Mathematical Modelling of Bacterial Metabolism

by

Pooi Kuhn Ip

B. Sc (Math.), B. Eng (Elec.)

A thesis submitted for the degree of

Doctor of Philosophy in Mathematics

Phenomics and Bioinformatics Research Centre
School of Information Technology and Mathematical Sciences
Division of Information Technology, Engineering and the Environment



University of
South Australia

October 2014

Contents

Contents iii

List of Figures v

List of Tables vi

Abbreviations vii

Summary viii

Declaration xi

Acknowledgments xiii

1 Introduction 1

2 Analysis of complex metabolic behaviour through pathway decomposition 5

Abstract 5
Background 5
Results 5
Conclusions 6

Background 7

Results and Discussion 10
Overview 10
Metabolic engineering of E. coli 12
Understanding the survival of MTB during infection 20

Conclusions 26

Methods 27
Genome-scale FBA modelling 27
Elementary mode decomposition 27
Characterization of optimal metabolic behaviour using given elementary modes 30
Implementation of FBA and elementary mode decomposition 31
Comparison to previous decomposition methods 31

Authors' contributions 32

Acknowledgements 32

Source code 32

3 Constraint-based modelling of heterologous pathways 33

Abstract 33

Author Summary 34

Introduction 35

Materials and Methods **39**
Genome-scale FBA modelling 39
Proportional flux forcing (PFF) 39
GDLS analysis 44
Construction of knockout mutants 44
Fermentation procedures 45
Fatty acid analysis 45

Results/Discussion **46**
Computational results of coupled PFF/GDLS analysis 46
Computational analysis of suggested engineering strategies 49
Computational results for combined PFF/GDLS using iJO1366 56
Experimental results from implementation of suggested engineering strategies 57
Comparison of PFF with OptForce and FSEOF 59

Acknowledgments **61**

4 Conclusion **62**

5 References **65**

Appendix A Statements of contribution and copyright licences **73**

Appendix B Scripting for Elementary mode decomposition **83**

Appendix C Additional GDLS results for *i*AF1260 and *i*JO1366 **92**

Appendix D Reaction listings **101**

Appendix E Elementary flux modes for acetate-producing *E. coli* knockout strategies **105**

Appendix F Characterising elementary flux modes for MTB modelling **124**

List of Figures

FIGURE 2-1: ELEMENTARY MODES USED BY MYCOBACTERIUM TUBERCULOSIS GROWING ON OCTADECENOATE AS THE SOLE CARBON SOURCE WITH AND WITHOUT ICL 22

FIGURE 3-1: BASIC PRINCIPLES OF PROPORTIONAL FLUX FORCING 42

FIGURE 3-2: THE EFFECT OF THE *FADE* GENE DELETION ON METABOLIC PATHWAYS IN *I*AF1260 FOR FFA AND BIOMASS ACCUMULATION. 52

FIGURE 3-3: THE EFFECT OF REMOVING REACTIONS RPE AND TALA ON THE PENTOSE PHOSPHATE PATHWAY AND FFA PRODUCTION IN *I*AF1260. 54

FIGURE 3-4: BIOMASS AND FA ACCUMULATION OF MG1655 PKS1, MG1655 Δ*FADE* PKS1, AND PKI8 PKS1. 58

List of Tables

TABLE 2-1 NUMBER OF CHANGES TO REACTIONS USED IN EACH *E. COLI* KNOCKOUT MUTANT COMPARED WITH THE WILD-TYPE (WT) 13

TABLE 2-2 ELEMENTARY MODES FOR ACETATE-PRODUCING *E. COLI* KNOCKOUT STRATEGIES[A]. 15

[A]METABOLITE ABBREVIATIONS: 12PPD-R, (R)-PROPANE-1,2-DIOL; AC, ACETATE; ETOH, ETHANOL; GLC, GLUCOSE; PI, PHOSPHATE; SUCC, SUCCINATE. 18

TABLE 3-1 STRAINS AND PLASMIDS. 37

TABLE 3-2 OLIGONUCLEOTIDE PRIMERS. 45

TABLE 3-3 KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*AF1260 WITH PFF MODEL OF '*TESA* WITH *A* = 0.2. THE STRATEGIES CHOSEN FOR EXPERIMENTAL VALIDATION ARE HIGHLIGHTED IN BOLD. 48

TABLE C-1: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*AF1260 WITH PFF MODEL OF '*TESA* WITH *A* = 0.1. 92

TABLE C-2: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*AF1260 WITH PFF MODEL OF '*TESA* WITH *A* = 0.3. 93

TABLE C-3: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*AF1260 WITH PFF MODEL OF '*TESA* WITH *A* = 0.4. 94

TABLE C-4: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*JO1366 WITH PFF MODEL OF '*TESA* WITH *A* = 0.1. 95

TABLE C-5: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*JO1366 WITH PFF MODEL OF '*TESA* WITH *A* = 0.2. 97

TABLE C-6: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*JO1366 WITH PFF MODEL OF '*TESA* WITH *A* = 0.3. 99

TABLE C-7: KNOCKOUT STRATEGIES FROM GDLS ANALYSIS OF *I*JO1366 WITH PFF MODEL OF '*TESA* WITH *A* = 0.4. 100

TABLE D-1: LISTING OF REACTIONS KNOCKED OUT IN ACETATE PRODUCING *E. COLI* KNOCKOUT STRATEGIES 101

TABLE D-2: LISTING OF *I*AF1260 (*E. COLI*) REACTIONS REFERRED TO IN THIS RESEARCH 101

TABLE D-3: LISTING OF ADDITIONAL *I*JO1366 (*E. COLI*) REACTIONS REFERRED TO IN RESEARCH 104

TABLE D-4: LISTING OF GSMN-TB (MTB) REACTIONS REFERRED TO IN THIS RESEARCH 104

TABLE E-1: LISTING OF ELEMENTARY FLUX MODES FOR ACETATE PRODUCTING *E. COLI* KNOCKOUT STRATEGIES 105

TABLE F-1: LISTING OF ELEMENTARY FLUX MODES CHARACTERISING METABOLIC FLUX DISTRIBUTIONS FOR MTB MODELLING. 123

Abbreviations

ATP	Adenosine-triphosphate
E. coli	*Escherichia coli*
EFM	Elementary Flux Mode
FBA	Flux Balance Analysis
FFA	Free Fatty Acid
FSEOF	Flux Scanning based on Enforced Objective Flux
GDBB	Genetic Design through Branch and Bound
GDLS	Genetic Design through Local Search
ICL	Isocitrate lyase
MILP	Mixed-Integer Linear Program
MoMA	Minimisation of Metabolic Adjustment
MTB	*Mycobacterium tuberculosis*
NGAM	Non-Growth Associated ATP Maintenance
PFF	Proportional Flux Forcing

***i*AF1260**	Genome-scale *E. coli* model (Feist et al. 2007)
***i*JO1366**	Genome-scale *E. coli* model (Orth et al. 2011)
GSMN-TB	Genome-scale MTB model (Beste et al. 2007)
***i*NJ661**	Genome-scale MTB model (Jamshidi and Palsson 2007)

Summary

The ubiquity of microbes in our lives has made research into microorganisms an important area of study. From their uses in industrial applications to their roles in human health and disease, the ability to characterise and ultimately manipulate microbes in advantageous ways is of great utility. In this respect, microbial metabolism is important for several reasons. The biochemical reactions that comprise metabolism determine the consumption and production of chemical compounds and thus, directly relate to efficiency of microbial industrial fermentation. Metabolism and its regulation is also a key determinant in the virulence and survival of pathogenic microbes.

The key to engineering microbial metabolism for a particular purpose is finding targets for manipulation or modification to study experimentally. Microbial metabolism is complex and effects of modifications to a metabolic pathway are not necessarily localised and may be indirect. Trying to study metabolism as a whole entity by investigating all potential sets of modifications experimentally is infeasible. Thus, mathematical models combined with computational search algorithms have been used to ease the experimental burden by directing laboratory efforts towards promising candidates in a systematic and efficient fashion.

Constraint-based modelling, where metabolism is represented as a directed stoichiometric network of constrained biochemical reactions, is one such widely used framework for predicting metabolic phenotype. Here, we present projects that consider two issues with the application of predictive modelling to metabolic engineering.

First, the complexity of genome-scale metabolic models affects our ability to interpret the predictions generated. A useful tool for understanding and manipulating cellular metabolism is decomposition into elementary flux modes, which systematically organise metabolic networks into potential pathways associated with biochemical functions. Its utility, however, is severely limited since the number of modes increases exponentially with the size of the network.

We developed a new method for decomposition of metabolic flux distributions into elementary flux modes. We demonstrated the utility of our method for metabolic engineering of *Escherichia coli* and for understanding the survival of *Mycobacterium tuberculosis* during infection.

Our method can easily operate on large, genome-scale networks with computational time improvements exceeding 2000-fold, since it does not require generation of all relevant modes of the metabolic network. The decompositions from our method are also useful for understanding complex flux distributions and debugging genome-scale models.

Second, the underlying model must successfully capture the effects of manipulations on the metabolic behaviour of an organism. With constraint-based modelling techniques, it is not immediately clear how to model the correct behaviour of inserted heterologous genes under artificial control.

We developed a modelling method that we call Proportional Flux Forcing (PFF) to model artificially induced enzymatic genes within constraint-based modelling schemes. We applied PFF in conjunction with flux balance analysis-based computational strain optimisation to yield non-obvious genetic manipulation strategies that significantly increase free fatty acid production in *Escherichia coli* with an artificially induced heterologous thioesterase. The two mutant strains constructed and successfully grown in this study show up to a five-fold increase in total free fatty acid yield while growing on glucose supplemented with casamino acids.

Declaration

This thesis presents work carried out by myself and does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text; and all substantive contributions by others to the work presented, including jointly authored publications, are clearly acknowledged.

POOI KUHN IP



08 / 10 / 2014

Acknowledgments

This work is dedicated to my wife, Jane,
my mother, Kin Chee,
my brothers, Lang and Shoane,
and to the memory of my father Kam Ming.

Special thanks go to my supervisor Assoc. Prof. Desmond Lun for all his ideas and thoughtful advice and discussion, for his encouragement and support, and for his patience and boundless enthusiasm. I hope to pay it forward in all future endeavours.

Many thanks also to my associate supervisor Dr. Belinda Chiera for her advice and support

Thanks must go to Prof. G. Charles Dismukes, Dr. L. Tiago Guerra, Dr. David J. Vinyard & Dr. Nicholas Skizim (Gas chromatography equipment, expertise and training); to Assoc. Prof. Albert Juhasz (Gas chromatography expertise and sample analysis training); to Assoc. Prof. Kwangwon Lee and Ass. Prof. Nir Yakoby (Microbiology laboratory equipment); to Kevin Abbey (Cluster computing expertise).

Thanks also to Prof. Andy Koronis, Prof. Stan Miklavci, the Phenomics and Bioinformatics Research Centre, and the School of Information Technology and Mathematical Sciences at the University of South Australia for supporting my candidacy; and to Prof. Joseph V. Martin, Prof. Benedetto Piccoli and the Center for Computational and Integrative Biology at Rutgers University for welcoming me so warmly.

Thanks to Dr. Caroline Colijn, Dr. Nir Yakoby, and Dr. Aaron Brandes for opportunities to collaborate while I was working on my Ph.D.

This work would not have been possible without the support of the Australian Government through an Australian Postgraduate Award.

Finally, thanks to all the new friends I have made in Australia and the United States. This adventure would not have been the same without all of you.

1 Introduction

Microbes are heavily involved in many aspects of our lives. They can act in both beneficial and detrimental ways. Microbial fermentation has a wide variety of industrial applications (Milne et al. 2009). These applications range from treatment of sewage and pollutants (Megharaj et al. 2011; Rawat et al. 2011; Urgun-Demirtas et al. 2006) to production of useful compounds such as antibiotics and chemical pre-cursors(Nogales et al. 2008) as well as biofuels for energy (Lee et al. 2008; Rawat et al. 2013). Microbial fermentation is used extensively in food production to make items such as beer, wine, bread, yoghurt and cheeses (Caplice and Fitzgerald 1999; Zhu et al. 2009). Intestinal bacteria are useful for regulating digestive health and the immune system, while pathogenic microbes cause disease and illness (Holmes et al. 2012; Neish 2009; Nicholson et al. 2012). The ubiquity of microbes in our lives has made research into microorganisms an important area of study. The aim is to characterise the microbes and ultimately to manipulate them in ways that would be advantageous for human purposes. For instance, microbes could be engineered to increase the yield and efficiency of an industrial process (Hong and Nielsen 2012; Papagianni 2012). In the case of pathogenic microbes, an understanding of their survival mechanisms could suggest potential drug targets for developing therapeutics (Beste and McFadden 2010; Boshoff and Lun 2010).

Microbial metabolism is important for several reasons including the following. First, metabolism determines the consumption and production of chemical compounds. Metabolism consists of all the biochemical reactions that occur within the organism and thus, directly relates to the efficiency of a microbe for industrial fermentation. Second, metabolism is a key determinant in the virulence and survival of pathogenic microbes. Metabolic regulation of the biochemical reactions allows an organism to

adapt to the surrounding environment, or a pathogen to its host (Eisenreich et al. 2010; Rohmer et al. 2011).

The key to engineering microbial metabolism for a particular purpose is finding targets for manipulation or modification to study experimentally. Traditionally, such genetic targets have been found by testing human intuition based on knowledge of metabolic pathways or discovered fortuitously in random mutant strains created via mutagenesis techniques. These strategies by themselves are inadequate for modern day metabolic engineering. Microbial metabolism is complex and effects of modifications to a metabolic pathway are not necessarily localized. Indirect effects elsewhere must also be accounted for. Trying to study metabolism as a whole entity by investigating all potential sets of modifications experimentally would be infeasible. Therefore, the ability to systematically find genetic targets from across the whole organism more efficiently, in terms of time and money, is of prime importance.

The use of mathematical models and computational search algorithms aims to harness computing power to ease the experimental burden by directing laboratory efforts towards promising candidates. Constraint-based modelling is one such widely used framework (Curran et al. 2012; Patil et al. 2004; Price et al. 2003). In this modelling paradigm, metabolism is represented as a directed stoichiometric network of constrained biochemical reactions. The network for any given organism is derived by building a metabolic reconstruction (Thiele and Palsson 2010), which is a curated organism-specific database with data drawn from sequencing and a variety of biochemical, genetic and genomic information sources. Techniques based on these network reconstructions, such as flux balance analysis (FBA) (Orth et al. 2010) and minimisation of metabolic adjustment (MoMA) (Shlomi et al. 2005), are able to give good predictions for measures of metabolic phenotype such as growth rates or essential genes (Feist et al. 2007), which in turn allows the range of predicted phenotypes to be searched computationally (Lun et al. 2009).

There are however some issues with the application of predictive modelling to metabolic engineering. As mentioned above, genome-scale metabolic models are complex systems. For example, the most recent published data on the metabolic network of the bacterium *Escherichia coli* K-12 MG1655, includes 1366 genes, 2251 metabolic reactions and 1136 metabolites (Orth et al. 2011). Models, especially for

less well studied organisms, may also contain gaps or incomplete knowledge (Orth and Palsson 2012). This complexity affects our ability to interpret the predictions generated. Understanding why an organism responds metabolically in a particular fashion to manipulation of its environment or its genetics is equally as important as finding useful manipulations.

Second, a model must successfully capture the effect of manipulations on the metabolic behaviour of an organism. Constraint-based modelling techniques, such as FBA, use constraints to limit the range of potential metabolic behaviours. These constraints typically provide bounds for maximum and minimum reaction fluxes designed to model the effect of characteristics such as thermodynamics, which define reaction reversibility and enzymatic rates, and substrate availability. Model annotations such as gene-associations allow genes and their respective enzymes to be linked to their respective reactions. In a straightforward manner, gene deletions and down-regulation can be implemented as additional constraints with the effects of these manipulations having been modelled reasonably successfully (Burgard et al. 2003; Edwards and Palsson 2000). When considering gene insertions, however, and in particular genes introduced under artificial control, it is not immediately clear how the constraints should be modified to predict the correct behaviour.

Here we present two research projects that provide solutions to the above issues when applying predictive metabolic modelling:

The first project tackles the complexity of genome-scale metabolic networks by considering how to decompose metabolic networks into simpler interacting components. The concept of elementary flux modes (Schuster et al. 1999; Schuster and Hilgetag 1994) was introduced as a systematic means of organizing metabolic networks into potential pathways associated with particular biochemical functions. A flux mode of a metabolic network is a flux vector that satisfies both the steady state condition on the network and also the irreversibility constraints on reaction flows. A flux mode is also considered elementary if the flux vector is minimal or non-decomposable—i.e. there exists no subset of reactions in the flux vector such that a flux vector on this subset is also a flux mode. A formal mathematical definition is available in Chapter 2.

While decomposition using elementary flux modes has proven to be a powerful tool for understanding and manipulating cellular metabolism (Stelling et al. 2002; Trinh et al. 2009), its utility is severely limited since the number of modes in a network increases exponentially with its size (Klamt and Stelling 2002). Here we present a new method for decomposition of metabolic flux distributions into elementary flux modes. Our method can easily operate on large, genome-scale networks since it does not require all relevant modes of the metabolic network to be generated. We illustrate the utility of our method for metabolic engineering of *Escherichia coli* and for understanding the survival of *Mycobacterium tuberculosis* during infection.

The second project examines the difficulties of modelling genes associated with heterologous pathways that have been introduced under artificial control. We present a modelling method that we call Proportional Flux Forcing (PFF) to model artificially induced enzymatic genes within a constraint-based modelling scheme. We apply PFF in conjunction with FBA-based computational strain optimisation methods to yield non-obvious genetic manipulation strategies.

2 Analysis of complex metabolic behaviour through pathway decomposition

This work has been published (Ip et al. 2011):

Ip K, Colijn C, Lun DS (2011) Analysis of complex metabolic behavior through pathway decomposition. BMC Syst Biol 5: 91.

The copyrights for the article are retained by the authors.

Additional description and analysis has been added to expand on various aspects of the research described in the above publication.

Abstract

Background

Understanding complex systems through decomposition into simple interacting components is a pervasive paradigm throughout modern science and engineering. For cellular metabolism, complexity can be reduced by decomposition into pathways with particular biochemical functions, and the concept of elementary flux modes provides a systematic way for organizing metabolic networks into such pathways. While decomposition using elementary flux modes has proven to be a powerful tool for understanding and manipulating cellular metabolism, its utility, however, is severely limited since the number of modes in a network increases exponentially with its size.

Results

Here, we present a new method for decomposition of metabolic flux distributions into elementary flux modes. Our method can easily operate on large, genome-scale networks since it does not require all relevant modes of the metabolic network to be

generated. We illustrate the utility of our method for metabolic engineering of *Escherichia coli* and for understanding the survival of *Mycobacterium tuberculosis* (MTB) during infection.

Conclusions

Our method can achieve computational time improvements exceeding 2000-fold and requires only several seconds to generate elementary mode decompositions on genome-scale networks. These improvements arise from not having to generate all relevant elementary modes prior to initiating the decomposition. The decompositions from our method are useful for understanding complex flux distributions and debugging genome-scale models.

Background

Computational analysis of cellular metabolism has recently gained increasing prominence and importance. In particular, genome-scale computational models capturing stoichiometric and thermodynamic constraints have been published for over 30 organisms (Milne et al. 2009) ranging from relatively simple prokaryotes such as *E. coli*, to complex eukaryotes such as *Homo sapiens* (Duarte et al. 2007; Ma et al. 2007). The application of such computational models is dependent on their accuracy and the tools developed for their analysis. The maintenance of model accuracy, or debugging, is an ongoing process by which model predictions are validated against experimental observations and discrepancies are identified and corrected. This process is clearly linked with the use and analysis of the models. These computational models can be analysed in a number of ways. Flux-balance analysis (FBA) and elementary mode analysis are perhaps two of the most popular and powerful.

FBA determines a distribution of steady-state reaction fluxes that satisfies the constraints of the model and that optimizes a biological objective function such as biomass or adenosine triphosphate (ATP) production (Kauffman et al. 2003). With appropriate constraints and a biological objective, FBA has been shown to be an effective method for prediction of phenotypes associated with genetic manipulations such as knockouts (Edwards et al. 2001; Feist et al. 2007) and of intracellular metabolic fluxes (Schuetz et al. 2007). A significant application for FBA, therefore, is metabolic engineering—using computational predictions of metabolic phenotype under genetic manipulations to guide the engineering of metabolically optimized strains (Alper et al. 2005b)—and computational methods have been devised to search the space of genetic manipulations *in silico* for those that yield the desired phenotype (Burgard et al. 2003; Lun et al. 2009; Pharkya et al. 2004; Pharkya and Maranas 2006; Rocha et al. 2008). But, while FBA has had a number of successful applications, the method gives little insight into its predictions, hindering human understanding and model debugging.

Elementary mode analysis, on the other hand, yields no explicit predictions of metabolic behaviour, and seeks primarily to allow understanding of an organism's metabolic capabilities. In elementary mode analysis, the elementary flux modes

(EFMs) that define minimal sets of reactions capable of operating at steady state are generated (Schuster and Hilgetag 1994). Elementary flux modes formalize the concept of a biochemical pathway (Schuster et al. 2000), and studying the modes associated with a given metabolic network has been shown to be effective for understanding its function, regulation, and robustness (Stelling et al. 2002; Trinh et al. 2009). The principal drawback of elementary mode analysis is that the number of EFMs in a network suffers from a combinatorial explosion (Klamt and Stelling 2002), and the use of complete sets of EFMs gives rise to problems with scalability when applied to genome-wide models (Acuna et al. 2009). For example, more than two million modes exist for a simple model of *E. coli* central metabolism consisting of 112 reactions (Gagneur and Klamt 2004), which increases to more than 26 million modes when the possible substrates are expanded (Terzer and Stelling 2008). *i*AF1260 (Feist et al. 2007), the most recent genome-scale metabolic network of *E. coli*, consists of 2077 reactions, and the number of reactions in *E. coli* metabolic models has increased steadily for the past two decades (Feist and Palsson 2008). Thus, the computation time and memory storage required to enumerate all the EFMs of full and detailed genome-scale metabolic networks are prohibitively large.

For many applications, however, our understanding of particular metabolic functions of an organism, such as its ability to produce a desired metabolite, is of greater interest than our understanding of its full metabolic capabilities. In this case, it is not necessary to know all the EFMs of the network, but simply those that combine to give rise to a particular behaviour. Previous approaches to this problem have relied on first computing all modes of the network (Poolman et al. 2004; Schwartz and Kanehisa 2005). More recently, an approach presented by de Figueiredo et al. (de Figueiredo et al. 2009), found biologically significant EFMs by identifying the K-shortest EFMs of the metabolic network without necessarily enumerating the whole set. Our motivation is similar in that we are not concerned with generating all the EFMs of a network. However, our approach differs greatly in that we wish to determine those elementary modes that combine to yield a given flux distribution. This flux distribution can consist of both measured and computationally predicted fluxes.

We present an algorithm to find the elementary modes that combine to produce any previously-specified flux distribution. This links the advantages of elementary mode analysis with the advantages of flux balance analysis, without requiring the

prohibitive computation of all elementary modes. Our method is therefore applicable to genome-scale models and, because it can take as an input any flux distribution, it can be connected to particular experimental conditions (Colijn et al. 2009) or genetic modifications (Lun et al. 2009).

To demonstrate the utility of our method and its applicability to genome-scale models, we apply it to genome-scale models of *E. coli* and MTB and show how its results can be used to interpret flux distributions related to the metabolic engineering of *E. coli* and to the survival of MTB during infection.

Results and Discussion

Overview

Our method takes a given steady-state metabolic flux distribution and the corresponding metabolic model, and produces a decomposition of the flux distribution into elementary flux modes. In this paper, we use flux distributions obtained by FBA, but our method can equally be applied to flux distributions obtained by alternative means, such as those derived from experimental measurement or obtained from genetic perturbation analysis methods such as MoMA (Segre et al. 2002) or ROOM (Shlomi et al. 2005). As an elementary flux mode is itself a set of reactions operating at steady state, any flux distribution composed entirely of elementary flux modes must necessarily operate at steady state too. Therefore, an input flux distribution derived from experimental measurement may need to be balanced first to produce a steady state flux distribution, either by regression to fit the measured data or alternative means.

Our method operates by first selecting the reaction with non-zero flux of maximum magnitude from the given flux distribution. The algorithm then uses mixed-integer linear programming to find an elementary flux mode that both contains the selected reaction and is contained in the given distribution. The contribution of this elementary flux mode to the given distribution is determined before it is removed to give an updated flux distribution. The updated flux distribution is used as the input distribution for the next iteration of the algorithm, and this procedure is repeated until the updated flux distribution is zero (see Methods for details).

Elementary mode decompositions are not unique. Our goal is to assist in biological interpretation, and we are primarily interested in obtaining a valid decomposition rather than any specific decomposition. A valid decomposition will arise irrespective of the choice made at each iteration of the reaction with non-zero flux, but how this choice is made determines the specific decomposition. Our choice has some desirable properties. As the elementary flux mode found by the mixed-integer linear program includes the chosen reaction, the flux through this reaction will upper bound the contribution of the elementary flux mode. By selecting the reaction with non-zero flux of maximum magnitude, we avoid the possibility of generating many elementary flux modes with very small weightings and hence minimal contribution to the flux

distribution. This choice also minimizes numerical instabilities arising from the calculations.

Although mixed-integer linear programming is NP-hard in general, some large mixed-integer linear programs (MILPs) can be solved with a modest amount of computation by solvers such as SCIP (Achterberg 2007), IBM ILOG CPLEX (International Business Machines Corp., Armonk, New York), and Gurobi (Gurobi Optimization, Houston, Texas). In particular, our algorithm, implemented using Gurobi, successfully terminates in at most several seconds in all the genome-scale applications mentioned in this paper.

By contrast, previous approaches to decomposing flux distributions into elementary modes (Poolman et al. 2004; Schwartz and Kanehisa 2005) have relied on first generating the complete set of relevant elementary modes, then calculating a weight distribution among these modes. Since elementary mode decompositions are not unique (Schwartz and Kanehisa 2005), the principal advantage of these previous approaches in comparison to ours is that they are capable of selecting a particular unique decomposition based on some criterion (typically the minimization of the Euclidean norm of the weight vector), while our approach simply generates a valid decomposition among all possibilities. It is not clear, however, that criteria for selecting the weight distribution, such as the minimization of the Euclidean norm, are biologically meaningful. As we will see with the examples in this paper, simply having a valid decomposition is in itself useful.

The principal drawback of these previous approaches is that generating the complete set of relevant elementary modes can be prohibitive, particularly for genome-scale models. To demonstrate this fact, we used efmtool (Terzer and Stelling 2008) to efficiently generate the relevant elementary modes for the genome-scale model of MTB by Jamshidi and Palsson (2007), *i*NJ661, for growth on Middlebrook 7H9—a standard growth medium for MTB. With the elementary modes generated by efmtool, we then applied the quadratic programming approach proposed by Schwartz and Kanehisa (2005) to obtain a distribution of weights to assign to these modes. The total computational time of this approach was 34 minutes and 27 seconds.

By contrast, our approach generates a valid elementary mode decomposition consisting of 19 modes in 1.0 seconds—a computational time improvement exceeding

2000-fold. *i*NJ661 consists of 1,028 reactions, which is modest for current genome-scale models (Milne et al. 2009). With the ever increasing size and complexity of genome-scale models (Feist and Palsson 2008) and the exponential manner in which the number of elementary modes increases with the size of the network, it follows that in some cases, the advantages offered by our approach may not simply be a several-thousand fold improvement in computational time, but rather the difference between practical feasibility and infeasibility.

Indeed, *i*NJ661v (Fang et al. 2010), a recent model that modifies *i*NJ661 with the aim of more accurately modelling MTB in *in vivo* infection, is only slightly larger than *i*NJ661, with 1,049 reactions. However, a more complex growth medium resulted in a significantly larger number of relevant elementary modes for the flux distribution obtained by FBA than that for *i*NJ661, and we were unable to generate them all using efmtool because of memory limitations. Using our decomposition method, we generated a valid elementary mode decomposition consisting of 27 modes in 2.5 seconds.

We should observe here that the length and complexity of an elementary mode is dependent on the boundaries of the network—what is considered internal and external to the network. In the case of whole genome metabolic networks, the generated elementary modes span the entire network. Note that the traditional sense of a metabolic pathway, such as glycolysis or the TCA cycle, is used to describe a grouping of related metabolic reactions and is typically localised to one part of the network only. The biomass objective ensures that the 'traditional' pathways for components like amino acids or nucleotides are captured in the elementary modes themselves.

Metabolic engineering of E. coli

To demonstrate the utility of our approach for metabolic engineering, we consider the metabolic engineering of *E. coli* for increased acetate production—a problem that has received attention owing to the value of acetic acid for its industrial and food uses (Causey et al. 2003). Knockout strategies for increased production of acetate based on FBA have previously been reported by Lun et al. (2009) using the genome-scale metabolic reconstruction of *E. coli* by Feist et al. (2007), *i*AF1260. These knockout strategies were generated using GDLS (Genetic Design through Local Search) (Lun et

al. 2009), an efficient heuristic for generating metabolic engineering strategies involving multiple knockouts from genome-scale models, extending the capabilities of the computationally-expensive optimal search proposed by OptKnock (Burgard et al. 2003). The strategies were chosen using FBA to have high predicted production of acetate whilst maintaining required energy production and growth to ensure viability. Specifically, a predicted non-growth-associated ATP maintenance (NGAM) flux of at least 8.39 mmol gDW^{-1} h^{-1} (mmol per gram dry cell weight per hour) and a biomass flux of at least 0.05 g h^{-1} (grams per hour) were required.

The proposed strategies make sense biologically (Lun et al. 2009) and include experimentally-tested knockouts for acetate production such as alcohol dehydrogenase and ATP synthase (Causey et al. 2003). However, the cause for the increased acetate production is not immediately clear from the metabolic phenotype predicted by FBA alone. Table 2-1 shows the number of reactions that have been added to or removed from the flux distributions of the gene-knockout mutants when compared with the wild-type flux distribution. As we can see, the changes in the flux distributions are quite extensive and, in particular, they do not result simply from shunts around the blockages created by the gene deletions. The inherent structure of the network, and hence the mechanisms by which predicted metabolic behaviour arises, is difficult to ascertain from the flux distribution as a whole. We, therefore, applied our decomposition algorithm to determine the elementary flux modes that make up the flux distributions under these knockout strategies. We took the best knockout strategies reported by Lun et al. (Lun et al. 2009) with numbers of knockouts ranging from 1 to 8. For the purposes of these strategies, a single knockout is considered to consist of all genes capable of catalysing a reaction, i.e. the enzyme

Gene Knockouts	**1**	**2**	**3**	**4**	**5**	**6**	**8**
Reactions with zero flux in WT and non-zero flux in mutant	12	12	22	25	25	29	36
Reactions with non-zero flux in WT and zero flux in mutant	9	9	16	21	23	17	24

Table 2-1 Number of changes to reactions used in each *E. coli* knockout mutant compared with the wild-type (WT)

complex or all complexes with the same metabolic function.

Table 2-2 shows the elementary flux modes we obtained. The listing of reactions that have been deleted as part of these knockout strategies is given in Appendix D, Table D-1. The complete modes are given in Appendix E. The flux distribution for each knockout mutant can be decomposed into two elementary modes, which together supply the energy and growth requirements of the organism. We emphasize that elementary mode decomposition will identify the structural components that are important to the metabolic phenotype.

The focus on the NGAM and biomass components reflects their biological importance to the underlying biology of the model. In fact, the requirement to supply NGAM flux and the desire to maximise growth are essentially competing against each other for the limited supply of glucose, and it makes sense that this could restrict the number of elementary modes necessary to characterise the distribution to two modes. The first mode only contributes to the required NGAM flux, and uses relatively few reactions. In comparison, the second mode is solely responsible for producing biomass and involves many more active reactions. This corresponds with the large number of metabolites required for biomass production.

Furthermore, as noted by Stelling et al. (2002), the unmodified *E. coli* metabolic network displays a degree of robustness as evidenced by the varied pathways by which biomass and NGAM are produced. Acetate, which is a by-product of growth, is produced to varying degrees by these pathways. Acetate flux is, in effect, coupled to the fluxes for the production of necessary metabolic components of NGAM and biomass. The knockouts restrict the available elementary modes and essentially force flux onto alternative pathways with increased coupling between the fluxes for acetate and the fluxes for NGAM and biomass components. The knockouts achieve this in a few different ways. Acetate flux is produced entirely by the reaction pair PTAr / ACKr, which generates acetate and ATP from acetyl-CoA (and phosphate ions). Thus, acetate production is a direct means of generating ATP. In fact, the NGAM EFMs identified in this analysis are equivalent to the most efficient NGAM producing EFMs available to the metabolic network. It makes sense that these types of EFMs will be utilised to meet the significant ATP requirements.

Table 2-2 Elementary modes for acetate-producing *E. coli* knockout strategies[a].

Knockouts (Number in brackets, Separated by semi-colons)	**Mode**	**Weight**	**Overall conversion** (Acetate production in bold)	**Biomass (g / mmol Glc)**	**NGAM (mmol / mmol Glc)**	**Number of reactions**
(0) None	1	2.92	1.000 Glc + 0.005 K + 0.315 NH_4 + 0.028 Pi + 0.007 SO_4 -> **0.785 Ac** + 1.661 CO_2 + 0.766 EtOH + 1.073 H + 1.675 H_2 + 0.010 Succ	0.029	0.2	412
	2	7.08	1.000 Glc + 0.297 H_2O + 0.004 K + 0.222 NH_4 + 0.020 Pi + 0.005 SO_4 → **0.849 Ac** + 1.762 CO_2 + 0.835 EtOH + 1.051 H + 1.772 H_2 + 0.007 Succ	0.021	1.1	414
	Combined		10.000 Glc + 2.101 H_2O + 0.041 K + 0.002 Mg_2 + 2.492 NH_4 + 0.222 Pi + 0.058 SO_4 -> **8.301 Ac** + 17.322 CO_2 + 8.152 EtOH + 10.578 H + 17.434 H_2 + 0.077 Succ	0.23	8.4	415
(1) frmA, adhE, adhP	1	4.18	1.000 Glc + 0.667 H_2O → 0.667 12PPD-R + **1.333 Ac** + 1.333 CO_2 + 1.333 H + 1.333 H_2	0	2	38
	2	5.82	1.000 Glc + 0.004 K + 0.228 NH_4 + 0.020 Pi + 0.005 SO_4 → 0.554 12PPD-R + **1.121 Ac** + 1.201 CO_2 + 1.330 H + 1.211 H_2 + 0.007 Succ	0.021	0.0045	414
	Combined		10.000 Glc + 2.788 H_2O + 0.022 K + 1.327 NH_4 + 0.118 Pi + 0.031 SO_4 -> 6.011 12PPD-R + **12.101** Ac + 12.563 CO_2 + 13.313 H + 12.623 H_2 + 0.041 Succ	0.12	8.4	418

(2) mhpF, adhE; ydfG	1	4.07	1.000 Glc + 0.750 H_2O → **1.500 Ac** + 1.500 CO_2 + 0.750 EtOH + 1.500 H + 1.500 H_2	0	2.1	54
	2	5.93	1.000 Glc + 0.047 H_2O + 0.004 K + 0.237 NH_4 + 0.021 Pi + 0.005 SO_4 → **1.296 Ac** + 1.306 CO_2 + 0.586 EtOH + 1.513 H + 1.353 H_2 + 0.007 Succ	0.022	0	417
	Combined		10.000 Glc + 3.331 H_2O + 0.023 K + 0.001 Mg_2 + 1.406 NH_4 + 0.125 Pi + 0.033 SO_4 -> **13.791 Ac** + 13.848 CO_2 + 6.529 EtOH + 15.075 H + 14.128 H_2 + 0.044 Succ	0.13	8.4	418
(3) atpABCDEFGH; frmA, adhE, adhP; pgi	1	2.82	1.000 Glc + 0.444 H_2O + 0.167 SO_4 → 0.444 12PPD-R + **1.556 Ac** + 1.556 CO_2 + 1.222 H + 1.556 H_2 + 0.167 H_2S	0	1.3	59
	2	7.18	1.000 Glc + 0.233 H_2O + 0.001 K + 0.077 NH_4 + 0.007 Pi + 0.159 SO_4 → 0.408 12PPD-R + **1.495 Ac** + 1.486 CO_2 + 1.251 H + 1.501 H_2 + 0.157 H_2S + 0.002 Succ	0.0071	0.64	418
	Combined		10.000 Glc + 2.926 H_2O + 0.009 K + 0.552 NH_4 + 0.049 Pi + 1.609 SO_4 -> 4.181 12PPD-R + **15.118 Ac** + 15.056 CO_2 + 12.431 H + 15.166 H_2 + 1.596 H_2S + 0.017 Succ	0.051	8.4	421
(4) atpABCDEFGH; frmA, adhE, adhP; pgi; yahI, ybcF, yqeA	1	2.91	1.000 Glc + 0.444 H_2O + 0.167 SO_4 → 0.444 12PPD-R + **1.556 Ac** + 1.556 CO_2 + 1.222 H + 1.556 H_2 + 0.167 H_2S	0	1.3	59
	2	7.09	1.000 Glc + 0.233 H2O + 0.001 K + 0.077 NH4 + 0.007 Pi + 0.159 SO4 → 0.408 12PPD-R + **1.495 Ac** + 1.486 CO2 + 1.251 H + 1.501 H2 + 0.157 H2S + 0.002 Succ	0.0071	0.64	416

	Combined		10.000 Glc + 2.946 H_2O + 0.009 K + 0.545 NH_4 + 0.049 Pi + 1.609 SO_4 -> 4.185 12PPD-R + **15.124 Ac** + 15.062 CO_2 + 12.428 H + 15.171 H_2 + 1.597 H_2S + 0.017 Succ	0.051	8.4	419
(5) atpABCDEFGH; frmA, adhE, adhP; pgi; tnaA; yahI, ybcF, yqeA	1	2.87	1.000 Glc + 0.444 H2O + 0.167 SO4 → 0.444 12PPD-R + **1.556 Ac** + 1.556 CO2 + 1.222 H + 1.556 H2 + 0.167 H2S	0	1.3	59
	2	7.13	1.000 Glc + 0.235 H2O + 0.001 K + 0.076 NH4 + 0.007 Pi + 0.158 SO4 → 0.408 12PPD-R + **1.496 Ac** + 1.486 CO2 + 1.252 H + 1.501 H2 + 0.157 H2S + 0.002 Succ	0.0071	0.64	413
	Combined		10.000 Glc + 2.949 H_2O + 0.009 K + 0.544 NH_4 + 0.049 Pi + 1.607 SO_4 -> 4.185 12PPD-R + **15.128 Ac** + 15.057 CO_2 + 12.435 H + 15.168 H_2 + 1.595 H_2S + 0.017 Succ	0.05	8.4	417
(6) atpABCDEFGH; galP; mhpF, adhE; pgi; pitAB; ydfG	1	6.53	1.000 Glc + 0.762 H2O → 0.071 12PPD-R + **1.690 Ac** + 1.357 CO2 + 0.524 EtOH + 1.690 H + 1.524 H2	0	1.3	77
	2	3.47	1.000 Glc + 0.280 H_2O + 0.003 K + 0.156 NH_4 + 0.014 Pi + 0.004 SO_4 → 0.114 12PPD-R + **1.404 Ac** + 1.304 CO_2 + 0.467 EtOH + 1.546 H + 1.388 H_2 + 0.005 Succ	0.014	0	424
	Combined		10.000 Glc + 5.945 H_2O + 0.009 K + 0.542 NH_4 + 0.048 Pi + 0.013 SO_4 -> 0.863 12PPD-R + **15.908 Ac** + 13.386 CO_2 + 5.042 EtOH + 16.403 H + 14.766 H_2 + 0.017 Succ	0.05	8.4	427

(8) (sapD or trkA or trkG), (sapD or trkA or trkH), kch, kup; atpABCDEFGH; galP; guaB; mhpF, adhE; pgi; pitAB; ydfG	1	6.53	1.000 Glc + 0.762 H_2O → 0.071 12PPD-R + **1.690 Ac** + 1.357 CO_2 + 0.524 EtOH + 1.690 H + 1.524 H_2	0	1.3	77
	2	3.47	1.000 Glc + 0.282 H_2O + 0.003 K + 0.155 NH_4 + 0.014 Pi + 0.004 SO_4 → 0.114 12PPD-R + **1.406 Ac** + 1.304 CO2 + 0.468 EtOH + 1.547 H + 1.389 H_2 + 0.005 Succ	0.014	0	424
	Combined		10.000 Glc + 5.953 H_2O + 0.009 K + 0.539 NH_4 + 0.048 Pi + 0.013 SO_4 -> 0.861 12PPD-R + **15.915 Ac** + 13.387 CO_2 + 5.043 EtOH + 16.407 H + 14.769 H_2 + 0.017 Succ	0.05	8.4	427

[a]Metabolite abbreviations: 12PPD-R, (R)-Propane-1,2-diol; Ac, acetate; EtOH, ethanol; Glc, glucose; Pi, phosphate; Succ, succinate.

The reactions ALCD2x and ACALD together allow the production of ethanol from acetyl-CoA. Targeting these reactions separately removes a by-product competing for acetyl-CoA, which is a precursor for many biomass components as well as acetate. In a more subtle fashion, removing ALCD2x allows ACALD to be reversed supplying more acetyl-CoA via acetaldehyde—central carbon flux is diverted through DRPA which increases acetaldehyde flux. Removing ACALD also decouples acetyle-CoA from acetaldehyde and allows reactions such as DRPA (for glyceraldehyde 3-phosphate) and THRAi (for pyruvate) that may generate excess acetaldehyde to be used.

Targeting PGI in glycolysis and GLCt2pp redirects the uptake of glucose through pathways that require more ATP—via fructose and ATP-driven glucose uptake. Flux is redirected along pathways such as EDA and SERD_L that generate more pyruvate to compensate for the increased ATP needs. Pyruvate is converted into acetyl-CoA and then acetate to supply ATP. This is clearest in the NGAM producing modes where ATP efficiency is paramount.

Other suggested deletions such as CBMKr and IMPD aim to increase the ATP demand or phosphate supply in the biomass EFM. There are also a variety of genes coding for transport proteins for potassium, phosphate and protons, which subtly shift the demand for ATP in the biomass EFM. An example is the gene PitAB (for PIt2rpp), which codes for phosphate transport and its removal forces a switch to ATP-driven phosphate uptake.

By finding and examining the elementary modes associated with the acetate-enhancing knockout strategies, we reveal how the energy and growth requirements are met by the NGAM and biomass EFMs and the mechanism responsible for the increased acetate flux, which was not apparent from the FBA analysis.

Most interestingly, when the modes are examined in terms of acetate production, we find that the most efficient modes are generally those that only contribute to NGAM flux. However, a biomass-producing mode is necessary to satisfy growth and viability requirements. Thus, the problem of selecting knockouts to maximise acetate production, given a limiting resource such as glucose, is not necessarily about finding a single optimal elementary mode. Rather, competing constraints demand that the chosen modes need to complement each other well. This can be seen in the reported decompositions for the various knockout mutants.

For example, of all the biomass-producing modes, the mode arising from the five-knockout mutant is the most efficient at 1.496 mmol of acetate per mmol of glucose. When more knockouts are allowed, the overall acetate production is improved despite a decrease in the acetate production efficiency of the biomass-producing mode. This decrease is offset by a shift towards using NGAM-producing modes with significantly more efficient production of acetate. For the six and eight knockout cases, the NGAM-producing mode generates 1.690 mmol of acetate per mmol of glucose.

Finally, as FBA does not necessarily yield a unique distribution, we implemented the recursive MILP algorithm from Lee et al. (Lee et al. 2000) to find alternate optima and then obtained decompositions for the corresponding flux distributions. Our results (not shown) indicate that the decomposition into two modes with distinct functions related to NGAM and biomass production is preserved for alternate optima. Thus, the decomposition of flux distributions into primarily NGAM-producing and biomass-producing modes is a robust quality.

Understanding the survival of MTB during infection

To illustrate another application of our approach, we consider the utilization of the glyoxylate shunt in MTB. The glyoxylate shunt enzyme isocitrate lyase (ICL), present in MTB as two isoforms, is believed to be required by microorganisms to utilize fatty acids as a source of carbon and energy in the absence of sugars such as glucose. It achieves this by providing an alternative pathway from isocitrate to malate and bypassing the two decarboxylation steps of the TCA cycle. Lipids and fatty-acids can be metabolised into acetyl-CoA and the glyoxylate shunt allows acetyl-CoA to be diverted towards the production of succinate and malate (and ultimately more complex carbohydrates) instead of being lost as carbon dioxide. It does, however, affect efficiency as two acetyl-CoA molecules are required compared to one for the TCA cycle. This shunt has previously been shown to be required for the *in viv*o growth and virulence of MTB (Munoz-Elias and McKinney 2005). Since MTB is believed to subsist on fatty acids during infection (Munoz-Elias and McKinney 2005; Schnappinger et al. 2003), it is argued that, by removing ICL, MTB is no longer able to utilize fatty acids for carbon and energy and therefore unable to grow *in vivo*. Indeed, strains of MTB absent in both isoforms of ICL are unable to grow on fatty acid substrates and unable to survive in macrophages and mice (Munoz-Elias and McKinney 2005).

Therefore, ICL has attracted significant attention as a promising drug target for treatment of MTB infection (Sharma et al. 2000).

It is possible, however, that given the robustness that is generally observed in metabolic networks (Stelling et al. 2002), alternative metabolic pathways may exist to allow strains without a supposedly essential functional enzyme like ICL to recover viability. Using our method, we can study the metabolic capabilities of MTB growing on fatty acid substrates and examine if and how these alternatives might be used. In particular, we determined the elementary modes used by MTB at differing uptake rates of octadecenoate using the genome-scale metabolic reconstruction of MTB by Beste et al. (2007) (see Figure 2-1).

The full elementary modes have been listed in Appendix F. For the growth conditions examined in this research, the limiting flux is the uptake of octadecanoate. Fluxes internal to the model do not hit their maximal flux magnitude limits, except for the special case of NGAM flux. The model simulates a 'fixed' NGAM flux to satisfy energy requirements to maintain cell viability. (In practice, this is simulated as a tightly bounded flux to assist in numerical stability.) The selection of elementary modes is thus a trade-off between maximising biomass growth whilst satisfying NGAM requirements given octadecanoate uptake. As octadecanoate uptake increases, new modes become available to satisfy the NGAM requirement and facilitate the change in growth mode.

We found that there exist modes that generate biomass and/or NGAM both using and not using ICL. At a given NGAM flux, the modes that use ICL generally produce biomass more efficiently than those that do not. However, for a given uptake of octadecanoate, the modes that produce the most biomass are those that do not use ICL. Therefore, the modes available to maximize the biomass production while maintaining a given NGAM requirement will depend on the supply flux of octadecenoate. When the supply is sufficiently high, the NGAM requirement is easily met by the high efficiency biomass producing modes that do not use ICL, but when it is lower, use of ICL allows the NGAM requirement to be met more efficiently and, hence, allows more biomass to be produced.

Figure 2-1: Elementary modes used by Mycobacterium tuberculosis growing on octadecenoate as the sole carbon source with and without ICL

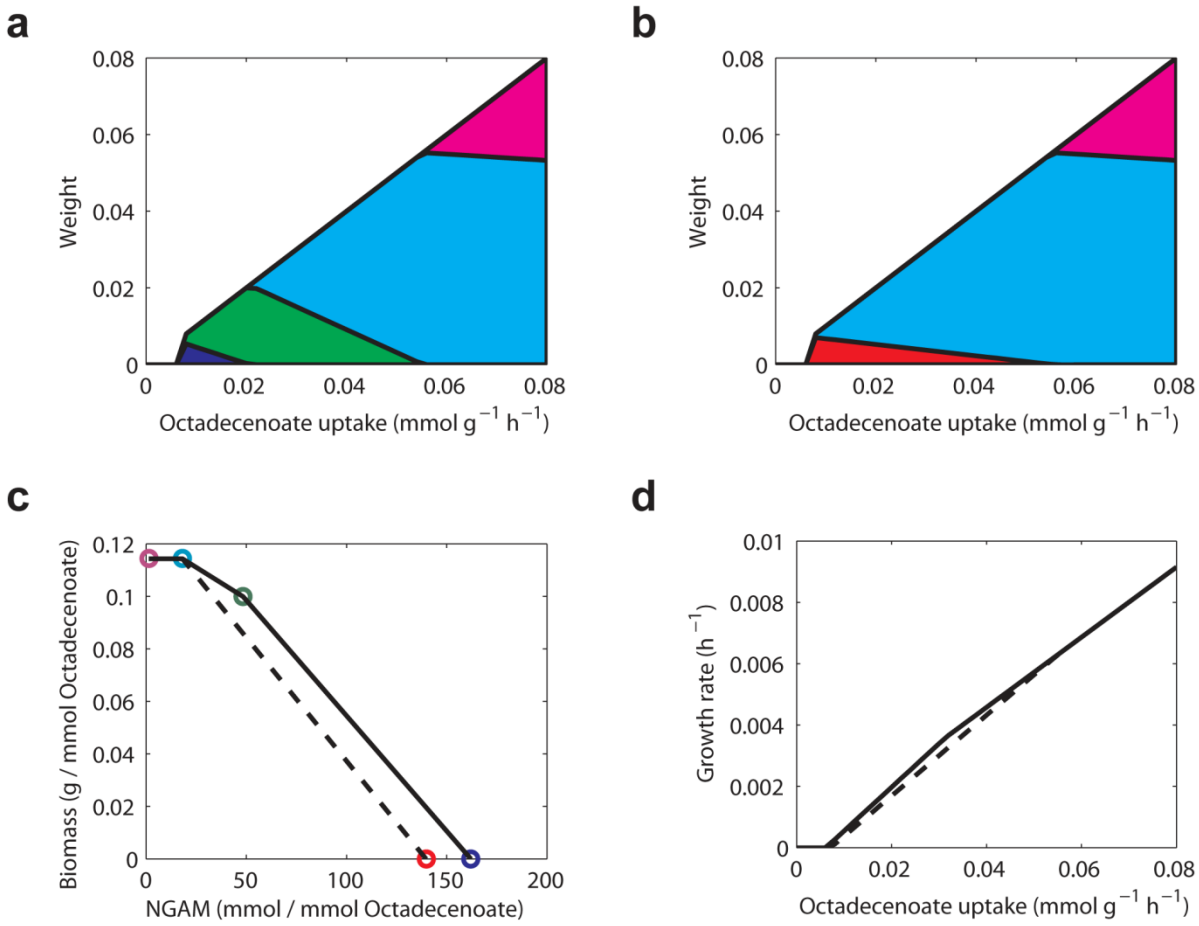


Figure 2-1 (cont.):

Five unique elementary modes are identified overall by applying our decomposition method to three representative flux distributions, and these elementary modes are used to characterize optimal metabolic behaviour for octadecenoate uptake varying from 0 to 0.08 mmol g^{-1} h^{-1}. The weights of these five elementary modes (coloured blue, green, red, cyan, and magenta) as octadecenoate uptake varies are shown stacked **(a)** with and **(b)** without ICL present. The modes coloured blue and green use ICL, while the remainder does not. Each mode is normalized so that the octadecenoate uptake of the mode is 1 mmol g^{-1} h^{-1}. **(c)** The biomass and NGAM generated by each mode with 1 mmol of octadecenoate. The region enclosed by the solid line is the space of achievable pairs of biomass and NGAM when ICL is present, while the region enclosed by the dotted line is that when ICL is not present. When octadecenoate is plentiful, the cyan and magenta modes can be used to meet the NGAM requirement and to produce biomass; when octadecenoate is more limited, the remaining modes are needed to meet the NGAM requirement and, with ICL present, this can be achieved more efficiently. **(d)** FBA-predicted growth rate under varying octadecenoate uptake with ICL present (solid line) and without it (dotted line).

Under the assumption that the metabolic reconstruction by Beste et al. is correct, our analysis implies that MTB is capable of producing both carbon and energy from fatty acids without the use of ICL but, for lower uptake rates of fatty acids, ICL allows for more efficient utilization of the fatty acids. Indeed, we found that the optimal growth rate predicted by FBA differs only slightly with ICL present and without it (see Figure 1d). This presents the intriguing possibility that MTB possesses the metabolic capability to grow on fatty acids without ICL, but does not do so after the knockout of ICL because it has not yet undergone the adaptive evolution necessary to make use of this metabolic capability. This possibility is consistent with the work of Fong and Palsson (Fong and Palsson 2004) showing that the growth rate of gene deletion strains of *E. coli* can change significantly after undergoing adaptive evolution. The possible existence of such inactive routes for metabolizing fatty acids without the use of ICL in MTB has been discussed elsewhere (Purohit et al. 2007) and, if true, would imply that MTB could rapidly evolve resistance to drugs inhibiting ICL.

Closer examination of the elementary modes reveals how MTB grows *in silico* without ICL and provides a testable means of confirming or rejecting the model's predictions. All the elementary modes that do not use ICL use malate synthase, the second of the two enzymes that form the glyoxylate shunt. In all of these modes, the full flux through malate synthase comes from a pathway that is essentially a 'hydroxyproline shunt' of the TCA cycle. Alpha-ketoglutarate is used to synthesise glutamate which is then converted to (S)-1-pyrroline-5-carboxylate, which is the typical product of the first step in the proline degradation pathway. From here, it is ultimately converted into pyruvate and glyoxylate via 1-pyrroline-5-carboxylate dehydrogenase (R139 – rocA) into 4-hydroxy-L-glutamate; 4-hydroxyglutamate transaminase (R140) into 4-hydroxy-2-ketoglutarate; and lastly 4-hydroxy-2-ketoglutaric aldolase (R141 – htrA). The annotations on this pathway indicate that these are probable predicted enzymes/reactions with mammalian analogs (Caspi et al. 2012). Indeed, for the final step, HtrA is a gene predicted to code for 4-hydroxy-2-oxoglutarate aldolase to complete the hydroxyproline degradation pathway in MTB using the Bayesian method of Green and Karp (2004). This pathway supplies the glyoxylate that is used as a substrate by malate synthase.

MTB and other mycobacteria have been observed to grow using hydroxyproline as a carbon source (Indira and Sirsi 1960; Krulwich and Pelliccione 1979), suggesting that

this pathway may indeed exist. Further studies confirming and characterizing this pathway will shed light on whether it does in fact provide MTB with a viable means of producing glyoxylate.

The elementary mode decomposition analysis we have presented sheds light on the mechanics of the FBA prediction that is difficult to obtain from the FBA results alone. The pathway using HtrA in combination with malate synthase can be used in GSMN-TB to generate biomass and NGAM. By characterising the range of metabolic behaviour over varying octadecanoate uptake levels in relation to a set of elementary modes, we can demonstrate how complementary elementary modes combine to supply the biomass and NGAM requirements of MTB. However, the space of possible biomass and NGAM production rates that can be achieved in this way is smaller than that which is possible using the glyoxylate shunt. At low octadecenoate uptake rates, this difference is important, leading to a lower biomass production to meet the NGAM requirement. Note that in contrast, FBA alone does not provide a means of breaking down a flux requirement like NGAM, and only observes that it has been satisfied.

It also demonstrates the utility of our method in identifying a potential source of the discrepancy between *in silico* predictions and observed experimental results. If growth is not experimentally possible, even after adaptive evolution, then it suggests that the model is incorrect and the likely error in the model comes from the inclusion of the predicted gene, HtrA.

While the other reaction in the shunt, malate synthase, is also considered a potential drug target (Dunn et al. 2009), we have not extended EFM decomposition analysis to include it. Standard FBA modelling predicts significant reductions in viability (growth reduced some 30-40 times slower) in contrast to the subtle differences in viability when considering ICL deletion. This research is focused on demonstrating the potential of effective EFM decomposition to analyse how an alternative pathway might be utilised and to identify potential discrepancies between predictions and experimental results.

Conclusions

We have presented a method for decomposing a given flux distribution into a set of constituent elementary modes. In contrast to previous approaches, our method does not require the initial generation of a full set of elementary modes, which is often computationally demanding and, in some cases, computationally infeasible for practical purposes. In a moderately-sized instance, we observed a computational time improvement of over 2000-fold using our method.

Overall, we see that elementary mode analysis offers a great deal that flux-balance analysis alone does not. FBA yields predictions of overall metabolic behaviour, while elementary mode analysis allows understanding of metabolic capabilities. By decomposing flux distributions obtained by FBA into elementary modes, we can gain insight into how metabolic networks achieve their predictions. We exploit modularity to decompose a complex entity into a simpler entity, which enables debugging and understanding and, ultimately, more sophisticated design and engineering.

Methods

Genome-scale FBA modelling

We work with the genome-scale model of *E. coli*, *i*AF1260. This model consists of three parts. First, from *m* metabolites and *n* reactions, we form an $m \times n$ stoichiometric matrix *S*, whose *ij*th element S_{ij} is the stoichiometric coefficient of metabolite *i* in reaction *j*. Second, the flux distribution *v*, whose *j*th element v_j is the flux though reaction *j*, is constrained by a lower bound vector *a*, whose *j*th element a_j is the lower bound of the flux through reaction *j*, and an upper bound vector *b*, whose *j*th element $b_j > 0$ is the upper bound of the flux through reaction *j*. Finally, the linear objective is formed by multiplying the fluxes by an objective vector *f*, whose *j*th element f_j is the weight of reaction *j* in the biological objective. Thus, a biologically optimal flux distribution is obtained by solving

$$\begin{aligned} \max \quad & f'v \\ \text{subject to} \quad & Sv = 0, \\ & a \le v \le b. \end{aligned} \tag{2-1}$$

Elementary mode decomposition

For a given flux vector v, we use $R(v) = \{i : v_i \neq 0\}$ to denote the set of indices of the reactions participating in v. We define an elementary flux mode using the following two definitions (Kauffman et al. 2003).

Definition: A flux mode, or mode, in a metabolic network specified by a stoichiometric matrix *S* and lower and upper bound constraints *a* and *b* is a non-zero $n \times 1$ vector *p* satisfying the following two conditions:

1 it is a valid steady-state flux distribution, i.e. $Sp = 0$;

2 irreversible reactions flow in the right direction, i.e. for all *j* such that $a_j \ge 0$, we have $p_j \ge 0$.

Definition: We say a flux mode is *elementary* if it is minimal among all flux modes, i.e. there does not exist any other flux mode such that $R(p') \subset R(p)$.

Before stating the algorithm, we require one further definition.

Definition: We say a flux mode *p conforms* to a flux distribution *v* if $v_j > 0$ for all reactions *j* with $p_j > 0$ and if $v_j < 0$ for all reactions *j* with $p_j < 0$.

Our interest is in finding elementary modes that conform to a given flux distribution *v* since it ensures that *v* is decomposed into elementary modes that have reactions flowing in the same direction as *v*, for the purposes of biological interpretation.

Our algorithm takes as input a flux distribution *v* in the feasible set of optimization problem (2-1) and outputs an elementary mode decomposition for *v*, i.e. a set of elementary flux modes $\{p^{(k)}\}$ that conform to *v* and a corresponding set of positive weights $\{w_k\}$ such that $v = \sum_{k=1}^{K} w_k p^{(k)}$. Suppose we have a flux distribution v that satisfies $Sv = 0$. Let $k := 1$, and $v^{(k)} := v$. Choose some j_k such that $v_{j_k}^{(k)} \neq 0$, and then solve the following mixed-integer linear program (MILP):

$$\begin{aligned} &\min \textstyle\sum_{j=0}^{n} q_j \\ &\text{subject to} Sp = 0, \\ &\quad p_{j_k} = \operatorname{sgn}(v_{j_k}^{(k)}), \\ &\quad -Mq_j \le p_j \le Mq_j, j = 1,\ldots,n, \\ &\quad p_j = 0,\ j \in \{j \mid v_j = 0\}, \\ &\quad p_j \le 0,\ j \in \{j \mid v_j < 0\}, \\ &\quad p_j \ge 0,\ j \in \{j \mid v_j > 0\}, \\ &\quad q_j \in \{0,1\},\ j = 1,\ldots,n, \end{aligned} \tag{2-2}$$

where M is a large constant and sgn is the sign function, taking the value 1 if its argument is positive and -1 if its argument is negative. This MILP is similar to that used by de Figueiredo et al. (de Figueiredo et al. 2009) for finding the shortest elementary modes in a metabolic network. Our purpose, however, differs significantly: we seek to decompose a given flux distribution into constituent elementary modes. The specific choice of j_k is unimportant: all choices such that $v_{j_k}^{(k)} \neq 0$ will generate a valid decomposition, though the specific decomposition achieved will likely vary with the choice. As discussed in the overview, we choose $j_k := \arg\max_j |v_j|$.

Let (p^*, q^*) be an optimal solution. We then set $p^{(k)} := p^*$ and $w_k := \min_{j \in \{j \mid p_j^{(k)} \neq 0\}} (v_j^{(k)} / p_j^{(k)})$. Finally, we set $v^{(k+1)} := v^{(k)} - w_k p^{(k)}$. If $v^{(k+1)}$ is the zero

vector, then we are done. Otherwise, we choose j_{k+1} such that $v_{j_{k+1}}^{(k+1)} \neq 0$, increment *k* by one, and repeat the above procedure.

The following proposition establishes that this algorithm generates the desired output. In brief, the algorithm works because, at each iteration, the MILP finds a flux mode where p_{j_k} is non-zero and where the number of non-zero elements in the flux mode is minimized. Because the number of non-zero elements in the flux mode is minimized, the flux mode is minimal and, hence, elementary. Since each reaction with non-zero flux must participate in at least one elementary mode in the decomposition, it does not matter how j_k is chosen, as long as $v_{j_k}^{(k)} \neq 0$. A valid decomposition will be generated regardless of how j_k is chosen, though the particular decomposition that is generated among the non-unique possibilities will depend on this choice.

Proposition: The elementary mode decomposition algorithm stated above terminates after a finite number of iterations *K* and generates a set of elementary flux modes $\{p^{(k)}\}$ that conform to *v* and a corresponding set of positive weights $\{w_k\}$ such that $v = \sum_{k=1}^{K} w_k p^{(k)}$.

Proof. First, to establish that each $p^{(k)}$ is in fact a mode, we observe that any $p^{(k)}$ generated as a solution of problem (2-2) will meet the steady state condition of the system. Problem (2-2) has a solution since $p = v^{(k)} / \left| v_{j_k}^{(k)} \right|$ and *q* such that $q_j = 1$ if $p_j \neq 0$ and $q_j = 0$ otherwise is in the feasible set of the problem. Further, by constraining the components of $p^{(k)}$ to have the same sign as the corresponding elements of v, we ensure that irreversible reactions flow in the right direction since, for any *j*, if $a_j \geq 0$ then $v_j \geq 0$, which sets the constraint $p_j \geq 0$ in problem (2-2).

Second, from the constraints of problem (2-2), we can see that $p^{(k)}$ conforms to *v*.

Lastly, we establish that each $p^{(k)}$ is minimal among all flux modes conforming to *v* and, therefore, elementary in the set of all such modes. We first observe that the optimal cost of problem (2-2) at iteration *k* is $\left| R(p^{(k)}) \right|$. Suppose there exists a mode p' that conforms to *v* with $R(p') \subset R(p^{(k)})$. If $p'_{j_k} \neq 0$, then we assume without loss of

generality that $p'_{j_k} = 1$, and (p', q'), where $q'_j = 1$ if $p'_j \neq 0$ and $q'_j = 0$ otherwise, is in the feasible set of problem (2-2) and $\sum_{j=1}^{n} q'_j = |R(p')| < |R(p^{(k)})|$, in contradiction with $p^{(k)}$ being an optimal solution. If $p'_{j_k} = 0$, then let $p'' = p^{(k)} - wp'$, where $w = \min_{j \in \{ j | p'_j \neq 0 \}} (p_j^{(k)} / p'_j)$. Now p'' is a mode that conforms to *v* with $R(p'') \subset R(p^{(k)})$ and (p'', q''), where $q''_j = 1$ if $p''_j \neq 0$ and $q''_j = 0$ otherwise, is in the feasible set of problem (2-2) and $\sum_{j=1}^{n} q''_j = |R(p'')| < |R(p^{(k)})|$, resulting in the same contradiction. Hence there does not exist a mode p' that conforms to *v* with $R(p') \subset R(p^{(k)})$, and we conclude that $p^{(k)}$ is elementary.

It is straightforward to see that $w_k > 0$ owing to its definition and that, after each iteration of the algorithm, $|R(v^{(k)})|$ will decrease by at least 1, i.e. $|R(v^{(k+1)})| < |R(v^{(k)})|$. Thus the algorithm will terminate after a finite number of iterations $K \leq |R(v)|$. □

Characterization of optimal metabolic behaviour using given elementary modes

When calculating the elementary mode decompositions for a range of related flux distributions, as in our MTB application, it is helpful to use only a subset of all elementary modes obtained, since it likely that a subset of the modes suffices to generate valid decompositions for all the distributions. To do so, we select a subset of *K* modes $\{p^{(1)}, \ldots, p^{(K)}\}$ out of all those obtained and use the following approach to determine if they suffice to support all the flux distributions. We successively remove modes from the subset to arrive at one that is minimal in the sense that no additional modes can be removed and still support all the flux distributions.

We represent each elementary flux mode as a column vector in a matrix $P = \left[p^{(1)} \quad \cdots \quad p^{(K)} \right]$ and define a non-negative weight vector $w = [w_1, \ldots, w_K]$ such that a flux distribution $v = Pw$. Substitution of $v = Pw$ into (1) gives a means of finding a biologically-optimal weight vector over the given set of elementary flux modes. Specifically, we solve

$$\begin{array}{ll} \max & f'Pw \\ \text{subject to} & a \leq Pw \leq b, \\ & w \geq 0. \end{array} \qquad (2\text{-}3)$$

If the biomass derived from solving (2-3) corresponds with that from (2-1), we conclude that the given elementary modes are sufficient to characterize the flux distribution of interest, and the given modes are utilized according to the weights w^* obtained from the optimal solution of (2-3).

Implementation of FBA and elementary mode decomposition

FBA and our elementary mode decomposition method were implemented using MATLAB R2010b and Gurobi 4.0. This implementation is available in Appendix B.

Comparison to previous decomposition methods

We used Gurobi 4.0 to solve optimization problem (2-1) to find a biologically optimal flux distribution for *i*NJ661 growing on Middlebrook 7H9. The resulting distribution v contained 507 non-zero components. The reactions j for which $v_j = 0$ were removed from the metabolic network, generating a reduced S that was used as input to efmtool. efmtool version 4.7.1 was used with default parameters in MATLAB R2010b to generate the elementary modes for the reduced metabolic network, resulting in a 507 × 131,558 matrix P containing all the elementary modes. Finally, the quadratic program

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} w_k^2 \\ \text{subject to} \quad & Pw = v, \\ & w \geq 0, \end{aligned}$$

as proposed by Schwartz and Kanehisa (2005) was solved using MOSEK 6.0.0.91 (MOSEK ApS, Copenhagen, Denmark).

efmtool generated the 131,558 elementary modes for the network in 1 minute 48 seconds, while the quadratic optimization step took 32 minutes and 39 seconds, resulting in a total computational time of 34 minutes and 27 seconds. Computations were carried out on the Mac OS X 10.6.4 platform using a computer with an Intel Core 2 Duo 2.53 GHz processor with 4 GB of RAM.

For *i*NJ661v, the biologically optimal flux distribution v obtained by solving optimization problem (2-1) contained 505 non-zero components. Again, a reduced S was generated by removing the reactions j for which $v_j = 0$, and the result used as input to efmtool. With a maximum heap size of 4 GB, efmtool failed before generating all elementary modes, with 657,447 modes generated.

Authors' contributions

DSL and CC conceived the project. DSL and KI designed and implemented the method and performed the computational experiments. DSL, CC, and KI wrote the paper. All authors have read and approved the final manuscript.

Acknowledgements

KI gratefully acknowledges support from the Australian Government through an Australian Postgraduate Award. The authors would like to thank Belinda Chiera and Melissa Yates for careful proofreading of the manuscript.

Source code

Source code for elementary mode decomposition method implemented using MATLAB and Gurobi. The MATLAB scripting is included as Appendix B.

3 Constraint-based modelling of heterologous pathways

This work has been published (Ip et al. 2014):

Ip K, Donoghue N, Kim MK, Lun DS. 2014. Constraint-based modeling of heterologous pathways: Application and experimental demonstration for overproduction of fatty acids in *Escherichia coli*. Biotechnol Bioeng 111(10):2056-66

A publisher permission statement has been included in Appendix A

Additional description and analysis has been added to expand on various aspects of the research described in the above publication.

Abstract

In a wide range of instances, constraint-based modelling has been shown to be useful for metabolic engineering. Constraint-based modelling allows the metabolic behaviour of mutant strains of an organism to be computationally predicted, thus allowing computational methods to efficiently identify genetic manipulation strategies to achieve specific design goals. Constraint-based modelling, however, is based on the premise of applying constraints to preclude certain behaviours, which makes sense for certain genetic manipulations (particularly knockouts and knockdowns), but not as much for others. In particular, when genes (such as those associated with a heterologous pathway) are introduced under artificial control, it is not clear how the constraints should be modified to predict the correct behaviour. In this paper, we introduce a modelling method that we call proportional flux forcing (PFF) to model artificially induced enzymatic genes. The model modifications introduced by PFF can be transformed into a

set of simple mass balance constraints, which allows computational methods for strain optimization based on flux balance analysis (FBA) to be utilised. We applied PFF to the metabolic engineering of *Escherichia coli* (*E. coli*) for free fatty acid (FFA) production—a metabolic engineering problem that has attracted significant attention because FFAs are a precursor to liquid transportation fuels such as biodiesel and biogasoline. We show that PFF used in conjunction with FBA-based computational strain optimization methods can yield non-obvious genetic manipulation strategies that significantly increase FFA production in *E. coli*. The two mutant strains constructed and successfully tested in this work had peak FFA yields of 0.050 g FFA/g carbon source (17.4% theoretical yield) and 0.035 g FFA/g carbon source (12.3% theoretical yield) when they were grown using a mixed carbon source of glucose and casamino acids in a ratio of 2-to-1. These yields represent increases of 5.4-fold and 3.8-fold, respectively, over the baseline strain.

Author Summary

Engineering the metabolism of living organisms is employed to produce a wide variety of useful chemicals, including pharmaceuticals, materials, and biofuels. In such engineering, an effective approach is to use computer modelling to predict outcomes resulting from genetic manipulations and to computationally search for those manipulations that lead to desired outcomes. A particular modelling paradigm that is effective and widely used is constraint-based modelling, but constraint-based modelling has certain difficulties when it comes to modelling genes under artificial expression control. We introduce a modelling method called proportional flux forcing (PFF) that models genes under artificial expression control and that is compatible with existing computational search methods for finding favourable genetic manipulations. We showed the effectiveness of this method by applying it to find genetic manipulations to increase free fatty acid (FFA) production in the bacterium *Escherichia coli*. *E. coli* requires inserting a gene under artificial expression control to produce FFAs, which are precursors to biofuels such as biodiesel and biogasoline. The production of FFAs has therefore attracted significant attention. Using PFF, we found genetic manipulation strategies that increase the total FFA yield of *E. coli* growing on glucose supplemented with casamino acids by up to five-fold.

Introduction

Constraint-based modelling has proved to be a powerful tool for metabolic engineering (Barrett et al. 2006; Curran et al. 2012; Oberhardt et al. 2009; Patil et al. 2004). In myriad cases, constraint-based modelling has been used to help identify metabolic engineering strategies that have been experimentally shown to be effective (Alper et al. 2005a; Alper et al. 2005b; Bro et al. 2006; Fong et al. 2005; Hjersted et al. 2007; Lee et al. 2007; Lee et al. 2005; Park et al. 2007; Wang et al. 2006; Yim et al. 2011). In addition, computational search methods have been developed that efficiently search through the engineering strategy space to identify those that are promising for particular design goals (Burgard et al. 2003; Egen and Lun 2012; Lun et al. 2009; Pharkya and Maranas 2006; Ranganathan et al. 2010; Rocha et al. 2008). In particular among these methods are those based on the OptKnock/OptReg (Burgard et al. 2003; Pharkya and Maranas 2006) framework that select knockouts and gene up- and down-regulations based directly on the predicted behaviour of mutant strains using constraint-based modelling. The effectiveness of these methods therefore depends significantly on the accuracy of the predictions of constraint-based modelling for both mutant and wild-type strains.

The ability of constraint-based modelling to accurately predict knockout phenotypes has been extensively tested and demonstrated (Fong et al. 2005; Segre et al. 2002; Shlomi et al. 2005), and it is straightforward to see how this ability should extend to predict phenotypes resulting from gene down-regulation. But for gene up-regulation and, in particular, for heterologous pathways that are inserted under the control of artificial promoters, it is less clear whether constraint-based modelling gives accurate predictions. The basic rationale of constraint-based modelling is that, rather than attempting to predict cellular behaviour from basic chemical or physics laws, known constraints on the behaviour are successively placed to narrow the space of possibilities (Covert et al. 2003; Palsson 2000). Finally, among the space of remaining possibilities, methods such as flux balance analysis (FBA) (Orth et al. 2010) are used to identify a specific prediction or set of predictions. Given this rationale, the way that knockouts and gene down-regulations should be simulated is clear: these genetic manipulations impose further constraints on the cell by limiting the possible flux through certain pathways. For genes whose expression is artificially induced, however, the simulation method is not clear. It is not that the space of constraints is changed by the induction of these

genes—it is that flux is *forced*. That is, the enzyme produced by the artificially-induced gene competes for substrate with enzymes under native control and forces metabolic flux in specific directions.

The concept of flux forcing for gene up-regulation is modelled in the OptReg framework (Pharkya and Maranas 2006) by finding the predicted metabolic flux range for each reaction in the wild type and simulating an up-regulation by constraining the lower bound of the up-regulated reaction to some range above its wild-type range. This approach has some shortcomings: first, for many reactions (and, indeed, for most heterologous production pathways), the predicted flux of the reaction in the wild type without any forcing is always 0, so it is unclear what the lower bound constraint should be; second, this approach is static in that the lower bound of the up-regulated reaction is fixed and does not respond to the flux changes caused by mutations.

Here, we introduce a constraint-based modelling method we call proportional flux forcing (PFF) that models the effect of substrate competition caused by artificially induced enzymatic genes by forcing a fixed fraction of the flux passing through the substrate into the alternative pathway formed by the induced gene. The basic principles of PFF are illustrated in Figure 3-1, which shows a metabolite *A* that is a substrate for a heterologous reaction R0 with flux v_0. PFF forces a fixed fraction α of the flux flowing into the substrate of artificially induced genes to be redirected. Importantly, the model modifications introduced by PFF can be transformed into a set of simple mass balance constraints. This allows FBA-based computational methods for strain optimization, including OptKnock (Burgard et al. 2003), OptReg (Pharkya and Maranas 2006), GDLS (Lun et al. 2009), and GDBB (Egen and Lun 2012) to be employed in conjunction with PFF. Thus, PFF can be used to model the existence of a heterologous production pathway that we know is needed for a certain engineering objective, and FBA-based computational methods for strain optimization can be used to direct flux toward the substrate (or substrates) of the production pathway.

We demonstrate the effectiveness of PFF for metabolic engineering by applying it to the metabolic engineering of *Escherichia coli* for free fatty acid (FFA) production. FFAs are a precursor of biodiesel and biogasoline (Lennen et al. 2010; Schirmer et al. 2010; Steen et al. 2010), and therefore much recent attention has focused on efficient microbial production of FFAs (Lennen and Pfleger 2012) focused on manipulating

metabolism to boost the production of FFA precursors, to block the downstream consumption of FFA by interrupting FFA oxidation pathways, and even to reverse the β-oxidation pathway. In all cases, FFA production in *E. coli* is typically induced by heterologous expression of either a modified form of the *E. coli* thioesterase *tesA* (Cho and Cronan 1995) or various plant thioesterases (Steen et al. 2010; Voelker and Davies 1994a). The expression of these thioesterases disrupts the fatty acid biosynthesis process and causes FFAs to be produced, which are secreted into the growth media. The thioesterase creates a metabolic sink, allowing FFAs, which would otherwise be incorporated into phospholipids and other cellular components at a limited rate, to accumulate. Thus, heterologous thioesterase expression is important for *E. coli* FFA production, and it is important to model its effect.

We modelled heterologous thioesterase expression in *E. coli* using PFF applied to *i*AF1260 (Feist et al. 2007), a recent genome-scale metabolic reconstruction of *E. coli* K-12 MG1655, and we used GDLS (Lun et al. 2009) to find knockout strategies predicted to increase FFA production. We constructed three knockout strains of *E. coli* K-12 MG1655 based on the results from GDLS: MG1655 Δ*fadE*, PKI8, and PKI10 (see Table 3-1). The knockout of *fadE* has been reported by several authors to increase FFA production in *E. coli* (Steen et al. 2010; Yu et al. 2011). This was one of the knockouts predicted by our analysis, and our implementation of it confirmed the effectiveness of

Strains	**Relevant genotype**			**Reference**
MG1655 Δ*fadE*	MG1655: Δ*fadE*			This study
PKI8	MG1655: Δ*rpe* Δ*sgcE* Δ*talAB*			This study
PKI10	MG1655: Δ*rpe* Δ*sgcE* Δ*talAB* Δ*fadE*			This study
Plasmids	**Replication origin**	**Expressed genes**	**Resistance**	**Reference**
pKS1	p15a	P_{lacUV5}: '*tesA*	Cam	(Steen et al. 2010)
pKD46	oriR101, repA101ts	P_{araB}: *γ-β-exo* (Red recombinase)	Amp	(Datsenko and Wanner 2000)
pKD13	oriR6K-gamma	Template plasmid	Amp, Kan	(Boshoff and Lun 2010; Datsenko and Wanner 2000)
pCP20	pSC101, Rep^{ts}	FLP^{+}, λ $cI857^{+}$	Amp, Cam	(Boshoff and Lun 2010; Cherepanov and Wackernagel 1995; Rawat et al. 2013)

knocking out *fadE* in MG1655. *E. coli* PKI8 is more interesting—it involves the knockout of four genes in the pentose phosphate pathway with no obvious connection to FFA production. These knockouts have not previously been used in connection to FFA production, and our experimental results (see Figure 3-4) show that they do indeed lead to a significant production increase (3.8-fold over the baseline strain). *E. coli* PKI10 combines the knockouts of PKI8 with a *fadE* knockout and is predicted to have increased FFA yield over PKI8. Unfortunately, we were unable to successfully culture *E. coli* PKI10 in defined media.

Although the yields obtained by our knockout strains are lower than those obtained by some recent FFA-producing *E. coli* strains (Lennen and Pfleger 2012), our experimental results nevertheless demonstrate that PFF coupled with FBA-based computational methods for strain optimization can successfully find effective metabolic engineering strategies and is a useful tool for optimizing strains that involve heterologous pathways. This approach may serve as a useful complement to other strain optimization approaches, allowing in particular for us to find genetic manipulations with no obvious connection to the target metabolite.

Materials and Methods

Genome-scale FBA modelling

We work with *i*AF1260, a recent genome-scale model of *E. coli* (Feist et al. 2007). This model consists of three parts. First, from m metabolites and n reactions, we form an $m \times n$ stoichiometric matrix S, whose ijth element S_{ij} is the stoichiometric coefficient of metabolite i in reaction j. Second, the flux distribution v, whose jth element v_j is the flux though reaction j, is constrained by a lower bound vector a, whose jth element a_j is the lower bound of the flux through reaction j, and an upper bound vector b, whose jth element b_j is the upper bound of the flux through reaction j. Finally, the linear objective is formed by multiplying the fluxes by an objective vector f, whose jth element f_j is the weight of reaction j in the biological objective. Thus, a biologically-optimal flux distribution is obtained by solving

$$\begin{aligned} &\max && f'v \\ &\text{subject to} && Sv = 0, \\ &&& a \le v \le b. \end{aligned} \tag{3-1}$$

Proportional flux forcing (PFF)

Suppose we have an artificially induced reaction whose substrate is metabolite i. Let the flux through this reaction be v_0. Moreover, suppose the reactions producing and consuming metabolite i are those in the sets $\Gamma_-(i)$ and $\Gamma_+(i)$, respectively. An example is given in Figure 3-1A with v_1 representing a reaction R1 producing metabolite A and v_2 representing a reaction R2 consuming metabolite A. If the heterologous pathway is represented by reaction R0 with flux v_0, then the PFF constraint is given by

$$v_0 \ge \alpha \left\{ \sum_{j \in \Gamma_-(i)} \max(v_j, 0) + \sum_{j \in \Gamma_+(i)} \max(-v_j, 0) \right\}. \tag{3-2}$$

where $\max(a, b)$ is the maximum of the values a and b. The term $\sum_{j \in \Gamma_-(i)} \max(v_j, 0)$ gives the summation of fluxes for metabolite A producing reactions with positive flux, and likewise, the term $\sum_{j \in \Gamma_+(i)} \max(-v_j, 0)$ gives the summation of fluxes for metabolite A consuming reactions with negative flux (i.e. flux in reverse direction). The complete

summation in the braces is therefore the total flux generating metabolite A, and the PFF constraint simply states that v0 must be at least a fraction α of the total flux through metabolite A.

For use in FBA-based computational methods, we convert this constraint to a set of mass balance constraints. First, we rewrite (3-2) by introducing new variables w_j for all $j \in \Gamma_-(i) \cup \Gamma_+(i)$ such that

$$v_0 = \alpha \sum_{j \in \Gamma_-(i) \cup \Gamma_+(i)} w_j, \tag{3-3}$$

and

$$\begin{aligned} w_j &\geq \max(v_j, 0), \quad && j \in \Gamma_-(i), \\ w_j &\geq \max(-v_j, 0), \quad && j \in \Gamma_+(i). \end{aligned} \tag{3-4}$$

Each of the inequalities in (3-4) can be represented as two inequalities. For $j \in \Gamma_-(i)$, $w_j \geq \max(v_j, 0)$ is equivalent to $w_j \geq 0$ and $w_j \geq v_j$, and for $j \in \Gamma_+(i)$, $w_j \geq \max(-v_j, 0)$ is equivalent to $w_j \geq 0$ and $w_j \geq -v_j$. Finally, we make use of the concept of slack variables (Bertsimas and Tsitsiklis 1997) to express each inequality constraint as an equality constraint and a nonnegative constraint. We introduce slack variables s_j for all $j \in \Gamma_-(i) \cup \Gamma_+(i)$ such that for $j \in \Gamma_-(i)$, we express $w_j \geq v_j$ as $w_j = v_j + s_j$ and $s_j \geq 0$, and for $j \in \Gamma_+(i)$, we express $w_j \geq -v_j$ as $w_j = -v_j + s_j$ and $s_j \geq 0$. Thus, the full set of constraints is given by:

$$\begin{aligned} v_0 &= \alpha \sum_{j \in \Gamma_-(i) \cup \Gamma_+(i)} w_j, \\ w_j &= v_j + s_j, \quad && j \in \Gamma_-(i), \\ w_j + v_j &= s_j, \quad && j \in \Gamma_+(i), \\ w_j, s_j &\geq 0, \quad && j \in \Gamma_-(i) \cup \Gamma_+(i). \end{aligned} \tag{3-5}$$

The mass balance constraints we seek arise naturally from (3-5) if we interpret each equality constraint as a mass balance constraint involving a new ‘fictional’ metabolite.

Figure 3-1A illustrates the additional reactions and the accompanying system of equations for a simple example metabolite *A*.

The mass balance constraints (3-5) can be used in the FBA optimization problem (3-1) and are equivalent to the PFF constraint (3-2). Figures 3-1B and 3-1C demonstrate how the PFF construction works to force a proportion α of the total flux through metabolite *A* to be siphoned off.

In this paper, we applied PFF to simulate the effect of the modified *E. coli* thioesterase '*tesA* (Cho and Cronan 1995) by forcing a fraction of the flux passing through fatty acyl-ACP compounds in *i*AF1260 through the following reaction:

$$\text{fatty acyl-ACP}_{[c]} \rightarrow \text{ACP}_{[c]} + \text{free fatty acid}_{[e]},$$

where the chain length of the free fatty acid is appropriate for the specific fatty acyl-ACP involved in the reaction, and the subscripts indicate the cellular compartment of the compound, with c representing the cytoplasm and e representing extracellular space. The specific fatty acyl-ACP substrates for '*tesA* that we used were octanoyl-ACP (C8:0-ACP), decanoyl-ACP (C10:0-ACP), dodecanoyl-ACP (C12:0-ACP), myristoyl-ACP (C14:0-ACP), cis-tetradec-7-enoyl ACP (C14:1-ACP), palmitoyl-ACP (C16:0-ACP), cis-hexadec-9-enoyl-ACP (C16:1-ACP), octadecanoyl-ACP (C18:0-ACP), cis-octadec-11-enoyl-ACP (C18:1-ACP). These substrates were chosen based on the observed FFA products of '*tesA* (Cho and Cronan 1995).

It should be noted that the biomass growth objective function is ultimately what directs the flow of flux through the metabolic network in this modelling. The optimisation applied in FBA to maximise biomass flux means that firstly, flux will only flow through a particular metabolite if it is coupled to the production of biomass / NGAM, and secondly, flux will be directed to minimise flux lost from biomass due to syphoning by a heterologous reaction. The *fadE* gene deletion modelling example discussed below in the results demonstrates both of these cases. The forcing reactions introduced into the model are entirely dependent on the heterologous reaction being modelled.

.

Figure 3-1: Basic principles of Proportional Flux Forcing

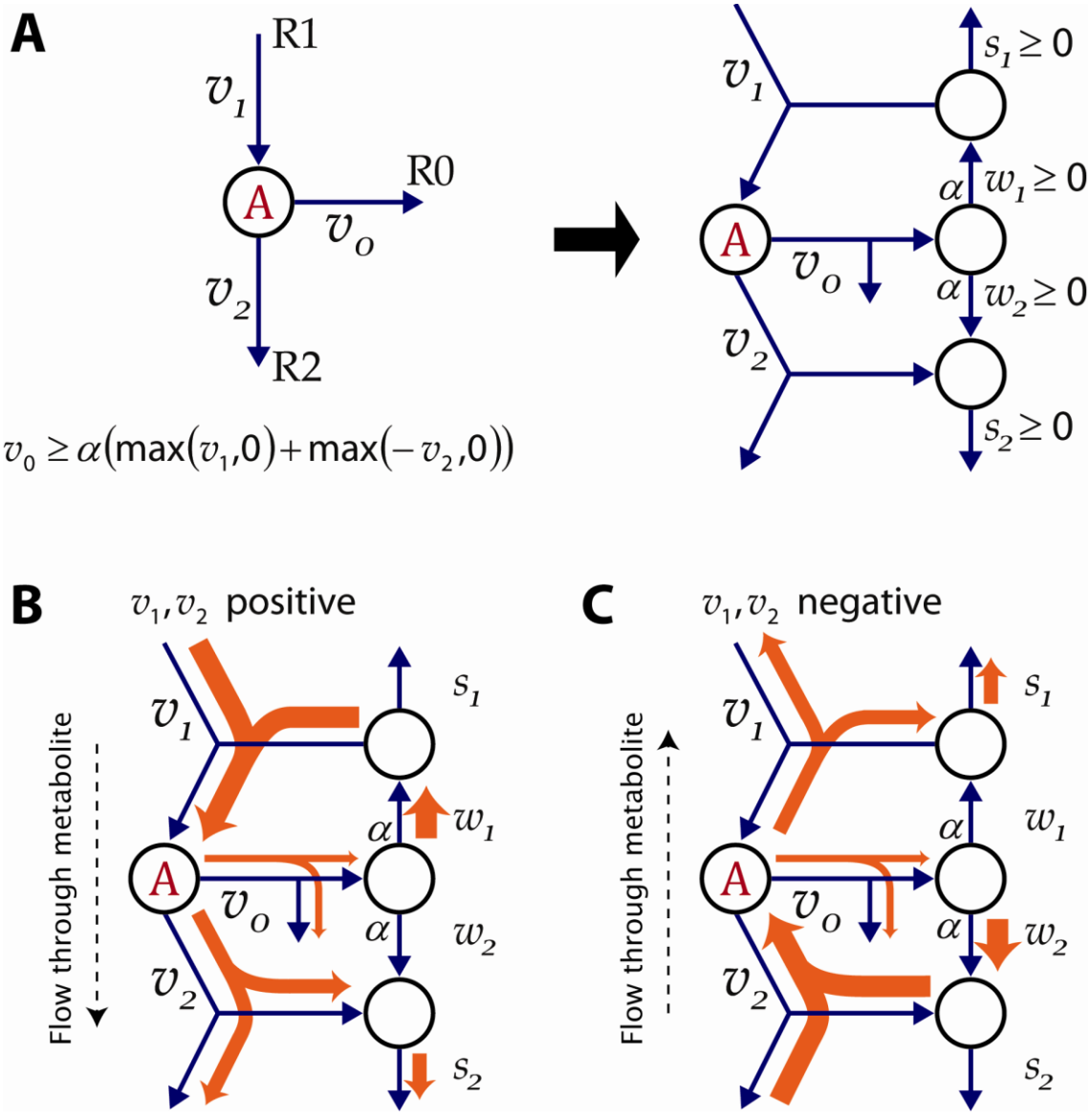


Figure 3-1 (cont.):
(A) The basic principles of proportional flux forcing (PFF) are illustrated through an example. Suppose a metabolite *A* is natively produced by reaction R1 with flux v_1 and consumed by reaction R2 with flux v_2, each of which may be reversible (i.e. v_1 and v_2 may be negative). Now suppose reaction R0 is artificially induced and carries flux v_0. We therefore impose the constraint shown that constrains v_0 to be at least some fraction α of the total flux flowing into *A*. This constraint can be transformed into a set of mass balance constraints for use in FBA-based computational methods, which results in the transformed network shown on the right. This transformation is described in detail in Materials and Methods. **(B)** When v_1 and v_2 are both positive, the action of the PFF construct to siphon off some fraction of the flux through the metabolite is given by the orange arrows. **(C)** In a similar fashion, flux is siphoned off when the flux through the metabolite is reversed and both v_1 and v_2 are negative.

GDLS analysis

GDLS version 1.11 (available at http://crab.rutgers.edu/~dslun/gdls) was applied to a modified model of *i*AF1260 that included the PFF modifications for '*tesA*. In the model, a lower bound of 0.3 h^{-1} was applied to the biomass flux to ensure viable mutants. GDLS was run with the following search parameters: the maximum number of (gene association) knockouts allowed *C* set to 20, the number of search paths stored *M* varied from 1 to 5, and the search neighbourhood size *k* varied from 1 to 3 (Lun et al. 2009).

Construction of knockout mutants

Knockout strains of *E. coli* K-12 MG1655 were constructed by λ Red mediated recombination (Datsenko and Wanner 2000). Oligonucleotide primers used in this study were ordered from GeneWorks Pty Ltd (Hindmarsh, South Australia, Australia) and are listed in Table 3-2. For each knockout target, the forward and reverse primers for the gene were used to amplify a linear cassette with plasmid pKD13 as the template. Recombination was induced using λ Red recombinase expressed from plasmid pKD46. The *kan* antibiotic marker was removed by expressing FLP recombinase encoded on plasmid pCP20. This λ Red mediated recombination process was repeated multiple times to obtain strains with multiple knockouts.

PKI8 pKS1 was adapted to grow on minimal media by subculturing at 1:1000 in baffled flasks over several hundred generations. Multiple PKI8 pKS1 replicates were maintained in parallel during this adaptation process. After approximately 150 and 300 generations, preliminary measurements of the potential FFA yields were taken for cultures inoculated from single colonies on MOPS minimal media plates, and the best producing strains were selected for further adaptation.

Fermentation procedures

For FFA yield experiments, strains were cultivated in MOPS minimal media (M2106, Teknova, Hollister, CA) supplemented with 25 mgL^{-1} chloramphenicol and 0.2% glucose and 0.1% casamino acids (J851, Amresco, Solon, OH) as the mixed carbon source. Starter cultures of the final representative strains were inoculated from single colonies on MOPS minimal media plates. Fermentation cultures were inoculated at 1:100 from starter cultures at OD600 ~ 0.5 and grown in baffled flasks kept at 37°C and shaken at 250 rpm.

Fatty acid analysis

Fatty acid analysis was performed as described by Voelker and Davies (Voelker and Davies 1994a). Briefly, their method converts FFAs into their corresponding fatty acid methyl esters (FAMEs), which are easier to store and analyse. A typical procedure is as follows:200 µL of a mixture of 0.5 mg/mL heptadeconoic acid (as an internal standard) in acetic acid was added to 5 mL liquid cultures that was then partitioned with 10 mL of a 1:1 (vol/vol) methanol:chloroform mixture. The lower chloroform layer was extracted and evaporated to near dryness using a nitrogen evaporator, redissolved in 1 mL of 5%

Table 3-2 Oligonucleotide primers.

Primer name	**Sequence (5'–3')**
fadEKO_fw	TCATCACAAGTGGTCAGACCTCCTACAAGTAAGGGGCTTT TCGTTATGATGTGTAGGCTGGAGCTGCTTC
fadEKO_rc	AAACGGAGCCTTTCGGCTCCGTTATTCATTTACGCGGCTT CAACTTTCCGATTCCGGGGATCCGTCGACC
rpeKO_fw	AGGAGTCGTTTCACCCGCGAAAAAATAATTCTCAAGGAG AAGCGGATGAAGTGTAGGCTGGAGCTGCTTC
rpeKO_rc	CAAAAGCGACGCCGCGAATATCTTCAAACTTATTCATGAC TTACCTTTGCATTCCGGGGATCCGTCGACC
sgcEKO_fw	AGACGATGCACTGTTTGCGCTGGTATCAGGTTAACGGAGG CGCTTATGATGTGTAGGCTGGAGCTGCTTC
sgcEKO_rc	TCTGTTGGCTCATCAGAAACAATCTCCTCTTACAGGGTGG CGAATTGAGCATTCCGGGGATCCGTCGACC
talAKO_fw	ATTAACGCACTCATCTAACACTTTACTTTTCAAGGAGTATT TCCTATGAAGTGTAGGCTGGAGCTGCTTC
talAKO_rc	TTCGGGACATATAACACTCCGTGGCTGGTTTATAGTTTGG CGGCAAGAAGATTCCGGGGATCCGTCGACC
talBKO_fw	ACCGGTTACATCCCCCTAACAAGCTGTTTAAAGAGAAATA CTATCATGACGTGTAGGCTGGAGCTGCTTC
talBKO_rc	GACCGACTTCCCGGTCACGCTAAGAATGATTACAGCAGAT CGCCGATCATATTCCGGGGATCCGTCGACC

(vol/vol) H_2SO_4 in methanol, and incubated for two hours in a 90°C water bath. Finally, 1 mL of 0.9% (wt/vol) NaCl in water was added followed by 300 µL of hexane. After thorough mixing, the top hexane layer was collected for analysis by gas chromatography (GC).

GC measurements were performed on a PerkinElmer (PE) Clarus 580 GC (PerkinElmer, Waltham, MA) equipped with a flame-ionization detector (FID) and a 30 m wax column (PE-WAX ETR-30M-.25MM-.25UM, PerkinElmer, Waltham, MA). The temperature program used was as follows: initial temperature of 100°C, held for 1 minute, then ramped at 30°C/min to a final temperature of 250°C and held for 8 minutes. The GC was calibrated using standards of C8–C22 FAMEs (18920-1AMP, Sigma-Aldrich, Saint Louis, MO) and C4-C24 FAMEs (Supelco 37 Component FAME Mix, Sigma-Aldrich, Saint Louis, MO), and integration analysis was performed using the TotalChrome software package (PerkinElmer, Waltham, MA).

This assay for quantifying fatty acids evaluates the total lipid content as opposed to free fatty acid content only, but it has been observed that most "excess" fatty acyl groups that accumulate in *E. coli* do so as FFAs (Davis et al. 2000; Lu et al. 2008a; Voelker and Davies 1994b). Thus, for the purposes of calculating theoretical yields, we assumed the FFA product was myristic acid—the FFA corresponding to the peak fatty acid product of our strains. We calculated theoretical yields using *i*AF1260 assuming a literature-derived amino acid composition for the casamino acids (Street et al. 1960). In this manner, we calculate the maximum yield of myristic acid from 1 g of our glucose and casamino acid carbon source mixture as 1.25 mol (0.28 g).

Results/Discussion

Computational results of coupled PFF/GDLS analysis

GDLS analysis for the PFF augmented *i*AF1260 genome-scale metabolic model was performed using a range of parameter settings (see Materials and Methods). Collated results for gene knockout strategies to optimize for fatty-acid production are shown in Table 3-3. The genes and corresponding reactions selected in the predicted strategies are involved in a range of metabolic pathways. Some of these deletion targets make sense as modifications to increase FFA production; either by interrupting a degradation pathway for the target fatty-acid products (*fadB*, *fadD*, *fadE* or *fadK*) or by eliminating

the production pathways of alternative products such as lactate (*dld* or *ldhA*). Indeed, *fadB*, *fadD*, and *fadE* have all been experimentally shown to increase FFA production in *E. coli* (Lu et al. 2008b; Steen et al. 2010; Youngquist et al. 2012). Other targets are less obvious. In particular, the genes *rpe*, *sgcE*, *talA*, and *talB* all lie in the pentose phosphate pathway with no clear connection to FFAs. The three sets of genetic modifications shown in bold in Table 3-3 were selected for experimental implementation and further analysis: *fadE* to examine disruption of the degradation pathway; *rpe*, *sgcE*, *talA*, and *talB* to examine disruption of the pentose phosphate pathway; and the combination of both.

For the results shown in Table 3-3, the thioesterase '*tesA* is modelled with the forcing factor α set to 0.2 across all its substrates. This means that 20% of the fluxes flowing through all the substrates of '*tesA* were redirected into the reactions catalysed by the '*tesA* product. An α value of 0.2 seemed to be a reasonable choice based on the observed effects of '*tesA* on growth rate and FFA production (Cho and Cronan 1995; Steen et al. 2010), and for the purpose of finding genetic modifications to increase production accumulation, the specific value of α that is chosen is not of great importance. We have included the results of searches performed using a variety of α values ranging from 0.1 to 0.4 in the Appendices (see Appendix C, Tables C-1, C-2, and C-3). While the specific flux values differ for each value of α, the single reaction knockout strategies found by GDLS propose deleting genes associated with reactions utilized in a pathway from C14:1-ACP to C14:0-ACP. Essentially, these are all equivalent to the *fadE* deletion mutant that we chose to study (see Figure 3-3A). Likewise, the double reaction knockout strategy involving the deletion of four genes related to the pentose phosphate pathway is found for all values of α.

Table 3-3 Knockout strategies from GDLS analysis of *i*AF1260 with PFF model of '*tesA* with α = 0.2. The strategies chosen for experimental validation are highlighted in bold.

Growth rate ($g\ h^{-1}$)	**FFA flux ($mmol\ gDW^{-1}\ h^{-1}$)**	**Number of knockouts**	**Number of gene associations**	**Gene associations***
0.838	0.336	0	0	None (wild type)
0.828	0.397	**1**	**1**	***fadE***
		2	1	(*fadK* OR *fadD*)
		1	1	*fadB*
		1	1	*tesB*
		1	1	*aas*
		3	2	*aas*, (*ldhA* OR *dld*)
		2	2	*aas*, *glpK*
0.592	0.553	**4**	**2**	**(*rpe* OR *sgcE*),(*talB* OR *talA*)**
0.585	0.618	**5**	**3**	***fadE*, (*rpe* OR *sgcE*), (*talB* OR *talA*)**
		5	3	*tesB*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*aas*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*fadB*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		7	4	*aas*, (*nudI* OR *dut*), (*rpe* OR *sgcE*), (*talB* OR *talA*)
		8	4	(*fadK* OR *fadD*), (*focA* OR *focB*), (*rpe* OR *sgcE*), (*talB* OR *talA*)

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

That some amount of flux is forced (i.e. $\alpha > 0$) is, however, necessary to achieving the predictions we found. If there is no flux forcing (i.e. $\alpha = 0$), the PFF inequality constraint (3-2) is reduced to a simple non-negative constraint (i.e. $v_0 \geq 0$). In this case, GDLS does not find any knockout strategies that increase FFA production for the iAF1260 model augmented with the heterologous pathway, and no FFA flux is predicted by FBA regardless of the knockouts performed. GDLS was again run using neighbourhood search sizes up to $k = 3$, which is the limit of what we searched due to computational running time. The reason for this behaviour is that if there is no flux forcing, then the maximization of biomass in FBA always results in carbon flux being directed entirely toward biomass production, as opposed to FFA production. This makes sense and highlights the fundamental reason that PFF is necessary: while for certain overproduction targets (e.g., ethanol, acetate, succinate) it is possible to couple biomass production and target production (Feist et al. 2010) so that production of the target is a necessary by-product of growth, for others (e.g., FFA) this appears not to be possible and instead, the goal of engineering is to direct flux toward the key substrates of a certain production pathway. In our case, we have essentially set up an optimization that allows us to find genetic manipulation strategies that direct flux to fatty acyl-ACP (fatty acyl-acyl carrier protein) compounds, which are the substrates of the modified thioesterase '*tesA*.

Computational analysis of suggested engineering strategies

The gene deletion strategies that we consider in this study are comprised of two basic sets of gene deletions: a single deletion of *fadE*, and a quad-deletion of *rpe*, *sgcE*, *talA*, and *talB*.

The first set of gene deletions is a single deletion of *fadE*, which is associated with acyl-CoA dehydrogenase (*i*AF1260 reactions: ACOAD1f – ACOAD8f). These reactions are associated with the fatty-acid beta-oxidation pathway, and typically, their disruption is considered to block the degradation pathway for the desired FA compounds. Our modelling shows that some of these reactions, specifically ACOAD6f – ACOAD8f, are used in pathways to convert C14:1-ACP, C16:1-ACP and C18:1-ACP to C14:0-ACP, C16:0-ACP and C18:0-ACP, respectively.

An example of this pathway from C14:1-ACP to C14:0-ACP is shown in Figure 3-2A. These pathways provide alternatives to the standard FA elongation pathway to generate

the longer even-chain saturated fatty acyl-ACPs (see Figure 3-2B). In the wild-type strain (see Figure 3-2C), the predicted flux distribution shows that these alternative pathways are used to minimize the flux directed through fatty acyl-ACP '*tesA* substrates (see Materials and Methods for the specific fatty acyl-ACP substrates for '*tesA* used in our model). In the *fadE*-deletion strain (see Figure 3-2D), the alternative pathways are interrupted and metabolic flux is redirected via C10:0 and C12:0 fatty acyl-ACP '*tesA* substrates.

The second set is a quad-deletion of *rpe* and *sgcE*, which is associated with ribose 5-phosphate 3-epimerase (*i*AF1260 reaction: RPE), and *talA* and *talB*, which is associated with transaldolase (*i*AF1260 reaction: TALA). The pentose phosphate pathway, which contains these two reactions, is primarily used for producing NADPH and precursors, such as erythrose-4-phosphate (E4P), for aromatic amino acids. These two particular functions can be seen in the metabolic flux distribution for the wild-type strain with pathways through the pentose phosphate pathway for the production of E4P (see Figure 3-3A) and the generation of NADPH (see Figure 3-3B). The effect of removing reactions RPE and TALA is seen in Figure 3-3C. It forces reaction TKT2 (transketolase) to run in the reverse direction in order to supply the required E4P for biomass components and generates sedoheptulose 7-phosphate (S7P) as a by-product. This S7P is in turn converted to core oligosaccharide lipid A via a pathway that consumes C12:0-ACP and C14:0-ACP and hence requires increased fatty acid biosynthesis. Furthermore, if the direction of TKT2 is reversed, the pentose phosphate pathway no longer has the ability to generate NADPH. More flux is directed through glycolysis, which improves the efficiency of NGAM production, but more of the NADH generated by the tricarboxylic acid cycle is converted to NADPH to supplement the modified pentose phosphate pathway. These effects, combined with the increased flux ultimately directed towards fatty acid biosynthesis, result in a significantly reduced biomass growth rate. The effect on the flux distribution through central carbon is summarized in Figure 3-3D.

In short, whereas the *fadE* deletion redirects flux through fatty acyl-ACP substrates in the fatty acid biosynthesis pathway, the modifications to the pentose phosphate pathway instead introduce a metabolite that consumes fatty acyl-ACP substrates and is coupled to biomass production. The two sets of reactions are not involved in the same pathways and the mechanisms by which they each alter metabolic flux are distinct. The

combination of both sets essentially blends the effect of both strategies to simultaneously introduce a biomass dependent by-product and redirect flux through FA biosynthesis, which gives rise to the increased predicted production of the five gene deletion PKI10 strain.

Figure 3-2: The effect of the *fadE* gene deletion on metabolic pathways in *i*AF1260 for FFA and biomass accumulation.

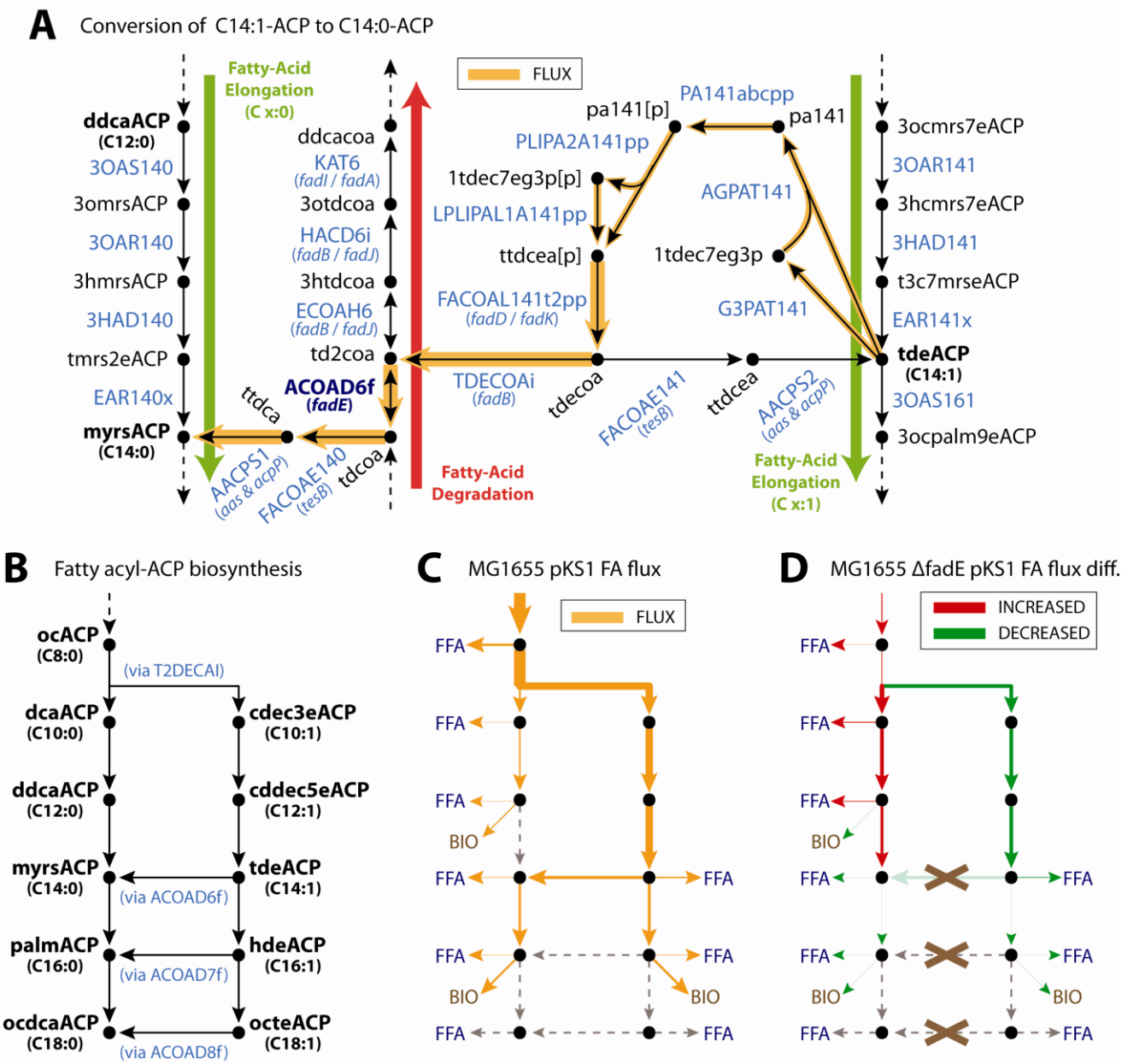


Figure 3-2 (cont.):
(A) The metabolic pathway (orange) present in *i*AF1260 that produces C14:0-ACP (myrsACP) from C14:1-ACP (tdeACP) is shown in relation to the fatty-acid beta-oxidation (red) and fatty-acid elongation (green) pathways. The metabolite names, reaction names and gene associations are taken directly from the model. The *fadE* gene, which is highlighted in bold, is one of several gene deletion candidates identified in our modelling (see Table 2) that are associated with reactions in this pathway. **(B)** A simplified metabolic network outlining fatty acid biosynthesis pathways for producing fatty acyl-ACPs available in *i*AF1260. **(C)** The predicted metabolic flux distribution through fatty acid biosynthesis in the wild-type strain model for a given rate of glucose uptake. The flux distribution has been overlaid on the network outline shown in panel (B). Flux directed towards biomass components (BIO in brown) and FFA production (FFA in blue) derived from these fatty acyl-ACP substrates are also shown. Arrow widths are proportional to pathway flux. **(D)** The changes in the predicted metabolic flux distribution shown caused by deleting the *fadE* gene. Red arrows indicate an increase in flux, while green arrows indicate a decrease. The interruption of the alternative biosynthesis pathways causes a redirection of flux through C10:0-ACP and C12:0-ACP, which are substrates of the modelled '*tesA*. This redirection results in a small increase in overall flux entering fatty acid biosynthesis but a slight decrease in biomass production due to the increased ratio of flux diverted to FFA production.

Figure 3-3: The effect of removing reactions RPE and TALA on the pentose phosphate pathway and FFA production in *i*AF1260.

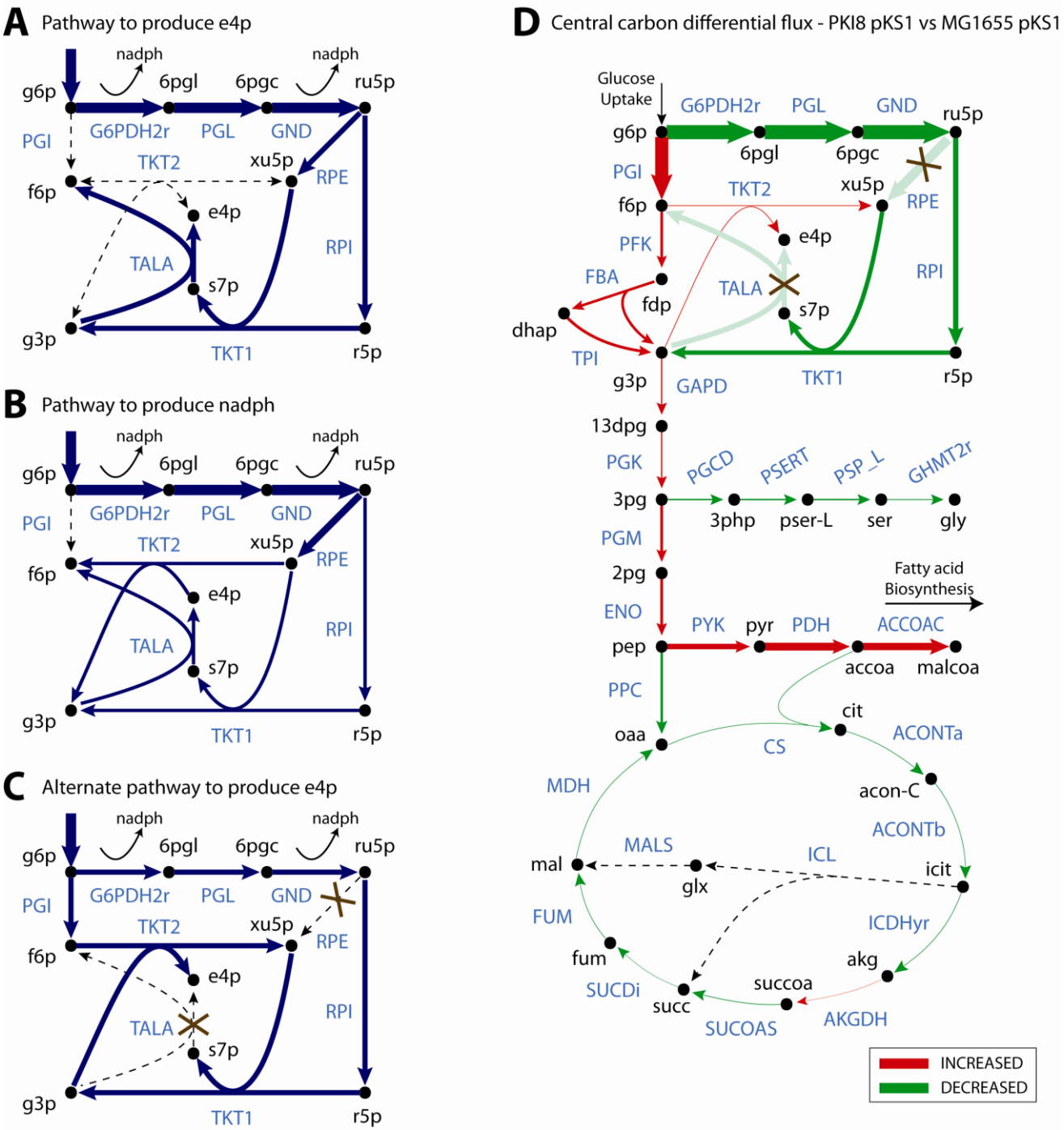


Figure 3-3 (cont.):
(All metabolite and reaction names are taken directly from the *i*AF1260 model. Red arrows indicate an increase in flux while green arrows indicate a decrease, and arrow widths are proportional to reaction flux.) The complete pentose phosphate pathway includes pathways that produce biomass precursors and generate NADPH for biosynthetic pathways such as fatty acid elongation. The removal of reactions RPE and TALA alters the available pathways and forces flux to be redirected to satisfy growth requirements. **(A)** The pathway used by the wild-type strain model to produce erythrose-4-phosphate (E4P), a biomass precursor for aromatic amino acids. **(B)** The pathway used by the wild-type strain model to produce the required NADPH for biosynthetic pathways. **(C)** In the absence of reactions RPE and TALA, the pathway used by the mutant strain to produce the required E4P instead relies on the reversal of the reaction TKT2. This pathway, however, generates a by-product, sedoheptulose 7-phosphate (S7P), which is converted to core oligosaccharide lipid A via a pathway that consumes fatty acyl-ACPs and requires increased fatty acid biosynthesis. In short, this mutation effectively increases the coupling between biomass production and fatty-acid biosynthesis. **(D)** The resulting changes in predicted metabolic flux through central carbon metabolism show redirected flux through glycolysis improving NGAM efficiency, but more of the NADH generated by the tricarboxylic acid cycle being converted to NADPH to compensate for the modified pentose phosphate pathway. When combined with the increased flux ultimately directed towards fatty acid biosynthesis via malonyl-CoA, the deletions result in a significantly reduced biomass growth rate.

Computational results for combined PFF/GDLS using iJO1366

The updated genome-scale *E. coli* model *i*JO1366 was published while we were performing our experimental work and thus was not available when we did our initial modelling. We repeated the implementation of PFF in *i*JO1366 and the results from our GDLS analysis are shown in the Appendices. Tables C-4, C-5, C-6 and C-7 give the PFF/GDLS analysis results for the *i*JO1366 model with values of $\alpha = 0.1, 0.2, 0.3$ and 0.4. The proposed strategies are broadly similar to those produced using the *i*AF1260 model. The *i*JO1366 model reclassifies some reactions in fatty-acid metabolism as reversible—specifically of interest, ACACT (formerly KAT), HACD, and CTECOAI (formerly TDECOAI, HDCOAI and ODECOAI)—and metabolic flux through fatty-acid biosynthesis is thus redirected when compared with that of *i*AF1260. Single reaction knockout strategies, however, are still predominantly focused on deletions in the fatty-acid degradation pathway.

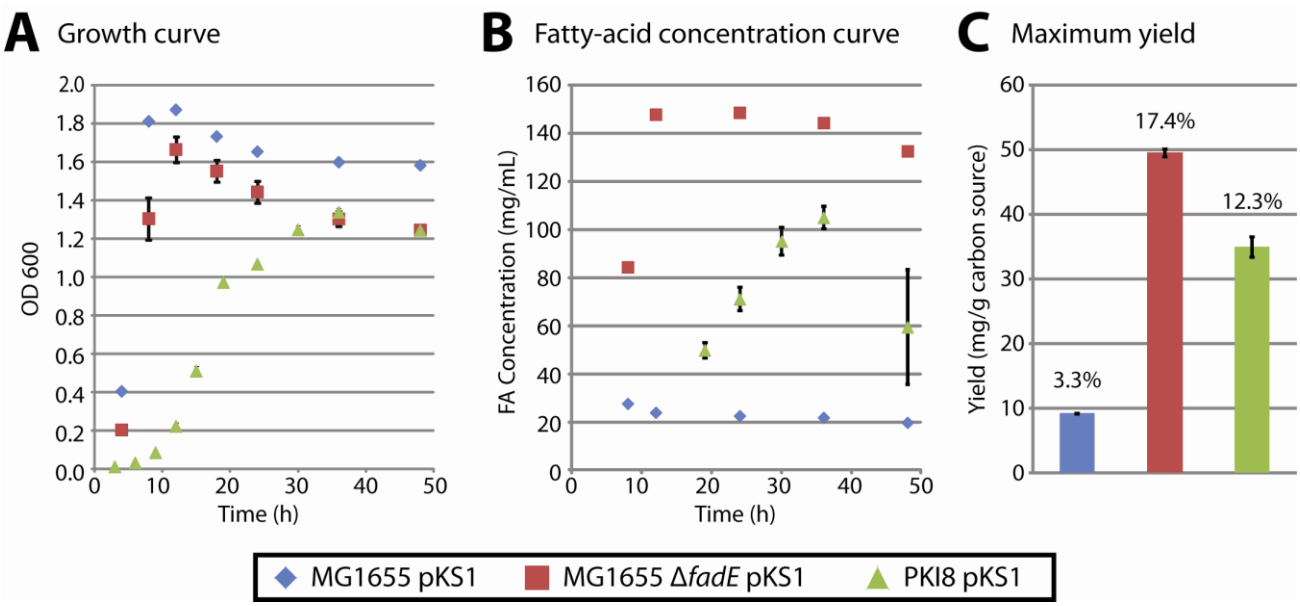
The double reaction knockout from the pentose phosphate pathway of RPE and TALA is no longer found as a separate deletion strategy. This is due to an alternative pathway created by the introduction of two reactions FBA3 and PFK3 in *i*JO1366. Interestingly, the results from the GDLS analysis do propose the deletion of RPE, TALA and one of FBA3 or PFK3 as a subset of a larger deletion strategy, which essentially reproduces the effect of the RPE-TALA deletion strategy found in *i*AF1260. Our experimental results nevertheless do indicate that the RPE-TALA deletion mutant (PKI8 pKS1) does indeed demonstrate a significant increase in FFA production.

Experimental results from implementation of suggested engineering strategies

Plasmid pKS1—which allows inducible expression of '*tesA*, the modified *E. coli* thioesterase that creates a metabolic sink for FFAs—was inserted into wild type MG1655 and three knockout strains suggested by our coupled PFF/GDLS analysis: MG1655 Δ*fadE*, PKI8, and PKI10 (see Tables 3-1 and 3-3). The strains were cultured in MOPS minimal media supplemented with 0.2% glucose and 0.1% casamino acids. Casamino acids were added to assist with the growth rate of the four and five gene deletion strains, as the modifications to the pentose phosphate pathway are known to affect the production of essential amino acids and inhibit growth rates in minimal media (Flores et al. 2004; Lyngstadaas et al. 1998; Sprenger 1995). '*tesA* expression was induced at OD600 ~ 0.4 by the addition of 1.0 mM isopropylthiogalactoside (IPTG). For up to 48 hours, samples were periodically removed to measure optical density and FFA titre (see Figure 3-4). We measured total fatty acids and consider them as good proxy for FFAs, since excess fatty acids in *E. coli* generally accumulate as FFAs (Davis et al. 2000; Lu et al. 2008b; Voelker and Davies 1994a). Although PKI10 pKS1 was predicted to have only slightly impaired growth rate compared to PKI8 pKS1, it did not grow in the supplemented MOPS minimal media in our experiment for unknown reasons. The remaining three strains all grew in the supplemented MOPS minimal media.

The peak FFA titre of MG1655 Δ*fadE* pKS1 and PKI8 pKS1 were significantly greater than that of MG1655 pKS1 (5.4-fold and 3.8-fold greater, respectively). The peak FFA titre of MG1655 Δ*fadE* pKS1 and PKI8 pKS1 were both reached within 24 hours after induction, corresponding to yields of 0.050 g FA/g carbon source (17.4% theoretical yield) and 0.035 g FFA/g carbon source (12.3% theoretical yield) respectively. As described in Materials and Methods, the calculated theoretical yield accounts for both the glucose and the supplementary casamino acids as the combined carbon source.

Figure 3-4: Biomass and FA accumulation of
MG1655 pKS1, MG1655 Δ*fadE* pKS1, and PKI8 pKS1.



The strains were grown in MOPS minimal media supplemented with 0.2% glucose and 0.1% casamino acids, and '*tesA* expression was induced at OD600 ~ 0.4 by addition of 1.0 mM isopropylthiogalactoside (IPTG). Measurements were taken at various times from 0 to 48 hrs after inoculation. Error bars represent the standard error of measurement of triplicate cultures. **(A)** The growth curves of the three strains as measured by optical density. **(B)** The corresponding FA concentration in culture of each strain. **(C)** The peak yields per gram of carbon source of each strain (MG1655 pKS1 at 8hrs; MG1655 Δ*fadE* pKS1 at 24hrs; PKI8 pKS1 at 36hrs) are shown together with their respective percentage of the maximum theoretical yield, which is calculated as described in Materials and Methods.

Based on previous work by other authors (Steen et al. 2010; Yu et al. 2011), we expected MG1655 Δ*fadE* pKS1 to have elevated FFA production compared to the wild type, and this was confirmed in our experimental results. PKI8, however, contains knockouts of four genes in the pentose phosphate pathway that have no obvious connection to FFA production and that have not previously been examined for this purpose. We found that PKI8 pKS1 did indeed have a significantly increased FFA yield compared to MG1655 pKS1 and moreover, as predicted, had a slower growth rate. That FFA titre and biomass are lower in PKI8 pKS1 compared to MG1655 Δ*fadE* pKS1 during the first 36 hours after induction is consistent with the lower predicted growth rate of PKI8 pKS1, but the lower growth rate also suggests a slower nutrient depletion rate, which would suggest the possibility of the strain ultimately producing more FFA than MG1655 Δ*fadE* pKS1. The growth and FFA production of PKI8 pKS1, however, appeared to stop around 24 hours after induction. At this time, both the FFA yield and optical density of the culture were lower than that of MG1655 Δ*fadE* pKS1 at its production peak, suggesting the possibility that the carbon source was not completely consumed. Measurements of glucose concentration (data not shown), however, showed that all strains consumed greater than 90% of the initial glucose by the time of their production peaks, indicating that the glucose at least was almost fully consumed. The peak FFA titre of PKI8 pKS1 nevertheless significantly exceeded that of the baseline strain, MG1655 pKS1.

Taken together, our experimental results show that PFF coupled with FBA-based computational methods for strain optimization can identify knockout targets, which may or may not be clearly linked with FFAs, for increasing FFA production. We have chosen, in this paper, to demonstrate the effectiveness of PFF for increasing *E. coli* FFA production, but it is a broadly applicable modelling method that can be used in any metabolic engineering problem that involves the insertion of artificially-induced genes. Moreover, while we have chosen to focus on knockout interventions, we can also consider gene up- and down-regulations by using PFF in conjunction with the OptReg framework.

Comparison of PFF with OptForce and FSEOF

OptForce (Ranganathan et al. 2010) and flux scanning based on enforced objective flux (FSEOF) (Choi et al. 2010) are computational methods for identifying metabolic engineering strategies for targeted overproduction based on constraint-based modelling.

Both methods differ significantly from the approach we take. As mentioned in the Results section on Computational results of coupled PFF/GDLS analysis, without some form of flux forcing, no objective flux is predicted by FBA regardless of the knockouts performed. Tools such as OptForce and FSEOF attempt to get around this by placing an artificial floor on the flux through the desired reaction and then searching for reaction fluxes in the rest of the network that need to change (either up or down) in order to satisfy this flux forcing requirement.

Specifically, OptForce looks at the predicted flux range for each reaction in the wild type strain and a hypothetical optimum overproducing strain and compares the two sets of flux ranges. Based on these flux ranges, it determines those reactions whose flux must change in an effective overproducing strain and, as a consequence, determines suggestions for engineering interventions. Similarly, FSEOF looks at the predicted intracellular flux distribution in the baseline strain and in a series of hypothetical overproducing strains between the baseline and the optimum. Based on the predicted intracellular flux changes among these strains, it identifies reactions whose fluxes increase as product flux is increased and uses these reactions to identify up-regulation targets.

In short, OptForce and FSEOF do not attempt to directly predict the effect of the insertion of heterologous genes nor the effect of genetic manipulations such as knockouts, gene down-regulations, and gene up-regulations on the growth and synthetic objective. Instead, they attempt to answer the question: given the desired behaviour (i.e. targeted level of overproduction), what modifications are necessary throughout the rest of the network to achieve it?

In contrast, PFF attempts to directly model the behaviour of the inserted heterologous genes—in this case, the effect of introducing a heterologous thioesterase in *E. coli*—by representing their effect as a set of mass balance constraints without setting a fixed minimum bound. This allows the application of computational methods that search through possible engineering strategies to find those with the greatest predicted production of the desired metabolite. In essence, this approach attempts to answer the question: given the predicted behaviour of the inserted heterologous genes, what modifications are sufficient to achieve overproduction?

OptForce has been shown to be effective for FFA production in *E. coli* (Ranganathan et al. 2012), while FSEOF has been shown to be effective for lycopene production in E. coli (Choi et al. 2010). In our study, we chose to focus exclusively on knockout interventions, while OptForce looked at knockouts and gene up- and down-regulations, and thus, the overall sets of genetic manipulations have little overlap. Nevertheless, some genetic manipulations such as the deletions of gene *fadE* (reaction ACOAD) or of reaction TALA, in combination with other manipulations, are suggested by both methods. Comparing the results of OptForce, FSEOF, and PFF in depth both for FFA production and in general is an interesting avenue for future work.

Acknowledgments

We thank Nathaniel Morrison, Andrea Paul, In Young Seo, and Fiona Wademan for laboratory assistance. We thank G. Charles Dismukes (Rutgers University) and Albert Juhasz (University of South Australia) for use of their GCs and for technical assistance with GC analysis. We thank Jay D. Keasling for the gift of plasmid pKS1.

4 Conclusion

Metabolic flux modelling is useful for interpreting metabolic capabilities and to search for engineering targets such as genetic manipulations to improve production efficiencies or drug targets for disease treatment. This research considered two issues related to modelling metabolism at a whole organism level and the use of constraint based modelling techniques to predict metabolic flux phenotypes.

The first issue concerns the complexity of metabolism. Complex networks can often be decomposed into smaller sub-networks which can be studied to help understand the action of the complete network. In the case of metabolic flux networks, one useful strategy is to decompose a metabolic flux network into elementary flux modes which are a systematic means of organising metabolic networks into potential pathways associated with biochemical functions. The number of elementary flux modes, however, grows exponentially with the size of the network and for genome-scale networks the number of modes can be prohibitive if decomposition requires generation of all possible modes beforehand.

In this research, we developed a new method for decomposition of metabolic flux distributions into elementary flux modes. We demonstrated the utility of our decomposition method with two examples. For the metabolic engineering of *Escherichia coli*, the obtained decompositions associated with acetate-enhancing knockout strategies reveal two important things: that the increased acetate flux is due to the knockouts forcing flux onto alternative pathways for producing the necessary energy and biomass requirements for viability and these pathways produce acetate as a side-product; and also that finding complementary modes is more important than finding a single optimal elementary mode. For understanding the survival of *Mycobacterium tuberculosis* during infection, the decomposition analysis highlights the effect of available modes on viability at low octadecanoate uptake rates, and also identifies a

potential source of discrepancy between *in silico* predictions and observed experimental results that should be experimentally testable.

Our decomposition method can easily operate on large, genome-scale networks since the generation of all relevant modes of the metabolic network is not required prior to decomposition. In particular, it can achieve computational time improvements exceeding 2000-fold and requires only a few seconds to generate decompositions on genome-scale networks. It should also increase the accessibility of elementary mode decomposition for constraint-based modelling research, allowing the potential benefits of elementary mode analysis, including enhanced insight into metabolic capabilities and more sophisticated design and engineering, to be exploited more widely.

The second issue concerns the effect of manipulations on the modelling. Predictions are most useful when the effects of manipulations are captured correctly in the model. In the case of inserted heterologous genes under artificial control, these artificially induced enzymatic genes effectively create substrate competition by forcing a fraction of flux passing through the substrate into the alternative pathways formed by the induced genes. With constraint-based modelling techniques, it is not immediately clear how to model this flux forcing behaviour.

In this research, we have developed a method which we call Proportional Flux Forcing (PFF) to model artificially induced enzymatic genes within constraint-based modelling schemes. We applied PFF in conjunction with FBA-based computational strain optimisation to yield non-obvious genetic manipulation strategies that significantly increase total free fatty acid yield in *E. coli* with an inserted heterologous thioesterase. The two mutant strains constructed and successfully tested in this work had peak FFA yields of 0.050 g FFA/g carbon source (17.4% theoretical yield) and 0.035 g FFA/g carbon source (12.3% theoretical yield) when they were grown using a mixed carbon source of glucose and casamino acids in a ratio of 2-to-1. These yields represent increases of 5.4-fold and 3.8-fold, respectively, over the baseline strain.

This research demonstrated that the PFF construct can be used to effectively model artificially induced enzymatic genes within constraint-based modelling schemes. The construct can potentially be used in conjunction with other computational methods such as OptForce and FSEOF that identify metabolic engineering strategies for targeted overproduction. By better capturing the metabolic action of the inserted heterologous

genes, the models should provide better predictions and improved understanding. The PFF construct also has the potential to be specifically tailored. For example, the various thioesterases commonly used in strains for biofuel production can be modelled with the specific profile of substrates that each thioesterase acts on allowing for customised design depending on application.

5 References

Achterberg T. 2007. Constraint Integer Programming. Berlin: Technische Universität Berlin.

Acuna V, Chierichetti F, Lacroix V, Marchetti-Spaccamela A, Sagot MF, Stougie L. 2009. Modes and cuts in metabolic networks: complexity and algorithms. Biosystems 95(1):51-60.

Alper H, Jin YS, Moxley JF, Stephanopoulos G. 2005a. Identifying gene targets for the metabolic engineering of lycopene biosynthesis in *Escherichia coli*. Metab Eng 7(3):155-64.

Alper H, Miyaoku K, Stephanopoulos G. 2005b. Construction of lycopene-overproducing *E. coli* strains by combining systematic and combinatorial gene knockout targets. Nat Biotechnol 23(5):612-6.

Barrett CL, Kim TY, Kim HU, Palsson BØ, Lee SY. 2006. Systems biology as a foundation for genome-scale synthetic biology. Curr Opin Biotechnol 17(5):488-492.

Bertsimas D, Tsitsiklis JN. 1997. Introduction to Linear Optimization: Athena Scientific.

Beste DJ, Hooper T, Stewart G, Bonde B, Avignone-Rossa C, Bushell ME, Wheeler P, Klamt S, Kierzek AM, McFadden J. 2007. GSMN-TB: a web-based genome-scale network model of *Mycobacterium tuberculosis* metabolism. Genome Biol 8(5):R89.

Beste DJ, McFadden J. 2010. System-level strategies for studying the metabolism of Mycobacterium tuberculosis. Mol Biosyst 6(12):2363-72.

Boshoff HI, Lun DS. 2010. Systems biology approaches to understanding mycobacterial survival mechanisms. Drug Discov Today Dis Mech 7(1):e75-e82.

Bro C, Regenberg B, Forster J, Nielsen J. 2006. *In silico* aided metabolic engineering of *Saccharomyces cerevisiae* for improved bioethanol production. Metab Eng 8(2):102-11.

Burgard AP, Pharkya P, Maranas CD. 2003. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol Bioeng 84(6):647-57.

Caplice E, Fitzgerald GF. 1999. Food fermentations: role of microorganisms in food production and preservation. Int J Food Microbiol 50(1-2):131-49.

Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA and others. 2012. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res 40(Database issue):D742-53.

Causey TB, Zhou S, Shanmugam KT, Ingram LO. 2003. Engineering the metabolism of *Escherichia coli* W3110 for the conversion of sugar to redox-neutral and oxidized products: homoacetate production. Proc Natl Acad Sci U S A 100(3):825-32.

Cherepanov PP, Wackernagel W. 1995. Gene disruption in *Escherichia coli*: TcR and KmR cassettes with the option of Flp-catalyzed excision of the antibiotic-resistance determinant. Gene 158(1):9-14.

Cho H, Cronan JE, Jr. 1995. Defective export of a periplasmic enzyme disrupts regulation of fatty acid synthesis. J Biol Chem 270(9):4216-9.

Choi HS, Lee SY, Kim TY, Woo HM. 2010. In silico identification of gene amplification targets for improvement of lycopene production. Applied and environmental microbiology 76(10):3097-3105.

Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng TY, Moody DB, Murray M, Galagan JE. 2009. Interpreting expression data with metabolic flux models: predicting *Mycobacterium tuberculosis* mycolic acid production. PLoS Comput Biol 5(8):e1000489.

Covert MW, Famili I, Palsson BØ. 2003. Identifying constraints that govern cell behavior: a key to converting conceptual to computational models in biology? Biotechnol Bioeng 84(7):763-72.

Curran KA, Crook NC, Alper HS. 2012. Using flux balance analysis to guide microbial metabolic engineering. Microbial Metabolic Engineering: Springer. p 197-216.

Datsenko KA, Wanner BL. 2000. One-step inactivation of chromosomal genes in *Escherichia coli* K-12 using PCR products. Proc Natl Acad Sci U S A 97(12):6640-5.

Davis MS, Solbiati J, Cronan JE. 2000. Overproduction of acetyl-CoA carboxylase activity increases the rate of fatty acid biosynthesis in Escherichia coli. Journal of Biological Chemistry 275(37):28593-28598.

de Figueiredo LF, Podhorski A, Rubio A, Kaleta C, Beasley JE, Schuster S, Planes FJ. 2009. Computing the shortest elementary flux modes in genome-scale metabolic networks. Bioinformatics 25(23):3158-65.

Duarte NC, Becker SA, Jamshidi N, Thiele I, Mo ML, Vo TD, Srivas R, Palsson BØ. 2007. Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc Natl Acad Sci U S A 104(6):1777-82.

Dunn MF, Ramirez-Trujillo JA, Hernandez-Lucas I. 2009. Major roles of isocitrate lyase and malate synthase in bacterial and fungal pathogenesis. Microbiology 155(Pt 10):3166-75.

Edwards JS, Ibarra RU, Palsson BØ. 2001. *In silico* predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. Nat Biotechnol 19(2):125-30.

Edwards JS, Palsson BØ. 2000. Metabolic flux balance analysis and the *in silico* analysis of *Escherichia coli* K-12 gene deletions. BMC Bioinformatics 1(1):1.

Egen D, Lun DS. 2012. Truncated branch and bound achieves efficient constraint-based genetic design. Bioinformatics 28(12):1619-23.

Eisenreich W, Dandekar T, Heesemann J, Goebel W. 2010. Carbon metabolism of intracellular bacterial pathogens and possible links to virulence. Nat Rev Microbiol 8(6):401-12.

Fang X, Wallqvist A, Reifman J. 2010. Development and analysis of an *in vivo*-compatible metabolic network of *Mycobacterium tuberculosis*. BMC Syst Biol 4(1):160.

Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BØ. 2007. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:121.

Feist AM, Palsson BØ. 2008. The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*. Nat Biotechnol 26(6):659-67.

Feist AM, Zielinski DC, Orth JD, Schellenberger J, Herrgard MJ, Palsson BØ. 2010. Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli. Metab Eng 12(3):173-86.

Flores S, de Anda-Herrera R, Gosset G, Bolívar FG. 2004. Growth-rate recovery of Escherichia coli cultures carrying a multicopy plasmid, by engineering of the pentose-phosphate pathway. Biotechnology and bioengineering 87(4):485-494.

Fong SS, Burgard AP, Herring CD, Knight EM, Blattner FR, Maranas CD, Palsson BØ. 2005. *In silico* design and adaptive evolution of *Escherichia coli* for production of lactic acid. Biotechnol Bioeng 91(5):643-8.

Fong SS, Palsson BØ. 2004. Metabolic gene-deletion strains of *Escherichia coli* evolve to computationally predicted growth phenotypes. Nat Genet 36(10):1056-8.

Gagneur J, Klamt S. 2004. Computation of elementary modes: a unifying framework and the new binary approach. BMC Bioinformatics 5:175.

Green ML, Karp PD. 2004. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC Bioinformatics 5:76.

Hjersted JL, Henson MA, Mahadevan R. 2007. Genome-scale analysis of *Saccharomyces cerevisiae* metabolism and ethanol production in fed-batch culture. Biotechnol Bioeng 97(5):1190-204.

Holmes E, Kinross J, Gibson GR, Burcelin R, Jia W, Pettersson S, Nicholson JK. 2012. Therapeutic modulation of microbiota-host metabolic interactions. Sci Transl Med 4(137):137rv6.

Hong KK, Nielsen J. 2012. Metabolic engineering of Saccharomyces cerevisiae: a key cell factory platform for future biorefineries. Cell Mol Life Sci 69(16):2671-90.

Indira M, Sirsi M. 1960. Nutritional studies on *Mycobacterium tuberculosis*—effect of amino acids on the '*in vitro*' growth of *Mycobacterium tuberculosis*. Indian Journal of Tuberculosis 7(4):75-82.

Ip K, Colijn C, Lun DS. 2011. Analysis of complex metabolic behavior through pathway decomposition. BMC Syst Biol 5(1):91.

Ip K, Donoghue N, Kim MK, Lun DS. 2014. Constraint-based modeling of heterologous pathways: Application and experimental demonstration for overproduction of fatty acids in Escherichia coli. Biotechnol Bioeng 111(10):2056-66.

Jamshidi N, Palsson BØ. 2007. Investigating the metabolic capabilities of *Mycobacterium tuberculosis* H37Rv using the *in silico* strain *i*NJ661 and proposing alternative drug targets. BMC Syst Biol 1:26.

Kauffman KJ, Prakash P, Edwards JS. 2003. Advances in flux balance analysis. Curr Opin Biotechnol 14(5):491-6.

Klamt S, Stelling J. 2002. Combinatorial complexity of pathway analysis in metabolic networks. Mol Biol Rep 29(1-2):233-6.

Krulwich TA, Pelliccione NJ. 1979. Catabolic pathways of coryneforms, nocardias, and mycobacteria. Annu Rev Microbiol 33:95-111.

Lee KH, Park JH, Kim TY, Kim HU, Lee SY. 2007. Systems metabolic engineering of *Escherichia coli* for L-threonine production. Mol Syst Biol 3:149.

Lee S, Phalakornkule C, Domach MM, Grossmann IE. 2000. Recursive MILP model for finding all the alternate optima in LP models for metabolic networks. Computers & Chemical Engineering 24(2-7):711-716.

Lee SJ, Lee DY, Kim TY, Kim BH, Lee J, Lee SY. 2005. Metabolic engineering of *Escherichia coli* for enhanced production of succinic acid, based on genome comparison and *in silico* gene knockout simulation. Appl Environ Microbiol 71(12):7880-7.

Lee SK, Chou H, Ham TS, Lee TS, Keasling JD. 2008. Metabolic engineering of microorganisms for biofuels production: from bugs to synthetic biology to fuels. Curr Opin Biotechnol 19(6):556-63.

Lennen RM, Braden DJ, West RA, Dumesic JA, Pfleger BF. 2010. A process for microbial hydrocarbon synthesis: Overproduction of fatty acids in *Escherichia coli* and catalytic conversion to alkanes. Biotechnol Bioeng 106(2):193-202.

Lennen RM, Pfleger BF. 2012. Engineering *Escherichia coli* to synthesize free fatty acids. Trends Biotechnol 30(12):659-67.

Lu X, Vora H, Khosla C. 2008a. Overproduction of free fatty acids in *E. coli*: Implications for biodiesel production. Metabolic Engineering 10(6):333-339.

Lu X, Vora H, Khosla C. 2008b. Overproduction of free fatty acids in *E. coli*: implications for biodiesel production. Metab Eng 10(6):333-9.

Lun DS, Rockwell G, Guido NJ, Baym M, Kelner JA, Berger B, Galagan JE, Church GM. 2009. Large-scale identification of genetic design strategies using local search. Mol Syst Biol 5:296.

Lyngstadaas A, Sprenger GA, Boye E. 1998. Impaired growth of an *Escherichia coli* rpe mutant lacking ribulose-5-phosphate epimerase activity. Biochim Biophys Acta 1381(3):319-30.

Ma H, Sorokin A, Mazein A, Selkov A, Selkov E, Demin O, Goryanin I. 2007. The Edinburgh human metabolic network reconstruction and its functional analysis. Mol Syst Biol 3:135.

Megharaj M, Ramakrishnan B, Venkateswarlu K, Sethunathan N, Naidu R. 2011. Bioremediation approaches for organic pollutants: a critical perspective. Environ Int 37(8):1362-75.

Milne CB, Kim PJ, Eddy JA, Price ND. 2009. Accomplishments in genome-scale *in silico* modeling for industrial and medical biotechnology. Biotechnol J 4(12):1653-70.

Munoz-Elias EJ, McKinney JD. 2005. *Mycobacterium tuberculosis* isocitrate lyases 1 and 2 are jointly required for in vivo growth and virulence. Nat Med 11(6):638-44.

Neish AS. 2009. Microbes in gastrointestinal health and disease. Gastroenterology 136(1):65-80.

Nicholson JK, Holmes E, Kinross J, Burcelin R, Gibson G, Jia W, Pettersson S. 2012. Host-gut microbiota metabolic interactions. Science 336(6086):1262-7.

Nogales J, Palsson BO, Thiele I. 2008. A genome-scale metabolic reconstruction of Pseudomonas putida KT2440: iJN746 as a cell factory. Bmc Systems Biology 2(1):79.

Oberhardt MA, Palsson BØ, Papin JA. 2009. Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320.

Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, Palsson BØ. 2011. A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism--2011. Mol Syst Biol 7(1):535.

Orth JD, Palsson BØ. 2012. Gap-filling analysis of the *i*JO1366 *Escherichia coli* metabolic network reconstruction for discovery of metabolic functions. BMC Syst Biol 6(1):30.

Orth JD, Thiele I, Palsson BØ. 2010. What is flux balance analysis? Nat Biotechnol 28(3):245-8.

Palsson BØ. 2000. The challenges of *in silico* biology. Nat Biotechnol 18(11):1147-50.

Papagianni M. 2012. Recent advances in engineering the central carbon metabolism of industrially important bacteria. Microb Cell Fact 11(1):50.

Park JH, Lee KH, Kim TY, Lee SY. 2007. Metabolic engineering of *Escherichia coli* for the production of L-valine based on transcriptome analysis and *in silico* gene knockout simulation. Proc Natl Acad Sci U S A 104(19):7797-802.

Patil KR, Åkesson M, Nielsen J. 2004. Use of genome-scale microbial models for metabolic engineering. Curr Opin Biotechnol 15(1):64-69.

Pharkya P, Burgard AP, Maranas CD. 2004. OptStrain: a computational framework for redesign of microbial production systems. Genome Res 14(11):2367-76.

Pharkya P, Maranas CD. 2006. An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. Metab Eng 8(1):1-13.

Poolman MG, Venkatesh KV, Pidcock MK, Fell DA. 2004. A method for the determination of flux in elementary modes, and its application to *Lactobacillus rhamnosus*. Biotechnol Bioeng 88(5):601-12.

Price ND, Papin JA, Schilling CH, Palsson BØ. 2003. Genome-scale microbial in silico models: the constraints-based approach. Trends Biotechnol 21(4):162-9.

Purohit HJ, Cheema S, Lal S, Raut CP, Kalia VC. 2007. In Search of Drug Targets for *Mycobacterium tuberculosis*. Infect Disord Drug Targets 7(3):245-50.

Ranganathan S, Suthers PF, Maranas CD. 2010. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. PLoS Comput Biol 6(4):e1000744.

Ranganathan S, Tee TW, Chowdhury A, Zomorrodi AR, Yoon JM, Fu Y, Shanks JV, Maranas CD. 2012. An integrated computational and experimental study for overproducing fatty acids in *Escherichia coli*. Metabolic Engineering 14(6):687-704.

Rawat I, Kumar RR, Mutanda T, Bux F. 2011. Dual role of microalgae: Phycoremediation of domestic wastewater and biomass production for sustainable biofuels production. Applied Energy 88(10):3411-3424.

Rawat I, Kumar RR, Mutanda T, Bux F. 2013. Biodiesel from microalgae: A critical evaluation from laboratory to large scale production. Applied Energy 103(0):444-467.

Rocha M, Maia P, Mendes R, Pinto JP, Ferreira EC, Nielsen J, Patil KR, Rocha I. 2008. Natural computation meta-heuristics for the *in silico* optimization of microbial strains. BMC Bioinformatics 9:499.

Rohmer L, Hocquet D, Miller SI. 2011. Are pathogenic bacteria just looking for food? Metabolism and microbial pathogenesis. Trends Microbiol 19(7):341-8.

Schirmer A, Rude MA, Li X, Popova E, del Cardayre SB. 2010. Microbial biosynthesis of alkanes. Science 329(5991):559-62.

Schnappinger D, Ehrt S, Voskuil MI, Liu Y, Mangan JA, Monahan IM, Dolganov G, Efron B, Butcher PD, Nathan C and others. 2003. Transcriptional Adaptation of *Mycobacterium tuberculosis* within Macrophages: Insights into the Phagosomal Environment. J Exp Med 198(5):693-704.

Schuetz R, Kuepfer L, Sauer U. 2007. Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. Mol Syst Biol 3:119.

Schuster S, Dandekar T, Fell DA. 1999. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends Biotechnol 17(2):53-60.

Schuster S, Fell DA, Dandekar T. 2000. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. Nat Biotechnol 18(3):326-32.

Schuster S, Hilgetag C. 1994. On elementary flux modes in biochemical reaction systems at steady state. Journal of Biological Systems 2(02):165-182.

Schwartz JM, Kanehisa M. 2005. A quadratic programming approach for decomposing steady-state metabolic flux distributions onto elementary modes. Bioinformatics 21 Suppl 2:ii204-5.

Segre D, Vitkup D, Church GM. 2002. Analysis of optimality in natural and perturbed metabolic networks. Proc Natl Acad Sci U S A 99(23):15112-7.

Sharma V, Sharma S, Hoener zu Bentrup K, McKinney JD, Russell DG, Jacobs WR, Jr., Sacchettini JC. 2000. Structure of isocitrate lyase, a persistence factor of *Mycobacterium tuberculosis*. Nat Struct Biol 7(8):663-8.

Shlomi T, Berkman O, Ruppin E. 2005. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. Proc Natl Acad Sci U S A 102(21):7695-700.

Sprenger GA. 1995. Genetics of pentose-phosphate pathway enzymes ofEscherichia coli K-12. Archives of microbiology 164(5):324-330.

Steen EJ, Kang Y, Bokinsky G, Hu Z, Schirmer A, McClure A, Del Cardayre SB, Keasling JD. 2010. Microbial production of fatty-acid-derived fuels and chemicals from plant biomass. Nature 463(7280):559-62.

Stelling J, Klamt S, Bettenbrock K, Schuster S, Gilles ED. 2002. Metabolic network structure determines key aspects of functionality and regulation. Nature 420(6912):190-3.

Street HE, Hughes JC, Lewis JS. 1960. Studies on the growth of excised roots: X. Individual amino acids and acid-hydrolysed casein as nitrogen sources for the growth of excised tomato roots. New Phytologist 59(3):273-287.

Terzer M, Stelling J. 2008. Large-scale computation of elementary flux modes with bit pattern trees. Bioinformatics 24(19):2229-35.

Thiele I, Palsson BØ. 2010. A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5(1):93-121.

Trinh CT, Wlaschin A, Srienc F. 2009. Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. Appl Microbiol Biotechnol 81(5):813-26.

Urgun-Demirtas M, Stark B, Pagilla K. 2006. Use of genetically engineered microorganisms (GEMs) for the bioremediation of contaminants. Crit Rev Biotechnol 26(3):145-64.

Voelker TA, Davies HM. 1994a. Alteration of the specificity and regulation of fatty acid synthesis of *Escherichia coli* by expression of a plant medium-chain acyl-acyl carrier protein thioesterase. J Bacteriol 176(23):7320-7.

Voelker TA, Davies HM. 1994b. Alteration of the specificity and regulation of fatty acid synthesis of Escherichia coli by expression of a plant medium-chain acyl-acyl carrier protein thioesterase. Journal of Bacteriology 176(23):7320-7327.

Wang Q, Chen X, Yang Y, Zhao X. 2006. Genome-scale *in silico* aided metabolic analysis and flux comparisons of *Escherichia coli* to improve succinate production. Appl Microbiol Biotechnol 73(4):887-94.

Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, Khandurina J, Trawick JD, Osterhout RE, Stephen R and others. 2011. Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. Nat Chem Biol 7(7):445-52.

Youngquist JT, Lennen RM, Ranatunga DR, Bothfeld WH, Marner WD, 2nd, Pfleger BF. 2012. Kinetic modeling of free fatty acid production in *Escherichia coli* based on continuous cultivation of a plasmid free strain. Biotechnol Bioeng 109(6):1518-27.

Yu X, Liu T, Zhu F, Khosla C. 2011. *In vitro* reconstitution and steady-state analysis of the fatty acid synthase from *Escherichia coli*. Proc Natl Acad Sci U S A 108(46):18643-8.

Zhu Y, Zhang Y, Li Y. 2009. Understanding the industrial application potential of lactic acid bacteria through genomics. Appl Microbiol Biotechnol 83(4):597-610.

Appendix A Statements of contribution and copyright licences

Analysis of complex metabolic behavior through pathway decomposition

Authors:

Kuhn Ip

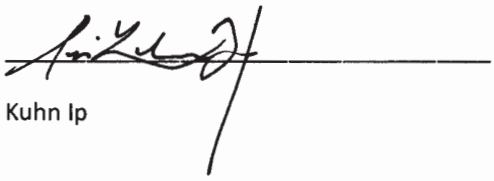
Dr. Caroline Colijn (Imperial College London, UK)

Dr. Desmond S Lun (Rutgers University, NJ, USA)

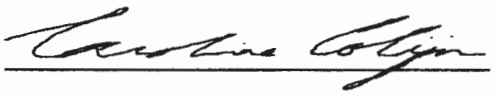
Statement of contribution

DSL and CC conceived the project. DSL and KI designed and implemented the method and performed the computational experiments. DSL, CC, and KI wrote the paper. All authors have read and approved the final manuscript.

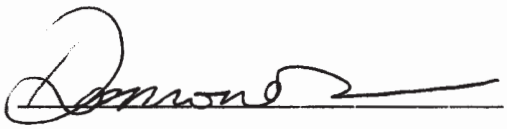
Signatures:



Kuhn Ip



Dr. Caroline Colijn



Dr. Desmond S Lun

Constraint-Based Modeling of Heterologous Pathways:

Application and Experimental Demonstration for Overproduction of Fatty Acids in *Escherichia coli*

Authors:

Kuhn Ip

Dr. Neil Donoghue

Min Kyung Kim (Rutgers University, NJ, USA)

Dr. Desmond S Lun (Rutgers University, NJ, USA)

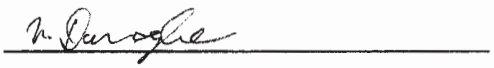
Statement of contribution

DSL conceived the proportional flux forcing method. KI, ND, and DSL designed and performed the experiments. KI, ND, and DSL created the single-gene knockout strain. KI and DSL created the four- and five-gene knockout strains. KI performed the fatty-acid quantification experiments. MKK assisted with strain adaptation. KI and DSL analysed the experimental data and performed computational analysis. KI and DSL wrote the paper. All authors have read and approved the final manuscript.

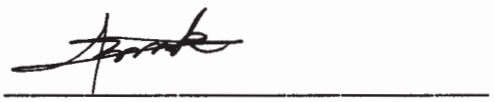
Signatures:



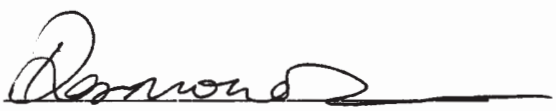
Kuhn Ip



Dr. Neil Donoghue



Min Kyung Kim



Dr. Desmond S Lun

JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS

Aug 28, 2014

This is a License Agreement between Pooi Kuhn Ip ("You") and John Wiley and Sons ("John Wiley and Sons") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by John Wiley and Sons, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3457991232364
License date	Aug 28, 2014
Licensed content publisher	John Wiley and Sons
Licensed content publication	Biotechnology & Bioengineering
Licensed content title	Constraint-based modeling of heterologous pathways: Application and experimental demonstration for overproduction of fatty acids in Escherichia coli
Licensed copyright line	© 2014 Wiley Periodicals, Inc.
Licensed content author	Kuhn Ip,Neil Donoghue,Min Kyung Kim,Desmond S. Lun
Licensed content date	Jun 16, 2014
Start page	2056
End page	2066
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Title of your thesis / dissertation	Mathematical modelling of bacterial metabolism
Expected completion date	Aug 2014
Expected size (number of pages)	80
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and

payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your Rightslink account (these are available at any time at http://myaccount.copyright.com).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.

- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this licence must be completed within two years of the date of the grant of this licence (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner. You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto.

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS,

INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.

- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.

- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.

- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.

- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY

concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.

- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.

- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.

- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses:: Creative Commons Attribution (CC-BY) license Creative Commons Attribution Non-Commercial (CC-BY-NC) license and Creative Commons Attribution Non-Commercial-NoDerivs (CC-BY-NC-ND) License. The license type is clearly identified on the article.

Copyright in any research article in a journal published as Open Access under a Creative Commons License is retained by the author(s). Authors grant Wiley a license to publish the article and identify itself as the original publisher. Authors also grant any third party the right to use the article freely as long as its integrity is maintained and its original authors, citation details and publisher are identified as follows: [Title of Article/Author/Journal Title and Volume/Issue. Copyright (c) [year] [copyright owner as specified in the Journal]. Links to the final article on Wiley's website are encouraged where applicable.

The Creative Commons Attribution License

The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-commercial re-use of an open access article, as long as the author is properly attributed.

The Creative Commons Attribution License does not affect the moral rights of authors, including without limitation the right not to have their work subjected to derogatory treatment. It also does not affect any other rights held by authors or third parties in the article, including without limitation the rights of privacy and publicity. Use of the article must not assert or imply, whether implicitly or explicitly, any connection with, endorsement or sponsorship of such use by the author, publisher or any other party associated with the article.

For any reuse or distribution, users must include the copyright notice and make clear to others that the article is made available under a Creative Commons Attribution license, linking to the relevant Creative Commons web page.

To the fullest extent permitted by applicable law, the article is made available as is and without representation or warranties of any kind whether express, implied, statutory or otherwise and including, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of defects, accuracy, or the presence or absence of errors.

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC) License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by non-commercial users

For non-commercial and non-promotional purposes, individual users may access, download, copy, display and redistribute to colleagues Wiley Open Access articles, as well as adapt, translate, text- and data-mine the content subject to the following conditions:

- The authors' moral rights are not compromised. These rights include the right of "paternity" (also known as "attribution" - the right for the author to be identified as such) and "integrity" (the right for the author not to have the work altered in such a way that the author's reputation or integrity may be impugned).

- Where content in the article is identified as belonging to a third party, it is the obligation of the user to ensure that any reuse complies with the copyright policies of the owner of that content.

- If article content is copied, downloaded or otherwise reused for non-commercial research and education purposes, a link to the appropriate bibliographic citation (authors, journal, article title, volume, issue, page numbers, DOI and the link to the definitive published version on **Wiley Online Library**) should be maintained. Copyright notices and disclaimers must not be deleted.

- Any translations, for which a prior translation agreement with Wiley has not been agreed, must prominently display the statement: "This is an unofficial translation of an article that appeared in a Wiley publication. The publisher has not endorsed this translation."

Use by commercial "for-profit" organisations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee. Commercial purposes include:

- Copying or downloading of articles, or linking to such articles for further redistribution, sale or licensing;
- Copying, downloading or posting by a site or service that incorporates advertising with such content;
- The inclusion or incorporation of article content in other works or services (other than normal quotations with an appropriate citation) that is then available for sale or licensing, for a fee (for example, a compilation produced for marketing purposes, inclusion in a sales pack)
- Use of article content (other than normal quotations with appropriate citation) by for-profit organisations for promotional purposes
- Linking to article content in e-mails redistributed for promotional, marketing or educational purposes;
- Use for the purposes of monetary reward by means of sale, resale, licence, loan, transfer or other form of commercial exploitation such as marketing products
- Print reprints of Wiley Open Access articles can be purchased from: corporatesales@wiley.com

Further details can be found on Wiley Online Library
http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

v1.9

You will be invoiced within 48 hours of this transaction date. You may pay your invoice by credit card upon receipt of the invoice for this transaction. Please follow instructions provided at that time.
To pay for this transaction now; please remit a copy of this document along with your payment. Payment should be in the form of a check or money order referencing your account number and this invoice number RLNK501389491.
Make payments to "COPYRIGHT CLEARANCE CENTER" and send to:

Copyright Clearance Center
Dept 001
P.O. Box 843006
Boston, MA 02284-3006
Please disregard electronic and mailed copies if you remit payment in advance.
Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Gratis licenses (referencing $0 in the Total field) are free. Please retain this printable license for your reference. No payment is required.

Appendix B Scripting for Elementary mode decomposition

```
README.txt

em_decomp

Requirements
------------

em_decomp requires MATLAB, libSBML (with the MATLAB interface),
Gurobi, and
Gurobi Mex.  It has been tested on MS Windows 7 and Mac OS X 10.6 with
MATLAB
R2010b, libSBML v4.3.0 and Gurobi v4.0.1.

libSBML is available from http://sbml.org/Software/libSBML.  It must
be built
to include the MATLAB interface so that the command 'TranslateSBML' is
in the
MATLAB path.

Gurobi is available from http://www.gurobi.com/ and is free for
academic use.
Gurobi Mex is a MATLAB interface for Gurobi that is available from
http://www.convexoptimization.com/wikimization/index.php/Gurobi_Mex:_A
_MATLAB_interface_for_Gurobi.
Gurobi Mex must be compiled and installed so that the command
'gurobi_mex' is
in the MATLAB path.  The solvers GLPK, SCIP, and CPLEX can also be
used (see
solve_milp.m), but these have not been tested.

Elementary mode decomposition of flux distribution in iNJ661
------------------------------------------------------------

As an example of the use of em_decomp, we illustrate how it can be
applied to
obtain an elementary mode decomposition of a flux distribution in
iNJ661, a
genome-scale model of Mycobacterium tuberculosis.

1. We first load the model using the script parse_sbml.m:

  >> fbamodel =
parse_sbml('iNJ661_biomass9_plus_vitamins_0.052_flux.xml');

When prompted with "Do you want to load the model anyway? Enter y/n",
respond
"y".
```

2. Now solve for an optimal flux distribution by FBA:

```
>> v = flux_balance(fbamodel);
```

3. Finally, we decompose the flux distribution v into elementary modes by

```
>> [P, w] = em_decomp(v, fbamodel);
```

This results in a 1028 x 20 matrix P containing the 20 elementary modes as
columns, and a 1 x 20 weight matrix w such that P * w' = v.

License

This is free software: you can redistribute it and/or modify it under the terms of the GNU Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This code is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Public License for more details.

Desmond Lun, November 2010 (dslun@camden.rutgers.edu)

```
% PARSE_SBML Parse SBML file
%    FBAMODEL = PARSE_SBML(SBMLFNAME) parses the SBML file
SBMLFNAME.txt
%    and returns the FBA model contained in it as FBAMODEL.
function fbamodel = parse_sbml(sbmlfname)

model = TranslateSBML(sbmlfname, 1);

fbamodel.nrxn = length(model.reaction);
fbamodel.nmetab = length(model.species);
fbamodel.S = sparse(fbamodel.nmetab, fbamodel.nrxn);    %
Stoichiometric matrix
fbamodel.rxns = cell(fbamodel.nrxn, 1);
fbamodel.metabs = cell(fbamodel.nmetab, 1);
fbamodel.f = zeros(fbamodel.nrxn, 1);       % biological objective
fbamodel.g = zeros(fbamodel.nrxn, 1);       % synthetic objective
fbamodel.vmin = zeros(fbamodel.nrxn, 1);
fbamodel.vmax = zeros(fbamodel.nrxn, 1);
fbamodel.present = true(fbamodel.nrxn, 1);  % reactions active in
model

boundary = zeros(fbamodel.nmetab, 1);

% metabolites & boundary conditions
for imetab = 1:fbamodel.nmetab
    fbamodel.metabs{imetab} = model.species(imetab).id;
    boundary(imetab) = model.species(imetab).boundaryCondition;
end

% parse reactions
for irxn = 1:fbamodel.nrxn
    fbamodel.rxns{irxn} = model.reaction(irxn).id;

    for ireactant = 1:length(model.reaction(irxn).reactant)

fbamodel.S(strcmp(model.reaction(irxn).reactant(ireactant).species,
fbamodel.metabs), irxn) ...
            = -model.reaction(irxn).reactant(ireactant).stoichiometry;
    end
    for iproduct = 1:length(model.reaction(irxn).product)

fbamodel.S(strcmp(model.reaction(irxn).product(iproduct).species,
fbamodel.metabs), irxn) ...
            = model.reaction(irxn).product(iproduct).stoichiometry;
    end

    for iparam = 1:length(model.reaction(irxn).kineticLaw.parameter)
        param = model.reaction(irxn).kineticLaw.parameter(iparam);
        if strcmp('LOWER_BOUND', param.id)
            fbamodel.vmin(irxn) = param.value;
        elseif strcmp('UPPER_BOUND', param.id)
            fbamodel.vmax(irxn) = param.value;
        elseif strcmp('OBJECTIVE_COEFFICIENT', param.id)
            fbamodel.f(irxn) = param.value;
        end
    end
end

fbamodel.metabs = fbamodel.metabs(~boundary);
fbamodel.S = fbamodel.S(~boundary, :);
fbamodel.nmetab = length(fbamodel.metabs);
```

```
% SOLVE_MILP Pass MILP to solver
%    [x, fval, exitflag] = solve_milp(f, A, b, Aeq, beq, lb, ub,
vartype, sense, options)
%    Select SOLVER by uncommenting.
function [x, fval, exitflag] = solve_milp(f, A, b, Aeq, beq, lb, ub,
vartype, sense, options)

% SOLVER = 'GLPK';
% SOLVER = 'SCIP';
% SOLVER = 'CPLEX';
SOLVER = 'Gurobi';

if nargin < 10
    options = optimset('Display', 'final');
end
if nargin < 9
    sense = 1;
end
if nargin < 8
    vartype = char('C' * ones(size(f)));
end
if nargin < 6
    lb = [];
    ub = [];
end
if nargin < 4
    Aeq = [];
    beq = [];
end

switch SOLVER
    case 'GLPK'
        param.tmlim = -1;
        param.save = 0;
        switch options.Display
            case 'off'
                param.msglev = 0;
            case 'iter'
                param.msglev = 3;
            case 'final'
                param.msglev = 2;
            case 'notify'
                param.msglev = 1;
        end

        ctype = [ char('U' * ones(size(A, 1), 1)); char ('S' *
ones(size(Aeq, 1), 1)) ];
        [x, fval, status] = glpk(f, [A; Aeq], [b; beq], lb, ub, ctype,
vartype, sense, param);

        if status == 180 || status == 151 || status == 171
            exitflag = 1;
        else
            exitflag = -1;
        end
    case 'SCIP'
        param = struct();
        switch options.Display
            case 'off'
                param.msglev = 0;
            case 'iter'
                param.msglev = 3;
            case 'final'
```

```
                param.msglev = 2;
            case 'notify'
                param.msglev = 1;
        end

        ctype = [ char('U' * ones(size(A, 1), 1)); char ('S' *
ones(size(Aeq, 1), 1)) ];
        if all(vartype == 'C')
            param.tmlim = -1;
            param.save = 0;
            [x, fval, exitflag] = glpk(f, [A; Aeq], [b; beq], lb, ub,
ctype, vartype, sense, param);
        else
            [x, fval, exitflag] = scip(f, [A; Aeq], [b; beq], lb, ub,
ctype, vartype, sense, param);
        end
    case 'CPLEX'
        cpoptions = cplexoptimset('Display', options.Display);
        if ~strcmp(options.Display, 'off')
            cpoptions = cplexoptimset('Diagnostics', 'on');
        end

        if all(vartype == 'C')
            [x, fval, exitflag] = cplexlp(sense * f, A, b, Aeq, beq,
lb, ub, [], cpoptions);
            fval = sense * fval;
        else
            vartype = reshape(vartype, 1, []);
            [x, fval, exitflag] = cplexmilp(sense * f, A, b, Aeq, beq,
[], [], [], lb, ub, vartype, [], cpoptions);
            fval = sense * fval;
        end
    case 'Gurobi'
        goptions = struct();
        goptions.FeasibilityTol = 1e-9;
        goptions.IntFeasTol = 1e-9;
        switch options.Display
            case 'off'
                goptions.Display = 0;
                goptions.DisplayInterval = 0;
            case 'iter'
                goptions.Display = 2;
                goptions.DisplayInterval = 5;
            case 'final'
                goptions.Display = 2;
                goptions.DisplayInterval = 0;
            case 'notify'
                goptions.Display = 1;
                goptions.DisplayInterval = 0;
        end

        contypes = [ char('<' * ones(size(A, 1), 1)); char ('=' *
ones(size(Aeq, 1), 1)) ];
        [x, fval, flag] = gurobi_mex(f, sense, [A; Aeq], [b; beq],
contypes, lb, ub, vartype, goptions);

        if flag == 2
            exitflag = 1;
        else
            exitflag = -1;
        end
end
end
```

```
% FLUX_BALANCE Flux-balance analysis of FBA model
%    [V, FMAX, FMIN] = FLUX_BALANCE(FBAMODEL) performs a basic flux-
balance
%    analysis of the FBA model FBAMODEL and returns a biomass-
maximizing
%    flux distribution in the vector V.  The maximimum and minimum
%    synthetic objective possible in a biomass-maximizing flux
distribution
%    is given in FMAX and FMIN, respectively.
%
%    [V, FMAX, FMIN] = FLUX_BALANCE(FBAMODEL, QUIET) performs the
analysis
%    and supresses screen output if QUIET is set to true.

function [v, fmax, fmin] = flux_balance(fbamodel, quiet)

if nargin < 2
    quiet = false;
end

nrxn   = fbamodel.nrxn;
nmetab = fbamodel.nmetab;

% setup basic flux-balance analysis
c = fbamodel.f;
lb = fbamodel.vmin;
ub = fbamodel.vmax;

Aeq = [ fbamodel.S
        sparse(1:nnz(~fbamodel.present), find(~fbamodel.present),
ones(nnz(~fbamodel.present), 1), nnz(~fbamodel.present), nrxn) ];
beq = [ zeros(nmetab, 1);
        zeros(nnz(~fbamodel.present), 1); ];

vartype = char('C' * ones(nrxn, 1));

options = optimset('Display', 'off');

[x, vbiomass, exitflag] = solve_milp(c, [], [], Aeq, beq, lb, ub,
vartype, -1, options);

if ~quiet
    fprintf('Biomass flux:    %f\n', vbiomass);
end

% calculating maximum and minimum synthetic objective possible
% assuming maximum biomass flux
if exitflag > 0
    v = x(1:nrxn);

    c = fbamodel.g;     % synthetic objective

    if any(c)
        % maximum possible synthetic flux
        lb_nobiomass = lb;
        ub_nobiomass = ub;
        lb_nobiomass(fbamodel.f > 0) = 0;
        ub_nobiomass(fbamodel.f < 0) = 0;
        [x, max_vsynth] = solve_milp(c, [], [], Aeq, beq,
lb_nobiomass, ub_nobiomass, vartype, -1, options);

        Aeq = [ Aeq;
```

```
                fbamodel.f' ];  % condition to ensure maximum biomass
in the max/min calculations
        beq = [ beq;
                vbiomass ];

        [x, fmin] = solve_milp(c, [], [], Aeq, beq, lb, ub, vartype,
1, options);
        [x, fmax, exitflag] = solve_milp(c, [], [], Aeq, beq, lb, ub,
vartype, -1, options);
        if exitflag > 0
            v = x(1:nrxn);

            if ~quiet
                fprintf('Synthetic flux:  [%f, %f] of %f\n', fmin,
fmax, max_vsynth);
            end
        end
    end
else
    fmin = NaN;     % error
    fmax = NaN;
    v = [];
end
```

```
% FLUX_BALANCE Flux-balance analysis of FBA model
%    [V, FMAX, FMIN] = FLUX_BALANCE(FBAMODEL) performs a basic flux-
balance
%    analysis of the FBA model FBAMODEL and returns a biomass-
maximizing
%    flux distribution in the vector V.  The maximimum and minimum
%    synthetic objective possible in a biomass-maximizing flux
distribution
%    is given in FMAX and FMIN, respectively.
%
%    [V, FMAX, FMIN] = FLUX_BALANCE(FBAMODEL, QUIET) performs the
analysis
%    and supresses screen output if QUIET is set to true.

function [v, fmax, fmin] = flux_balance(fbamodel, quiet)

if nargin < 2
    quiet = false;
end

nrxn   = fbamodel.nrxn;
nmetab = fbamodel.nmetab;

% setup basic flux-balance analysis
c = fbamodel.f;
lb = fbamodel.vmin;
ub = fbamodel.vmax;

Aeq = [ fbamodel.S
        sparse(1:nnz(~fbamodel.present), find(~fbamodel.present),
ones(nnz(~fbamodel.present), 1), nnz(~fbamodel.present), nrxn) ];
beq = [ zeros(nmetab, 1);
        zeros(nnz(~fbamodel.present), 1); ];

vartype = char('C' * ones(nrxn, 1));

options = optimset('Display', 'off');

[x, vbiomass, exitflag] = solve_milp(c, [], [], Aeq, beq, lb, ub,
vartype, -1, options);

if ~quiet
    fprintf('Biomass flux:    %f\n', vbiomass);
end

% calculating maximum and minimum synthetic objective possible
% assuming maximum biomass flux
if exitflag > 0
    v = x(1:nrxn);

    c = fbamodel.g;     % synthetic objective

    if any(c)
        % maximum possible synthetic flux
        lb_nobiomass = lb;
        ub_nobiomass = ub;
        lb_nobiomass(fbamodel.f > 0) = 0;
        ub_nobiomass(fbamodel.f < 0) = 0;
        [x, max_vsynth] = solve_milp(c, [], [], Aeq, beq,
lb_nobiomass, ub_nobiomass, vartype, -1, options);

        Aeq = [ Aeq;
```

```
                fbamodel.f' ];  % condition to ensure maximum biomass
in the max/min calculations
        beq = [ beq;
                vbiomass ];

        [x, fmin] = solve_milp(c, [], [], Aeq, beq, lb, ub, vartype,
1, options);
        [x, fmax, exitflag] = solve_milp(c, [], [], Aeq, beq, lb, ub,
vartype, -1, options);
        if exitflag > 0
            v = x(1:nrxn);

            if ~quiet
                fprintf('Synthetic flux:  [%f, %f] of %f\n', fmin,
fmax, max_vsynth);
            end
        end
    end
else
    fmin = NaN;     % error
    fmax = NaN;
    v = [];
end
```

Appendix C Additional GDLS results for *i*AF1260 and *i*JO1366

Table C-1: Knockout strategies from GDLS analysis of *i*AF1260 with PFF model of '*tesA* with α = 0.1.

Growth rate (g h^{-1})	FFA flux (mmol gDW^{-1} h^{-1})	Number of knockouts	Number of gene associations	Gene associations*
0.891	0.135	0	0	None (wild type)
0.889	0.154	1	1	*fadE*
		2	1	(*fadK* OR *fadD*)
		1	1	*fadB*
		1	1	*tesB*
		1	1	(*aas* AND *acpP*)
		2	2	*fadB*, *uraA*
0.651	0.234	4	2	(*rpe* OR *sgcE*), (*talB* OR *talA*)
0.649	0.256	5	3	*fadE*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*fadB*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	(*aas* AND *acpP*), (*rpe* OR *sgcE*), (*talB* OR *talA*)
		6	3	(*fadK* OR *fadD*), (*rpe* OR *sgcE*), (*talB* OR *talA*)

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Table C-2: Knockout strategies from GDLS analysis of *i*AF1260 with PFF model of '*tesA* with α = 0.3.

Growth rate (g h^{-1})	FFA flux (mmol gDW^{-1} h^{-1})	Number of knockouts	Number of gene associations	Gene associations*
0.761	0.640	0	0	None (wild type)
0.737	0.787	1	1	*fade*
		2	1	(*fadK* OR *fadD*)
		1	1	*fadB*
		1	1	*pldA*
		1	1	(*aas* AND *acpP*)
		3	2	*fadE*, (*ldhA* OR *dld*)
0.516	0.985	4	2	(*rpe* OR *sgcE*), (*talB* OR *talA*)
0.499	1.124	5	3	*fadE*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*pldA*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	(*aas* AND *acpP*), (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*fadB*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		6	3	(*fadK* OR *fadD*), (*rpe* OR *sgcE*), (*talB* OR *talA*)

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Table C-3: Knockout strategies from GDLS analysis of *i*AF1260 with PFF model of '*tesA* with α = 0.4.

Growth rate (g h^{-1})	**FFA flux (mmol gDW^{-1} h^{-1})**	**Number of knockouts**	**Number of gene associations**	**Gene associations***
0.652	1.099	0	0	None (wild type)
0.600	1.406	1	1	*fadE*
		2	1	(*fadK* OR *fadD*)
		1	1	*fadB*
		1	1	*tesB*
		1	1	(*aas* AND *acpP*)
		1	1	*pldA*
		2	2	*aas*, *glpK*
0.419	1.557	4	2	(*rpe* OR *sgcE*), (*talB* OR *talA*)
0.388	1.806	5	3	*fadE*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*pldA*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		5	3	*fadB*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		6	3	(*fadK* OR *fadD*), (*rpe* OR *sgcE*), (*talB* OR *talA*)
		6	4	*fadB*, *pgsA*, (*rpe* OR *sgcE*), (*talB* OR *talA*)
		6	4	(*aas* AND *acpP*), *fadI*, (*rpe* OR *sgcE*), (*talB* OR *talA*)

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Table C-4: Knockout strategies from GDLS analysis of *i*JO1366 with PFF model of '*tesA* with α = 0.1.

Growth rate (g h^{-1})	FFA flux (mmol gDW^{-1} h^{-1})	Number of knockouts	Number of gene associations	Gene associations*
0.975	0.034	0	0	None (wild type)
0.936	0.133	2	1	(*fadB* OR *fadJ*)
0.934	0.153	1	1	*fadE*
		2	1	(*aas* OR *acpP*)
		1	1	*tesB*
0.320	0.168	2	1	(*rpiA* OR *rpiB*)
0.320	0.168	3	2	(*rpiA* OR *rpiB*), *hemF*
0.930	0.169	2	2	*fadE*, *fabF*
0.318	0.181	3	2	*fadB*, (*rpiA* OR *rpiB*)
0.923	0.195	2	2	*fadE*, *fabA*
0.912	0.219	3	2	(*fadB* OR *fadJ*), *fabA*
		3	2	(*fadI* OR *fadA*), *fabA*
		4	3	(*fadB* OR *fadJ*), *fabA*, *nagA*
		6	3	(*fadB* OR *fadJ*), *fabA*, (*gabT* OR *puuE*), (*mphF* OR *adhE*)
		6	4	(*fadB* OR *fadJ*), *fabA*, (*gabD* OR *sad*)
0.681	0.301	8	5	(*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*
		8	5	(*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*
		8	5	(*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*
0.758	0.356	8	3	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***
		8	3	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*mphF* OR *adhE*), *nuoA-M***
0.501	0.756	11	6	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***, *rplT*, *aceA*, *ppc*
0.468	0.789	13	6	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***, (*tdcA* OR *ackA* OR *purT*), *cyoA-D****, *tpiA*
		12	6	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*mphF* OR *adhE*), *nuoA-M***, (*tdcA* OR *ackA* OR *purT*), *cyoA-D****, *tpiA*
0.310	0.845	13	6	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*),

				*nuoA-M****, (*pta* OR *eutD*), (*pntA* OR *pntB*), *atp_synthase******
		14	6	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***, (*tdcA* OR *ackA* OR *purT*) , (*pntA* OR *pntB*), *atp_synthase******
0.322	1.023	16	9	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***, (*tdcA* OR *ackA* OR *purT*), *cyoA-D****, *umpG*, *ppc*, *tpiA*, *sdhA_D*****
0.305	1.050	15	9	(*ompF* OR *ompN* OR *ompC* OR *phoE*), (*adhP* OR *adhE* OR *adhC*), *nuoA-M***, *rplT*, *aceA*, *ppc*, *cyoA-D****, (*pta* OR *eutD*), *tpiA*

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

**(nuoA AND nuoB AND nuoC AND nuoD AND nuoE AND nuoF AND nuoG AND nuoH AND nuoI AND nuoJ AND nuoK AND nuoL AND nuoM)

***(*cyoA* AND *cyoB* AND *cyoC* AND *cyoD*)

****(*sdhA* AND *sdhB* AND *sdhC* AND *sdhD*)

*****(((*atpF* and *atpG* and *atpH*) and (*atpA* and *atpB* and *atpC* and *atpD* and *atpE*) and *atpI*) or ((*atpF* and *atpG* and *atpH*) and (*atpA* and *atpB* and *atpC* and *atpD* and *atpE*)))

Table C-5: Knockout strategies from GDLS analysis of *i*JO1366 with PFF model of '*tesA* with α = 0.2.

Growth rate (g h^{-1})	FFA flux (mmol gDW^{-1} h^{-1})	Number of knockouts	Number of gene associations	Gene associations*
0.960	0.075	0	0	None (wild type)
0.880	0.336	2	1	(*fadB* OR *fadJ*)
0.302	0.357	2	1	(*rpiA* OR *rpiB*)
0.870	0.400	1	1	(*aas* AND *acpP*)
0.870	0.400	1	1	*fadE*
		1	1	*tesB*
		3	2	*tesB*, (*fadK* OR *fadD*)
		3	2	*tesB*, (*pykA* OR *pykF*)
0.867	0.405	3	2	*tesB*, (*pykA* OR *pykF*)
0.857	0.451	2	2	*fadE*, *fabF*
0.838	0.528	2	2	*fadE*, *fabA*
0.811	0.607	3	2	(*fadB* OR *fadJ*), *fabA*
		4	3	(*fadB* OR *fadJ*), *fabA*, *acs*
		4	3	(*fadB* OR *fadJ*), *fabA*, (*eutC* AND *eutB*)
		5	3	(*fadB* OR *fadJ*), *fabA*, (*dld* OR *ldhA*)
		5	4	(*fadB* OR *fadJ*), *fabA*, (*yhfW* OR *deoB*)
		3	2	(*fadI* OR *fadA*), *fabA*
		4	3	(*fadI* OR *fadA*), *fabA*, *acs*
		5	3	(*fadI* OR *fadA*), *fabA*, *gpt*
		4	3	(*fadI* OR *fadA*), *fabA*, *nagK*
		5	3	(*fadI* OR *fadA*), *fabA*, (*nudC* OR *ligA*)
		6	4	(*fadI* OR *fadA*), *fabA*, (*pfkA* OR *pfkB*), *tdk*
		8	4	(*fadK* OR *fadD*), (*focA* OR *focB*), (*rpe* OR *sgcE*), (*talB* OR *talA*)
0.789	0.683	5	4	(*fadB* OR *fadJ*), *fabA*, *aas*, *teas*
		5	4	(*fadI* OR *fadA*), *fabA*, *aas*, *teas*
0.594	0.757	8	5	(*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*
		10	6	(*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*, (*dld* OR *ldhA*)
		10	6	(*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*, (*dld* OR *ldhA*)
		8	5	(*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*
		8	5	(*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*

0.581	0.810	11	8	*aas*, (*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*, *tesA*, *anmk*
		11	8	*aas*, (*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*, *tesA*, *deoD*
		12	8	*aas*, (*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*, *tesA*, (*yhfW* OR *deoB*)
		12	8	*aas*, (*fadB* OR *fadJ*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*, *tesA*, (*yhfW* OR *deoB*)
		10	7	*aas*, *(fadB OR fadJ)*, *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*, *tesA*,
		11	8	*aas*, (*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*, *tesA*, *deoD*

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Table C-6: Knockout strategies from GDLS analysis of *i*JO1366 with PFF model of '*tesA* with α = 0.3.

Growth rate (g h^{-1})	FFA flux (mmol gDW^{-1} h^{-1})	Number of knockouts	Number of gene associations	Gene associations*
0.943	0.127	0	0	None (wild type)
0.797	0.647	2	1	(*fadB* OR *fadJ*)
		2	1	(*fadI* OR *fadA*)
0.771	0.801	1	1	*tesB*
		1	1	*fadE*
		1	1	(*aas* AND *acpP*)
0.767	0.809	3	2	*tesB*, (*rpiA* OR *rpiB*)
0.742	0.927	2	2	*fadE*, *fabF*
0.705	1.083	2	2	*fadE*, *fabA*
0.651	1.272	3	2	(*fadB* OR *fadJ*), *fabA*
		3	2	(*fadI* OR *fadA*), *fabA*
		4	3	(*fadI* OR *fadA*), *fabA*, *codA*
		4	3	(*fadB* OR *fadJ*), *fabA*, *aas*, (*acpP* OR *plsX*)
0.606	1.438	5	4	(*fadB* OR *fadJ*), *fabA*, *aas*, *tesA*
		5	4	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*
		6	5	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, *ldcA*
		6	5	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, *kbl*
		7	5	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, (*gabT* OR *puuE*)
0.447	1.544	10	7	(*fadB* OR *fadJ*), *fabA*, *aas*, *tesA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*
		10	7	(*fadB* OR *fadJ*), *fabA*, *aas*, *tesA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*
		11	8	(*fadB* OR *fadJ*), *fabA*, *aas*, *tesA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*, *ldcA*
		10	7	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *pfkA*
		10	7	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Table C-7: Knockout strategies from GDLS analysis of *i*JO1366 with PFF model of '*tesA* with α = 0.4.

Growth rate (g h^{-1})	**FFA flux (mmol gDW^{-1} h^{-1})**	**Number of knockouts**	**Number of gene associations**	**Gene associations***
0.920	0.192	0	0	None (wild type)
0.677	1.125	2	1	(*fadB* OR *fadJ*)
		2	1	(*fadI* OR *fadA*)
0.624	1.441	1	1	*tesB*
		1	1	*fade*
		1	1	(*aas* AND *acpP*)
0.620	1.452	3	2	*tesB*, (*rpiA* OR *rpiB*)
0.569	1.693	2	2	*fadE*, *fabF*
0.516	1.937	2	2	*fadE*, *fabA*
0.430	2.259	3	2	(*fadB* OR *fadJ*), *fabA*
		3	2	(*fadI* OR *fadA*), *fabA*
		4	3	(*fadI* OR *fadA*), *fabA*, *asnB*
		4	3	(*fadB* OR *fadJ*), *fabA*, argO
0.318	2.340	8	5	(*fadI* OR *fadA*), *fabA*, (*rpe* OR *sgcE*), (*talB* OR *talA*), *fbaA*
0.369	2.493	5	4	(*fadB* OR *fadJ*), *fabA*, *aas*, *teas*
		5	4	(*fadI* OR *fadA*), *fabA*, *aas*, *teas*
		6	5	(*fadB* OR *fadJ*), *fabA*, *aas*, *tesA*, *sstT*
		6	5	(*fadI* OR *fadA*), *fabA*, *aas*, *tesA*, *argO*

*Each gene association consists of one or more genes that map to a unique enzymatic function. In a gene association, genes are related to each other by AND or OR relationships. An AND relationship indicates that both genes are required for the function (and hence only one of the genes need to be knocked out to remove the function), and an OR relationship indicates that only one of the genes are needed for the function (and hence both genes need to knocked out to remove the function).

Appendix D Reaction listings

Table D-1: Listing of reactions knocked out in acetate producing *E. coli* knockout strategies

(See in Chapter 2) Taken from *i*AF1260. Compartments given in brackets.

Knockout Set	Reactions	Reaction equations
frmA, adhE, adhP	ALCD2x	[c] : etoh + nad <==> acald + h + nadh
mhpF, adhE	ACALD	[c] : acald + coa + nad <==> accoa + h + nadh
ydfG	ATHRDHr	[c] : athr-L + nadp <==> 2aobut + h + nadph
	DSERDHr	[c] : nadp + ser-D <==> 2amsa + h + nadph
	LSERDHr	[c] : nadp + ser-L <==> 2amsa + h + nadph
atpABCDEFGH	ATPS4rpp	adp[c] + (4) h[p] + pi[c] <==> atp[c] + (3) h[c] + h2o[c]
Pgi	PGI	[c] : g6p <==> f6p
yahI, ybcF, yqeA	CBMKr	[c] : atp + co2 + nh4 <==> adp + cbp + (2) h
tnaA	TRPAS2	[c] : h2o + trp-L <==> indole + nh4 + pyr
galP	GALt2pp,	gal[p] + h[p] --> gal[c] + h[c]
	GLCt2pp	glc-D[p] + h[p] --> glc-D[c] + h[c]
pitAB	PIt2rpp	h[p] + pi[p] <==> h[c] + pi[c]
(sapD or trkA or trkG), (sapD or trkA or trkH), kch, kup	Kt2pp	h[p] + k[p] --> h[c] + k[c]
guaB	IMPD	[c] : h2o + imp + nad --> h + nadh + xmp

Table D-2: Listing of *i*AF1260 (*E. coli*) reactions referred to in this research

Name	Reaction	Gene Association
3HAD140	[c] : 3hmrsACP --> h2o + tmrs2eACP	*fabZ* OR *fabA*
3HAD141	[c] : 3hcmrs7eACP --> h2o + t3c7mrseACP	*fabZ* OR *fabA*
3OAR140	[c] : 3omrsACP + h + nadph <==> 3hmrsACP + nadp	*fabG*
3OAR141	[c] : 3ocmrs7eACP + h + nadph --> 3hcmrs7eACP + nadp	*fabG*
3OAS140	[c] : ddcaACP + h + malACP --> 3omrsACP + ACP + co2	*fabF* OR *fabB*
3OAS141	[c] : cddec5eACP + h + malACP --> 3ocmrs7eACP + ACP + co2	*fabF* OR *fabB*
3OAS161	[c] : h + malACP + tdeACP --> 3ocpalm9eACP + ACP +	*fabB*

	co2	
AACPS1	[c] : ACP + atp + ttdca --> amp + myrsACP + ppi	*acpP* AND *aas*
AACPS2	[c] : ACP + atp + ttdcea --> amp + ppi + tdeACP	*acpP* AND *aas*
ACOAD1f	[c] : btcoa + fad <==> b2coa + fadh2	*fadE*
ACOAD2f	[c] : fad + hxcoa <==> fadh2 + hx2coa	*fadE*
ACOAD3f	[c] : fad + occoa <==> fadh2 + oc2coa	*fadE*
ACOAD4f	[c] : dcacoa + fad <==> dc2coa + fadh2	*fadE*
ACOAD5f	[c] : ddcacoa + fad <==> dd2coa + fadh2	*fadE*
ACOAD6f	[c] : fad + tdcoa <==> fadh2 + td2coa	*fadE*
ACOAD7f	[c] : fad + pmtcoa <==> fadh2 + hdd2coa	*fadE*
ACOAD8f	[c] : fad + stcoa <==> fadh2 + od2coa	*fadE*
ACONTa	[c] : cit <==> acon-C + h2o	*acnB* OR *acnA*
ACONTb	[c] : acon-C + h2o <==> icit	*acnB OR acnA*
AGPAT141	[c] : 1tdec7eg3p + tdeACP --> ACP + pa141	*plsC*
AKGDH	[c] : akg + coa + nad --> co2 + nadh + succoa	*lpd* AND *sucA* AND *sucB*
CS	[c] : accoa + h2o + oaa --> cit + coa + h	*gltA*
EAR140x	[c] : h + nadh + tmrs2eACP --> myrsACP + nad	*fabI*
EAR141x	[c] : h + nadh + t3c7mrseACP --> nad + tdeACP	*fabI*
ECOAH6	[c] : 3htdcoa <==> h2o + td2coa	*fadJ* OR *fadB*
EDA	[c] : 2ddg6p --> g3p + pyr	*eda*
EDD	[c] : 6pgc --> 2ddg6p + h2o	*edd*
ENO	[c] : 2pg <==> h2o + pep	*eno*
FACOAE140	[c] : h2o + tdcoa --> coa + h + ttdca	*tesB*
FACOAE141	[c] : h2o + tdecoa --> coa + h + ttdcea	*tesB*
FACOAL141t2pp	atp[c] + coa[c] + h[p] + ttdcea[p] --> amp[c] + h[c] + ppi[c] + tdecoa[c]	*fadK* OR *fadD*
FBA	[c] : fdp <==> dhap + g3p	*ydjI* OR *baeR* OR *fbaA*
FUM	[c] : fum + h2o <==> mal-L	*fumC* OR *fumA* OR *fum B*
G3PAT141	[c] : glyc3p + tdeACP --> 1tdec7eg3p + ACP	*plsB*
G6PDH2r	[c] : g6p + nadp <==> 6pgl + h + nadph	*zwf*
GAPD	[c] : g3p + nad + pi <==> 13dpg + h + nadh	*gapA*
GND	[c] : 6pgc + nadp --> co2 + nadph + ru5p-D	*gnd*
HACD6i	[c] : 3htdcoa + nad --> 3otdcoa + h + nadh	*fadJ* OR *fadB*
ICDHyr	[c] : icit + nadp <==> akg + co2 + nadph	*icd*
ICL	[c] : icit --> glx + succ	*aceA*
KAT6	[c] : 3otdcoa + coa --> accoa + ddcacoa	*fadI* OR *fadA*
LPLIPAL1A141pp	[p] : 1tdec7eg3p + h2o --> glyc3p + h + ttdcea	*tesA*
MALS	[c] : accoa + glx + h2o --> coa + h + mal-L	*glcB* OR *aceB*
MDH	[c] : mal-L + nad <==> h + nadh + oaa	*mdh*
PA141abcpp	atp[c] + h2o[c] + pa141[c] --> adp[c] + h[c] + pa141[p] + pi[c]	*msbA*
PDH	[c] : coa + nad + pyr --> accoa + co2 + nadh	*aceE AND aceF* AND *lpd*
PFK	[c] : atp + f6p --> adp + fdp + h	*pfkB* OR *pfkA*
PGCD	[c] : 3pg + nad --> 3php + h + nadh	*serA*
PGI	[c] : g6p <==> f6p	*pgi*
PGK	[c] : 3pg + atp <==> 13dpg + adp	*pgk*

PGL	[c] : 6pgl + h2o --> 6pgc + h	*pgl*
PGM	[c] : 2pg <==> 3pg	*gpmA OR gpmM OR ytjC*
PLIPA2A141pp	[p] : h2o + pa141 --> 1tdec7eg3p + h + ttdcea	*pldA*
PPC	[c] : co2 + h2o + pep --> h + oaa + pi	*ppc*
PSERT	[c] : 3php + glu-L --> akg + pser-L	*serC*
PYK	[c] : adp + h + pep --> atp + pyr	*pykF OR pykA*
RPE	[c] : ru5p-D <==> xu5p-D	*rpe* OR *sgcE*
RPI	[c] : r5p <==> ru5p-D	*rpiA OR rpiB*
SUCDi	[c] : q8 + succ --> fum + q8h2	*sdhC* AND *sdhD* AND *sdhA AND sdhB*
SUCOAS	[c] : atp + coa + succ <==> adp + pi + succoa	*sucC AND sucD*
T2DECAI	[c] : tdec2eACP <==> cdec3eACP	*fabA*
TALA	[c] : g3p + s7p <==> e4p + f6p	*talB* OR *talA*
TDECOAI	[c] : tdecoa --> td2coa	*fadB*
TKT1	[c] : r5p + xu5p-D <==> g3p + s7p	*tktB* OR *tktA*
TKT2	[c] : e4p + xu5p-D <==> f6p + g3p	*tktB OR tktA*
TPI	[c] : dhap <==> g3p	*tpiA*

Table D-3: Listing of additional *i*JO1366 (*E. coli*) reactions referred to in research

Name	Reaction	Gene Association
ACACT1r	[c] : aacoa + coa <==> (2) accoa	atoB OR fadI OR fadA
ACACT2r	[c] : 3ohcoa + coa <==> accoa + btcoa	fadI OR fadA
ACACT3r	[c] : 3oocoa + coa <==> accoa + hxcoa	fadI OR fadA
ACACT4r	[c] : 3odcoa + coa <==> accoa + occoa	fadI OR fadA
ACACT5r	[c] : 3oddcoa + coa <==> accoa + dcacoa	fadI OR fadA
ACACT6r	[c] : 3otdcoa + coa <==> accoa + ddcacoa	fadI OR fadA
ACACT7r	[c] : 3ohdcoa + coa <==> accoa + tdcoa	fadI OR fadA
ACACT8r	[c] : 3oodcoa + coa <==> accoa + pmtcoa	fadI OR fadA
CTECOAI6	[c] : tdecoa <==> td2coa	fabB
CTECOAI7	[c] : hdcoa <==> hdd2coa	fabB
CTECOAI8	[c] : odecoa <==> od2coa	fabB
FBA3	[c] : s17bp <==> dhap + e4p	fbaA
HACD1	[c] : 3hbcoa + nad <==> aacoa + h + nadh	fadB OR fadJ
HACD2	[c] : 3hhcoa + nad <==> 3ohcoa + h + nadh	fadB OR fadJ
HACD3	[c] : 3hocoa + nad <==> 3oocoa + h + nadh	fadB OR fadJ
HACD4	[c] : 3hdcoa + nad <==> 3odcoa + h + nadh	fadB OR fadJ
HACD5	[c] : 3hddcoa + nad <==> 3oddcoa + h + nadh	fadB OR fadJ
HACD6	[c] : 3htdcoa + nad <==> 3otdcoa + h + nadh	fadB OR fadJ
HACD7	[c] : 3hhdcoa + nad <==> 3ohdcoa + h + nadh	fadB OR fadJ
HACD8	[c] : 3hodcoa + nad <==> 3oodcoa + h + nadh	fadB OR fadJ
PFK_3	[c] : atp + s7p --> adp + h + s17bp	pfkA

Table D-4: Listing of GSMN-TB (MTB) reactions referred to in this research

Name	Reaction	Gene Association
HtrA (R141)	1 HYDROXYAKG = 1 GLX + 1 PYR	htrA (Rv1223)
ICL (R102)	1 ICIT = 1 GLX + 1 SUCC	Icl1 OR (icl2a AND icl2b) (Rv0467) OR (Rv1915 AND Rv1916)
MALS (R103)	1 GLX + 1 ACCOA = 1 MAL + 1 COA	glcB (Rv1837c)
NGAM (R129)	1 ATP = 1 ADP + 1 PI	mgtC (Rv1811)

Appendix E Elementary flux modes for acetate-producing *E. coli* knockout strategies

Table E-1: Listing of elementary flux modes for acetate producting *E. coli* knockout strategies

(See Chapter 2) Elementary modes have been normalised to glucose uptake flux. Only reactions used in the elementary modes have been included. Flux values in bold have been rounded to 4 decimal places, with true zero flux values grayed out. Reaction names and formulas copied from the *i*AF1260—see Feist et al. (2007) for full reaction information. Fluxes of interest have been highlighted in orange.

	Knockouts		0		1		2		3		4		5		6		8	
	w	2.9197	7.0803	4.1818	5.8182	4.0679	5.9321	2.8199	7.1801	2.9118	7.0882	2.8714	7.1286	6.5256	3.4744	6.5256	3.4744
Name	Reaction			NGAM	BIO	NGAM	BIO	NGAM	BIO	NGAM	BIO	NGAM	BIO	NGAM	BIO	NGAM	BIO
EX_12ppd-R(e)	[e] : 12ppd-R <==>	0.0000	0.0000	**0.6667**	**0.5539**	0.0000	0.0000	**0.4444**	**0.4078**	**0.4444**	**0.4078**	**0.4444**	**0.4080**	**0.0714**	**0.1143**	**0.0714**	**0.1136**
EX_ac(e)	[e] : ac <==>	**0.7850**	**0.8488**	**1.3333**	**1.1215**	**1.5000**	**1.2962**	**1.5556**	**1.4947**	**1.5556**	**1.4947**	**1.5556**	**1.4955**	**1.6905**	**1.4036**	**1.6905**	**1.4055**
EX_ca2(e)	[e] : ca2 <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**-0.0001**	0.0000	**-0.0001**
EX_cl(e)	[e] : cl <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**-0.0001**	0.0000	**-0.0001**
EX_co2(e)	[e] : co2 <==>	**1.6610**	**1.7616**	**1.3333**	**1.2009**	**1.5000**	**1.3058**	**1.5556**	**1.4860**	**1.5556**	**1.4860**	**1.5556**	**1.4856**	**1.3571**	**1.3039**	**1.3571**	**1.3040**
EX_cobalt2(e)	[e] : cobalt2 <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
EX_cu2(e)	[e] : cu2 <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
EX_etoh(e)	[e] : etoh <==>	**0.7661**	**0.8355**	0.0000	0.0000	**0.7500**	**0.5863**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.5238**	**0.4673**	**0.5238**	**0.4678**
EX_fe2(e)	[e] : fe2 <==>	**-0.0002**	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**
EX_fe3(e)	[e] : fe3 <==>	**-0.0002**	**-0.0001**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**
EX_glc(e)	[e] : glc-D <==>	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**
EX_glyclt(e)	[e] : glyclt <==>	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EX_h(e)	[e] : h <==>	**1.0732**	**1.0514**	**1.3333**	**1.3298**	**1.5000**	**1.5127**	**1.2222**	**1.2513**	**1.2222**	**1.2513**	**1.2222**	**1.2521**	**1.6905**	**1.5460**	**1.6905**	**1.5473**
EX_h2(e)	[e] : h2 <==>	**1.6751**	**1.7715**	**1.3333**	**1.2112**	**1.5000**	**1.3529**	**1.5556**	**1.5012**	**1.5556**	**1.5012**	**1.5556**	**1.5012**	**1.5238**	**1.3879**	**1.5238**	**1.3888**
EX_h2o(e)	[e] : h2o <==>	0.0000	**-0.2967**	**-0.6667**	0.0000	**-0.7500**	**-0.0473**	**-0.4444**	**-0.2330**	**-0.4444**	**-0.2330**	**-0.4444**	**-0.2347**	**-0.7619**	**-0.2800**	**-0.7619**	**-0.2823**
EX_h2s(e)	[e] : h2s <==>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.1667**	**0.1568**	**0.1667**	**0.1568**	**0.1667**	**0.1566**	0.0000	0.0000	0.0000	0.0000
EX_k(e)	[e] : k <==>	**-0.0052**	**-0.0037**	0.0000	**-0.0038**	0.0000	**-0.0039**	0.0000	**-0.0013**	0.0000	**-0.0013**	0.0000	**-0.0013**	0.0000	**-0.0026**	0.0000	**-0.0026**
EX_mg2(e)	[e] : mg2 <==>	**-0.0002**	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**
EX_mn2(e)	[e] : mn2 <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
EX_mobd(e)	[e] : mobd <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
EX_nh4(e)	[e] : nh4 <==>	**-0.3154**	**-0.2218**	0.0000	**-0.2281**	0.0000	**-0.2370**	0.0000	**-0.0768**	0.0000	**-0.0768**	0.0000	**-0.0763**	0.0000	**-0.1559**	0.0000	**-0.1552**
EX_pi(e)	[e] : pi <==>	**-0.0281**	**-0.0198**	0.0000	**-0.0203**	0.0000	**-0.0211**	0.0000	**-0.0068**	0.0000	**-0.0068**	0.0000	**-0.0068**	0.0000	**-0.0139**	0.0000	**-0.0138**
EX_so4(e)	[e] : so4 <==>	**-0.0073**	**-0.0051**	0.0000	**-0.0053**	0.0000	**-0.0055**	**-0.1667**	**-0.1586**	**-0.1667**	**-0.1586**	**-0.1667**	**-0.1584**	0.0000	**-0.0036**	0.0000	**-0.0036**
EX_succ(e)	[e] : succ <==>	**0.0098**	**0.0069**	0.0000	**0.0071**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0048**	0.0000	**0.0048**
EX_zn2(e)	[e] : zn2 <==>	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
12PPDRtex	12ppd-R[e] <==> 12ppd-R[p]	0.0000	0.0000	**-0.6667**	**-0.5539**	0.0000	0.0000	**-0.4444**	**-0.4078**	**-0.4444**	**-0.4078**	**-0.4444**	**-0.4080**	**-0.0714**	**-0.1143**	**-0.0714**	**-0.1136**
12PPDRtpp	12ppd-R[p] <==> 12ppd-R[c]	0.0000	0.0000	**-0.6667**	**-0.5539**	0.0000	0.0000	**-0.4444**	**-0.4078**	**-0.4444**	**-0.4078**	**-0.4444**	**-0.4080**	**-0.0714**	**-0.1143**	**-0.0714**	**-0.1136**
3HAD100	[c] : 3hdecACP --> h2o + tdec2eACP	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3HAD120	[c] : 3hddecACP --> h2o + tddec2eACP	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
3HAD121	[c] : 3hcddec5eACP --> h2o + t3c5ddeceACP	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3HAD140	[c] : 3hmrsACP --> h2o + tmrs2eACP	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**
3HAD141	[c] : 3hcmrs7eACP --> h2o + t3c7mrseACP	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**

3HAD160	[c] : 3hpalmACP --> h2o + tpalm2eACP	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**
3HAD161	[c] : 3hcpalm9eACP --> h2o + t3c9palmeACP	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3HAD40	[c] : 3haACP --> but2eACP + h2o	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3HAD60	[c] : 3hhexACP --> h2o + thex2eACP	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3HAD80	[c] : 3hoctACP --> h2o + toct2eACP	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAR100	[c] : 3odecACP + h + nadph <==> 3hdecACP + nadp	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAR120	[c] : 3oddecACP + h + nadph <==> 3hddecACP + nadp	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
3OAR121	[c] : 3ocddec5eACP + h + nadph --> 3hcddec5eACP + nadp	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3OAR140	[c] : 3omrsACP + h + nadph <==> 3hmrsACP + nadp	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
3OAR141	[c] : 3ocmrs7eACP + h + nadph --> 3hcmrs7eACP + nadp	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3OAR160	[c] : 3opalmACP + h + nadph <==> 3hpalmACP + nadp	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**
3OAR161	[c] : 3ocpalm9eACP + h + nadph --> 3hcpalm9eACP + nadp	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3OAR40	[c] : actACP + h + nadph <==> 3haACP + nadp	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAR60	[c] : 3ohexACP + h + nadph <==> 3hhexACP + nadp	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAR80	[c] : 3ooctACP + h + nadph <==> 3hoctACP + nadp	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAS100	[c] : h + malACP + ocACP --> 3odecACP + ACP + co2	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAS120	[c] : dcaACP + h + malACP --> 3oddecACP + ACP + co2	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
3OAS121	[c] : cdec3eACP + h + malACP --> 3ocddec5eACP + ACP + co2	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3OAS140	[c] : ddcaACP + h + malACP --> 3omrsACP + ACP + co2	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
3OAS141	[c] : cddec5eACP + h + malACP --> 3ocmrs7eACP + ACP + co2	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
3OAS160	[c] : h + malACP + myrsACP --> 3opalmACP + ACP + co2	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**
3OAS161	[c] : h + malACP + tdeACP --> 3ocpalm9eACP + ACP + co2	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**

3OAS60	[c] : butACP + h + malACP --> 3ohexACP + ACP + co2	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
3OAS80	[c] : h + hexACP + malACP --> 3ooctACP + ACP + co2	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
5DOAN	[c] : dad-5 + h2o --> 5drib + ade	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
A5PISO	[c] : ru5p-D <==> ara5p	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
ACALD	[c] : acald + coa + nad <==> accoa + h + nadh	**-0.7661**	**-0.8355**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0118**	0.0000	**0.0118**	0.0000	**0.0122**	0.0000	0.0000	0.0000	0.0000
ACCOAC	[c] : accoa + atp + hco3 --> adp + h + malcoa + pi	**0.0706**	**0.0496**	0.0000	**0.0510**	0.0000	**0.0530**	0.0000	**0.0172**	0.0000	**0.0172**	0.0000	**0.0171**	0.0000	**0.0349**	0.0000	**0.0347**
ACGK	[c] : acglu + atp --> acg5p + adp	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ACGS	[c] : accoa + glu-L --> acglu + coa + h	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ACHBS	[c] : 2obut + h + pyr --> 2ahbut + co2	**0.0085**	**0.0060**	0.0000	**0.0061**	0.0000	**0.0064**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0042**	0.0000	**0.0042**
ACKr	[c] : ac + atp <==> actp + adp	**-0.7680**	**-0.8368**	**-1.3333**	**-1.1092**	**-1.5000**	**-1.2834**	**-1.5556**	**-1.4905**	**-1.5556**	**-1.4905**	**-1.5556**	**-1.4914**	**-1.6905**	**-1.3952**	**-1.6905**	**-1.3972**
ACLS	[c] : h + (2) pyr --> alac-S + co2	**0.0256**	**0.0180**	0.0000	**0.0185**	0.0000	**0.0192**	0.0000	**0.0062**	0.0000	**0.0062**	0.0000	**0.0062**	0.0000	**0.0126**	0.0000	**0.0126**
ACOATA	[c] : ACP + accoa <==> acACP + coa	**0.0104**	**0.0073**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0025**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ACODA	[c] : acorn + h2o --> ac + orn	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ACONTa	[c] : cit <==> acon-C + h2o	**0.0315**	**0.0221**	0.0000	**0.0227**	0.0000	**0.0236**	0.0000	**0.0077**	0.0000	**0.0077**	0.0000	**0.0076**	0.0000	**0.0155**	0.0000	**0.0155**
ACONTb	[c] : acon-C + h2o <==> icit	**0.0315**	**0.0221**	0.0000	**0.0227**	0.0000	**0.0236**	0.0000	**0.0077**	0.0000	**0.0077**	0.0000	**0.0076**	0.0000	**0.0155**	0.0000	**0.0155**
ACOTA	[c] : acorn + akg <==> acg5sa + glu-L	**-0.0087**	**-0.0061**	0.0000	**-0.0063**	0.0000	**-0.0065**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0043**	0.0000	**-0.0043**
ACt2rpp	ac[p] + h[p] <==> ac[c] + h[c]	**-0.7850**	**-0.8488**	**-1.3333**	**-1.1215**	**-1.5000**	**-1.2962**	**-1.5556**	**-1.4947**	**-1.5556**	**-1.4947**	**-1.5556**	**-1.4955**	**-1.6905**	**-1.4036**	**-1.6905**	**-1.4055**
ACtex	ac[e] <==> ac[p]	**-0.7850**	**-0.8488**	**-1.3333**	**-1.1215**	**-1.5000**	**-1.2962**	**-1.5556**	**-1.4947**	**-1.5556**	**-1.4947**	**-1.5556**	**-1.4955**	**-1.6905**	**-1.4036**	**-1.6905**	**-1.4055**
ADCL	[c] : 4adcho --> 4abz + h + pyr	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ADCS	[c] : chor + gln-L --> 4adcho + glu-L	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ADK1	[c] : amp + atp <==> (2) adp	**0.0885**	**0.0622**	0.0000	**0.0448**	**0.7500**	**0.5964**	**0.3333**	**0.3353**	**0.3333**	**0.3287**	**0.3333**	**0.3282**	**0.3571**	**0.4206**	**0.3571**	**0.4201**
ADK3	[c] : amp + gtp <==> adp + gdp	**-0.0223**	**-0.0156**	0.0000	**-0.0167**	0.0000	**-0.0173**	**-0.1667**	**-0.1624**	**-0.1667**	**-0.1624**	**-0.1667**	**-0.1622**	**-0.3571**	**-0.4013**	**-0.3571**	**-0.4009**
ADNK1	[c] : adn + atp --> adp + amp + h	**0.0000**	**0.0000**	0.0000	**0.0000**	**0.7500**	**0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ADPT	[c] : ade + prpp --> amp + ppi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
ADSK	[c] : aps + atp --> adp + h + paps	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1585**	**0.1667**	**0.1585**	**0.1667**	**0.1583**	0.0000	**0.0036**	0.0000	**0.0035**
ADSL1r	[c] : dcamp <==> amp + fum	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0066**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ADSL2r	[c] : 25aics <==> aicar + fum	**0.0131**	**0.0092**	0.0000	**0.0094**	0.0000	**0.0098**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0064**
ADSS	[c] : asp-L + gtp + imp --> dcamp + gdp + (2) h + pi	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0066**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
AGPAT160	[c] : 1hdecg3p + palmACP --> ACP + pa160	**0.0019**	**0.0013**	0.0000	**0.0013**	0.0000	**0.0014**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0009**	0.0000	**0.0009**
AGPAT161	[c] : 1hdec9eg3p + hdeACP --> ACP + pa161	**0.0022**	**0.0015**	0.0000	**0.0016**	0.0000	**0.0017**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0011**	0.0000	**0.0011**
AGPR	[c] : acg5sa + nadp + pi <==> acg5p + h + nadph	**-0.0087**	**-0.0061**	0.0000	**-0.0063**	0.0000	**-0.0065**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0043**	0.0000	**-0.0043**

AHCYSNS	[c] : ahcys + h2o --> ade + rhcys	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
AICART	[c] : 10fthf + aicar <==> fprica + thf	**0.0158**	**0.0111**	0.0000	**0.0115**	0.0000	**0.0119**	0.0000	**0.0039**	0.0000	**0.0039**	0.0000	**0.0038**	0.0000	**0.0078**	0.0000	**0.0078**
AIRC2	[c] : air + atp + hco3 --> 5caiz + adp + h + pi	**0.0131**	**0.0092**	0.0000	**0.0094**	0.0000	**0.0098**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0064**
AIRC3	[c] : 5aizc <==> 5caiz	**-0.0131**	**-0.0092**	0.0000	**-0.0094**	0.0000	**-0.0098**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0065**	0.0000	**-0.0064**
ALAALAr	[c] : (2) ala-D + atp <==> adp + alaala + h + pi	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
ALAR	[c] : ala-L <==> ala-D	**0.0012**	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
ALATA_L	[c] : akg + ala-L <==> glu-L + pyr	**-0.0171**	**-0.0120**	0.0000	**-0.0123**	0.0000	**-0.0128**	0.0000	**-0.0042**	0.0000	**-0.0042**	0.0000	**-0.0041**	0.0000	**-0.0084**	0.0000	**-0.0084**
ALCD2x	[c] : etoh + nad <==> acald + h + nadh	**-0.7661**	**-0.8355**	0.0000	0.0000	**-0.7500**	**-0.5863**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**-0.5238**	**-0.4673**	**-0.5238**	**-0.4678**
AMPMS2	[c] : air + h2o + nad --> 4ampm + (2) for + (3) h + nadh	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
ANPRT	[c] : anth + prpp --> ppi + pran	**0.0017**	**0.0012**	0.0000	**0.0012**	0.0000	**0.0012**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0008**	0.0000	**0.0008**
ANS	[c] : chor + gln-L --> anth + glu-L + h + pyr	**0.0017**	**0.0012**	0.0000	**0.0012**	0.0000	**0.0012**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0008**	0.0000	**0.0008**
APRAUR	[c] : 5apru + h + nadph --> 5aprbu + nadp	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
ARGSL	[c] : argsuc <==> arg-L + fum	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ARGSS	[c] : asp-L + atp + citr-L --> amp + argsuc + h + ppi	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
ASAD	[c] : aspsa + nadp + pi <==> 4pasp + h + nadph	**-0.0313**	**-0.0220**	0.0000	**-0.0226**	0.0000	**-0.0600**	0.0000	**-0.0195**	0.0000	**-0.0195**	0.0000	**-0.0197**	**-0.1667**	**-0.0925**	**-0.1667**	**-0.0932**
ASNS2	[c] : asp-L + atp + nh4 --> amp + asn-L + h + ppi	**0.0071**	**0.0050**	0.0000	**0.0051**	0.0000	**0.0053**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0035**	0.0000	**0.0035**
ASP1DC	[c] : asp-L + h --> ala-B + co2	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ASPCT	[c] : asp-L + cbp --> cbasp + h + pi	**0.0097**	**0.0068**	0.0000	**0.0070**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0023**	0.0000	**0.0048**	0.0000	**0.0048**
ASPK	[c] : asp-L + atp <==> 4pasp + adp	**0.0313**	**0.0220**	0.0000	**0.0226**	0.0000	**0.0600**	0.0000	**0.0195**	0.0000	**0.0195**	0.0000	**0.0197**	**0.1667**	**0.0925**	**0.1667**	**0.0932**
ASPO5	[c] : asp-L + fum --> h + iasp + succ	**0.0001**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ASPTA	[c] : akg + asp-L <==> glu-L + oaa	**-0.0856**	**-0.0602**	0.0000	**-0.0619**	0.0000	**-0.1008**	0.0000	**-0.0327**	0.0000	**-0.0327**	0.0000	**-0.0329**	**-0.1667**	**-0.1193**	**-0.1667**	**-0.1199**
ATPM (NGAM)	[c] : atp + h2o --> adp + h + pi	**0.1978**	**1.1034**	**2.0000**	**0.0045**	**2.0625**	0.0000	**1.3333**	**0.6449**	**1.3333**	**0.6359**	**1.3333**	**0.6399**	**1.2857**	0.0000	**1.2857**	0.0000
ATPPRT	[c] : atp + prpp --> ppi + prbatp	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
ATPS4rpp	adp[c] + (4) h[p] + pi[c] <==> atp[c] + (3) h[c] + h2o[c]	**-0.1055**	0.0000	0.0000	**-0.2147**	**-0.1875**	**-0.4062**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
BPNT	[c] : h2o + pap --> amp + pi	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1585**	**0.1667**	**0.1585**	**0.1667**	**0.1583**	0.0000	**0.0036**	0.0000	**0.0035**
CA2tex	ca2[e] <==> ca2[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
CAt6pp	ca2[c] + na1[p] <==> ca2[p] + na1[c]	**-0.0001**	**-0.0001**	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**-0.0001**	0.0000	**-0.0001**
CBMKr	[c] : atp + co2 + nh4 <==> adp + cbp + (2) h	**0.0183**	**0.0129**	0.0000	**0.0133**	0.0000	**0.0138**	0.0000	**0.0045**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0091**	0.0000	**0.0090**
CBPS	[c] : (2) atp + gln-L + h2o + hco3 -->	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0045**	0.0000	**0.0044**	0.0000	0.0000	0.0000	0.0000

	(2) adp + cbp + glu-L + (2) h + pi																
CDPMEK	[c] : 4c2me + atp --> 2p4c2me + adp + h	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CHORM	[c] : chor --> pphn	**0.0092**	**0.0065**	0.0000	**0.0066**	0.0000	**0.0069**	0.0000	**0.0022**	0.0000	**0.0022**	0.0000	**0.0022**	0.0000	**0.0045**	0.0000	**0.0045**
CHORS	[c] : 3psme --> chor + pi	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
CHRPL	[c] : chor --> 4hbz + pyr	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CLt3_2pp	(2) cl[p] + h[c] --> (2) cl[c] + h[p]	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CLtex	cl[e] <==> cl[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
CO2tex	co2[e] <==> co2[p]	**-1.6610**	**-1.7616**	**-1.3333**	**-1.2009**	**-1.5000**	**-1.3058**	**-1.5556**	**-1.4860**	**-1.5556**	**-1.4860**	**-1.5556**	**-1.4856**	**-1.3571**	**-1.3039**	**-1.3571**	**-1.3040**
CO2tpp	co2[p] <==> co2[c]	**-1.6610**	**-1.7616**	**-1.3333**	**-1.2009**	**-1.5000**	**-1.3058**	**-1.5556**	**-1.4860**	**-1.5556**	**-1.4860**	**-1.5556**	**-1.4856**	**-1.3571**	**-1.3039**	**-1.3571**	**-1.3040**
COBALT2tex	cobalt2[e] <==> cobalt2[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
COBALT2tpp	cobalt2[p] --> cobalt2[c]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CPPPGO2	[c] : (2) amet + cpppg3 --> (2) co2 + (2) dad-5 + (2) met-L + pppg9	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CS	[c] : accoa + h2o + oaa --> cit + coa + h	**0.0315**	**0.0221**	0.0000	**0.0227**	0.0000	**0.0236**	0.0000	**0.0077**	0.0000	**0.0077**	0.0000	**0.0076**	0.0000	**0.0155**	0.0000	**0.0155**
CTPS2	[c] : atp + gln-L + h2o + utp --> adp + ctp + glu-L + (2) h + pi	**0.0047**	**0.0033**	0.0000	**0.0034**	0.0000	**0.0035**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0023**	0.0000	**0.0023**
CU2tex	cu2[e] <==> cu2[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CU2tpp	cu2[p] --> cu2[c]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
CYSS	[c] : acser + h2s --> ac + cys-L + h	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	0.0000	**0.0018**	0.0000	**0.0018**	0.0000	**0.0017**	0.0000	**0.0036**	0.0000	**0.0035**
CYSTL	[c] : cyst-L + h2o --> hcys-L + nh4 + pyr	**0.0045**	**0.0032**	0.0000	**0.0033**	0.0000	**0.0034**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
CYTK1	[c] : atp + cmp <==> adp + cdp	**0.0053**	**0.0037**	0.0000	**0.0038**	0.0000	**0.0040**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0026**	0.0000	**0.0026**
DADK	[c] : atp + damp <==> adp + dadp	0.0000	0.0000	0.0000	0.0000	**-0.7500**	**-0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DALAt2pp	ala-D[p] + h[p] --> ala-D[c] + h[c]	**0.0004**	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0002**	0.0000	**0.0002**
DAPDC	[c] : 26dap-M + h --> co2 + lys-L	**0.0100**	**0.0071**	0.0000	**0.0073**	0.0000	**0.0075**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0050**	0.0000	**0.0049**
DAPE	[c] : 26dap-LL <==> 26dap-M	**0.0109**	**0.0076**	0.0000	**0.0078**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0053**
DASYN160	[c] : ctp + h + pa160 --> cdpdhdecg + ppi	**0.0019**	**0.0013**	0.0000	**0.0013**	0.0000	**0.0014**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0009**	0.0000	**0.0009**
DASYN161	[c] : ctp + h + pa161 --> cdpdhdec9eg + ppi	**0.0022**	**0.0015**	0.0000	**0.0016**	0.0000	**0.0017**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0011**	0.0000	**0.0011**
DB4PS	[c] : ru5p-D --> db4p + for + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DDPA	[c] : e4p + h2o + pep --> 2dda7p + pi	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
DHAD1	[c] : 23dhmb --> 3mob + h2o	**0.0256**	**0.0180**	0.0000	**0.0185**	0.0000	**0.0192**	0.0000	**0.0062**	0.0000	**0.0062**	0.0000	**0.0062**	0.0000	**0.0126**	0.0000	**0.0126**
DHAD2	[c] : 23dhmp --> 3mop + h2o	**0.0085**	**0.0060**	0.0000	**0.0061**	0.0000	**0.0064**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0042**	0.0000	**0.0042**
DHAPT	[c] : dha + pep --> dhap + pyr	0.0000	0.0000	**0.3333**	**0.2736**	**0.5000**	**0.4542**	**0.2222**	0.0000	0.0000	0.0000	**0.2222**	0.0000	0.0000	0.0000	0.0000	0.0000
DHDPRy	[c] : 23dhdp + h + nadph --> nadp + thdp	**0.0109**	**0.0076**	0.0000	**0.0078**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0053**

DHDPS	[c] : aspsa + pyr --> 23dhdp + h + (2) h2o	**0.0109**	**0.0076**	0.0000	**0.0078**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0053**
DHFR	[c] : dhf + h + nadph <==> nadp + thf	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
DHFS	[c] : atp + dhpt + glu-L --> adp + dhf + h + pi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DHNPA2	[c] : dhnpt --> 6hmhpt + gcald	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DHORD5	[c] : dhor-S + mqn8 --> mql8 + orot	**0.0097**	**0.0068**	0.0000	**0.0070**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0023**	0.0000	**0.0048**	0.0000	**0.0048**
DHORTS	[c] : dhor-S + h2o <==> cbasp + h	**-0.0097**	**-0.0068**	0.0000	**-0.0070**	0.0000	**-0.0073**	0.0000	**-0.0024**	0.0000	**-0.0024**	0.0000	**-0.0023**	0.0000	**-0.0048**	0.0000	**-0.0048**
DHPPDA2	[c] : 25drapp + h + h2o --> 5apru + nh4	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DHPS2	[c] : 4abz + 6hmhptpp --> dhpt + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DHPTDCs	[c] : dhptd --> h2o + hmfurn	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DHQS	[c] : 2dda7p --> 3dhq + pi	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
DHQTi	[c] : 3dhq --> 3dhsk + h2o	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
D4HBA	[c] : 4hba -->	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
D5DRIB	[c] : 5drib -->	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DHMFURN	[c] : hmfurn -->	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DMATT	[c] : dmpp + ipdp --> grdp + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DMPPS	[c] : h + h2mb4p + nadh --> dmpp + h2o + nad	0.0000	0.0000	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DNMPPA	[c] : dhpmp + h2o --> dhnpt + pi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DNTPPA	[c] : ahdt + h2o --> dhpmp + h + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DPCOAK	[c] : atp + dpcoa --> adp + coa + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DPR	[c] : 2dhp + h + nadph --> nadp + pant-R	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DRPA	[c] : 2dr5p --> acald + g3p	0.0000	0.0000	0.0000	0.0000	**0.7500**	**0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3571**	**0.3903**	**0.3571**	**0.3900**
DTMPK	[c] : atp + dtmp <==> adp + dtdp	**0.0008**	**0.0005**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
DURIPP	[c] : duri + pi <==> 2dr1p + ura	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3571**	**0.3903**	**0.3571**	**0.3900**
DXPRIi	[c] : dxyl5p + h + nadph --> 2me4p + nadp	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
DXPS	[c] : g3p + h + pyr --> co2 + dxyl5p	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
E4PD	[c] : e4p + h2o + nad <==> 4per + (2) h + nadh	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
EAR100x	[c] : h + nadh + tdec2eACP --> dcaACP + nad	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
EAR120x	[c] : h + nadh + tddec2eACP --> ddcaACP + nad	**0.0060**	**0.0042**	0.0000	**0.0043**	0.0000	**0.0045**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0015**	0.0000	**0.0030**	0.0000	**0.0030**
EAR121x	[c] : h + nadh + t3c5ddeceACP --> cddec5eACP + nad	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
EAR140x	[c] : h + nadh + tmrs2eACP -->	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**

	myrsACP + nad																
EAR141x	[c] : h + nadh + t3c7mrseACP --> nad + tdeACP	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
EAR160x	[c] : h + nadh + tpalm2eACP --> nad + palmACP	**0.0037**	**0.0026**	0.0000	**0.0027**	0.0000	**0.0028**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0018**	0.0000	**0.0018**
EAR161x	[c] : h + nadh + t3c9palmeACP --> hdeACP + nad	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
EAR40x	[c] : but2eACP + h + nadh --> butACP + nad	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
EAR60x	[c] : h + nadh + thex2eACP --> hexACP + nad	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
EAR80x	[c] : h + nadh + toct2eACP --> nad + ocACP	**0.0104**	**0.0073**	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
EDA	[c] : 2ddg6p --> g3p + pyr	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.6667**	**0.7670**	**0.6667**	**0.7670**	**0.6667**	**0.7669**	**0.7619**	**0.7273**	**0.7619**	**0.7276**
EDD	[c] : 6pgc --> 2ddg6p + h2o	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.6667**	**0.7670**	**0.6667**	**0.7670**	**0.6667**	**0.7669**	**0.7619**	**0.7273**	**0.7619**	**0.7276**
ENO	[c] : 2pg <==> h2o + pep	**1.8814**	**1.9166**	**1.3333**	**1.3604**	**1.5000**	**1.5809**	**0.8889**	**0.8081**	**0.8889**	**0.8081**	**0.8889**	**0.8081**	**0.9286**	**0.8636**	**0.9286**	**0.8645**
ETOHt2rpp	etoh[p] + h[p] <==> etoh[c] + h[c]	**-0.7661**	**-0.8355**	0.0000	0.0000	**-0.7500**	**-0.5863**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**-0.5238**	**-0.4673**	**-0.5238**	**-0.4678**
ETOHtex	etoh[e] <==> etoh[p]	**-0.7661**	**-0.8355**	0.0000	0.0000	**-0.7500**	**-0.5863**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**-0.5238**	**-0.4673**	**-0.5238**	**-0.4678**
F6PA	[c] : f6p <==> dha + g3p	0.0000	0.0000	**0.3333**	**0.2736**	**0.5000**	**0.4542**	**0.2222**	0.0000	0.0000	0.0000	**0.2222**	0.0000	0.0000	0.0000	0.0000	0.0000
FBA	[c] : fdp <==> dhap + g3p	**0.9747**	**0.9822**	**0.6667**	**0.7081**	0.0000	**0.1602**	**0.1111**	**0.2268**	**0.3333**	**0.2268**	**0.1111**	**0.2269**	0.0000	0.0000	0.0000	0.0000
FCLT	[c] : fe2 + ppp9 --> (2) h + pheme	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
FE2tex	fe2[e] <==> fe2[p]	**0.0002**	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
FE2tpp	fe2[p] --> fe2[c]	**0.0002**	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
FE3abcpp	atp[c] + fe3[p] + h2o[c] --> adp[c] + fe3[c] + h[c] + pi[c]	**0.0002**	**0.0001**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
FE3tex	fe3[e] <==> fe3[p]	**0.0002**	**0.0001**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
FHL	[c] : for + h --> co2 + h2	**1.6751**	**1.7715**	**1.3333**	**1.2112**	**1.5000**	**1.3529**	**1.5556**	**1.5012**	**1.5556**	**1.5012**	**1.5556**	**1.5012**	**1.5238**	**1.3879**	**1.5238**	**1.3888**
FLDR	[c] : fldox + h + nadph --> fldrd + nadp	0.0000	0.0000	0.0000	**0.0017**	0.0000	**0.0018**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	0.0000	0.0000	0.0000
FMNAT	[c] : atp + fmn + h --> fad + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
FRD2	[c] : fum + mql8 --> mqn8 + succ	**0.0097**	**0.0068**	0.0000	**0.0070**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0023**	0.0000	**0.0048**	0.0000	**0.0048**
FUM	[c] : fum + h2o <==> mal-L	**0.0207**	**0.0145**	0.0000	**0.0150**	0.0000	**0.0155**	0.0000	**0.0050**	0.0000	**0.0050**	0.0000	**0.0050**	0.0000	**0.0102**	0.0000	**0.0102**
G1PACT	[c] : accoa + gam1p --> acgam1p + coa + h	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
G1SAT	[c] : glu1sa <==> 5aop	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
G3PAT160	[c] : glyc3p + palmACP --> 1hdecg3p + ACP	**0.0019**	**0.0013**	0.0000	**0.0013**	0.0000	**0.0014**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0009**	0.0000	**0.0009**
G3PAT161	[c] : glyc3p + hdeACP --> 1hdec9eg3p + ACP	**0.0022**	**0.0015**	0.0000	**0.0016**	0.0000	**0.0017**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0011**	0.0000	**0.0011**
G3PD2	[c] : glyc3p + nadp <==> dhap + h + nadph	**-0.0041**	**-0.0029**	0.0000	**-0.0029**	0.0000	**-0.0031**	0.0000	**-0.0010**	0.0000	**-0.0010**	0.0000	**-0.0010**	0.0000	**-0.0020**	0.0000	**-0.0020**

G5SADs	[c] : glu5sa --> 1pyr5c + h + h2o	**0.0065**	**0.0045**	0.0000	**0.0047**	0.0000	**0.0049**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0032**	0.0000	**0.0032**
G5SD	[c] : glu5p + h + nadph --> glu5sa + nadp + pi	**0.0065**	**0.0045**	0.0000	**0.0047**	0.0000	**0.0049**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0032**	0.0000	**0.0032**
G6PDH2r	[c] : g6p + nadp <==> 6pgl + h + nadph	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.6667**	**0.7670**	**0.6667**	**0.7670**	**0.6667**	**0.7669**	**0.7619**	**0.7273**	**0.7619**	**0.7276**
GAPD	[c] : g3p + nad + pi <==> 13dpg + h + nadh	**1.9300**	**1.9508**	**1.3333**	**1.3955**	**1.5000**	**1.5809**	**0.8889**	**0.8081**	**0.8889**	**0.8081**	**0.8889**	**0.8081**	**0.9286**	**0.8636**	**0.9286**	**0.8645**
GARFT	[c] : 10fthf + gar <==> fgam + h + thf	**0.0098**	**0.0069**	0.0000	**0.0071**	0.0000	**-0.0291**	0.0000	**-0.0094**	0.0000	**-0.0094**	0.0000	**-0.0098**	**-0.1667**	**-0.0721**	**-0.1667**	**-0.0730**
GART	[c] : atp + for + gar --> adp + fgam + h + pi	**0.0032**	**0.0023**	0.0000	**0.0023**	0.0000	**0.0389**	0.0000	**0.0126**	0.0000	**0.0126**	0.0000	**0.0129**	**0.1667**	**0.0786**	**0.1667**	**0.0794**
GCALDD	[c] : gcald + h2o + nad --> glyclt + (2) h + nadh	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GF6PTA	[c] : f6p + gln-L --> gam6p + glu-L	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
GHMT2r	[c] : ser-L + thf <==> gly + h2o + mlthf	**0.0310**	**0.0218**	0.0000	**0.0224**	0.0000	**-0.0132**	0.0000	**-0.0043**	0.0000	**-0.0043**	0.0000	**-0.0047**	**-0.1667**	**-0.0617**	**-0.1667**	**-0.0626**
GK1	[c] : atp + gmp <==> adp + gdp	**0.0071**	**0.0050**	0.0000	**0.0051**	0.0000	**0.0053**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0035**	0.0000	**0.0035**
GLCabcpp	atp[c] + glc-D[p] + h2o[c] --> adp[c] + glc-D[c] + h[c] + pi[c]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.2381**	**0.2727**	**0.2381**	**0.2724**
GLCptspp	glc-D[p] + pep[c] --> g6p[c] + pyr[c]	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.6667**	**0.7670**	**0.6667**	**0.7670**	**0.6667**	**0.7669**	**0.7619**	**0.7273**	**0.7619**	**0.7276**
GLCt2pp	glc-D[p] + h[p] --> glc-D[c] + h[c]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3333**	**0.2330**	**0.3333**	**0.2330**	**0.3333**	**0.2331**	0.0000	0.0000	0.0000	0.0000
GLCtex	glc-D[e] <==> glc-D[p]	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
GLNS	[c] : atp + glu-L + nh4 --> adp + gln-L + h + pi	**0.0529**	**0.0372**	0.0000	**0.0382**	0.0000	**0.0397**	0.0000	**0.0129**	0.0000	**0.0173**	0.0000	**0.0172**	0.0000	**0.0261**	0.0000	**0.0260**
GLU5K	[c] : atp + glu-L --> adp + glu5p	**0.0065**	**0.0045**	0.0000	**0.0047**	0.0000	**0.0049**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0032**	0.0000	**0.0032**
GLUDy	[c] : glu-L + h2o + nadp <==> akg + h + nadph + nh4	**-0.2482**	**-0.1745**	0.0000	**-0.1794**	0.0000	**-0.1864**	0.0000	**-0.0605**	0.0000	**-0.0605**	0.0000	**-0.0605**	**-0.1667**	**-0.1757**	**-0.1667**	**-0.1760**
GLUPRT	[c] : gln-L + h2o + prpp --> glu-L + ppi + pram	**0.0131**	**0.0092**	0.0000	**0.0095**	0.0000	**0.0098**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0064**
GLUR	[c] : glu-D <==> glu-L	**-0.0008**	**-0.0006**	0.0000	**-0.0006**	0.0000	**-0.0006**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0004**	0.0000	**-0.0004**
GLUTRR	[c] : glutrna + h + nadph --> glu1sa + nadp + trnaglu	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
GLUTRS	[c] : atp + glu-L + trnaglu --> amp + glutrna + ppi	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
GLYCLTt2rpp	glyclt[p] + h[p] <==> glyclt[c] + h[c]	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GLYCLTtex	glyclt[e] <==> glyclt[p]	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GMPS2	[c] : atp + gln-L + h2o + xmp --> amp + glu-L + gmp + (2) h + ppi	**0.0071**	**0.0050**	0.0000	**0.0051**	0.0000	**0.0053**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0035**	0.0000	**0.0035**
GRTT	[c] : grdp + ipdp --> frdp + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
GRXR	[c] : grxox + (2) gthrd --> grxrd + gthox	**0.0031**	**0.0022**	0.0000	**0.0006**	**0.7500**	**0.5504**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

GTHOr	[c] : gthox + h + nadph <==> (2) gthrd + nadp	**0.0031**	**0.0022**	0.0000	**0.0006**	**0.7500**	**0.5504**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GTPCI	[c] : gtp + h2o --> ahdt + for + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GTPCII2	[c] : gtp + (3) h2o --> 25drapp + for + (2) h + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
H2Otex	h2o[e] <==> h2o[p]	0.0000	**0.2967**	**0.6667**	0.0000	**0.7500**	**0.0473**	**0.4444**	**0.2330**	**0.4444**	**0.2330**	**0.4444**	**0.2347**	**0.7619**	**0.2800**	**0.7619**	**0.2823**
H2Otpp	h2o[p] <==> h2o[c]	0.0000	**0.2967**	**0.6667**	0.0000	**0.7500**	**0.0473**	**0.4444**	**0.2330**	**0.4444**	**0.2330**	**0.4444**	**0.2347**	**0.7619**	**0.2800**	**0.7619**	**0.2823**
H2St1pp	h2s[c] --> h2s[p]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.1667**	**0.1568**	**0.1667**	**0.1568**	**0.1667**	**0.1566**	0.0000	0.0000	0.0000	0.0000
H2Stex	h2s[e] <==> h2s[p]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**-0.1667**	**-0.1568**	**-0.1667**	**-0.1568**	**-0.1667**	**-0.1566**	0.0000	0.0000	0.0000	0.0000
H2tex	h2[e] <==> h2[p]	**-1.6751**	**-1.7715**	**-1.3333**	**-1.2112**	**-1.5000**	**-1.3529**	**-1.5556**	**-1.5012**	**-1.5556**	**-1.5012**	**-1.5556**	**-1.5012**	**-1.5238**	**-1.3879**	**-1.5238**	**-1.3888**
H2tpp	h2[p] <==> h2[c]	**-1.6751**	**-1.7715**	**-1.3333**	**-1.2112**	**-1.5000**	**-1.3529**	**-1.5556**	**-1.5012**	**-1.5556**	**-1.5012**	**-1.5556**	**-1.5012**	**-1.5238**	**-1.3879**	**-1.5238**	**-1.3888**
HBZOPT	[c] : 4hbz + octdp --> 3ophb + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
HCO3E	[c] : co2 + h2o <==> h + hco3	**0.0837**	**0.0588**	0.0000	**0.0605**	0.0000	**0.0628**	0.0000	**0.0204**	0.0000	**0.0248**	0.0000	**0.0247**	0.0000	**0.0413**	0.0000	**0.0412**
HEX7	[c] : atp + fru --> adp + f6p + h	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3333**	**0.2330**	**0.3333**	**0.2330**	**0.3333**	**0.2331**	**0.2381**	**0.2727**	**0.2381**	**0.2724**
HISTD	[c] : h2o + histd + (2) nad --> (3) h + his-L + (2) nadh	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
HISTP	[c] : h2o + hisp --> histd + pi	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
HMBS	[c] : h2o + (4) ppbng --> hmbil + (4) nh4	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
HPPK2	[c] : 6hmhpt + atp --> 6hmhptpp + amp + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HSDy	[c] : hom-L + nadp <==> aspsa + h + nadph	**-0.0204**	**-0.0144**	0.0000	**-0.0148**	0.0000	**-0.0518**	0.0000	**-0.0168**	0.0000	**-0.0168**	0.0000	**-0.0171**	**-0.1667**	**-0.0871**	**-0.1667**	**-0.0879**
HSK	[c] : atp + hom-L --> adp + h + phom	**0.0159**	**0.0112**	0.0000	**0.0115**	0.0000	**0.0484**	0.0000	**0.0157**	0.0000	**0.0157**	0.0000	**0.0160**	**0.1667**	**0.0849**	**0.1667**	**0.0857**
HSST	[c] : hom-L + succoa --> coa + suchms	**0.0045**	**0.0032**	0.0000	**0.0033**	0.0000	**0.0034**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
HSTPT	[c] : glu-L + imacp --> akg + hisp	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
Htex	h[e] <==> h[p]	**-1.0732**	**-1.0514**	**-1.3333**	**-1.3298**	**-1.5000**	**-1.5127**	**-1.2222**	**-1.2513**	**-1.2222**	**-1.2513**	**-1.2222**	**-1.2521**	**-1.6905**	**-1.5460**	**-1.6905**	**-1.5473**
HXAND	[c] : h2o + hxan + nad --> h + nadh + xan	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0035**
ICDHyr	[c] : icit + nadp <==> akg + co2 + nadph	**0.0315**	**0.0221**	0.0000	**0.0227**	0.0000	**0.0236**	0.0000	**0.0077**	0.0000	**0.0077**	0.0000	**0.0076**	0.0000	**0.0155**	0.0000	**0.0155**
IG3PS	[c] : gln-L + prlp --> aicar + eig3p + glu-L + h	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
IGPDH	[c] : eig3p --> h2o + imacp	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
IGPS	[c] : 2cpr5p + h --> 3ig3p + co2 + h2o	**0.0017**	**0.0012**	0.0000	**0.0012**	0.0000	**0.0012**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0008**	0.0000	**0.0008**
ILETA	[c] : akg + ile-L <==> 3mop + glu-L	**-0.0085**	**-0.0060**	0.0000	**-0.0061**	0.0000	**-0.0064**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0021**	0.0000	**-0.0042**	0.0000	**-0.0042**
IMPC	[c] : h2o + imp <==> fprica	**-0.0158**	**-0.0111**	0.0000	**-0.0115**	0.0000	**-0.0119**	0.0000	**-0.0039**	0.0000	**-0.0039**	0.0000	**-0.0038**	0.0000	**-0.0078**	0.0000	**-0.0078**
IMPD	[c] : h2o + imp + nad --> h + nadh + xmp	**0.0071**	**0.0050**	0.0000	**0.0051**	0.0000	**0.0053**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0017**	0.0000	**0.0035**	0.0000	0.0000

IPDDI	[c] : ipdp <==> dmpp	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
IPDPS	[c] : h + h2mb4p + nadh --> h2o + ipdp + nad	**0.0001**	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
IPMD	[c] : 3c2hmp + nad --> 3c4mop + h + nadh	**0.0132**	**0.0093**	0.0000	**0.0095**	0.0000	**0.0099**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0065**
IPPMIa	[c] : 3c2hmp <==> 2ippm + h2o	**-0.0132**	**-0.0093**	0.0000	**-0.0095**	0.0000	**-0.0099**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0065**	0.0000	**-0.0065**
IPPMIb	[c] : 2ippm + h2o <==> 3c3hmp	**-0.0132**	**-0.0093**	0.0000	**-0.0095**	0.0000	**-0.0099**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0032**	0.0000	**-0.0065**	0.0000	**-0.0065**
IPPS	[c] : 3mob + accoa + h2o --> 3c3hmp + coa + h	**0.0132**	**0.0093**	0.0000	**0.0095**	0.0000	**0.0099**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0065**
K2L4Aabcpp	atp[c] + h2o[c] + kdo2lipid4[c] --> adp[c] + h[c] + kdo2lipid4[p] + pi[c]	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
K2L4Atex	kdo2lipid4[p] --> kdo2lipid4[e]	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
Kabcpp	atp[c] + h2o[c] + k[p] --> adp[c] + h[c] + k[c] + pi[c]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0026**
KARA1	[c] : 23dhmb + nadp <==> alac-S + h + nadph	**-0.0256**	**-0.0180**	0.0000	**-0.0185**	0.0000	**-0.0192**	0.0000	**-0.0062**	0.0000	**-0.0062**	0.0000	**-0.0062**	0.0000	**-0.0126**	0.0000	**-0.0126**
KARA2	[c] : 2ahbut + h + nadph <==> 23dhmp + nadp	**0.0085**	**0.0060**	0.0000	**0.0061**	0.0000	**0.0064**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0042**	0.0000	**0.0042**
KAS14	[c] : acACP + h + malACP --> ACP + actACP + co2	**0.0104**	**0.0073**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0025**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
KAS15	[c] : accoa + h + malACP --> actACP + co2 + coa	0.0000	0.0000	0.0000	**0.0075**	0.0000	**0.0078**	0.0000	0.0000	0.0000	**0.0025**	0.0000	**0.0025**	0.0000	**0.0051**	0.0000	**0.0051**
KDOCT2	[c] : ctp + kdo --> ckdo + ppi	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
KDOPP	[c] : h2o + kdo8p --> kdo + pi	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
KDOPS	[c] : ara5p + h2o + pep --> kdo8p + pi	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
Kt2pp	h[p] + k[p] --> h[c] + k[c]	**0.0052**	**0.0037**	0.0000	**0.0038**	0.0000	**0.0039**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0026**	0.0000	0.0000
Ktex	k[e] <==> k[p]	**0.0052**	**0.0037**	0.0000	**0.0038**	0.0000	**0.0039**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0013**	0.0000	**0.0026**	0.0000	**0.0026**
LALDO2x	[c] : h + mthgxl + nadh --> lald-D + nad	0.0000	0.0000	**0.6667**	**0.5539**	0.0000	0.0000	**0.4444**	**0.4078**	**0.4444**	**0.4078**	**0.4444**	**0.4080**	**0.0714**	**0.1143**	**0.0714**	**0.1136**
LCARR	[c] : h + lald-D + nadh <==> 12ppd-R + nad	0.0000	0.0000	**0.6667**	**0.5539**	0.0000	0.0000	**0.4444**	**0.4078**	**0.4444**	**0.4078**	**0.4444**	**0.4080**	**0.0714**	**0.1143**	**0.0714**	**0.1136**
LEUTAi	[c] : 4mop + glu-L --> akg + leu-L	**0.0132**	**0.0093**	0.0000	**0.0095**	0.0000	**0.0099**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0065**
LPADSS	[c] : lipidX + u23ga --> h + lipidAds + udp	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
MCOATA	[c] : ACP + malcoa <==> coa + malACP	**0.0706**	**0.0496**	0.0000	**0.0510**	0.0000	**0.0530**	0.0000	**0.0172**	0.0000	**0.0172**	0.0000	**0.0171**	0.0000	**0.0349**	0.0000	**0.0347**
MCTP1App	[p] : murein5p5p --> ala-D + murein5px4p	**0.0004**	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0002**	0.0000	**0.0002**
MDH	[c] : mal-L + nad <==> h + nadh + oaa	**0.0207**	**0.0145**	0.0000	**0.0150**	0.0000	**0.0155**	0.0000	**0.0050**	0.0000	**0.0050**	0.0000	**0.0050**	0.0000	**0.0102**	0.0000	**0.0102**

MECDPDH2	[c] : 2mecdp + nadh --> h2mb4p + h2o + nad	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MECDPS	[c] : 2p4c2me --> 2mecdp + cmp	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MEPCT	[c] : 2me4p + ctp + h --> 4c2me + ppi	**0.0001**	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
METAT	[c] : atp + h2o + met-L --> amet + pi + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
METS	[c] : 5mthf + hcys-L --> h + met-L + thf	**0.0045**	**0.0032**	0.0000	**0.0033**	0.0000	**0.0034**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
MG2t3_2pp	(2) h[c] + mg2[p] <==> (2) h[p] + mg2[c]	**0.0002**	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
MG2tex	mg2[e] <==> mg2[p]	**0.0002**	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**
MGSA	[c] : dhap --> mthgxl + pi	0.0000	0.0000	**0.6667**	**0.5539**	0.0000	0.0000	**0.4444**	**0.4078**	**0.4444**	**0.4078**	**0.4444**	**0.4080**	**0.0714**	**0.1143**	**0.0714**	**0.1136**
MN2tpp	mn2[p] --> mn2[c]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MNtex	mn2[e] <==> mn2[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MOAT	[c] : ckdo + lipidA --> cmp + h + kdolipid4	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
MOAT2	[c] : ckdo + kdolipid4 --> cmp + h + kdo2lipid4	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
MOBDabcpp	atp[c] + h2o[c] + mobd[p] --> adp[c] + h[c] + mobd[c] + pi[c]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MOBDtex	mobd[e] <==> mobd[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
MOHMT	[c] : 3mob + h2o + mlthf --> 2dhp + thf	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MPTG	(2) uaagmda[c] --> (2) h[c] + murein5p5p[p] + (2) udcpdp[c]	**0.0004**	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0002**	0.0000	**0.0002**
MTHFC	[c] : h2o + methf <==> 10fthf + h	**0.0257**	**0.0181**	0.0000	**0.0186**	0.0000	**-0.0172**	0.0000	**-0.0056**	0.0000	**-0.0056**	0.0000	**-0.0059**	**-0.1667**	**-0.0643**	**-0.1667**	**-0.0652**
MTHFD	[c] : mlthf + nadp <==> methf + nadph	**0.0257**	**0.0181**	0.0000	**0.0186**	0.0000	**-0.0172**	0.0000	**-0.0056**	0.0000	**-0.0056**	0.0000	**-0.0059**	**-0.1667**	**-0.0643**	**-0.1667**	**-0.0652**
MTHFR2	[c] : (2) h + mlthf + nadh --> 5mthf + nad	**0.0045**	**0.0032**	0.0000	**0.0033**	0.0000	**0.0034**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
NADK	[c] : atp + nad --> adp + h + nadp	**0.0000**	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NADS1	[c] : atp + dnad + nh4 --> amp + h + nad + ppi	**0.0001**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NDPK2	[c] : atp + udp <==> adp + utp	**0.0117**	**0.0082**	0.0000	**0.0090**	0.0000	**0.0093**	0.0000	**0.0030**	0.0000	**0.0030**	0.0000	**0.0030**	0.0000	**0.0058**	0.0000	**0.0057**
NDPK3	[c] : atp + cdp <==> adp + ctp	**0.0045**	**0.0032**	0.0000	**0.0038**	0.0000	**0.0040**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
NDPK4	[c] : atp + dtdp <==> adp + dttp	**0.0008**	**0.0005**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
NDPK5	[c] : atp + dgdp <==> adp + dgtp	**0.0008**	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0004**	0.0000	**0.0004**
NDPK6	[c] : atp + dudp <==> adp + dutp	0.0000	0.0000	0.0000	**-0.0006**	0.0000	**-0.0006**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	0.0000	0.0000	0.0000
NDPK7	[c] : atp + dcdp <==> adp + dctp	**0.0008**	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
NDPK8	[c] : atp + dadp <==> adp + datp	**0.0008**	**0.0005**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0004**	0.0000	**0.0004**

NH4tex	nh4[e] <==> nh4[p]	**0.3154**	**0.2218**	0.0000	**0.2281**	0.0000	**0.2370**	0.0000	**0.0768**	0.0000	**0.0768**	0.0000	**0.0763**	0.0000	**0.1559**	0.0000	**0.1552**
NH4tpp	nh4[p] <==> nh4[c]	**0.3154**	**0.2218**	0.0000	**0.2281**	0.0000	**0.2370**	0.0000	**0.0768**	0.0000	**0.0768**	0.0000	**0.0763**	0.0000	**0.1559**	0.0000	**0.1552**
NNATr	[c] : atp + h + nicrnt <==> dnad + ppi	**0.0001**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NNDPR	[c] : (2) h + prpp + quln --> co2 + nicrnt + ppi	**0.0001**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NTD1	[c] : dump + h2o --> duri + pi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3571**	**0.3903**	**0.3571**	**0.3900**
NTD11	[c] : h2o + imp --> ins + pi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0035**
NTD6	[c] : damp + h2o --> dad-2 + pi	0.0000	0.0000	0.0000	0.0000	**0.7500**	**0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
OCBT	[c] : cbp + orn <==> citr-L + h + pi	**0.0087**	**0.0061**	0.0000	**0.0063**	0.0000	**0.0065**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0043**	0.0000	**0.0043**
OCTDPS	[c] : frdp + (5) ipdp --> octdp + (5) ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
OHPBAT	[c] : glu-L + ohpb <==> akg + phthr	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
OMCDC	[c] : 3c4mop + h --> 4mop + co2	**0.0132**	**0.0093**	0.0000	**0.0095**	0.0000	**0.0099**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0032**	0.0000	**0.0065**	0.0000	**0.0065**
OMPDC	[c] : h + orot5p --> co2 + ump	**0.0097**	**0.0068**	0.0000	**0.0070**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0023**	0.0000	**0.0048**	0.0000	**0.0048**
OPHBDC	[c] : 3ophb + h --> 2oph + co2	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
OPHHX3	[c] : 2oph + (2) atp + (3) h2o + nad --> 2ohph + (2) adp + (3) h + nadh + (2) pi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
ORPT	[c] : orot5p + ppi <==> orot + prpp	**-0.0097**	**-0.0068**	0.0000	**-0.0070**	0.0000	**-0.0073**	0.0000	**-0.0024**	0.0000	**-0.0024**	0.0000	**-0.0023**	0.0000	**-0.0048**	0.0000	**-0.0048**
P5CR	[c] : 1pyr5c + (2) h + nadph --> nadp + pro-L	**0.0065**	**0.0045**	0.0000	**0.0047**	0.0000	**0.0049**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0016**	0.0000	**0.0032**	0.0000	**0.0032**
PANTS	[c] : ala-B + atp + pant-R --> amp + h + pnto-R + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PAPPT3	[c] : udcpp + ugmda --> uagmda + ump	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
PAPSR	[c] : paps + trdrd --> (2) h + pap + so3 + trdox	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1585**	**0.1667**	**0.1585**	**0.1667**	**0.1583**	0.0000	**0.0036**	0.0000	**0.0035**
PDX5PO2	[c] : nad + pdx5p --> h + nadh + pydx5p	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
PDX5PS	[c] : dxyl5p + nad + phthr --> co2 + h + (2) h2o + nadh + pdx5p + pi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
PE160abcpp	atp[c] + h2o[c] + pe160[c] --> adp[c] + h[c] + pe160[p] + pi[c]	**0.0012**	**0.0009**	0.0000	**0.0009**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
PE161abcpp	atp[c] + h2o[c] + pe161[c] --> adp[c] + h[c] + pe161[p] + pi[c]	**0.0014**	**0.0010**	0.0000	**0.0010**	0.0000	**0.0011**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0007**	0.0000	**0.0007**
PERD	[c] : 4per + nad <==> h + nadh + ohpb	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
PFK	[c] : atp + f6p --> adp + fdp + h	**0.9747**	**0.9822**	**0.6667**	**0.7081**	0.0000	**0.1602**	**0.1111**	**0.2268**	**0.3333**	**0.2268**	**0.1111**	**0.2269**	0.0000	0.0000	0.0000	0.0000
PFL	[c] : coa + pyr --> accoa + for	**1.6783**	**1.7737**	**1.3333**	**1.2135**	**1.5000**	**1.3918**	**1.5556**	**1.5138**	**1.5556**	**1.5138**	**1.5556**	**1.5142**	**1.6905**	**1.4665**	**1.6905**	**1.4681**
PGAMT	[c] : gam1p <==> gam6p	**-0.0028**	**-0.0019**	0.0000	**-0.0020**	0.0000	**-0.0021**	0.0000	**-0.0007**	0.0000	**-0.0007**	0.0000	**-0.0007**	0.0000	**-0.0014**	0.0000	**-0.0014**
PGCD	[c] : 3pg + nad --> 3php + h + nadh	**0.0486**	**0.0342**	0.0000	**0.0351**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

PGI	[c] : g6p <==> f6p	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PGK	[c] : 3pg + atp <==> 13dpg + adp	-1.9300	-1.9508	-1.3333	-1.3955	-1.5000	-1.5809	-0.8889	-0.8081	-0.8889	-0.8081	-0.8889	-0.8081	-0.9286	-0.8636	-0.9286	-0.8645
PGL	[c] : 6pgl + h2o --> 6pgc + h	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.6667	0.7670	0.6667	0.7670	0.6667	0.7669	0.7619	0.7273	0.7619	0.7276
PGM	[c] : 2pg <==> 3pg	-1.8814	-1.9166	-1.3333	-1.3604	-1.5000	-1.5809	-0.8889	-0.8081	-0.8889	-0.8081	-0.8889	-0.8081	-0.9286	-0.8636	-0.9286	-0.8645
PHETA1	[c] : akg + phe-L <==> glu-L + phpyr	-0.0051	-0.0036	0.0000	-0.0037	0.0000	-0.0039	0.0000	-0.0013	0.0000	-0.0013	0.0000	-0.0012	0.0000	-0.0025	0.0000	-0.0025
PIt2rpp	h[p] + pi[p] <==> h[c] + pi[c]	0.0281	0.0198	0.0000	0.0203	0.0000	0.0211	0.0000	0.0068	0.0000	0.0068	0.0000	0.0068	0.0000	0.0000	0.0000	0.0000
PItex	pi[e] <==> pi[p]	0.0281	0.0198	0.0000	0.0203	0.0000	0.0211	0.0000	0.0068	0.0000	0.0068	0.0000	0.0068	0.0000	0.0139	0.0000	0.0138
PIuabcpp	atp[c] + h2o[c] + pi[p] --> adp[c] + h[c] + (2) pi[c]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0139	0.0000	0.0138
PMDPHT	[c] : 5aprbu + h2o --> 4r5au + pi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PMPK	[c] : 4ampm + atp --> 2mahmp + adp	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PNTK	[c] : atp + pnto-R --> 4ppan + adp + h	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PPBNGS	[c] : (2) 5aop --> h + (2) h2o + ppbng	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PPC	[c] : co2 + h2o + pep --> h + oaa + pi	0.0964	0.0678	0.0000	0.0697	0.0000	0.1089	0.0000	0.0353	0.0000	0.0353	0.0000	0.0355	0.1667	0.1246	0.1667	0.1252
PPCDC	[c] : 4ppcys + h --> co2 + pan4p	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PPKr	[c] : atp + pi <==> adp + ppi	-0.0912	-0.0641	0.0000	-0.0659	0.0000	-0.0685	-0.1667	-0.1790	-0.1667	-0.1790	-0.1667	-0.1787	0.0000	-0.0451	0.0000	-0.0484
PPM	[c] : r1p <==> r5p	0.0000	0.0000	0.0000	-0.0197	-0.7500	-0.5703	0.0000	0.0000	0.0000	-0.0066	0.0000	-0.0066	-0.3571	-0.4038	-0.3571	-0.4034
PPM2	[c] : 2dr1p <==> 2dr5p	0.0000	0.0000	0.0000	0.0000	0.7500	0.5498	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3571	0.3903	0.3571	0.3900
PPNCL2	[c] : 4ppan + ctp + cys-L --> 4ppcys + cmp + h + ppi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PPND	[c] : nad + pphn --> 34hpp + co2 + nadh	0.0040	0.0028	0.0000	0.0029	0.0000	0.0030	0.0000	0.0010	0.0000	0.0010	0.0000	0.0010	0.0000	0.0020	0.0000	0.0020
PPNDH	[c] : h + pphn --> co2 + h2o + phpyr	0.0051	0.0036	0.0000	0.0037	0.0000	0.0039	0.0000	0.0013	0.0000	0.0013	0.0000	0.0012	0.0000	0.0025	0.0000	0.0025
PPPGO3	[c] : (3) fum + pppg9 --> ppp9 + (3) succ	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PRAGSr	[c] : atp + gly + pram <==> adp + gar + h + pi	0.0131	0.0092	0.0000	0.0095	0.0000	0.0098	0.0000	0.0032	0.0000	0.0032	0.0000	0.0032	0.0000	0.0065	0.0000	0.0064
PRAIi	[c] : pran --> 2cpr5p	0.0017	0.0012	0.0000	0.0012	0.0000	0.0012	0.0000	0.0004	0.0000	0.0004	0.0000	0.0004	0.0000	0.0008	0.0000	0.0008
PRAIS	[c] : atp + fpram --> adp + air + (2) h + pi	0.0131	0.0092	0.0000	0.0095	0.0000	0.0098	0.0000	0.0032	0.0000	0.0032	0.0000	0.0032	0.0000	0.0065	0.0000	0.0064
PRAMPC	[c] : h2o + prbamp --> prfp	0.0028	0.0019	0.0000	0.0020	0.0000	0.0021	0.0000	0.0007	0.0000	0.0007	0.0000	0.0007	0.0000	0.0014	0.0000	0.0014
PRASCS	[c] : 5aizc + asp-L + atp --> 25aics + adp + h + pi	0.0131	0.0092	0.0000	0.0094	0.0000	0.0098	0.0000	0.0032	0.0000	0.0032	0.0000	0.0032	0.0000	0.0065	0.0000	0.0064
PRATPP	[c] : h2o + prbatp --> h + ppi + prbamp	0.0028	0.0019	0.0000	0.0020	0.0000	0.0021	0.0000	0.0007	0.0000	0.0007	0.0000	0.0007	0.0000	0.0014	0.0000	0.0014
PRFGS	[c] : atp + fgam + gln-L + h2o --> adp + fpram + glu-L + h + pi	0.0131	0.0092	0.0000	0.0095	0.0000	0.0098	0.0000	0.0032	0.0000	0.0032	0.0000	0.0032	0.0000	0.0065	0.0000	0.0064
PRMICI	[c] : prfp <==> prlp	0.0028	0.0019	0.0000	0.0020	0.0000	0.0021	0.0000	0.0007	0.0000	0.0007	0.0000	0.0007	0.0000	0.0014	0.0000	0.0014
PROt2rpp	h[p] + pro-L[p] <==> h[c] + pro-L[c]	-0.0001	-0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

PROt4pp	na1[p] + pro-L[p] --> na1[c] + pro-L[c]	**0.0001**	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PRPPS	[c] : atp + r5p <==> amp + h + prpp	**0.0272**	**0.0192**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0066**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PSCVT	[c] : pep + skm5p <==> 3psme + pi	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
PSD160	[c] : h + ps160 --> co2 + pe160	**0.0019**	**0.0013**	0.0000	**0.0013**	0.0000	**0.0014**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0009**	0.0000	**0.0009**
PSD161	[c] : h + ps161 --> co2 + pe161	**0.0022**	**0.0015**	0.0000	**0.0016**	0.0000	**0.0017**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0011**	0.0000	**0.0011**
PSERT	[c] : 3php + glu-L --> akg + pser-L	**0.0486**	**0.0342**	0.0000	**0.0351**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PSP_L	[c] : h2o + pser-L --> pi + ser-L	**0.0486**	**0.0342**	0.0000	**0.0351**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PSSA160	[c] : cdpdhdecg + ser-L --> cmp + h + ps160	**0.0019**	**0.0013**	0.0000	**0.0013**	0.0000	**0.0014**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0009**	0.0000	**0.0009**
PSSA161	[c] : cdpdhdec9eg + ser-L --> cmp + h + ps161	**0.0022**	**0.0015**	0.0000	**0.0016**	0.0000	**0.0017**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0005**	0.0000	**0.0011**	0.0000	**0.0011**
PTAr	[c] : accoa + pi <==> actp + coa	**0.7680**	**0.8368**	**1.3333**	**1.1092**	**1.5000**	**1.2834**	**1.5556**	**1.4905**	**1.5556**	**1.4905**	**1.5556**	**1.4914**	**1.6905**	**1.3952**	**1.6905**	**1.3972**
PTPATi	[c] : atp + h + pan4p --> dpcoa + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PUNP1	[c] : adn + pi <==> ade + r1p	**0.0000**	**0.0000**	0.0000	**0.0000**	**-0.7500**	**-0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PUNP2	[c] : dad-2 + pi <==> 2dr1p + ade	0.0000	0.0000	0.0000	0.0000	**0.7500**	**0.5498**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PUNP5	[c] : ins + pi <==> hxan + r1p	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0035**
PYK	[c] : adp + h + pep --> atp + pyr	**0.7614**	**0.8322**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.2222**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PYNP2r	[c] : pi + uri <==> r1p + ura	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**-0.3571**	**-0.3903**	**-0.3571**	**-0.3900**
QULNS	[c] : dhap + iasp --> (2) h2o + pi + quln	**0.0001**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
R15BPK	[c] : atp + r15bp --> adp + prpp	0.0000	0.0000	0.0000	**0.0197**	0.0000	**0.0205**	0.0000	0.0000	0.0000	**0.0066**	0.0000	**0.0066**	0.0000	**0.0135**	0.0000	**0.0169**
R1PK	[c] : atp + r1p --> adp + h + r15bp	0.0000	0.0000	0.0000	**0.0197**	0.0000	**0.0205**	0.0000	0.0000	0.0000	**0.0066**	0.0000	**0.0066**	0.0000	**0.0135**	0.0000	**0.0169**
RBFK	[c] : atp + ribflv --> adp + fmn + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
RBFSa	[c] : 4r5au + db4p --> dmlz + (2) h2o + pi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
RBFSb	[c] : (2) dmlz --> 4r5au + ribflv	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
RHCCE	[c] : rhcys --> dhptd + hcys-L	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
RNDR1	[c] : adp + trdrd --> dadp + h2o + trdox	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0004**	0.0000	**0.0004**
RNDR1b	[c] : adp + grxrd --> dadp + grxox + h2o	**0.0008**	**0.0005**	0.0000	**0.0006**	**0.7500**	**0.5504**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNDR2	[c] : gdp + trdrd --> dgdp + h2o + trdox	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0004**	0.0000	**0.0004**
RNDR2b	[c] : gdp + grxrd --> dgdp + grxox + h2o	**0.0008**	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNDR3	[c] : cdp + trdrd --> dcdp + h2o + trdox	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
RNDR3b	[c] : cdp + grxrd --> dcdp + grxox + h2o	**0.0008**	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNDR4	[c] : trdrd + udp --> dudp + h2o +	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3571**	**0.3907**	**0.3571**	**0.3903**

	trdox																
RNDR4b	[c] : grxrd + udp --> dudp + grxox + h2o	**0.0008**	**0.0005**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNTR1c	[c] : atp + fldrd --> datp + fldox + h2o	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	0.0000	0.0000	0.0000
RNTR2c	[c] : fldrd + gtp --> dgtp + fldox + h2o	0.0000	0.0000	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	0.0000	0.0000	0.0000
RNTR3c	[c] : ctp + fldrd --> dctp + fldox + h2o	0.0000	0.0000	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNTR4c	[c] : fldrd + utp --> dutp + fldox + h2o	0.0000	0.0000	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	0.0000	0.0000	0.0000
RPE	[c] : ru5p-D <==> xu5p-D	**-0.0226**	**-0.0159**	0.0000	**-0.0163**	**-0.5000**	**-0.3835**	0.0000	**-0.0055**	0.0000	**-0.0055**	0.0000	**-0.0055**	**-0.2381**	**-0.2714**	**-0.2381**	**-0.2711**
RPI	[c] : r5p <==> ru5p-D	**-0.0214**	**-0.0151**	0.0000	**-0.0155**	**-0.5000**	**-0.3826**	0.0000	**-0.0052**	0.0000	**-0.0052**	0.0000	**-0.0052**	**-0.2381**	**-0.2708**	**-0.2381**	**-0.2705**
SADT2	[c] : atp + gtp + h2o + so4 --> aps + gdp + pi + ppi	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1585**	**0.1667**	**0.1585**	**0.1667**	**0.1583**	0.0000	**0.0036**	0.0000	**0.0035**
SDPDS	[c] : h2o + sl26da --> 26dap-LL + succ	**0.0109**	**0.0076**	0.0000	**0.0078**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0053**
SDPTA	[c] : akg + sl26da <==> glu-L + sl2a6o	**-0.0109**	**-0.0076**	0.0000	**-0.0078**	0.0000	**-0.0082**	0.0000	**-0.0026**	0.0000	**-0.0026**	0.0000	**-0.0026**	0.0000	**-0.0054**	0.0000	**-0.0053**
SERAT	[c] : accoa + ser-L <==> acser + coa	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	0.0000	**0.0018**	0.0000	**0.0018**	0.0000	**0.0017**	0.0000	**0.0036**	0.0000	**0.0035**
SERD_L	[c] : ser-L --> nh4 + pyr	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.1667**	**0.0530**	**0.1667**	**0.0539**
SHCHD2	[c] : dscl + nad --> h + nadh + scl	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
SHCHF	[c] : fe2 + scl --> (3) h + sheme	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
SHK3Dr	[c] : 3dhsk + h + nadph <==> nadp + skm	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
SHKK	[c] : atp + skm --> adp + h + skm5p	**0.0109**	**0.0076**	0.0000	**0.0079**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0054**
SHSL1	[c] : cys-L + suchms --> cyst-L + h + succ	**0.0045**	**0.0032**	0.0000	**0.0033**	0.0000	**0.0034**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
SO4tex	so4[e] <==> so4[p]	**0.0073**	**0.0051**	0.0000	**0.0053**	0.0000	**0.0055**	**0.1667**	**0.1586**	**0.1667**	**0.1586**	**0.1667**	**0.1584**	0.0000	**0.0036**	0.0000	**0.0036**
SUCCt3pp	h[p] + succ[c] --> h[c] + succ[p]	**0.0098**	**0.0069**	0.0000	**0.0071**	0.0000	**0.0073**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0024**	0.0000	**0.0048**	0.0000	**0.0048**
SUCCtex	succ[e] <==> succ[p]	**-0.0098**	**-0.0069**	0.0000	**-0.0071**	0.0000	**-0.0073**	0.0000	**-0.0024**	0.0000	**-0.0024**	0.0000	**-0.0024**	0.0000	**-0.0048**	0.0000	**-0.0048**
SUCOAS	[c] : atp + coa + succ <==> adp + pi + succoa	**0.0154**	**0.0108**	0.0000	**0.0111**	0.0000	**0.0115**	0.0000	**0.0037**	0.0000	**0.0037**	0.0000	**0.0037**	0.0000	**0.0076**	0.0000	**0.0076**
SULabcpp	atp[c] + h2o[c] + so4[p] --> adp[c] + h[c] + pi[c] + so4[c]	**0.0073**	**0.0051**	0.0000	**0.0053**	0.0000	**0.0055**	**0.1667**	**0.1586**	**0.1667**	**0.1586**	**0.1667**	**0.1584**	0.0000	**0.0036**	0.0000	**0.0036**
SULRi	[c] : (5) h + (3) nadph + so3 --> (3) h2o + h2s + (3) nadp	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1585**	**0.1667**	**0.1585**	**0.1667**	**0.1583**	0.0000	**0.0036**	0.0000	**0.0035**
T2DECAI	[c] : tdec2eACP <==> cdec3eACP	**0.0044**	**0.0031**	0.0000	**0.0032**	0.0000	**0.0033**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0011**	0.0000	**0.0022**	0.0000	**0.0022**
TALA	[c] : g3p + s7p <==> e4p + f6p	**-0.0059**	**-0.0041**	0.0000	**-0.0042**	**-0.2500**	**-0.1877**	0.0000	**-0.0014**	0.0000	**-0.0014**	0.0000	**-0.0014**	**-0.1190**	**-0.1330**	**-0.1190**	**-0.1329**
TDSK	[c] : atp + lipidAds --> adp + h + lipidA	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**

THD2pp	(2) h[p] + nadh[c] + nadp[c] --> (2) h[c] + nad[c] + nadph[c]	**0.4285**	**0.3014**	0.0000	**0.3098**	**0.7500**	**0.9811**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.2619**	**0.1588**	**0.2619**	**0.1607**
THDPS	[c] : h2o + succoa + thdp --> coa + sl2a6o	**0.0109**	**0.0076**	0.0000	**0.0078**	0.0000	**0.0082**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0026**	0.0000	**0.0054**	0.0000	**0.0053**
THRAi	[c] : thr-L --> acald + gly	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0365**	0.0000	**0.0118**	0.0000	**0.0118**	0.0000	**0.0122**	**0.1667**	**0.0770**	**0.1667**	**0.0778**
THRD_L	[c] : thr-L --> 2obut + nh4	**0.0085**	**0.0060**	0.0000	**0.0061**	0.0000	**0.0064**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0021**	0.0000	**0.0042**	0.0000	**0.0042**
THRS	[c] : h2o + phom --> pi + thr-L	**0.0159**	**0.0112**	0.0000	**0.0115**	0.0000	**0.0484**	0.0000	**0.0157**	0.0000	**0.0157**	0.0000	**0.0160**	**0.1667**	**0.0849**	**0.1667**	**0.0857**
THRt2rpp	h[p] + thr-L[p] <==> h[c] + thr-L[c]	0.0000	0.0000	0.0000	**-0.0001**	0.0000	**-0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**-0.0001**	0.0000	**-0.0001**
THRt4pp	na1[p] + thr-L[p] --> na1[c] + thr-L[c]	0.0000	0.0000	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0001**	0.0000	**0.0001**
THZPSN	[c] : atp + cys-L + dxyl5p + tyr-L --> 4hba + 4mpetz + ala-L + amp + co2 + h + h2o + ppi	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
TKT1	[c] : r5p + xu5p-D <==> g3p + s7p	**-0.0059**	**-0.0041**	0.0000	**-0.0042**	**-0.2500**	**-0.1877**	0.0000	**-0.0014**	0.0000	**-0.0014**	0.0000	**-0.0014**	**-0.1190**	**-0.1330**	**-0.1190**	**-0.1329**
TKT2	[c] : e4p + xu5p-D <==> f6p + g3p	**-0.0167**	**-0.0118**	0.0000	**-0.0121**	**-0.2500**	**-0.1958**	0.0000	**-0.0041**	0.0000	**-0.0041**	0.0000	**-0.0040**	**-0.1190**	**-0.1384**	**-0.1190**	**-0.1382**
TMDS	[c] : dump + mlthf --> dhf + dtmp	**0.0008**	**0.0005**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
TMPK	[c] : atp + thmmp --> adp + thmpp	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
TMPPP	[c] : 2mahmp + 4mpetz + h --> ppi + thmmp	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
TPI	[c] : dhap <==> g3p	**0.9705**	**0.9793**	**0.3333**	**0.4248**	**0.5000**	**0.6113**	**-0.1111**	**-0.1820**	**-0.1111**	**-0.1820**	**-0.1111**	**-0.1821**	**-0.0714**	**-0.1164**	**-0.0714**	**-0.1156**
TRDR	[c] : h + nadph + trdox --> nadp + trdrd	**0.0072**	**0.0051**	0.0000	**0.0052**	0.0000	**0.0054**	**0.1667**	**0.1587**	**0.1667**	**0.1587**	**0.1667**	**0.1585**	**0.3571**	**0.3954**	**0.3571**	**0.3950**
TRPAS2	[c] : h2o + trp-L <==> indole + nh4 + pyr	**-0.0017**	**-0.0012**	0.0000	**-0.0012**	0.0000	**-0.0012**	0.0000	**-0.0004**	0.0000	**-0.0004**	0.0000	0.0000	0.0000	**-0.0008**	0.0000	**-0.0008**
TRPS2	[c] : indole + ser-L --> h2o + trp-L	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0004**	0.0000	0.0000	0.0000	0.0000
TRPS3	[c] : 3ig3p --> g3p + indole	**0.0017**	**0.0012**	0.0000	**0.0012**	0.0000	**0.0012**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0008**	0.0000	**0.0008**
TYRTA	[c] : akg + tyr-L <==> 34hpp + glu-L	**-0.0040**	**-0.0028**	0.0000	**-0.0029**	0.0000	**-0.0030**	0.0000	**-0.0010**	0.0000	**-0.0010**	0.0000	**-0.0010**	0.0000	**-0.0020**	0.0000	**-0.0020**
U23GAAT	[c] : 3hmrsACP + u3hga --> ACP + h + u23ga	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
UAAGDS	[c] : 26dap-M + atp + uamag --> adp + h + pi + ugmd	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UAGAAT	[c] : 3hmrsACP + uacgam <==> ACP + u3aga	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
UAGCVT	[c] : pep + uacgam --> pi + uaccg	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UAGDP	[c] : acgam1p + h + utp --> ppi + uacgam	**0.0028**	**0.0019**	0.0000	**0.0020**	0.0000	**0.0021**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0007**	0.0000	**0.0014**	0.0000	**0.0014**
UAGPT3	[c] : uacgam + uagmda --> h + uaagmda + udp	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UAMAGS	[c] : atp + glu-D + uama --> adp + h + pi + uamag	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UAMAS	[c] : ala-L + atp + uamr --> adp + h +	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**

	pi + uama																
UAPGR	[c] : h + nadph + uaccg --> nadp + uamr	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UDCPDP	[c] : h2o + udcpdp --> h + pi + udcpp	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UDCPDPS	[c] : frdp + (8) ipdp --> (8) ppi + udcpdp	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
UGMDDS	[c] : alaala + atp + ugmd --> adp + h + pi + ugmda	**0.0008**	**0.0006**	0.0000	**0.0006**	0.0000	**0.0006**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0002**	0.0000	**0.0004**	0.0000	**0.0004**
UHGADA	[c] : h2o + u3aga --> ac + u3hga	**0.0011**	**0.0008**	0.0000	**0.0008**	0.0000	**0.0009**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0003**	0.0000	**0.0006**	0.0000	**0.0006**
UMPK	[c] : atp + ump <==> adp + udp	**0.0111**	**0.0078**	0.0000	**0.0080**	0.0000	**0.0083**	0.0000	**0.0027**	0.0000	**0.0027**	0.0000	**0.0027**	**0.3571**	**0.3958**	**0.3571**	**0.3954**
UPP3MT	[c] : (2) amet + uppg3 --> (2) ahcys + dscl + h	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
UPP3S	[c] : hmbil --> h2o + uppg3	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
UPPDC1	[c] : (4) h + uppg3 --> (4) co2 + cpppg3	**0.0000**	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
URIDK2r	[c] : atp + dump <==> adp + dudp	**-0.0008**	**-0.0005**	0.0000	**-0.0006**	0.0000	**-0.0006**	0.0000	**-0.0002**	0.0000	**-0.0002**	0.0000	**-0.0002**	**-0.3571**	**-0.3907**	**-0.3571**	**-0.3903**
URIK2	[c] : gtp + uri --> gdp + h + ump	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3571**	**0.3903**	**0.3571**	**0.3900**
USHD	[c] : h2o + u23ga --> (2) h + lipidX + ump	**0.0006**	**0.0004**	0.0000	**0.0004**	0.0000	**0.0004**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0003**	0.0000	**0.0003**
VALTA	[c] : akg + val-L <==> 3mob + glu-L	**-0.0124**	**-0.0087**	0.0000	**-0.0090**	0.0000	**-0.0093**	0.0000	**-0.0030**	0.0000	**-0.0030**	0.0000	**-0.0030**	0.0000	**-0.0061**	0.0000	**-0.0061**
XPPT	[c] : prpp + xan --> ppi + xmp	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0035**
XYLI2	[c] : glc-D <==> fru	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3333**	**0.2330**	**0.3333**	**0.2330**	**0.3333**	**0.2331**	**0.2381**	**0.2727**	**0.2381**	**0.2724**
Zn2tex	zn2[e] <==> zn2[p]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
ZN2tpp	zn2[p] --> zn2[c]	**0.0001**	**0.0001**	0.0000	**0.0001**	0.0000	**0.0001**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**	0.0000	**0.0000**
Biomass	‡	**0.0293**	**0.0206**	0.0000	**0.0211**	0.0000	**0.0220**	0.0000	**0.0071**	0.0000	**0.0071**	0.0000	**0.0071**	0.0000	**0.0145**	0.0000	**0.0144**

‡Biomass = (0.000223) 10fthf[c] + (0.000223) 2ohph[c] + (0.513700) ala-L[c] + (0.000223) amet[c] + (0.295800) arg-L[c] + (0.241100) asn-L[c] + (0.241100) asp-L[c] + (59.984000) atp[c] + (0.004737) ca2[c] + (0.004737) cl[c] + (0.000576) coa[c] + (0.003158) cobalt2[c] + (0.133500) ctp[c] + (0.003158) cu2[c] + (0.091580) cys-L[c] + (0.026170) datp[c] + (0.027020) dctp[c] + (0.027020) dgtp[c] + (0.026170) dttp[c] + (0.000223) fad[c] + (0.007106) fe2[c] + (0.007106) fe3[c] + (0.263200) gln-L[c] + (0.263200) glu-L[c] + (0.612600) gly[c] + (0.215100) gtp[c] + (54.462000) h2o[c] + (0.094740) his-L[c] + (0.290500) ile-L[c] + (0.177600) k[c] + (0.450500) leu-L[c] + (0.343200) lys-L[c] + (0.153700) met-L[c] + (0.007895) mg2[c] + (0.000223) mlthf[c] + (0.003158) mn2[c] + (0.003158) mobd[c] + (0.001831) nad[c] + (0.000447) nadp[c] + (0.011843) nh4[c] + (0.022330) pe160[c] + (0.026320) pe161[c] + (0.175900) phe-L[c] + (0.000223) pheme[c] + (0.221100) pro-L[c] + (0.000223) pydx5p[c] + (0.000223) ribflv[c] + (0.215800) ser-L[c] + (0.000223) sheme[c] + (0.003948) so4[c] + (0.000223) thf[c] + (0.000223) thmpp[c] + (0.253700) thr-L[c] + (0.056840) trp-L[c] + (0.137900) tyr-L[c] + (0.000055) udcpdp[c] + (0.144100) utp[c] + (0.423200) val-L[c] + (0.003158) zn2[c] + (0.019450) kdo2lipid4[e] + (0.013890) murein5px4p[p] + (0.041480) pe160[p] + (0.048890) pe161[p] --> (59.810000) adp[c] + (59.810000) h[c] + (59.806000) pi[c] + (0.773900) ppi[c]

Appendix F Characterising elementary flux modes for MTB modelling

Table F-1: Listing of elementary flux modes characterising metabolic flux distributions for MTB modelling.

(See Chapter 2) Elementary modes have been normalised to octadecanoate uptake flux. Only reactions used in the elementary modes have been included. Flux values in bold have been rounded to 4 decimal places, with true zero flux values grayed out. Reaction names and formulas copied from the GSMN-TB model—see Beste et al. (2007) for full reaction information. Fluxes of interest and those discussed in the text have been highlighted in orange.

Name	Reaction	Blue	Green	Red	Cyan	Magenta
EX_BIOTINxt	1 BIOTINxt =	0.0000	**-0.0003**	0.0000	**-0.0003**	**-0.0045**
EX_CITxt	1 CITxt =	0.0000	**-0.0001**	0.0000	**-0.0001**	**-0.0001**
EX_CO2xt	1 CO2xt =	**26.8007**	**6.7054**	**26.8007**	**3.7961**	**3.7406**
EX_CO-IIxt	1 CO-IIxt =	0.0000	**0.0000**	0.0000	0.0000	0.0000
EX_FE3xt	1 FE3xt =	0.0000	**-0.0001**	0.0000	**-0.0001**	**-0.0001**
EX_NH3xt	1 NH3xt =	**-0.0268**	**-4.9406**	**-0.0268**	**-5.6521**	**-5.6523**
EX_O2xt	1 O2xt =	**-39.7635**	**-17.6979**	**-35.2844**	**-12.3807**	**-12.2976**
EX_9-OCTADECENOATExt	1 9-OCTADECENOATExt =	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**
EX_PIxt	1 PIxt =	**-0.0178**	**-2.7638**	**-0.0178**	**-3.1616**	**-3.1617**
EX_SLFxt	1 SLFxt =	0.0000	**-0.0091**	0.0000	**-0.0105**	**-0.0105**
R11	1 MAN6P = 1 F6P	0.0000	**-0.0340**	0.0000	**-0.0389**	**-0.0389**
R12	1 MAN1P = 1 MAN6P	0.0000	**-0.0340**	0.0000	**-0.0389**	**-0.0389**
R13	1 MAN1P + 1 GTP = 1 GDPMAN + 1 PI	0.0000	**0.0340**	0.0000	**0.0389**	**0.0389**
R14	1 GDPMAN + 1 DPP = 1 GDP + 1 PPM	0.0000	**0.0266**	0.0000	**0.0305**	**0.0305**
R15	1 GDPMAN = 1 GDPDHDOMAN	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R16	1 GDPDHDOMAN = 1 GDPDHDOGAL	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R17	1 GDPDHDOGAL + 1 NADPH = 1 GDPFUC + 1 NADP	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R20	1 UDPG = 1 UDPGAL	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R21	1 G1P + 1 UTP = 1 UDPG + 1 PPI	0.0000	**0.0302**	0.0000	**0.0346**	**5.3067**
R22	1 bDG6P = 1 IP	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R23	1 IP = 1 PI + 1 MI	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R24	1 MLT = 2 GLC	0.0000	**0.0275**	0.0000	**0.0314**	**0.0314**
R27	1 G6P + 1 UDPG = 1 TRE6P + 1 UDP	0.0000	**0.0252**	0.0000	**0.0289**	**5.3010**
R28	1 MLT + 1 PI = 1 bDG1P + 1 bDGLC	0.0000	0.0000	0.0000	0.0000	**5.2721**
R29	1 bDG1P = 1 bDG6P	0.0000	0.0000	0.0000	0.0000	**5.2721**
R30	1 MLT = 1 TRE	0.0000	**-0.0275**	0.0000	**-0.0314**	**-5.3035**
R31	1 TRE6P = 1 PI + 1 TRE	0.0000	**0.0226**	0.0000	**0.0259**	**5.2980**
R32	1 FOR + 1 NAD = 1 NADH + 1 CO2	0.0000	**0.0065**	0.0000	**0.0074**	**0.0074**
R34	1 bDGLC + 1 ATP = 1 bDG6P + 1 ADP	0.0000	0.0000	0.0000	0.0000	**5.2721**
R37	1 G6P = 1 bDG6P	0.0000	**-0.0571**	0.0000	**-0.0653**	**-10.6095**
R38	1 G6P = 1 G1P	0.0000	**0.0319**	0.0000	**0.0365**	**5.3086**
R39	1 bDG6P = 1 F6P	0.0000	**-0.0600**	0.0000	**-0.0687**	**-0.0687**

R43	1 FDP = 1 F6P + 1 PI	0.0059	0.8009	0.0059	0.9161	0.9161
R44	1 FDP = 1 DHAP + 1 G3P	-0.0059	-0.8009	-0.0059	-0.9161	-0.9161
R45	1 DHAP = 1 G3P	-0.0059	-0.8241	-0.0059	-0.9426	-0.9426
R46	1 G3P + 1 NAD + 1 PI = 1 13PDG + 1 NADH	-0.0238	-1.9862	-0.0238	-2.2705	-2.2705
R48	1 13PDG + 1 ADP = 1 3PG + 1 ATP	-0.0238	-1.9862	-0.0238	-2.2705	-2.2705
R49	1 3PG = 1 2PG	-0.0327	-3.3862	-0.0327	-3.8719	-3.8719
R50	1 2PG = 1 PEP	-0.0327	-3.3862	-0.0327	-3.8719	-3.8719
R51	1 PEP + 1 ADP = 1 ATP + 1 PYR	8.9256	1.2829	0.0000	0.0000	0.0000
R54	1 NAD + 1 DLIPO = 1 LIPO + 1 NADH	8.9167	1.1276	8.9167	0.0000	0.0000
R55	1 OA + 1 ACCOA = 1 CIT + 1 COA	8.9584	4.9317	8.9584	4.3487	4.3487
R56	1 CIT = 1 CISACONITATE	8.9584	4.9318	8.9584	4.3488	4.3488
R57	1 CISACONITATE = 1 ICIT	8.9584	4.9318	8.9584	4.3488	4.3488
R58	1 ICIT + 1 NADP = 1 AKG + 1 NADPH + 1 CO2	0.0000	0.0725	8.9584	4.3488	4.3488
R63	1 SUCCOA + 1 ADP + 1 PI = 1 SUCC + 1 COA + 1 ATP	0.0000	-0.0249	0.0000	-0.0104	-0.0104
R64	1 FUM = 1 MAL	17.8751	7.7873	8.9167	2.0813	2.0814
R65	1 MAL + 1 NAD = 1 OA + 1 NADH	26.8334	12.6466	17.8751	6.3472	6.3473
R69	1 PYR + 1 ATP + 1 PI = 1 PEP + 1 AMP + 1 PPI	0.0000	0.0000	0.0327	4.0915	0.0000
R73	1 ACAL + 1 NADH = 1 ETH + 1 NAD	0.0000	0.0000	0.0000	-1.2494	-1.1386
R74	1 ACAL + 1 NADPH = 1 ETH + 1 NADP	0.0000	0.0000	0.0000	1.2494	1.1386
R75	1 AC + 1 ATP = 1 ACETYLP + 1 ADP	0.0000	0.0089	0.0000	0.0101	0.0101
R76	1 COA + 1 ACETYLP = 1 ACCOA + 1 PI	0.0000	0.0089	0.0000	0.0101	0.0101
R78	1 PROPIONYLP + 1 COA = 1 PROPIONYLCOA + 1 PI	0.0000	0.0000	0.0000	0.0202	0.0202
R82	1 PROPIONYLCOA + 1 OA = 1 2METHYLCITRATE + 1 COA	8.9167	1.1259	8.9167	0.0000	0.0000
R83	1 2METHYLCITRATE = 1 2METHYLCISACONITATE	8.9167	1.1259	8.9167	0.0000	0.0000
R84	1 2METHYLCISACONITATE = 1 METHYLISOCITRATE	8.9167	1.1259	8.9167	0.0000	0.0000
R85	1 METHYLISOCITRATE = 1 SUCC + 1 PYR	8.9167	1.1259	8.9167	0.0000	0.0000
R90	1 RL5P = 1 X5P	-0.0059	-0.6968	-0.0059	-0.7969	-0.7969
R91	1 RL5P = 1 R5P	0.0059	0.6925	0.0059	0.7919	0.7919
R92	1 R1P = 1 R5P	0.0000	-0.0002	0.0000	-0.0002	-0.0002
R94	1 G3P + 1 S7P + 0.001 THI = 1 X5P + 1 R5P	0.0030	0.3415	0.0030	0.3905	0.3905
R96	1 G3P + 1 F6P + 0.001 THI = 1 X5P + 1 E4P	0.0030	0.3553	0.0030	0.4064	0.4064
R97	1 G3P + 1 S7P = 1 E4P + 1 F6P	-0.0030	-0.3415	-0.0030	-0.3905	-0.3905
R100	1 PYR + 1 BIOTIN-CO2 = 1 OA + 0.999 BIOTIN	0.0000	0.0000	0.0000	0.0000	4.1469
R101	1 GDP + 1 PEP + 1 CO2 = 1 OA + 1 GTP	-8.9584	-4.7018	0.0000	0.1822	-3.9093
R102 (ICL)	1 ICIT = 1 GLX + 1 SUCC	8.9584	4.8593	0.0000	0.0000	0.0000
R103 (MALS)	1 GLX + 1 ACCOA = 1 MAL + 1 COA	8.9584	4.8593	8.9584	4.2660	4.2660
R106	1 NADP + 1 METTHF = 1 METHF + 1 NADPH	0.0089	1.8143	0.0089	2.0758	2.0758
R107	1 METHF = 1 FTHF	0.0089	1.8143	0.0089	2.0758	2.0758
R108	1 FADH2 + 1 METTHF = 1 MTHF + 1 FAD	0.0000	0.0139	0.0000	0.0158	0.0158
R109	1 SLF + 1 ATP = 1 APS + 1 PPI	0.0000	0.0091	0.0000	0.0105	0.0105
R110	1 APS + 1 ATP = 1 PAPS + 1 ADP	0.0000	0.0002	0.0000	0.0002	0.0002
R111	1 APS + 1 RTHIO = 1 AMP + 1 H2SO3 + 1 OTHIO	0.0000	0.0090	0.0000	0.0103	0.0103
R112	1 H2SO3 + 6 FERO = 6 FERI + 1 H2S	0.0000	0.0090	0.0000	0.0103	0.0103
R113	1 PAP = 1 AMP + 1 PI	0.0000	0.0002	0.0000	0.0002	0.0002
R119	1 MK + 1 NADH = 1 MKH2 + 1 NAD + 4 H	44.7353	20.2853	44.7353	16.7056	16.5394
R122	1 MK + 1 SUCC = 1 MKH2 + 1 FUM	25.8751	13.9566	16.9167	7.9856	7.9855
R123	1 FAD + 1 SUCC = 1 FADH2 + 1 FUM	-8.0000	-7.9710	-8.0000	-7.9669	-7.9669
R124	1 FE3 + 1 NADPH = 1 FE2 + 1 NADP	0.0000	0.0001	0.0000	0.0001	0.0001
R125	1 MKH2 + 2 HEME-FE3 = 1 MK + 2 HEME-FE2	79.5271	35.3703	70.5687	24.6920	24.5258
R126	2 HEME-FE2 + 0.5 O2 = 2 HEME-FE3 + 6 H	79.5271	35.3703	70.5687	24.6920	24.5258
R128	1 ADP + 1 PI + 4 H = 1 ATP	164.2804	73.9624	150.8510	55.4458	56.3622
R129 (NGAM)	1 ATP = 1 ADP + 1 PI	162.1347	48.4211	139.7142	18.0885	0.0000
R131	1 PPI = 2 PI + 1 H	1.0178	2.4866	1.0506	6.8086	12.1361
R133	1 NADP + 2 FERO = 2 FERI + 1 NADPH	0.0000	-0.0269	0.0000	-0.0308	-0.0308
R134	2 H2O2 = 1 O2	0.0000	0.0047	0.0000	0.0054	0.0054
R139	1 P5C + 1 NAD = 1 HYDROXYGLU + 1 NADH	0.0000	0.0000	8.9584	4.2660	4.2660
R140	1 HYDROXYGLU + 1 AKG = 1 HYDROXYAKG + 1 GLU	0.0000	0.0000	8.9584	4.2660	4.2660
R141 (HtrA)	1 HYDROXYAKG = 1 GLX + 1 PYR	0.0000	0.0000	8.9584	4.2660	4.2660
R144	1 ORN + 1 AKG = 1 GLUGSAL + 1 GLU	0.0000	0.0120	0.0000	0.0000	4.2797
R145	1 CAP + 1 ORN = 1 PI + 1 CITR	0.0000	0.0149	0.0000	0.0170	0.0170
R146	1 ORN = 1 PTRSC + 1 CO2	0.0000	0.0000	0.0000	0.0001	0.0001
R147	1 SAM + 1 PTRSC = 1 5MTA + 1 SPRMD	0.0000	0.0000	0.0000	0.0001	0.0001
R149	1 ILE + 1 AKG = 1 GLU + 1 OMVAL	-8.9167	-1.1359	-8.9167	-0.0095	0.0459
R150	1 LIPO + 1 OXOMETHYLPENTANOATE = 1 2MBUTLIPO + 1 CO2	8.9167	1.1276	8.9167	0.0000	0.0000
R151	1 2MBUTLIPO + 1 COA = 1 2MBUTCOA + 1 DLIPO	8.9167	1.1276	8.9167	0.0000	0.0000
R152	1 MK + 1 2MBUTCOA = 1 MKH2 + 1 2MBECOA	8.9167	1.1276	8.9167	0.0000	0.0000
R153	1 2MBECOA = 1 HMBUTCOA	8.9167	1.1276	8.9167	0.0000	0.0000

R154	1 HMBUTCOA + 1 NAD = 1 MAACCOA + 1 NADH	**8.9167**	**1.1276**	**8.9167**	0.0000	0.0000
R155	1 COA + 1 MAACCOA = 1 PROPIONYLCOA + 1 ACCOA	**8.9167**	**1.1276**	**8.9167**	0.0000	0.0000
R156	1 PROPIONYLCOA + 1 BIOTIN-CO2 = 1 SMMALONYLCOA + 0.999 BIOTIN	0.0000	0.0000	0.0000	**0.0182**	**0.0182**
R157	1 SMMALONYLCOA = 1 RMMALONYLCOA	0.0000	**-0.0159**	0.0000	0.0000	0.0000
R159	1 SUCCOA + 0.001 COB-III = 1 RMMALONYLCOA	0.0000	**0.0159**	0.0000	0.0000	0.0000
R176	1 ASP = 1 bALA + 1 CO2	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R178	1 THR + 1 NAD = 1 2A3KB + 1 NADH	0.0000	**0.0000**	0.0000	0.0000	0.0000
R179	1 2A3KB = 1 AMINOACETONE + 1 CO2	0.0000	**0.0000**	0.0000	0.0000	0.0000
R180	1 AMINOACETONE + 1 NADH = 1 1-AMINO-PROPAN-2-OL + NAD	0.0000	**0.0000**	0.0000	0.0000	0.0000
R183	1 PI + 1 OBUT + 1 O2 = 1 PROPIONYLP + 1 CO2	0.0000	0.0000	0.0000	**0.0202**	**0.0202**
R184	1 SAP + 1 THF = 1 NH3 + 1 METTHF + 1 DHLIPOYLPROTEIN	0.0000	**0.4534**	0.0000	**0.5190**	**0.5191**
R185	1 SER + 1 THF = 1 METTHF + 1 GLY	**0.0089**	**1.3798**	**0.0089**	**1.5782**	**1.5783**
R186	1 GLY + 1 LIPOYLPROTEIN = 1 SAP + 1 CO2	0.0000	**0.4534**	0.0000	**0.5190**	**0.5191**
R187	1 NAD + 1 DHLIPOYLPROTEIN = 1 NADH + 1 LIPOYLPROTEIN	0.0000	**0.4534**	0.0000	**0.5190**	**0.5191**
R190	1 TYR + 1 AKG = 1 4HPP + 1 GLU	0.0000	0.0000	0.0000	**-0.0046**	0.0000
R195	1 AKG + 1 PHE = 1 PHPYR + 1 GLU	0.0000	0.0000	0.0000	**-0.0064**	0.0000
R197	1 GLN + 1 AKG + 1 NADH = 2 GLU + 1 NAD	**8.9167**	**3.4423**	**8.9167**	**3.9595**	**4.0151**
R198	1 NH3 + 1 AKG + 1 NADPH = 1 GLU + 1 NADP	**-8.9078**	0.0000	**-8.9078**	0.0000	0.0000
R199	1 GLU + 1 ATP = 1 GLUP + 1 ADP	0.0000	0.0000	**8.9584**	**4.2797**	0.0000
R200	1 NADPH + 1 GLUP = 1 GLUGSAL + 1 NADP + 1 PI	0.0000	0.0000	**8.9584**	**4.2797**	0.0000
R201	1 GLUGSAL = 1 P5C	0.0000	**0.0120**	**8.9584**	**4.2797**	**4.2797**
R202	2 P5C + 1 NADPH + 1 NADH = 2 PRO + 1 NADP + 1 NAD	0.0000	**0.0060**	0.0000	**0.0069**	**0.0069**
R204	1 NAGLU + 1 ATP = 1 NAGLUP + 1 ADP	0.0000	**0.0269**	0.0000	**0.0171**	**4.2967**
R205	1 NADPH + 1 NAGLUP = 1 NAGLUS + 1 NADP + 1 PI	0.0000	**0.0269**	0.0000	**0.0171**	**4.2967**
R206	1 NAGLUS + 1 GLU = 1 NAORN + 1 AKG	0.0000	**0.0269**	0.0000	**0.0171**	**4.2967**
R207	1 NAORN + 1 GLU = 1 NAGLU + 1 ORN	0.0000	**0.0269**	0.0000	**0.0171**	**4.2967**
R208	1 GLN + 2 ATP + 1 CO2 = 1 CAP + 1 GLU + 2 ADP + 1 PI	0.0000	**0.0240**	0.0000	**0.0275**	**0.0275**
R209	1 ASP + 1 CITR + 1 ATP = 1 AMP + 1 ARGSUCC + 1 PPI	0.0000	**0.0149**	0.0000	**0.0170**	**0.0170**
R210	1 ARGSUCC = 1 ARG + 1 FUM	0.0000	**0.0149**	0.0000	**0.0170**	**0.0170**
R211	2 PYR + 0.001 THI = 1 ACLAC + 1 CO2	**8.9167**	**1.1760**	**8.9167**	**0.0554**	0.0000
R212	1 ACLAC + 1 NADPH = 1 DHMVA + 1 NADP	**8.9167**	**1.1760**	**8.9167**	**0.0554**	0.0000
R213	1 DHMVA = 1 OMVAL	**8.9167**	**1.1760**	**8.9167**	**0.0554**	0.0000
R214	1 GLU + 1 OMVAL = 1 VAL + 1 AKG	0.0000	**0.0178**	0.0000	**0.0204**	**0.0204**
R215	1 THR = 1 OBUT + 1 NH3	0.0000	0.0000	0.0000	**0.0143**	**0.0697**
R216	1 PYR + 1 OBUT = 1 ABUT + 1 CO2	0.0000	0.0000	0.0000	0.0000	**0.0554**
R217	1 ABUT + 1 NADPH = 1 DIHYDROXYMETHYLPENTANOATE + 1 NADP	0.0000	0.0000	0.0000	0.0000	**0.0554**
R218	1 DIHYDROXYMETHYLPENTANOATE = 1 OXOMETHYLPENTANOATE	0.0000	0.0000	0.0000	0.0000	**0.0554**
R219	1 GLU + 1 OXOMETHYLPENTANOATE = 1 ILE + 1 AKG	**-8.9167**	**-1.1276**	**-8.9167**	0.0000	**0.0554**
R220	1 OMVAL + 1 ACCOA = 1 IPPMAL + 1 COA	0.0000	**0.0192**	0.0000	**0.0220**	**0.0220**
R221	1 IPPMAL = 1 CBHCAP	0.0000	**0.0192**	0.0000	**0.0220**	**0.0220**
R222	1 CBHCAP + 1 NAD = 1 OICAP + 1 NADH	0.0000	**0.0192**	0.0000	**0.0220**	**0.0220**
R223	1 OICAP = 1 MOP + 1 CO2	0.0000	**0.0192**	0.0000	**0.0220**	**0.0220**
R224	1 MOP + 1 GLU = 1 LEU + 1 AKG	0.0000	**0.0192**	0.0000	**0.0220**	**0.0220**
R225	1 OA + 1 GLU = 1 ASP + 1 AKG	0.0000	**1.8872**	0.0000	**2.1807**	**2.2362**
R226	1 ASP + 1 GLN + 1 ATP = 1 GLU + 1 ASN + 1 AMP + 1 PPI	0.0000	**0.0046**	0.0000	**0.0052**	**0.0052**
R227	1 ASP + 1 ATP = 1 ASP4P + 1 ADP	0.0000	**0.0247**	0.0000	**0.0485**	**0.1039**
R228	1 ASP4P + 1 NADPH = 1 ASPSA + 1 NADP + 1 PI	0.0000	**0.0247**	0.0000	**0.0485**	**0.1039**
R229	1 ASPSA + 1 NADPH = 1 HSER + 1 NADP	0.0000	**0.0156**	0.0000	**0.0381**	**0.0935**
R230	1 HSER + 1 ATP = 1 PHSER + 1 ADP	0.0000	**0.0119**	0.0000	**0.0279**	**0.0833**
R231	1 PHSER = 1 THR + 1 PI	0.0000	**0.0119**	0.0000	**0.0279**	**0.0833**
R232	1 PYR + 1 ASPSA = 1 DIHYDRODIPICOLINATE	0.0000	**0.0091**	0.0000	**0.0104**	**0.0104**
R233	2 DIHYDRODIPICOLINATE + 1 NADPH + 1 NADH = 2 TETRAHYDRODIPICOLINATE + 1 NADP + 1 NAD	0.0000	**0.0045**	0.0000	**0.0052**	**0.0052**
R234	1 TETRAHYDRODIPICOLINATE + 1 SUCCOA = 1 SAOPIM + 1 COA	0.0000	**0.0091**	0.0000	**0.0104**	**0.0104**
R235	1 SAOPIM + 1 GLU = 1 SDAPIM + 1 AKG	0.0000	**0.0091**	0.0000	**0.0104**	**0.0104**
R236	1 SDAPIM = 1 SUCC + 1 DAPIM	0.0000	**0.0091**	0.0000	**0.0104**	**0.0104**
R237	1 DAPIM = 1 MDAPIM	0.0000	**0.0091**	0.0000	**0.0104**	**0.0104**
R238	1 MDAPIM = 1 LYS + 1 CO2	0.0000	**0.0041**	0.0000	**0.0047**	**0.0047**
R245	1 MTHF + 1 HCYS = 1 MET + 1 THF	0.0000	**0.0139**	0.0000	**0.0158**	**0.0158**
R246	1 PYR + 1 NADH + 1 NH3 = 1 ALA + 1 NAD	0.0000	**0.0432**	0.0000	**0.0495**	**0.0495**
R247	1 3PG + 1 NAD = 1 PHP + 1 NADH	**0.0089**	**1.4000**	**0.0089**	**1.6014**	**1.6014**
R248	1 GLU + 1 PHP = 1 3PSER + 1 AKG	**0.0089**	**1.4000**	**0.0089**	**1.6014**	**1.6014**

R249	1 3PSER = 1 SER + 1 PI	**0.0089**	**1.4000**	**0.0089**	**1.6014**	**1.6014**
R250	1 SER + 1 ACCOA = 1 ASER + 1 COA	0.0000	**0.0051**	0.0000	0.0000	0.0000
R251	1 H2S + 1 ASER = 1 CYS + 1 AC	0.0000	**0.0051**	0.0000	0.0000	0.0000
R252	1 ACCOA + 1 HSER = 1 OAHSER + 1 COA	0.0000	**0.0037**	0.0000	**0.0101**	**0.0101**
R254	1 OAHSER + 1 H2S = 1 AC + 1 HCYS	0.0000	**0.0037**	0.0000	**0.0101**	**0.0101**
R257	1 SER + 1 HCYS = 1 LLCT	0.0000	0.0000	0.0000	**0.0058**	**0.0058**
R258	1 LLCT = 1 OBUT + 1 NH3 + 1 CYS	0.0000	0.0000	0.0000	**0.0058**	**0.0058**
R259	1 MET + 1 ATP = 1 SAM + 1 PPI + 1 PI	0.0000	**0.0103**	0.0000	**0.0117**	**0.0117**
R260	1 SAH = 1 ADN + 1 HCYS	0.0000	**0.0101**	0.0000	**0.0115**	**0.0115**
R261	1 PRPP + 1 ATP = 1 PRBATP + 1 PPI	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R262	1 PRBATP = 1 PRBAMP + 1 PPI	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R263	1 PRBAMP = 1 PRFP	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R264	1 PRFP = 1 PRLP	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R265	1 GLN + 1 PRLP = 1 GLU + 1 DIMGP + 1 AICAR	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R266	1 DIMGP = 1 IMACP	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R267	1 IMACP + 1 GLU = 1 HISOLP + 1 AKG	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R268	1 HISOLP = 1 HISOL + 1 PI	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R269	1 HISOL + 2 NAD = 1 HIS + 2 NADH	0.0000	**0.0044**	0.0000	**0.0051**	**0.0051**
R270	1 PEP + 1 E4P = 1 3DDAH7P + 1 PI	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R271	1 3DDAH7P = 1 DQT + 1 PI	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R272	1 DQT = 1 DHSK	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R273	1 NADPH + 1 DHSK = 1 SME + 1 NADP	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R274	1 SME + 1 ATP = 1 SME3P + 1 ADP	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R275	1 PEP + 1 SME3P = 1 3PSME + 1 PI	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R276	1 3PSME = 1 CHOR + 1 PI	0.0000	**0.0139**	0.0000	**0.0159**	**0.0159**
R277	1 CHOR + 1 GLN = 1 AN + 1 GLU + 1 PYR	0.0000	**0.0028**	0.0000	**0.0032**	**0.0032**
R278	1 AN + 1 PRPP = 1 NPRAN + 1 PPI	0.0000	**0.0028**	0.0000	**0.0032**	**0.0032**
R279	1 NPRAN = 1 CPAD5P	0.0000	**0.0028**	0.0000	**0.0032**	**0.0032**
R280	1 CPAD5P = 1 IGP + 1 CO2	0.0000	**0.0028**	0.0000	**0.0032**	**0.0032**
R281	1 SER + 1 IGP = 1 TRP + 1 G3P	0.0000	**0.0028**	0.0000	**0.0032**	**0.0032**
R282	1 CHOR = 1 PHEN	0.0000	**0.0096**	0.0000	**0.0109**	**0.0109**
R283	1 PHEN = 1 PHPYR + 1 CO2	0.0000	**0.0056**	0.0000	**0.0064**	**0.0064**
R284	1 PHPYR + 1 GLU = 1 AKG + 1 PHE	0.0000	**0.0056**	0.0000	0.0000	**0.0064**
R285	1 PHEN + 1 NAD = 1 4HPP + 1 NADH + 1 CO2	0.0000	**0.0040**	0.0000	**0.0046**	**0.0046**
R286	1 4HPP + 1 GLU = 1 TYR + 1 AKG	0.0000	**0.0040**	0.0000	0.0000	**0.0046**
R287	1 R5P + 1 ATP = 1 PRPP + 1 AMP	**0.0089**	**1.0339**	**0.0089**	**1.1824**	**1.1824**
R288	1 ASP + 1 CAP = 1 CAASP + 1 PI	0.0000	**0.0092**	0.0000	**0.0105**	**0.0105**
R289	1 CAASP = 1 DOROA	0.0000	**0.0092**	0.0000	**0.0105**	**0.0105**
R290	1 O2 + 1 DOROA = 1 OROA + 1 H2O2	0.0000	**0.0092**	0.0000	**0.0105**	**0.0105**
R291	1 OROA + 1 PRPP = 1 OMP + 1 PPI	0.0000	**0.0092**	0.0000	**0.0105**	**0.0105**
R292	1 OMP = 1 UMP + 1 CO2	0.0000	**0.0092**	0.0000	**0.0105**	**0.0105**
R293	1 UMP + 1 ATP = 1 UDP + 1 ADP	0.0000	**0.0143**	0.0000	**0.0164**	**0.0164**
R294	1 UDP + 1 ATP = 1 ADP + 1 UTP	0.0000	**0.0475**	0.0000	**0.0544**	**5.3265**
R296	1 NH3 + 1 UTP + 1 ATP = 1 CTP + 1 ADP + 1 PI	0.0000	**0.0049**	0.0000	**0.0056**	**0.0056**
R297	1 RTHIO + 1 CDP + O2 = 1 OTHIO + 1 DCDP	0.0000	**0.0023**	0.0000	**0.0027**	**0.0027**
R299	1 DCDP + 1 ATP = 1 DCTP + 1 ADP	0.0000	**0.0023**	0.0000	**0.0027**	**0.0027**
R301	1 DUTP = 1 DUMP + 1 PPI	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R302	1 DUMP + 1 FADH2 + 1 METTHF = 1 DTMP + 1 THF + 1 FAD	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R304	1 DTMP + 1 ATP = 1 DTDP + 1 ADP	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R305	1 DTDP + 1 ATP = 1 DTTP + 1 ADP	0.0000	**0.0029**	0.0000	**0.0033**	**0.0033**
R306	1 CDP + 1 ATP = 1 CTP + 1 ADP	0.0000	**0.0122**	0.0000	**0.0140**	**0.0140**
R307	1 CMP + 1 ATP = 1 ADP + 1 CDP	0.0000	**0.0145**	0.0000	**0.0166**	**0.0166**
R310	1 OTHIO + 1 NADPH = 1 RTHIO + 1 NAD	0.0000	**0.0168**	0.0000	**0.0193**	**0.0193**
R313	1 GLN + 1 PRPP = 1 GLU + 1 PRAM + 1 PPI	**0.0089**	**0.9058**	**0.0089**	**1.0358**	**1.0358**
R314	1 ATP + 1 GLY + 1 PRAM = 1 GAR + 1 ADP + 1 PI	**0.0089**	**0.9058**	**0.0089**	**1.0358**	**1.0358**
R315	1 FTHF + 1 GAR = 1 FGAR + 1 THF	**0.0089**	**0.9058**	**0.0089**	**1.0358**	**1.0358**
R316	1 FGAR + 1 GLN + 1 ATP = 1 GLU + 1 FGAM + 1 ADP + 1 PI	**0.0089**	**0.9058**	**0.0089**	**1.0358**	**1.0358**
R317	1 FGAM + 1 ATP = 1 AIR + 1 ADP + 1 PI	**0.0089**	**0.9058**	**0.0089**	**1.0358**	**1.0358**
R318	1 AIR + 1 HCO3 + ATP = 1 CAIR + ADP + PI	0.0000	**0.9040**	0.0000	**1.0349**	**1.0350**
R319	1 CAIR + 1 ASP + 1 ATP = 1 SAICAR + 1 ADP + 1 PI	0.0000	**0.9040**	0.0000	**1.0349**	**1.0350**
R320	1 SAICAR = 1 FUM + 1 AICAR	0.0000	**0.9040**	0.0000	**1.0349**	**1.0350**
R321	1 FTHF + 1 AICAR = 1 PRFICA + 1 THF	0.0000	**0.9084**	0.0000	**1.0400**	**1.0401**
R322	1 PRFICA = 1 IMP	0.0000	**0.9084**	0.0000	**1.0400**	**1.0401**
R328	1 GMP + 1 DATP = 1 GDP + 1 DADP	0.0000	**-0.0012**	0.0000	**-0.0014**	**-0.0014**
R329	1 DUDP + 1 ATP = 1 ADP + 1 DUTP	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R330	1 RTHIO + 1 UDP + O2 = 1 DUDP + 1 OTHIO	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R332	1 5MTA + 1 PI = 1 AD + MTR1P	0.0000	**0.0000**	0.0000	**0.0001**	**0.0001**

R338	1 DA + 1 PI = 1 AD + 1 DR1P	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R338b	1 DR1P = R5P	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R340	1 AD + 1 R1P = 1 ADN + 1 PI	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R349	1 ASP + 1 IMP + 1 GTP = 1 GDP + 1 ASUC + 1 PI	0.0000	**0.8991**	0.0000	**1.0293**	**1.0294**
R350	1 ASUC = 1 FUM + 1 AMP	0.0000	**0.8991**	0.0000	**1.0293**	**1.0294**
R351	1 AMP + 1 ATP = 2 ADP	**1.0089**	**3.2755**	**1.0416**	**7.7131**	**7.7687**
R351b	1 ADN + 1 ATP = 1 AMP + 1 ADP	0.0000	**0.0103**	0.0000	**0.0117**	**0.0117**
R352	1 RTHIO + 1 ADP + O2 = 1 DADP + 1 OTHIO	0.0000	**0.0012**	0.0000	**0.0014**	**0.0014**
R355	1 IMP + 1 NAD = 1 XMP + 1 NADH	0.0000	**0.0093**	0.0000	**0.0107**	**0.0107**
R356	1 XMP + 1 GLN + 1 ATP = 1 GLU + 1 GMP + 1 AMP + 1 PPI	0.0000	**0.0093**	0.0000	**0.0107**	**0.0107**
R357	1 GMP + 1 ATP = 1 GDP + 1 ADP	0.0000	**0.0105**	0.0000	**0.0121**	**0.0121**
R358	1 GDP + 1 ATP = 1 ADP + 1 GTP	**8.9584**	**5.6411**	0.0000	**0.8932**	**4.9847**
R359	1 GDP + 1 RTHIO + O2 = 1 OTHIO + 1 DGDP	0.0000	**0.0023**	0.0000	**0.0027**	**0.0027**
R361	1 ATP + 1 DGDP = 1 DGTP + 1 ADP	0.0000	**0.0023**	0.0000	**0.0027**	**0.0027**
R362	1 ASP + 1 FUM = 1 SUCC + 1 ISUCC	0.0000	**0.0163**	0.0000	**0.0186**	**0.0186**
R364	1 DHAP + 1 ISUCC = 1 QA + 1 PI	0.0000	**0.0163**	0.0000	**0.0186**	**0.0186**
R365	1 QA + 1 PRPP = 1 NACN + 1 PPI + 1 CO2	0.0000	**0.0163**	0.0000	**0.0186**	**0.0186**
R368	1 NACN + 1 ATP = 1 NAAD + 1 PPI	0.0000	**0.0163**	0.0000	**0.0186**	**0.0186**
R369	1 NH3 + 1 NAAD + 1 ATP = 1 AMP + 1 NAD + 1 PPI	0.0000	**0.0163**	0.0000	**0.0186**	**0.0186**
R377	1 NAD + 1 ATP = 1 NADP + 1 ADP	0.0000	**0.0321**	0.0000	**0.0368**	**0.0368**
R381	1 GTP = 1 D6RP5P + 1 FOR + 1 PPI	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R382	1 D6RP5P = 1 A6RP5P + 1 NH3	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R383	1 A6RP5P + 1 NADPH = 1 A6RP5P2 + 1 NADP	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R384	1 A6RP5P2 = 1 A6RP + 1 PI	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R385	1 RL5P = 1 DB4P + 1 FOR	0.0000	**0.0043**	0.0000	**0.0050**	**0.0050**
R386	1 DB4P + 1 A6RP = 1 D8RL + 1 PI	0.0000	**0.0043**	0.0000	**0.0050**	**0.0050**
R387	2 D8RL = 1 RIBFLAV + 1 A6RP	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R388	1 RIBFLAV + 1 ATP = 1 ADP + 1 FMN	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
R389	1 ATP + 1 FMN = 1 FAD + 1 PPI	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R390	1 OMVAL + 1 METTHF = 1 AKP + 1 THF	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R391	1 AKP + 1 NADPH = 1 PANT + 1 NADP	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R392	1 PANT + 1 bALA + 1 ATP = 1 PNTO + 1 AMP + 1 PPI	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R393	1 PNTO + 1 ATP = 1 4PPNTO + 1 ADP	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R394	1 4PPNTO + 1 CYS + 1 CTP = 1 4PPNCYS + 1 CMP + 1 PPI	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R395	1 4PPNCYS = 1 4PPNTE + 1 CO2	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R396	1 4PPNTE + 1 ATP = 1 DPCOA + 1 PPI	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R397	1 DPCOA + 1 ATP = 1 ADP + 1 COA	0.0000	**0.0031**	0.0000	**0.0035**	**0.0035**
R398	1 COA = 1 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R411	1 CYS + 1 FE2 = 1 FERO + 1 ALA	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
R412	1 CHOR = 1 ICHOR	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R413	1 AKG + 1 ICHOR = 1 PYR + 1 SHCHC + 1 CO2	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R414	1 SHCHC = 1 OSB	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R415	1 OSB + 1 COA + 1 ATP = 1 OSBCOA + 1 AMP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R416	1 OSBCOA = 1 DHN + 1 COA	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R417	1 NPP + 1 DHN = 1 DMK + 1 PPI + 1 CO2	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R418	1 SAM + 1 DMK = 1 MKH2 + 1 SAH	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R425	1 CO2 = 1 HCO3	0.0000	**1.1740**	0.0000	**1.3622**	**5.5092**
R426	1 BIOTIN + 1 ATP = 1 BIOTIN-BCCP + 1 AMP + 1 PPI	0.0000	**0.2700**	0.0000	**0.3273**	**4.4742**
R427	1 BIOTIN-BCCP + 1 HCO3 + 1 ATP = 1 BIOTIN-CO2 + 1 ADP + 1 PI	0.0000	**0.2700**	0.0000	**0.3273**	**4.4742**
R427b	1 OCTANOYL-ACP = 1 APO-LIPO + 1 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R427c	1 APO-LIPO + 2 SAM + 2 H2S = 1 LIPO + 2 DA + 2 MET	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R428	1 ATP + 1 GLU = 1 TRNA-GLU + 1 AMP + 1 PPI	0.0000	**0.0009**	0.0000	**0.0009**	**0.0009**
R429	1 TRNA-GLU + 1 NADPH = 1 GLU1SEMIALD + 1 NADP	0.0000	**0.0009**	0.0000	**0.0009**	**0.0009**
R430	1 GLU1SEMIALD = 1 5-AMINO-LEVULINATE	0.0000	**0.0009**	0.0000	**0.0009**	**0.0009**
R431	2 5-AMINO-LEVULINATE = 1 PORPHOBILILOGIN	0.0000	**0.0005**	0.0000	**0.0005**	**0.0005**
R432	4 PORPHOBILILOGIN = 1 HYDROXYMETHYLBILANE + 4 NH3	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R433	1 HYDROXYMETHYLBILANE = 1 UROPORPHYRINOGEN-III	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R434	1 UROPORPHYRINOGEN-III = 1 COPROPORPHYRINOGEN-III + 4 CO2	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R435	1 COPROPORPHYRINOGEN-III + 2 O2 = 1 PROTOPORPHYRINOGEN + 2 H2O2 + 2 CO2	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R437	1 PROTOPORPHYRINOGEN + 1.5 O2 = 1 PROTOPORPHYRIN-IX	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R438	1 PROTOPORPHYRIN-IX + 1 FE2 = 1 PROTOHEME-FE2	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R439	1 PROTOHEME-FE2 + 1 FPP = 1 HEME-FE2 + 1 PPI	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**

R440	1 UROPORPHYRINOGEN-III + 2 SAM = 1 PRECORIN-2 + 2 SAH	0.0000	**0.0000**	0.0000	0.0000	0.0000
R441	1 PRECORIN-2 + 1 SAM = 1 PRECORIN-3A + 1 SAH	0.0000	**0.0000**	0.0000	0.0000	0.0000
R442	1 PRECORIN-3A + O2 = 1 PRECORIN-3B	0.0000	**0.0000**	0.0000	0.0000	0.0000
R443	1 PRECORIN-3B + SAM = 1 PRECORIN-4B + 1 SAH	0.0000	**0.0000**	0.0000	0.0000	0.0000
R444	1 PRECORIN-4B + SAM = 1 PRECORIN-5 + 1 SAH	0.0000	**0.0000**	0.0000	0.0000	0.0000
R445	1 PRECORIN-5 + SAM = 1 PRECORIN-6X + 1 SAH + 1 AC	0.0000	**0.0000**	0.0000	0.0000	0.0000
R446	1 PRECORIN-6X + 1 NADPH = 1 PRECORIN-6Y + 1 NADP	0.0000	**0.0000**	0.0000	0.0000	0.0000
R447	1 PRECORIN-6Y + 2 SAM = 1 PRECORIN-8X + 2 SAH + CO2	0.0000	**0.0000**	0.0000	0.0000	0.0000
R448	1 PRECORIN-8X = 1 HYDROGENOBYRINATE	0.0000	**0.0000**	0.0000	0.0000	0.0000
R449	1 HYDROGENOBYRINATE + 2 ATP + 2 GLN = 1 HYDROGENOBYRINATE-A-C-DIAMIDE + 2 PI + 2 ADP + 2 GLU	0.0000	**0.0000**	0.0000	0.0000	0.0000
R450	1 HYDROGENOBYRINATE-A-C-DIAMIDE + 1 ATP + 1 CO-II = 1 COB-II-RINICACID-A-C-DIAMIDE + 1 ADP + 1 PI	0.0000	**0.0000**	0.0000	0.0000	0.0000
R451	1 COB-II-RINICACID-A-C-DIAMIDE + 1 NADH = 1 COB-I-RINICACID-A-C-DIAMIDE + 1 NAD	0.0000	**0.0000**	0.0000	0.0000	0.0000
R452	1 COB-I-RINICACID-A-C-DIAMIDE + 1 ATP = 1 ADENOSYL-COBRINICACID-A-C-DIAMIDE + 1 PPI	0.0000	**0.0000**	0.0000	0.0000	0.0000
R453	1 ADENOSYL-COBRINICACID-A-C-DIAMIDE + 4 ATP + 4 GLN = 1 ADENOSYL-COBRATE + 4 ADP + 4 PI + 4 GLU	0.0000	**0.0000**	0.0000	0.0000	0.0000
R454	1 ADENOSYL-COBRATE + 1 1-AMINO-PROPAN-2-OL = 1 ADENOSYLCOBINAMIDE	0.0000	**0.0000**	0.0000	0.0000	0.0000
R455	1 ADENOSYLCOBINAMIDE + 1 ATP = 1 ADENOSYLCOBINIMIDE-P + 1 ADP	0.0000	**0.0000**	0.0000	0.0000	0.0000
R456	1 ADENOSYLCOBINIMIDE-P + GTP = 1 ADENOSYLCOBINAMIDE-GDP + 1 PPI	0.0000	**0.0000**	0.0000	0.0000	0.0000
R457	1 ADENOSYLCOBINAMIDE-GDP = 1 COB-I + 1 GMP	0.0000	**0.0000**	0.0000	0.0000	0.0000
R458	1 COB-II + NADH = COB-I + NAD	0.0000	**0.0000**	0.0000	0.0000	0.0000
R459	1 COB-III + NADH = COB-II + NAD	0.0000	**0.0000**	0.0000	0.0000	0.0000
R466	1 TS-PROTEIN = 1 TS-COSH-TYR	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R467	1 AIR = AHMMP	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R468	1 AHMMP + 1 ATP = 1 AMPMP + 1 ADP	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R469	1 AMPMP + 1 ATP = 1 MAHMPPP + 1 ADP	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R470	1 DX5P + 1 TS-COSH-TYR = HEMT + 1 TS-PROTEIN	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R471	1 HEMT + 1 ATP = 1 MPET + 1 ADP	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R472	1 MAHMPPP + 1 MPET = 1 THIP + 1 PPI	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R473	1 THIP = 1 THI + 1 PI	**0.0089**	**0.0019**	**0.0089**	**0.0009**	**0.0008**
R486	2 COA = 1 MBT-HOLO	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R495	1 ACCOA + 1 BIOTIN-CO2 = 1 MALCOA + 0.999 BIOTIN	0.0000	**0.2675**	0.0000	**0.3063**	**0.3063**
R496	1 ACCOA + 1 ACP = 1 COA + 1 ACACP	0.0000	**0.0248**	0.0000	**0.0284**	**0.0284**
R497b	4 MALACP + 8 NADPH + 1 ACACP = 8 NADP + 1 OCTANOYL-ACP + 4 ACP + 4 CO2	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R499	7 MALACP + 14 NADPH + 1 ACACP = 1 HEXADECANOYL-ACP + 14 NADP + 7 ACP + 7 CO2	0.0000	**0.0141**	0.0000	**0.0162**	**0.0162**
R501	8 MALACP + 16 NADPH + 1 ACACP = 1 OCTADECANOYL-ACP + 16 NADP + 8 ACP + 8 CO2	0.0000	**0.0012**	0.0000	**0.0014**	**0.0014**
R503	1 ACACP + 9 MALACP + 18 NADPH = 1 EICOSANOYL-ACP + 18 NADP + 9 ACP + 9 CO2	0.0000	**0.0038**	0.0000	**0.0044**	**0.0044**
R504	1 ACACP + 10 MALACP + 20 NADPH = 1 DOCOSANOYL-ACP + 20 NADP + 10 ACP + 10 CO2	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R505	1 ACACP + 11 MALACP + 22 NADPH = 1 TETRACOSANOYL-ACP + 22 NADP + 11 ACP + 11 CO2	0.0000	**0.0013**	0.0000	**0.0015**	**0.0015**
R506	1 ACACP + 12 MALACP + 24 NADPH = 1 HEXACOSANOYL-ACP + 24 NADP + 12 ACP + 12 CO2	0.0000	**0.0033**	0.0000	**0.0038**	**0.0038**
R507	1 ACACP + 13 MALACP + 26 NADPH = 1 OCTACOSANOYL-ACP + 26 NADP + 13 ACP + 13 CO2	0.0000	**0.0004**	0.0000	**0.0005**	**0.0005**
R508	1 HEXACOSANOYL-COA + 1 BIOTIN-CO2 = 1 HEXACOSANOYL-COA-CO2 + 1 BIOTIN	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R511	1 HEXADECANOYL-ACP + 1 COA = 1 HEXADECANOYL-COA + 1 ACP	0.0000	**0.0023**	0.0000	**0.0026**	**0.0026**
R513	1 OCTADECANOYL-ACP + 1 COA = 1 OCTADECANOYL-COA + 1 ACP	0.0000	**0.0012**	0.0000	**0.0014**	**0.0014**
R515	1 EICOSANOYL-ACP + 1 COA = 1 EICOSANOYL-COA + 1 ACP	0.0000	**0.0034**	0.0000	**0.0039**	**0.0039**
R516	1 DOCOSANOYL-ACP + 1 COA = 1 DOCOSANOYL-COA + 1 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R517	1 TETRACOSANOYL-ACP + 1 COA = 1 TETRACOSANOYL-COA + 1 ACP	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**

R518	1 HEXACOSANOYL-ACP + 1 COA = 1 HEXACOSANOYL-COA + 1 ACP	0.0000	**0.0033**	0.0000	**0.0038**	**0.0038**
R520	1 NONADECANOYL-ACP + 1 COA = 1 NONADECANOYL-COA + 1 ACP	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**
R522	1 PENTADECANOYL-ACP + 1 COA = 1 PENTADECANOYL-COA + 1 ACP	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**
R533	1 PROPIONYLCOA + 1 ACP = 1 PROPIONYLACP + 1 COA	0.0000	**0.0017**	0.0000	**0.0020**	**0.0020**
R534	6 MALACP + 12 NADPH + 1 PROPIONYLACP = 12 NADP + 6 ACP + 1 PENTADECANOYL-ACP + 6 CO2	0.0000	**0.0017**	0.0000	**0.0020**	**0.0020**
R535	1 MALACP + 2 NADPH + 1 PENTADECANOYL-ACP = 2 NADP + 1 ACP + 1 HEPTADECANOYL-ACP + 1 CO2	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**
R536	1 MALACP + 2 NADPH + 1 HEPTADECANOYL-ACP = 2 NADP + 1 ACP + 1 NONADECANOYL-ACP + 1 CO2	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**
R544	1 9-OCTADECENOATE + 1 COA + 1 ATP = 1 9-OCTADECENOYL-COA + 1 AMP + 1 PPI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
R560	1 DODECANOYL-COA + 1 FAD + 1 NAD + COA = 1 DECANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R561	1 DECANOYL-COA + 1 FAD + 1 NAD + COA = 1 OCTANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R562	1 OCTANOYL-COA + 1 FAD + 1 NAD + COA = 1 HEXANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R563	1 HEXANOYL-COA + 1 FAD + 1 NAD + COA = 1 BUTANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9982**	**1.0000**	**0.9980**	**0.9980**
R564	1 BUTANOYL-COA + 1 FAD + 1 NAD + COA = 1 FADH2 + 1 NADH + 2 ACCOA	**1.0000**	**0.9982**	**1.0000**	**0.9980**	**0.9980**
R565	1 9-OCTADECENOYL-COA + 1 FAD + 1 NAD + COA = 1 9-HEXADECENOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9991**	**1.0000**	**0.9990**	**0.9990**
R566	1 9-HEXADECENOYL-COA + 1 FAD + 1 NAD + COA = 1 CIS-DELTA-5-TETRADECANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R567	1 CIS-DELTA-5-TETRADECANOYL-COA + 1 FAD + 1 NAD + COA = 1 CIS-DELTA-3-DODECANOYL-COA + 1 FADH2 + 1 NADH + 1 ACCOA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R568	1 CIS-DELTA-3-DODECANOYL-COA = 1 TRANS-DELTA-2-DODECANOYL-COA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R569	1 TRANS-DELTA-2-DODECANOYL-COA = 1 TRANS-DELTA-2-HYDROXY-DODECANOYL-COA	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R570	1 TRANS-DELTA-2-HYDROXY-DODECANOYL-COA + 1 NAD = 1 DODECANOYL-COA + 1 NADH	**1.0000**	**0.9983**	**1.0000**	**0.9980**	**0.9980**
R571	1 PA = 1 DAG + 1 PI	0.0000	**0.0019**	0.0000	**0.0021**	**0.0021**
R583	0.2 HEXADECANOYL-COA + 0.1 9-HEXADECENOYL-COA + 0.1 OCTADECANOYL-COA + 0.1 9-OCTADECENOYL-COA + 0.1 EICOSANOYL-COA + 0.1 TETRACOSANOYL-COA + 0.1 HEXACOSANOYL-COA + 0.1 NONADECANOYL-COA + 0.1 PENTADECANOYL-COA + 1 GL3P = 1 AGL3P + 1 COA	0.0000	**0.0035**	0.0000	**0.0040**	**0.0040**
R587	1 AGL3P + 0.2 HEXADECANOYL-COA + 0.1 9-HEXADECENOYL-COA + 0.1 OCTADECANOYL-COA + 0.1 9-OCTADECENOYL-COA + 0.1 EICOSANOYL-COA + 0.1 TETRACOSANOYL-COA + 0.1 HEXACOSANOYL-COA + 0.1 NONADECANOYL-COA + 0.1 PENTADECANOYL-COA = 1 PA + 1 COA	0.0000	**0.0035**	0.0000	**0.0040**	**0.0040**
R588	1 PA + 1 CTP = 1 CDPDG + 1 PPI	0.0000	**0.0016**	0.0000	**0.0019**	**0.0019**
R589	1 CDPDG + 1 SER = 1 CMP + 1 PS	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R591	1 DHAP + 1 NADH = 1 NAD + 1 GL3P	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R592	1 CDPDG + 1 GL3P = 1 PGP + 1 CMP	0.0000	**0.0005**	0.0000	**0.0005**	**0.0005**
R593	1 PGP = 1 PI + 1 PG	0.0000	**0.0005**	0.0000	**0.0005**	**0.0005**
R594	1 HEXADECANOYL-ACP + 1 GL3P = 1 ACP + 1 A160GL3P	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R595	1 A160GL3P + 1 TUBERCULOSTEROYL-ACP = 1 PA-TBA + 1 ACP	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R596	1 CTP + 1 PA-TBA = 1 PPI + 1 CDPDAG-TBA	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R597	1 MI + 1 CDPDAG-TBA = 1 CMP + 1 PI-TBA	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R598	1 GDPMAN + 1 PI-TBA = 1 GDP + 1 AC1PIM1	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R599	1 HEXADECANOYL-ACP + 1 AC1PIM1 = 1 ACP + 1 AC2PIM1	0.0000	**0.0003**	0.0000	**0.0004**	**0.0004**
R602	1 GDPMAN + 1 AC1PIM1 = 1 GDP + 1 AC1PIM2	0.0000	**0.0026**	0.0000	**0.0030**	**0.0030**
R603	1 HEXADECANOYL-ACP + 1 AC1PIM2 = 1 ACP + 1 AC2PIM2	0.0000	**0.0016**	0.0000	**0.0019**	**0.0019**
R604	1 HEXADECANOYL-ACP + 1 AC2PIM2 = 1 ACP + 1 AC3PIM2	0.0000	**0.0013**	0.0000	**0.0015**	**0.0015**
R606	1 GDPMAN + 1 AC1PIM2 = 1 GDP + 1 AC1PIM3	0.0000	**0.0010**	0.0000	**0.0011**	**0.0011**

R607	1 HEXADECANOYL-ACP + 1 AC1PIM3 = 1 ACP + 1 AC2PIM3	0.0000	**0.0010**	0.0000	**0.0011**	**0.0011**
R608	1 HEXADECANOYL-ACP + 1 AC2PIM3 = 1 ACP + 1 AC3PIM3	0.0000	**0.0010**	0.0000	**0.0011**	**0.0011**
R610	1 PPM + 1 AC3PIM3 = 1 DPP + 1 AC3PIM4	0.0000	**0.0010**	0.0000	**0.0011**	**0.0011**
R611	1 PPM + 1 AC3PIM4 = 1 DPP + 1 AC3PIM5	0.0000	**0.0010**	0.0000	**0.0011**	**0.0011**
R612	1 HEXADECANOYL-ACP + 1 AC3PIM5 = 1 ACP + 1 AC4PIM5	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R613	1 PPM + 1 AC4PIM5 = 1 DPP + 1 AC4PIM6	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R614	1 PPM + 1 AC4PIM6 = 1 DPP + 1 AC4PIM7	0.0000	**0.0003**	0.0000	**0.0004**	**0.0004**
R615	1 HEXANOYL-COA + 5 MALCOA + 5 SMMALONYLCOA + 20 NADPH = 1 PENTA-METHYL-TRICONTANOYL + 10 CO2 + 20 NADP + 11 COA	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
R616	1 PENTA-METHYL-TRICONTANOYL + 1 COA = 1 PENTA-METHYL-TRICONTANOYL-COA	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
R617	1 PENTA-METHYL-TRICONTANOYL-COA + 1 ATP = 1 PHOSPHO-PENTA-METHYL-TRICONTANOYL-COA + 1 ADP	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
R618	1 MALCOA + 1 ACP = 1 MALACP + 1 COA	0.0000	**0.2590**	0.0000	**0.2965**	**0.2965**
R619	1 SMMALONYLCOA + 1 ACP = 1 RMMALONYLACP + 1 COA	0.0000	**0.0009**	0.0000	**0.0010**	**0.0010**
R620	1 EICOSANOYL-COA + 1 MALACP = 1 BETA-KETO-C22-ACYL-ACP + 1 CO2 + COA	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R621	1 BETA-KETO-C22-ACYL-ACP + 1 NADPH = 1 D-3-HYDROXY-C22-ACYL-ACP + 1 NADP	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R622	1 D-3-HYDROXY-C22-ACYL-ACP = 1 TRANS-DELTA-2-ENOYL-C22-ACYL-ACP	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R623	1 TRANS-DELTA-2-ENOYL-C22-ACYL-ACP = 1 CIS-DELTA-3-ENOYL-C22-ACYL-ACP	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R624	1 CIS-DELTA-3-ENOYL-C22-ACYL-ACP + 5 MALACP + 10 NADPH = 1 CIS-DELTA-13-ENOYL-C32-ACYL-ACP + 5 CO2 + 10 NADP + 5 ACP	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R625	1 CIS-DELTA-13-ENOYL-C32-ACYL-ACP + 1 MALACP + 1 NADPH = 1 CIS-DELTA-2-3,15-ENOYL-C34-ACYL-ACP + 1 NADP + 1 CO2 + 1 ACP	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R626	1 CIS-DELTA-2-3,15-ENOYL-C34-ACYL-ACP + 8 MALACP + 16 NADPH = 1 CIS-DELTA-2-19,31-ENOYL-C50-ACYL-ACP + 8 CO2 + 16 NADP + 8 ACP	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R627	1 CIS-DELTA-2-19,31-ENOYL-C50-ACYL-ACP + 2 SAM + 2 NADP = 1 CYCLOPROPYL-MEROACYL-ACP + 2 SAH + 2 NADPH	0.0000	**0.0020**	0.0000	**0.0023**	**0.0023**
R628	1 CIS-DELTA-3-ENOYL-C22-ACYL-ACP + 8 MALACP + 16 NADPH = 1 CIS-DELTA-19-ENOYL-C38-ACYL-ACP + 8 CO2 + 16 NADP + 8 ACP	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R629	1 CIS-DELTA-19-ENOYL-C38-ACYL-ACP + 1 MALACP + 1 NADPH = 1 CIS-DELTA-2-3,21-ENOYL-C40-ACYL-ACP + 1 CO2 + 1 NADP + 1 ACP	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R630	1 CIS-DELTA-2-3,21-ENOYL-C40-ACYL-ACP + 8 MALACP + 16 NADPH = 1 CIS-DELTA-2-19,37-ENOYL-C56-ACYL-ACP + 8 CO2 + 16 NADP + 8 ACP	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R631	1 CIS-DELTA-2-19,37-ENOYL-C56-ACYL-ACP + 1 SAM = 1 CIS-DELTA-37-ENOYL-METHY-HYDROXY-C56-ACYL-ACP + 1 SAH	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R632	1 CIS-DELTA-37-ENOYL-METHY-HYDROXY-C56-ACYL-ACP + 2 SAM + 2 NADP = 1 CIS-METHOXY-MEROACYL-ACP + 2 SAH + 2 NADPH	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R633	1 CIS-DELTA-37-ENOYL-METHY-HYDROXY-C56-ACYL-ACP + 1 SAM + 2 NADP = 1 CIS-KETO-MEROACYL-ACP + 1 SAH + 2 NADPH	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R634	1 CIS-DELTA-37-ENOYL-METHY-HYDROXY-C56-ACYL-ACP + 1 SAM + 1 NADP = 1 CIS-DELTA-37-ENOYL-DIMETHY-HYDROXY-C56-ACYL-ACP + 1 SAH + 1 NADPH	0.0000	**0.0002**	0.0000	**0.0003**	**0.0003**
R635	1 CIS-DELTA-37-ENOYL-DIMETHY-HYDROXY-C56-ACYL-ACP + 2 SAM + 2 NADP = 1 TRANS-METHOXY-MEROACYL-ACP + 2 SAH + 2 NADPH	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R636	1 CIS-DELTA-37-ENOYL-DIMETHY-HYDROXY-C56-ACYL-ACP + 1 SAM + 2 NADP = 1 TRANS-KETO-MEROACYL-ACP + 1 SAH + 2 NADPH	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R637	0.80 CYCLOPROPYL-MEROACYL-ACP + 0.05 CIS-METHOXY-MEROACYL-ACP + 0.05 TRANS-METHOXY-MEROACYL-ACP + 0.05 CIS-KETO-MEROACYL-ACP + 0.05	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**

	TRANS-KETO-MEROACYL-ACP + 1 ATP = 1 MEROACYL-AMP + 1 ACP + 1 PI					
R638	1 MEROACYL-AMP + 1 HEXACOSANOYL-COA-CO2 = 1 C78-3-OXY-MYCOLATE-ENZ + 1 CO2 + 1 AMP + 1 COA	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R639	1 C78-3-OXY-MYCOLATE-ENZ + 1 NADPH = 1 C78MYCOLATE-ENZ	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R640	1 C78MYCOLATE-ENZ + 1 PPM = 1 C78MYCOLATE-PP	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R641	1 C78MYCOLATE-PP + 1 TRE6P = 1 TREHALOSEMONOMYCOLATE-P + 1 PPM	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R642	1 TREHALOSEMONOMYCOLATE-P = 1 TREHALOSEMONOMYCOLATE(CY) + 1 PI	0.0000	**0.0025**	0.0000	**0.0028**	**0.0028**
R643	1 COA = 1 MAS	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R644	0.25 EICOSANOYL-ACP + 0.25 DOCOSANOYL-ACP + 0.25 TETRACOSANOYL-ACP + 0.25 OCTACOSANOYL-ACP + MAS = 1 ACYL-MAS + 1 ACP	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R645	1 ACYL-MAS + 4 SMMALONYLCOA + 8 NADPH = 1 MYCOCEROSOYL-MAS + 8 NADP + 4 CO2 + 4 COA	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R646	1 HEXADECANOYL-COA + 1 MAS = 1 HEXADECANOYL-MAS + 1 COA	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
R647	1 HEXADECANOYL-MAS + 6 SMMALONYLCOA + 12 NADPH = 1 PTHIOCERANOYL-COA + 12 NADP + 6 CO2 + 5 COA + 1 MAS	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R648	1 HEXADECANOYL-MAS + 8 SMMALONYLCOA + 15 NADPH = 1 HYDROXYPHTHIOCERANOYL-COA + 15 NADP + 8 CO2 + 7 COA + 1 MAS	0.0000	**0.0003**	0.0000	**0.0004**	**0.0004**
R649	1 SMMALONYLCOA + 1 HEXADECANOYL-ACP + 2 NADPH = 2 NADP + 1 COA + 1 CO2 + 1 TUBERCULOSTEROYL-ACP	0.0000	**0.0030**	0.0000	**0.0034**	**0.0034**
R650	1 CHOR = 1 HBA + 1 PYR	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R651	1 DOCOSANOYL-COA + ATP + 3 MALCOA + 2 SMMALONYLCOA + 5 NADPH = 1 PHTHIODIOLONE-A + 5 CO2 + 5 NADP + AMP + PPI	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R652	1 PHTHIODIOLONE-A + 1 NADPH = 1 PTT + 1 NADP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R653	1 PTT + 1 SAM = 1 PTC + 1 SAH	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R654	1 PTC + 2 MYCOCEROSOYL-MAS = 1 DIM-CYTO + 2 MAS	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R655	1 HBA + 8 MALCOA + 16 NADPH + 1 ATP = 1 PHPAA-COA + 7 COA + 9 CO2 + 1 AMP + 1 PPI	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R656	1 PHPAA-COA + 3 MALCOA + 2 SMMALONYLCOA = 1 PHENOLPHTHIODIOLONE-A + 5 CO2 + 5 COA	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R657	1 PHENOLPHTHIODIOLONE-A + NADPH = 1 PPTT + NADP	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R658	1 PPTT + 1 SAM = 1 PPTC + 1 SAH	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R659	1 PPTC + 2 MYCOCEROSOYL-MAS = 1 PHDIM-CYTO + 2 MAS	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R660	1 PHDIM-CYTO = 1 PHDIM	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R661	1 PHDIM + 1 DTDPRHAM = 1 PHDIMRHAM1 + 1 DTDP	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R662	1 PHDIMRHAM1 + 1 DTDPRHAM = 1 PHDIMRHAM2 + 1 DTDP	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R663	1 PHDIMRHAM2 + 1 SAM = 1 MYCOSIDE_B + 1 SAH	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R668	1 TRE + 1 PAPS = 1 TRE-S + 1 PAP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R669	1 TRE-S + 0.333 HEXADECANOYL-COA + 0.333 OCTADECANOYL-COA + 0.333 EICOSANOYL-COA = 1 ACYL-TRE-S-COA	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R670	1 ACYL-TRE-S-COA + 1 HYDROXYPHTHIOCERANOYL-COA = 1 SL1278-CYTO + 2 COA	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R671	1 SL1278-CYTO = 1 SL1278-WALL	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R672	1 OCTADECANOYL-COA + 3 RMMALONYLACP + 6 NADPH = 1 MYCOLIPANOIC-ACP + 3 CO2 + 6 NADP + 1 COA + 2 ACP	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
R673	1 OCTADECANOYL-COA + 3 RMMALONYLACP + 5 NADPH = 1 MYCOLIPENOYL-ACP + 3 CO2 + 5 NADP + 1 COA + 2 ACP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R674	1 OCTADECANOYL-COA + 3 RMMALONYLACP + 4 NADPH = 1 MYCOLIPDIENOYL-ACP + 3 CO2 + 4 NADP + 1 COA + 2 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R678	1 G3P + 1 PYR = 1 DX5P + 1 CO2	**0.0089**	**0.0087**	**0.0089**	**0.0087**	**0.0087**
R679	1 DX5P + 1 NADPH = 1 MDE4P + 1 NADP	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R680	1 MDE4P + 1 CTP = 1 CDPMDE + 1 PPI	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R681	1 CDPMDE + 1 ATP = 1 2PCDPMDE + 1 ADP	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R682	1 2PCDPMDE = 1 MDECPP + 1 CMP	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R683	1 MDECPP + 1 NADH = 1 HMB4PP + 1 NAD	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**

R684	1 HMB4PP + 1 NADH = 1 IPP + 1 NAD	0.0000	**0.0069**	0.0000	**0.0079**	**0.0079**
R685	1 IPP = 1 DMPP	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R686	1 IPP + 1 DMPP = 1 GPP + 1 PPI	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R687	1 IPP + 1 GPP = 1 FPP + 1 PPI	0.0000	**0.0008**	0.0000	**0.0009**	**0.0009**
R688	1 IPP + 1 FPP = 1 GGPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R689	1 IPP + 1 GGPP = 1 PPPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R690	1 IPP + 1 PPPP = 1 HPPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R691	1 IPP + 1 HPPP = 1 HEPPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R692	1 IPP + 1 HEPPP = 1 OPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R693	1 IPP + 1 OPP = 1 NPP + 1 PPI	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
R695	1 PRPP + 1 DPP = 1 PPI + 1 DPPPR	0.0000	**0.0954**	0.0000	**0.1092**	**0.1092**
R696	1 DPPPR = 1 PI + 1 DPPR	0.0000	**0.0954**	0.0000	**0.1092**	**0.1092**
R697	1 DPPR = 1 ARAFDPP	0.0000	**0.0954**	0.0000	**0.1092**	**0.1092**
R698	1 ALA = 1 DALA	0.0000	**0.0100**	0.0000	**0.0114**	**0.0114**
R699	2 DALA + ATP = 1 ALAALA + 1 PI + 1 ADP	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R700	1 DPP + 1 UDPNAG = 1 NAGDPP + 1 UMP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R701	1 NAGDPP + 1 DTDPRHAM = 1 RHAMNAGDPP + 1 DTDP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R702	1 RHAMNAGDPP + 30 UDPGALF = 1 GALACTANDPP + 30 UDP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R705	1 GALACTANDPP + 1 ARAFDPP = 1 ARA[1]GALACTANDPP + 1 DPP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R706	1 ARA[1]GALACTANDPP + 70 ARAFDPP = 1 ARABINOGALACTANDPP + 70 DPP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R707	1 G1P + 1 DTTP = 1 DTDPGLC + 1 PPI	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R708	1 DTDPGLC = 1 DTDP4DH6DGLC	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R709	1 DTDP4DH6DGLC = 1 DTDP4DH6DMAN	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R710	1 DTDP4DH6DMAN + 1 NADPH = 1 DTDPRHAM + 1 NADP	0.0000	**0.0017**	0.0000	**0.0019**	**0.0019**
R711	1 UDPGAL = 1 UDPGALF	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R712	1 GLN + 1 F6P = 1 GLU + 1 GA6P	**-8.9346**	**-5.3172**	**-8.9346**	**-6.1033**	**-6.1588**
R713	1 GLU = 1 DGLU	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R714	1 GA6P = 1 F6P + 1 NH3	**-8.9346**	**-5.3273**	**-8.9346**	**-6.1149**	**-6.1704**
R715	1 GA1P = 1 GA6P	0.0000	**-0.0101**	0.0000	**-0.0116**	**-0.0116**
R716	1 ACCOA + 1 GA1P = 1 COA + 1 NAGA1P	0.0000	**0.0101**	0.0000	**0.0116**	**0.0116**
R717	1 NAGA1P + 1 UTP = 1 UDPNAG + PPI	0.0000	**0.0101**	0.0000	**0.0116**	**0.0116**
R718	1 UDPNAG + 1 PEP = 1 UDPNAGPEE	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R719	1 UDPNAGPEE + 1 NADPH = 1 UDPNAM + 1 NADP	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R722	1 UDPNAM + 1 NADPH = 1 UDPNGM + 1 NADP	0.0000	**0.0035**	0.0000	**0.0040**	**0.0040**
R723	0.7 UDPNGM + 0.3 UDPNAM + 1 ATP + 1 ALA = 1 UDP[NAM:NGM]ALA + 1 ADP + 1 PI	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R724	1 UDP[NAM:NGM]ALA + 1 ATP + 1 DGLU = 1 UDP[NAM:NGM]ALAGLU + 1 ADP + 1 PI	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R725	1 UDP[NAM:NGM]ALAGLU + 1 ATP + 1 MDAPIM = 1 UDP[NAM:NGM]AGMDAPIM + 1 ADP + 1 PI	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R726	1 UDP[NAM:NGM]AGMDAPIM + 1 ALAALA + 1 ATP = 1 UDP[NAM:NGM]AGMDAPIMAA + 1 ADP + 1 PI	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R727	1 UDP[NAM:NGM]AGMDAPIMAA + 1 DPP = 1 LIPID1 + 1 UMP	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R728	1 LIPID1 + 1 UDPNAG = 1 LIPID2 + 1 UDP	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R729	1 LIPID2 + 1 NH3 + 1 ATP = 1 LIPID2-AMIDATED + 1 ADP + 1 PI	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R730	1 LIPID2-AMIDATED = 1 PEPTIDOGLYCAN + 1 DPP	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
R731	30 PEPTIDOGLYCAN + 1 ARABINOGALACTANDPP = 1 ARABINOGALACTAN_PEPTIDOGLYCAN + 1 DPP	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
R800	1 NH3xt = 1 NH3	**0.0268**	**4.9406**	**0.0268**	**5.6521**	**5.6523**
R804	1 O2xt = 1 O2	**39.7635**	**17.6979**	**35.2844**	**12.3807**	**12.2976**
R805	1 CO2xt = 1 CO2	**-26.8007**	**-6.7054**	**-26.8007**	**-3.7961**	**-3.7406**
R838	1 PIxt + 1 ATP = 2 PI + 1 ADP	**0.0178**	**2.7638**	**0.0178**	**3.1616**	**3.1617**
R841	1 SLFxt + 1 ATP = 1 SLF + 1 ADP + 1 PI	0.0000	**0.0091**	0.0000	**0.0105**	**0.0105**
R918	1 9-OCTADECENOATExt = 1 9-OCTADECENOATE	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
R924	1 CITxt + 1 FE3xt = 1 CIT + 1 FE3	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
R925	1 BIOTINxt = 1 BIOTIN	0.0000	**0.0003**	0.0000	**0.0003**	**0.0045**
R926	1 CO-IIxt = 1 CO-II	0.0000	**0.0000**	0.0000	0.0000	0.0000
PROT	0.389 ILE + 0.832 VAL + 0.173 MET + 1.324 ALA + 0.208 HIS + 0.192 LYS + 0.899 LEU + 0.696 ARG + 0.537 ASP + 0.541 SER + 0.429 GLU + 0.283 GLN + 0.556 THR + 0.132 TRP + 0.187 TYR + 0.094 CYS + 0.213 ASN + 40.74 ATP + 0.958 GLY + 0.56 PRO + 0.26 PHE = PROTEIN	0.0000	**0.0214**	0.0000	**0.0245**	**0.0245**
RNA	0.641 UTP + 0.708 CTP + 1.98 ATP + 1.13 GTP = RNA + 1.25 ADP + 1.25 PI + 3.21 PPI	0.0000	**0.0036**	0.0000	**0.0041**	**0.0041**

DNA	0.557 DTTP + 1.061 DGTP + 1.061 DCTP + 4.44 ATP + 0.557 DATP = DNA + 4.44 ADP + 4.44 PI + 3.24 PPI	0.0000	**0.0022**	0.0000	**0.0025**	**0.0025**
SM_MOL	0.188 NAD + 0.168 NADP + 0.163 COA + 0.012 ACP + 0.012 LIPO + 0.012 MBT-HOLO + 0.146 MK + 0.274 FMN + 0.159 FAD + 0.163 MAS + 0.01 HEME-FE2 + 0.01 HEME-FE3 + 0.001 FE2 + 0.001 FE3 + 0.001 FERO + 0.001 FERI + 0.01 SPRMD + 0.01 MTR1P = SMALLMOLECULES	0.0000	**0.0050**	0.0000	**0.0057**	**0.0057**
PE	1.23 PS = 1 PE + 1.23 CO2	0.0000	**0.0006**	0.0000	**0.0007**	**0.0007**
TAG	0.220 HEXADECANOYL-COA + 0.110 9-HEXADECENOYL-COA + 0.110 OCTADECANOYL-COA + 0.110 9-OCTADECENOYL-COA + 0.110 EICOSANOYL-COA + 0.110 TETRACOSANOYL-COA + 0.110 HEXACOSANOYL-COA + 0.110 NONADECANOYL-COA + 0.110 PENTADECANOYL-COA + 1.16 DAG = 1.10 PI + 1 TAGbio + 1.10 COA	0.0000	**0.0016**	0.0000	**0.0018**	**0.0018**
PIMS	0.082 AC2PIM1 + 0.082 AC2PIM2 + 0.082 AC3PIM5 + 0.082 AC4PIM6 + 0.082 AC4PIM7 = 1 PIMS	0.0000	**0.0040**	0.0000	**0.0046**	**0.0046**
LAM	0.064 LM + 4.5 ARAFDPP = 4.5 DPP + 1 LAM	0.0000	**0.0186**	0.0000	**0.0213**	**0.0213**
MAPC	0.008 ARABINOGALACTAN_PEPTIDOGLYCAN + 0.238 TREHALOSEMONOMYCOLATE = 1 MAPC + 0.238 TRE	0.0000	**0.0208**	0.0000	**0.0238**	**0.0238**
P-L-GLX	3.4 GLU + 3.4 GLN = 1 P-L-GLX	0.0000	**0.0035**	0.0000	**0.0040**	**0.0040**
CL	0.647 CDPDG + 0.647 PG = 0.647 CMP + 1 CL	0.0000	**0.0007**	0.0000	**0.0008**	**0.0008**
LM	3.60 PPM + 0.200 AC3PIM2 = 3.60 DPP + 1 LM	0.0000	**0.0066**	0.0000	**0.0075**	**0.0075**
TDM	0.522 TREHALOSEMONOMYCOLATE(CY) = 1 TREHALOSEDIMYCOLATE + 0.522 TRE	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
TMM	0.480 TREHALOSEMONOMYCOLATE(CY) + 0.480 ATP = 1 TREHALOSEMONOMYCOLATE + 0.480 ADP + 0.480 PI	0.0000	**0.0050**	0.0000	**0.0058**	**0.0058**
DAT	1.38 MYCOLIPENOYL-ACP + 0.690 TRE6P = 1 DIACYLTREHALOSE + 1.38 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
PAT	0.334 MYCOLIPANOIC-ACP + 0.668 MYCOLIPENOYL-ACP + 0.668 MYCOLIPDIENOYL-ACP + 0.334 TRE6P = 1 POLYACYLTREHALOSE + 1.67 ACP	0.0000	**0.0001**	0.0000	**0.0001**	**0.0001**
MPD	1.01 PHOSPHO-PENTA-METHYL-TRICONTANOYL-COA + 1.01 GDPMAN = 1 MPD + 1.01 GDP + 1.01 COA	0.0000	**0.0000**	0.0000	**0.0000**	**0.0000**
DIM	0.436 DIM-CYTO = 1 DIM	0.0000	**0.0002**	0.0000	**0.0002**	**0.0002**
PGL	0.259 MYCOSIDE_B + 0.259 GDPFUC + 0.259 DTDP + 0.778 SAM = 1 PGL-TB + 0.778 SAH	0.0000	**0.0029**	0.0000	**0.0033**	**0.0033**
SL-1	0.325 SL1278-WALL + 0.325 HYDROXYPHTHIOCERANOYL-COA + 0.325 PTHIOCERANOYL-COA = 1 SL-1 + 0.650 COA	0.0000	**0.0005**	0.0000	**0.0006**	**0.0006**
GLUCAN	5.5 GLC = 1 GLUCAN	0.0000	**0.0100**	0.0000	**0.0114**	**0.0114**
BIOMASS1	0.214 PROTEIN + 0.036 RNA + 0.022 DNA + 0.050 SMALLMOLECULES + 0.006 PE + 0.016 TAGbio + 0.040 PIMS + 0.186 LAM + 0.208 MAPC + 0.035 P-L-GLX + 0.007 CL + 0.054 LM + 0.001 TREHALOSEDIMYCOLATE + 0.001 TREHALOSEMONOMYCOLATE + 0.001 POLYACYLTREHALOSE + 0.001 DIACYLTREHALOSE + 0.0001 MPD + 0.002 DIM + 0.029 PGL-TB + 0.005 SL-1 + 0.1 GLUCAN + 47 ATP = 47 ADP + 47 PI	0.0000	**0.0999**	0.0000	**0.1143**	**0.1143**