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], co2[e], cobalt2[e], colipap[e], crn[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], dca[e], dcmp[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dopa[e], doxrbcn[e], dtmp[e], duri[e], eca4colipa[e], enlipa[e], etha[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fecrm[e], feenter[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcr[e], glcur[e], glu-L[e], glyald[e], glyb[e], glyc3p[e], gmp[e], gsn[e], gthox[e], gtp[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], his-L[e], hom-L[e], hxa[e], imp[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lys-L[e], mal-L[e], malt[e], maltttr[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], mg2[e], mincyc[e], minohp[e], mn2[e], mobd[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orot[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppal[e], pppn[e], ppt[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], tartr-L[e], tcynt[e], thm[e], thym[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpgal[e], urea[e], uri[e], val-L[e], xan[e], xtsn[e], xylu-L[e], zn2[e]
73	1484	HBZOPT	Hydroxybenzoate octaprenyltransferase	23camp[e], 23cump[e], 26dap-M[e], 3ump[e], 5dglcn[e], ade[e], agm[e], ala-D[e], amp[e], butso3[e], csn[e], dcmp[e], ddca[e], dmso[e], fald[e], fe3dhbzs[e], fecrm[e], feoxam-un[e], fruur[e], galctn-L[e], glcur[e], gln-L[e], glyc3p[e], gmp[e], lcts[e], lyx-L[e], maltpt[e], maltttr[e], man6p[e], melib[e], pacald[e], pro-L[e], progly[e], sulfac[e], tartr-L[e], udpacgal[e], urea[e]
74	1507	HISTD	histidinol dehydrogenase	his-L[e]
75	1508	HISTP	histidinol-phosphatase	his-L[e]

76	1520	HPPK2	6-hydroxymethyl-dihydropterin pyrophosphokinase	12ppd-S[e], 3ump[e], 5dglcn[e], acgal1p[e], acgam1p[e], ade[e], arab-L[e], arg-L[e], btn[e], cpgn[e], cytd[e], dha[e], din[e], fe3hox-un[e], fum[e], g3pe[e], g3ps[e], gam[e], gam6p[e], glcr[e], glcur[e], h2[e], his-L[e], ins[e], malt[e], malthx[e], man[e], nac[e], no2[e], octa[e], pacald[e], ppa[e], ppal[e], ptrc[e], pydx[e], r5p[e], taur[e], thymd[e], ttdcea[e], tyr-L[e], uacgam[e], udpglcur[e]
77	1528	HSK	homoserine kinase	thr-L[e], thrp[e]
78	1529	HSST	homoserine O-succinyltransferase	23ccmp[e], 23dappa[e], 2ddglcn[e], 3hpppn[e], 5dglcn[e], ac[e], agm[e], arg-L[e], cmp[e], fe3dhbzs[e], glcur1p[e], glu-L[e], gly[e], hdca[e], his-L[e], leu-L[e], malthx[e], maltpt[e], man[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mmet[e], mnl[e], ptrc[e], pyr[e], ser-D[e], thm[e], tyrp[e], udpacgal[e], xylu-L[e]
79	1530	HSTPT	histidinol-phosphate transaminase	his-L[e]
80	1548	ICDHyr	isocitrate dehydrogenase (NADP)	12ppd-R[e], 15dap[e], 23cgmp[e], 34dhpac[e], 3amp[e], 3hpp[e], 3ump[e], 4hoxpacd[e], 5mtr[e], ac[e], acgal[e], acgam1p[e], acmum[e], acolipa[e], acser[e], ade[e], ag[e], akg[e], alaala[e], anhgm[e], arbt[e], arg-L[e], aso3[e], ca2[e], cbi[e], cbl1[e], chol[e], chtbs[e], cl[e], cm[e], co2[e], cobalt2[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], cu2[e], cyan[e], dad-2[e], damp[e], ddca[e], dgsn[e], dha[e], dimp[e], dms[e], doxrbcn[e], eca4colipa[e], enlipa[e], fe2[e], fe3[e], feenter[e], feoxam-un[e], fruur[e], fum[e], fusa[e], g3pc[e], g3pe[e], galur[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], gln-L[e], glu-L[e], glyb[e], glyc-R[e], glyclt[e], gsn[e], gthox[e], gthrd[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], hom-L[e], hxan[e], idon-L[e], inost[e], isetac[e], k[e], lac-D[e], lcts[e], lipa[e], lipa_cold[e], lys-L[e], lyx-L[e], mal-L[e], meoh[e], met-D[e], mg2[e], mincyc[e], mn2[e], mobd[e], na1[e], nh4[e], ni2[e], no[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], orn[e], pheme[e], pi[e], ppa[e], pro-L[e], progly[e], quin[e], rfamp[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], taur[e], tcynt[e], thym[e], tma[e], tmao[e], trp-L[e], tsul[e], ttrcyc[e], tungs[e], uacgam[e], udpacgal[e], zn2[e]
81	1558	IG3PS	Imidazole-glycerol-3-phosphate synthase	his-L[e]
82	1559	IGPDH	imidazoleglycerol-phosphate dehydratase	his-L[e]
83	1560	IGPS	indole-3-glycerol-phosphate synthase	indole[e], trp-L[e]
84	1561	ILETA	isoleucine transaminase	ile-L[e]
85	1581	IPMD	3-isopropylmalate dehydrogenase	leu-L[e]

86	1582	IPPMIa	3-isopropylmalate dehydratase	leu-L[e]
87	1583	IPPMIb	2-isopropylmalate hydratase	leu-L[e]
88	1584	IPPS	2-isopropylmalate synthase	leu-L[e]
89	1589	KARA1	ketol-acid reductoisomerase (2,3-dihydroxy-3-methylbutanoate)	val-L[e]
90	1590	KARA2	ketol-acid reductoisomerase (2-Acetolactate)	ile-L[e]
91	1624	LEUTAi	leucine transaminase (irreversible)	leu-L[e]
92	1710	MALCOAMT	Malonyl-CoA methyltransferase	btn[e]
93	1771	METAT	methionine adenosyltransferase	3hcinnm[e], acgam1p[e], acmana[e], ddca[e], dmso[e], dopa[e], g3pc[e], glcur1p[e], glyc-R[e], ile-L[e], indole[e], nac[e], succ[e], tre[e], uacgam[e]
94	1776	METS	methionine synthase	mmet[e]

95	1836	MOHMT	3-methyl-2-oxobutanoate hydroxymethyltransferase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e],

				thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
96	1849	MTHFR2	5,10-methylenetetrahydrofolate reductase (NADH)	mmet[e]

97	1870	NADS1	NAD synthase (nh3)	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-

				D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
98	1886	NH4tpp	ammonia reversible transport (periplasm)	23dappa[e], 3amp[e], 3gmp[e], ade[e], adn[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], arg-L[e], asn-L[e], asp-L[e], cgly[e], csn[e], cytd[e], dad-2[e], dgmp[e], dgsn[e], etha[e], gln-L[e], glu-L[e], gly[e], gmp[e], gsn[e], gthrd[e], gua[e], hxan[e], ins[e], orn[e], progly[e], ser-D[e], ser-L[e], thr-L[e], xan[e], xtsn[e]

99	1900	NNATr	nicotinate-nucleotide adenylyltransferase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-

				D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

100	1902	NNDPR	nicotinate-nucleotide diphosphorylase (carboxylating)	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-

				D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
101	1979	OCBT	ornithine carbamoyltransferase	arg-L[e]
102	1983	OCTDPS	Octaprenyl pyrophosphate synthase	12ppd-S[e], 3hpppn[e], acgal1p[e], alaala[e], butso3[e], cd2[e], cpgn-un[e], cynt[e], dad-2[e], dca[e], dha[e], enter[e], fald[e], fe3dhbzs[e], fe3hox[e], glcur1p[e], glu-L[e], glyc-R[e], his-L[e], hxan[e], lac-D[e], lac-L[e], leu-L[e], man6p[e], metsox-R-L[e], metsox-S-L[e], mso3[e], no[e], orn[e], pydxn[e], pyr[e], thr-L[e], tyrp[e], uri[e]
103	1985	OGMEACPD	3-Oxo-glutaryl-[ACP] methyl ester dehydratase	btn[e]
104	1986	OGMEACPR	3-Oxo-glutaryl-[ACP] methyl ester reductase	btn[e]
105	1987	OGMEACPS	3-Oxo-glutaryl-[ACP] methyl ester synthase	btn[e]
106	1988	OHPBAT	O-Phospho-4-hydroxy-L-threonine:2-oxoglutarate aminotransferase	pydam[e], pydx[e], pydxn[e]

107	1991	OMCDC	2-Oxo-4-methyl-3-carboxypentanoate decarboxylation	leu-L[e]
108	1994	OMPDC	orotidine-5-phosphate decarboxylase	23ccmp[e], 23cump[e], 3cmp[e], 3ump[e], cmp[e], csn[e], cytd[e], dcmp[e], dcyt[e], dump[e], duri[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], uri[e]
109	1998	OPHBDC	Octaprenyl-hydroxybenzoate decarboxylase	23camp[e], 23cump[e], 26dap-M[e], 3ump[e], 5dglcn[e], ade[e], agm[e], ala-D[e], amp[e], butso3[e], csn[e], dcmp[e], ddca[e], dmso[e], fald[e], fe3dhbzs[e], fecrm[e], feoxam-un[e], fruur[e], galctn-L[e], glcur[e], gln-L[e], glyc3p[e], gmp[e], lcts[e], lyx-L[e], maltpt[e], maltttr[e], man6p[e], melib[e], pacald[e], pro-L[e], progly[e], sulfac[e], tartr-L[e], udpacgal[e], urea[e]
110	2001	OPMEACPD	3-Oxo-pimeloyl-[ACP] methyl ester dehydratase	btn[e]
111	2002	OPMEACPR	3-Oxo-pimeloyl-[ACP] methyl ester reductase	btn[e]
112	2003	OPMEACPS	3-Oxo-pimeloyl-[ACP] methyl ester synthase	btn[e]
113	2009	ORPT	orotate phosphoribosyltransferase	23ccmp[e], 23cump[e], 3cmp[e], 3ump[e], cmp[e], csn[e], cytd[e], dcmp[e], dcyt[e], dump[e], duri[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], uri[e]
114	2015	P5CR	pyrroline-5-carboxylate reductase	pro-L[e], progly[e]

115	2027	PANTS	pantothenate synthase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e],

				thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
116	2050	PDX5PS	Pyridoxine 5-phosphate synthase	pydam[e], pydx[e], pydxn[e]
117	2061	PERD	Erythronate 4-phosphate (4per) dehydrogenase	pydam[e], pydx[e], pydxn[e]
118	2078	PGK	phosphoglycerate kinase	etoh[e]
119	2081	PGM	phosphoglycerate mutase	3ump[e], 4abut[e], akg[e], ala-L[e], asn-L[e], asp-L[e], cit[e], fe3dcit[e], galct-D[e], glcr[e], gln-L[e], glu-L[e], glyc-R[e], gthrd[e], lipoate[e], mal-L[e], orn[e], pro-L[e], sbt-D[e], tartr-D[e]
120	2113	PHETA1	phenylalanine transaminase	phe-L[e]

121	2164	PMDPHT	pyrimidine phosphatase	12ppd-R[e], 12ppd-S[e], 15dap[e], 23camp[e], 23cump[e], 23dappa[e], 2ddglcn[e], 34dhpac[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn-fe3[e], arg-L[e], ascb-L[e], aso3[e], asp-L[e], but[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], co2[e], cobalt2[e], colipap[e], crn[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], dca[e], dcmp[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dopa[e], doxrbcn[e], dtmp[e], duri[e], eca4colipa[e], enlipa[e], etha[e], ethso3[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fecrm[e], feenter[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galt[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcr[e], glcur[e], glu-L[e], glyald[e], glyb[e], glyc3p[e], gmp[e], gsn[e], gthox[e], gtp[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], his-L[e], hom-L[e], hxa[e], imp[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lys-L[e], mal-L[e], malt[e], maltttr[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], mg2[e], mincyc[e], minohp[e], mn2[e], mobd[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orot[e], pacald[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppal[e], pppn[e], ppt[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], tartr-L[e], tcynt[e], thm[e], thym[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpgal[e], urea[e], uri[e], val-L[e], xan[e], xtsn[e], xylu-L[e], zn2[e]
122	2165	PMEACPE	Pimeloyl-[ACP] methyl ester esterase	btn[e]
123	2166	PMPK	phosphomethylpyrimidine kinase	thm[e]

124	2167	PNTK	pantothenate kinase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e],

				thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

125	2182	PPCDC	phosphopantothenoylcysteine decarboxylase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e],

				thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

126	2190	PPNCL2	phosphopantothenate-cysteine ligase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e],

				thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]
127	2191	PPND	prephenate dehydrogenase	tyr-L[e], tyrp[e]
128	2192	PPNDH	prephenate dehydratase	phe-L[e]
129	2203	PRAIi	phosphoribosylanthranilate isomerase (irreversible)	indole[e], trp-L[e]
130	2204	PRAMPC	phosphoribosyl-AMP cyclohydrolase	his-L[e]
131	2205	PRASCSi	phosphoribosylaminoimidazole succinocarboxamide synthase	23camp[e], 23cgmp[e], 3amp[e], 3gmp[e], ade[e], adn[e], amp[e], dad-2[e], damp[e], dgmp[e], dgsn[e], dimp[e], din[e], gmp[e], gsn[e], gua[e], hxan[e], imp[e], ins[e], xan[e], xmp[e], xtsn[e]
132	2206	PRATPP	phosphoribosyl-ATP pyrophosphatase	his-L[e]
133	2208	PRMICI	1-(5-phosphoribosyl)-5-[(5-phosphoribosylamino)methylideneamino)imidazole-4-carboxamide isomerase	his-L[e]

134	2220	PSCVT	3-phosphoshikimate 1-carboxyvinyltransferase	ala-B[e]
135	2242	PTPATi	pantetheine-phosphate adenylyltransferase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e],

				so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

136	2276	QULNS	quinolinate synthase	12ppd-R[e], 12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23cump[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3cmp[e], 3gmp[e], 3hcinnm[e], 3hpp[e], 3hpppn[e], 3ump[e], 4abut[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], ac[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], acnam[e], acolipa[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], alaala[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], ascb-L[e], asn-L[e], aso3[e], asp-L[e], btn[e], but[e], butso3[e], ca2[e], cbi[e], cbl1[e], cd2[e], cgly[e], chol[e], chtbs[e], cit[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], csn[e], cu[e], cu2[e], cyan[e], cynt[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], dcyt[e], ddca[e], dgmp[e], dgsn[e], dha[e], dimp[e], din[e], dms[e], dmso[e], dopa[e], doxrbcn[e], dtmp[e], dump[e], duri[e], eca4colipa[e], enlipa[e], enter[e], etha[e], ethso3[e], etoh[e], f6p[e], fald[e], fe2[e], fe3[e], fe3dcit[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], fecrm[e], fecrm-un[e], feenter[e], feoxam[e], feoxam-un[e], for[e], fru[e], frulys[e], fruur[e], fuc-L[e], fum[e], fusa[e], g1p[e], g3pc[e], g3pe[e], g3pg[e], g3pi[e], g3ps[e], g6p[e], gal[e], gal-bD[e], gal1p[e], galct-D[e], galctn-D[e], galctn-L[e], galt[e], galur[e], gam[e], gam6p[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcr[e], glcur[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyald[e], glyb[e], glyc[e], glyc-R[e], glyc2p[e], glyc3p[e], glyclt[e], gmp[e], gsn[e], gthox[e], gthrd[e], gtp[e], gua[e], h[e], h2[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hdcea[e], hg2[e], his-L[e], hom-L[e], hxa[e], hxan[e], idon-L[e], ile-L[e], imp[e], indole[e], inost[e], ins[e], isetac[e], k[e], kdo2lipid4[e], lac-D[e], lac-L[e], lcts[e], leu-L[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], lyx-L[e], mal-D[e], mal-L[e], malt[e], malthx[e], maltpt[e], malttr[e], maltttr[e], man[e], man6p[e], manglyc[e], melib[e], meoh[e], met-D[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], minohp[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no[e], no2[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], octa[e], orn[e], orot[e], pacald[e], peamn[e], phe-L[e], pheme[e], pi[e], pnto-R[e], ppa[e], ppal[e], pppn[e], ppt[e], pro-L[e], progly[e], psclys[e], pser-L[e], ptrc[e], pydam[e], pydx[e], pydxn[e], pyr[e], quin[e], r5p[e], rfamp[e], rib-D[e], rmn[e], sbt-D[e], sel[e], ser-D[e], ser-L[e], skm[e], slnt[e], so2[e], so3[e], so4[e], spmd[e], succ[e], sucr[e], sulfac[e], tartr-

				D[e], tartr-L[e], taur[e], tcynt[e], thm[e], thr-L[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], trp-L[e], tsul[e], ttdca[e], ttdcea[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], uacgam[e], udpacgal[e], udpg[e], udpgal[e], udpglcur[e], ump[e], ura[e], urea[e], uri[e], val-L[e], xan[e], xmp[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

137	2282	RBFK	riboflavin kinase	12ppd-S[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 23dappa[e], 26dap-M[e], 2ddglcn[e], 34dhpac[e], 3amp[e], 3hpp[e], 4hoxpacd[e], 5mtr[e], ac[e], acgal[e], acgal1p[e], acmana[e], acmum[e], acolipa[e], acser[e], ag[e], akg[e], amp[e], arab-L[e], arbt[e], arbtn[e], arbtn-fe3[e], arg-L[e], aso3[e], but[e], butso3[e], ca2[e], cbi[e], chol[e], cl[e], cmp[e], co2[e], cobalt2[e], colipap[e], crn[e], cu2[e], cyan[e], cytd[e], dad-2[e], dgmp[e], dms[e], dmso[e], dopa[e], dump[e], eca4colipa[e], enlipa[e], fe2[e], fe3[e], fe3dcit[e], fru[e], g3pe[e], g3pi[e], g6p[e], galt[e], gbbtn[e], gdp[e], glc-D[e], glu-L[e], gly[e], glyb[e], glyc-R[e], gthox[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hg2[e], hom-L[e], hxa[e], inost[e], isetac[e], k[e], lipa[e], lipa_cold[e], mal-D[e], maltpt[e], man[e], man6p[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mmet[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], nmn[e], no2[e], o16a4colipa[e], o2[e], o2s[e], orn[e], pacald[e], pheme[e], pi[e], pppn[e], ppt[e], pro-L[e], psclys[e], pydx[e], rmn[e], sel[e], slnt[e], so2[e], so3[e], so4[e], sucr[e], tartr-L[e], tcynt[e], thym[e], tma[e], tsul[e], ttdca[e], tungs[e], udpacgal[e], urea[e], xtsn[e], xyl-D[e], xylu-L[e], zn2[e]

138	2283	RBFSa	riboflavin synthase	12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 26dap-M[e], 34dhpac[e], 3amp[e], 3hpp[e], 3hpppn[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acmana[e], acmum[e], acnam[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arg-L[e], aso3[e], asp-L[e], btn[e], but[e], ca2[e], cbi[e], cgly[e], chol[e], chtbs[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], cu[e], cu2[e], cyan[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgsn[e], dha[e], dimp[e], dms[e], doxrbcn[e], dump[e], duri[e], eca4colipa[e], enlipa[e], etha[e], etoh[e], f6p[e], fe2[e], fe3[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], feoxam[e], fru[e], frulys[e], fusa[e], g3pc[e], g3pg[e], gal-bD[e], gal1p[e], galctn-L[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyb[e], glyc[e], glyc2p[e], gmp[e], gthox[e], gthrd[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hg2[e], his-L[e], hom-L[e], hxa[e], idon-L[e], inost[e], k[e], lac-L[e], lcts[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], man6p[e], manglyc[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], orot[e], pacald[e], peamn[e], pheme[e], pi[e], ppa[e], ppal[e], pppn[e], ppt[e], psclys[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sbt-D[e], sel[e], ser-D[e], slnt[e], so2[e], so3[e], so4[e], succ[e], tartr-L[e], tcynt[e], thm[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], tsul[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], udpacgal[e], udpgal[e], ump[e], ura[e], urea[e], uri[e], xmp[e], xyl-D[e], zn2[e]

139	2284	RBFSb	riboflavin synthase	12ppd-S[e], 14glucan[e], 15dap[e], 23camp[e], 23ccmp[e], 23cgmp[e], 26dap-M[e], 34dhpac[e], 3amp[e], 3hpp[e], 3hpppn[e], 4hoxpacd[e], 5dglcn[e], 5mtr[e], acac[e], acald[e], acgal[e], acgal1p[e], acgam[e], acmana[e], acmum[e], acnam[e], acser[e], ade[e], adn[e], adocbl[e], ag[e], agm[e], akg[e], ala-B[e], ala-D[e], ala-L[e], all-D[e], alltn[e], amp[e], anhgm[e], arab-L[e], arbt[e], arg-L[e], aso3[e], asp-L[e], btn[e], but[e], ca2[e], cbi[e], cgly[e], chol[e], chtbs[e], cl[e], cm[e], cmp[e], co2[e], cobalt2[e], colipa[e], colipap[e], cpgn[e], cpgn-un[e], crn[e], crn-D[e], cu[e], cu2[e], cyan[e], cys-D[e], cys-L[e], cytd[e], dad-2[e], damp[e], dca[e], dcmp[e], ddca[e], dgsn[e], dha[e], dimp[e], dms[e], doxrbcn[e], dump[e], duri[e], eca4colipa[e], enlipa[e], etha[e], etoh[e], f6p[e], fe2[e], fe3[e], fe3dhbzs[e], fe3hox[e], fe3hox-un[e], feoxam[e], fru[e], frulys[e], fusa[e], g3pc[e], g3pg[e], gal-bD[e], gal1p[e], galctn-L[e], gbbtn[e], gdp[e], glc-D[e], glcn[e], glcur1p[e], gln-L[e], glu-L[e], gly[e], glyb[e], glyc[e], glyc2p[e], gmp[e], gthox[e], gthrd[e], h[e], h2o[e], h2o2[e], h2s[e], hacolipa[e], halipa[e], hdca[e], hg2[e], his-L[e], hom-L[e], hxa[e], idon-L[e], inost[e], k[e], lac-L[e], lcts[e], lipa[e], lipa_cold[e], lipoate[e], lys-L[e], man6p[e], manglyc[e], meoh[e], met-D[e], metsox-R-L[e], metsox-S-L[e], mg2[e], mincyc[e], mn2[e], mnl[e], mobd[e], mso3[e], n2o[e], na1[e], nac[e], nh4[e], ni2[e], no[e], no3[e], novbcn[e], o16a4colipa[e], o2[e], o2s[e], ocdca[e], ocdcea[e], orot[e], pacald[e], peamn[e], pheme[e], pi[e], ppa[e], ppal[e], pppn[e], ppt[e], psclys[e], pydam[e], pyr[e], quin[e], r5p[e], rfamp[e], rmn[e], sbt-D[e], sel[e], ser-D[e], slnt[e], so2[e], so3[e], so4[e], succ[e], tartr-L[e], tcynt[e], thm[e], thrp[e], thym[e], thymd[e], tma[e], tmao[e], tre[e], tsul[e], ttrcyc[e], tungs[e], tym[e], tyr-L[e], tyrp[e], udpacgal[e], udpgal[e], ump[e], ura[e], urea[e], uri[e], xmp[e], xyl-D[e], zn2[e]
140	2323	SADT2	Sulfate adenyltransferase	ac[e], alltn[e], arab-L[e], arbtn-fe3[e], butso3[e], cgly[e], cpgn[e], cu[e], cys-D[e], cys-L[e], damp[e], dcmp[e], dgsn[e], dopa[e], enter[e], ethso3[e], frulys[e], g3pi[e], gal-bD[e], glu-L[e], gthrd[e], gtp[e], hdcea[e], ile-L[e], isetac[e], lac-D[e], lcts[e], leu-L[e], lipoate[e], mal-D[e], mal-L[e], malttr[e], man[e], melib[e], mso3[e], ocdca[e], orot[e], peamn[e], phe-L[e], pro-L[e], succ[e], sulfac[e], taur[e], thm[e], tre[e], ttdcea[e], udpg[e], udpgal[e], xan[e]

141	2341	SERAT	serine O-acetyltransferase	3cmp[e], 3gmp[e], acnam[e], ade[e], adn[e], ala-L[e], amp[e], anhgm[e], arg-L[e], cd2[e], cgly[e], cpgn[e], crn-D[e], cys-L[e], dcyt[e], ddca[e], dha[e], dump[e], fru[e], glcn[e], glcur[e], glyc-R[e], gsn[e], gthrd[e], hdcea[e], hxa[e], indole[e], isetac[e], leu-L[e], lipoate[e], mmet[e], nac[e], no2[e], ppal[e], progly[e], pydx[e], pyr[e], sbt-D[e], ttdcea[e], tym[e], tyrp[e], ump[e], val-L[e], xmp[e]
142	2355	SHK3Dr	shikimate dehydrogenase	skm[e]
143	2356	SHKK	shikimate kinase	ala-B[e]
144	2357	SHSL1	O-succinylhomoserine lyase (L-cysteine)	23ccmp[e], 23dappa[e], 2ddglcn[e], 3hpppn[e], 5dglcn[e], ac[e], agm[e], arg-L[e], cmp[e], fe3dhbzs[e], glcur1p[e], glu-L[e], gly[e], hdca[e], his-L[e], leu-L[e], malthx[e], maltpt[e], man[e], met-L[e], metsox-R-L[e], metsox-S-L[e], mmet[e], mnl[e], ptrc[e], pyr[e], ser-D[e], thm[e], tyrp[e], udpacgal[e], xylu-L[e]
145	2394	SULRi	sulfite reductase (NADPH2)	14glucan[e], 23cgmp[e], ac[e], acgal1p[e], acgam[e], acgam1p[e], acmana[e], acmum[e], adocbl[e], agm[e], ala-B[e], asn-L[e], asp-L[e], btn[e], cgly[e], chtbs[e], cm[e], cpgn-un[e], cys-D[e], cys-L[e], damp[e], dca[e], dcyt[e], ddca[e], dgsn[e], doxrbcn[e], dtmp[e], duri[e], fe3dcit[e], feenter[e], frulys[e], fum[e], fusa[e], g3ps[e], glyald[e], glyc-R[e], glyc3p[e], gthrd[e], ins[e], lac-D[e], lipoate[e], malt[e], mincyc[e], novbcn[e], peamn[e], pppn[e], ptrc[e], rfamp[e], sbt-D[e], ttrcyc[e], tyrp[e], ura[e], xtsn[e]
146	2430	THRD_L	L-threonine deaminase	ile-L[e]
147	2432	THRS	threonine synthase	thr-L[e], thrp[e]
148	2441	THZPSN3	thiazole phosphate synthesis	thm[e]
149	2452	TMDS	thymidylate synthase	dtmp[e], thymd[e]
150	2455	TMPPP	thiamine-phosphate diphosphorylase	thm[e]
151	2483	TYRL	tyrosine lyase	thm[e]
152	2487	TYRTA	tyrosine transaminase	tyr-L[e], tyrp[e]
153	2506	UDCPDP	undecaprenyl-diphosphatase	14glucan[e], 23cgmp[e], 3cmp[e], acald[e], acmum[e], ala-B[e], alaala[e], all-D[e], but[e], cmp[e], cys-L[e], dca[e], ddca[e], din[e], duri[e], ethso3[e], fald[e], fecrm[e], g3pc[e], g3pe[e], g3ps[e], glcn[e], glcr[e], glcur[e], gln-L[e], glyc[e], glyc2p[e], gmp[e], hdcea[e], hxan[e], mal-L[e], maltpt[e], manglyc[e], peamn[e], phe-L[e], ppt[e], progly[e], ptrc[e], pydx[e], rmn[e], ser-L[e], taur[e], ura[e], val-L[e], xyl-D[e]

Note: Reaction numbers, reaction names, reaction descriptions, and metabolite names come from the iJO1366 model[12]

APPENDIX E

Table E.1: Essential reactions predicted using Minimization of Metabolic Adjustment method on iJO1366[12,24]

#	iJO1366 Rxn #	iJO1366 Rxn Name
1	1	DM_4CRSOL
2	2	DM_5DRIB
3	4	DM_AMOB
4	5	DM_MTHTHF
5	74	EX_ca2(e)
6	82	EX_cl(e)
7	85	EX_co2(e)
8	86	EX_cobalt2(e)
9	95	EX_cu2(e)
10	187	EX_h2o(e)
11	206	EX_k(e)
12	228	EX_meoh(e)
13	233	EX_mg2(e)
14	237	EX_mn2(e)
15	239	EX_mobd(e)
16	244	EX_nh4(e)
17	245	EX_ni2(e)
18	252	EX_o2(e)
19	263	EX_pi(e)
20	291	EX_so4(e)
21	332	EX_zn2(e)
22	426	3OAR140
23	438	3OAS140
24	460	5DOAN
25	463	A5PISO
26	498	ACCOAC
27	509	ACGK
28	510	ACGS
29	511	ACHBS
30	513	ACLS
31	532	ACODA
32	536	ACONTa
33	537	ACONTb
34	538	ACOTA
35	554	ADCL
36	555	ADCS

37	575	ADSK
38	576	ADSL1r
39	577	ADSL2r
40	578	ADSS
41	598	AGPAT160
42	599	AGPAT161
43	602	AGPR
44	604	AHCYSNS
45	605	AICART
46	606	AIRC2
47	607	AIRC3
48	613	ALAALAr
49	616	ALAR
50	656	AMAOTr
51	658	AMPMS2
52	665	ANPRT
53	666	ANS
54	668	AOXSr2
55	688	APRAUR
56	708	ARGSL
57	709	ARGSS
58	714	ASAD
59	728	ASP1DC
60	729	ASPCT
61	730	ASPK
62	736	ASPTA
63	749	ATPPRT
64	750	ATPS4rpp
65	755	BMOCOS
66	756	BMOGDS1
67	757	BMOGDS2
68	758	BPNT
69	763	BTS5
70	773	CA2tex
71	776	CAt6pp
72	800	CDPMEK
73	811	CHORM
74	812	CHORS
75	813	CHRPL
76	836	CLt3_2pp
77	837	CLtex

78	842	CO2tex
79	843	CO2tpp
80	846	COBALT2tex
81	847	COBALT2tpp
82	861	CPMPS
83	878	CS
84	888	CTPS2
85	892	CU2tex
86	893	CU2tpp
87	906	CYSS
88	908	CYSTL
89	943	DAPDC
90	944	DAPE
91	950	DASYN160
92	951	DASYN161
93	955	DB4PS
94	956	DBTS
95	969	DDPA
96	976	DHAD1
97	977	DHAD2
98	986	DHDPRy
99	987	DHDPS
100	988	DHFR
101	989	DHFS
102	994	DHNPA2r
103	999	DHORTS
104	1001	DHPPDA2
105	1002	DHPS2
106	1003	DHPTDCs2
107	1007	DHQS
108	1008	DHQTi
109	1015	DMATT
110	1025	DNMPPA
111	1026	DNTPPA
112	1031	DPCOAK
113	1032	DPR
114	1043	DTMPK
115	1052	DXPRIi
116	1053	DXPS
117	1055	E4PD
118	1100	EGMEACPR

119	1102	ENO
120	1106	EPMEACPR
121	1157	FCLT
122	1220	FMNAT
123	1251	G1PACT
124	1255	G1SAT
125	1267	G3PD2
126	1280	G5SADs
127	1315	GAPD
128	1319	GCALDD
129	1326	GF6PTA
130	1333	GK1
131	1359	GLNS
132	1371	GLUPRT
133	1372	GLUR
134	1374	GLUTRR
135	1375	GLUTRS
136	1421	GMPS2
137	1437	GRTT
138	1452	GTPCI
139	1453	GTPCII2
140	1468	H2Otex
141	1469	H2Otpp
142	1484	HBZOPT
143	1487	HCO3E
144	1507	HISTD
145	1508	HISTP
146	1515	HMBS
147	1520	HPPK2
148	1527	HSDy
149	1528	HSK
150	1529	HSST
151	1530	HSTPT
152	1548	ICDHyr
153	1553	ICYSDS
154	1558	IG3PS
155	1559	IGPDH
156	1560	IGPS
157	1561	ILETA
158	1566	IMPC
159	1581	IPMD

160	1582	IPPMIa
161	1583	IPPMIb
162	1584	IPPS
163	1587	K2L4Aabcpp
164	1588	K2L4Aabctex
165	1589	KARA1
166	1590	KARA2
167	1593	KDOCT2
168	1594	KDOPP
169	1595	KDOPS
170	1600	Ktex
171	1624	LEUTAi
172	1641	LPADSS
173	1710	MALCOAMT
174	1744	MCOATA
175	1746	MCTP1App
176	1763	MECDPDH5
177	1764	MECDPS
178	1768	MEOHtex
179	1769	MEOHtrpp
180	1770	MEPCT
181	1771	METAT
182	1776	METS
183	1787	MG2tex
184	1826	MNtex
185	1827	MOADSUx
186	1828	MOAT
187	1829	MOAT2
188	1831	MOBDabcpp
189	1832	MOBDtex
190	1834	MOCOS
191	1836	MOHMT
192	1838	MPTAT
193	1839	MPTG
194	1841	MPTS
195	1842	MPTSS
196	1849	MTHFR2
197	1850	MTHTHFSs
198	1870	NADS1
199	1878	NDPK2
200	1880	NDPK4

201	1885	NH4tex
202	1886	NH4tpp
203	1890	NI2tex
204	1900	NNATr
205	1902	NNDPR
206	1975	O2tex
207	1976	O2tpp
208	1979	OCBT
209	1983	OCTDPS
210	1985	OGMEACPD
211	1986	OGMEACPR
212	1987	OGMEACPS
213	1988	OHPBAT
214	1991	OMCDC
215	1994	OMPDC
216	1998	OPHBDC
217	2001	OPMEACPD
218	2002	OPMEACPR
219	2003	OPMEACPS
220	2009	ORPT
221	2015	P5CR
222	2027	PANTS
223	2042	PAPPT3
224	2050	PDX5PS
225	2055	PE160abcpp
226	2056	PE161abcpp
227	2061	PERD
228	2075	PGAMT
229	2078	PGK
230	2081	PGM
231	2113	PHETA1
232	2119	PItex
233	2164	PMDPHT
234	2165	PMEACPE
235	2166	PMPK
236	2167	PNTK
237	2180	PPBNGS
238	2182	PPCDC
239	2190	PPNCL2
240	2191	PPND
241	2192	PPNDH

242	2201	PRAGSr
243	2202	PRAIS
244	2203	PRAIi
245	2204	PRAMPC
246	2205	PRASCSi
247	2206	PRATPP
248	2207	PRFGS
249	2208	PRMICI
250	2220	PSCVT
251	2224	PSD160
252	2225	PSD161
253	2235	PSSA160
254	2236	PSSA161
255	2242	PTPATi
256	2276	QULNS
257	2282	RBFK
258	2283	RBFSa
259	2284	RBFSb
260	2292	RHCCE
261	2323	SADT2
262	2329	SDPDS
263	2330	SDPTA
264	2341	SERAT
265	2353	SHCHD2
266	2354	SHCHF
267	2355	SHK3Dr
268	2356	SHKK
269	2357	SHSL1
270	2366	SO4tex
271	2394	SULRi
272	2414	TDSK
273	2419	THDPS
274	2430	THRD_L
275	2432	THRS
276	2441	THZPSN3
277	2452	TMDS
278	2454	TMPK
279	2455	TMPPP
280	2483	TYRL
281	2487	TYRTA
282	2492	U23GAAT

283	2493	UAAGDS
284	2499	UAGAAT
285	2500	UAGCVT
286	2501	UAGDP
287	2502	UAGPT3
288	2503	UAMAGS
289	2504	UAMAS
290	2505	UAPGR
291	2506	UDCPDP
292	2507	UDCPDPS
293	2524	UGMDDS
294	2525	UHGADA
295	2531	UMPK
296	2535	UPP3MT
297	2536	UPP3S
298	2537	UPPDC1
299	2552	USHD
300	2583	Zn2tex