NTNU
Det skapende universitet

A method for dynamic flux balance analysis with global constraints

Emil Karlsen

Bioteknologi (5 årig)
Innlevert: mai 2017
Hovedveileder: Eivind Almaas, IBT

Norges teknisk-naturvitenskapelige universitet
Institutt for bioteknologi og matvitenskap

Acknowledgment

I am grateful towards my supervisor, Professor Eivind Almaas, who had an uncanny ability to see when I wasn't doing my best, and who wasn't shy of pushing me harder when he did. I could hardly have asked for a better captain at the helm.

I am grateful towards Pål Røynestad, whose hard work deciphering the MOMENT model and formulating the easy-to-implement ircFBA model made this work possible.

Thanks to my parents, for instilling in me a great joy in reading, and for keeping my curiosity alive by answering truthfully "I don't know." when I asked difficult, and often silly, questions. And thanks to them for instilling in me a sense that I can become anything I want to be, and for letting me decide for myself what that is.

Thanks to my girlfriend Silje, who patiently listened as I rambled on about subjects she lacked the background to understand, and to my friend Håvard, who patiently listened as I rambled on about subjects I lacked the background to explain. And thanks to him for teaching me how to write a report; this would be a far more byzantine mess of arcane ramblings without his trusty advice and concise commentary. Self-deprecating quips aside, it would easily double my page count if I were to express to all my friends the gratitude they so greatly deserve, and the NTNU press is expensive. The people who welcomed me to Trondheim, the people who welcome me at home, and the people who welcome me at the V&A office: thank you for bearing with me. You all mean so very much to me, and I would not be where, what, or who I am today without you. And I mean that as a compliment!

And last but not least I'd like to direct a big thanks to Marvin Rausand and the guys over at the RAMS wiki for publically making available the latex template used to write this thesis. Without it, I would surely have spent many more hours making a much uglier product.

E.K.

Summary

The utility of modeling in biology is on the rise. Hardware and software is becoming more and more sophisticated, and as the flood of data from high-throughput experimental techniques grows, the necessity and potential sophistication of models grows with it. Flux balance analysis (FBA) has remained one of the most successful and popular methods for modeling cellular metabolism in systems biology for many years, and has spawned a range of offshoots and derivatives.

In this project, two existing forms of FBA were combined into a new one. The first, dynamic FBA (dFBA), allows the modeling of cellular metabolism over time, and how it reacts with its environment. It has proven useful in optimizing industrial batch conditions for microbial production of biomolecules, and for probing the transients of cellular metabolism. The other method, internally constrained FBA (ircFBA) has shown a remarkably accuracy in predicting cellular metabolic behavior across a wide range of growth-media with less situation-specific knowledge and tuning required than standard FBA.

These two methods were combined into a new framework, dubbed dynamic internally constrained FBA (dircFBA). This was done in an attempt at capturing the utility of dFBA and the accuracy of ircFBA. The method, dircFBA, was applied to an extensive genome-scale model of *E. coli.* After testing of the new model's predictions against experimental data retrieved from literature, and analyzing the results, the attempt was deemed a success. The model's predictions improved upon those made by its predecessors, and permits further insight into the dynamics of cell metabolism interacting with a changing environment.

Furthermore, dircFBA provides a solid framework for future expansion, promising even greater fidelity in future extensions.

Sammendrag

(summary in Norwegian)

Nytteverdien til modellering i biologi øker stadig. Maskinvare og programvare blir stadig mer sofistikert, og i møte med den stadig voksende strømmen av data fra høy-volum eksperimentelle teknikker vokser også modellers potensielle sofistikasjon og nødvendighet. Fluksbalanseanalyse (FBA) har vært en av de mest suksessfulle og populære metodene for modellering av cellulær metabolisme i systembiologi i en årrekke, og har gitt opphav til en rekke avskudd og derivater.

I dette prosjektet har to eksisterende former for FBA blitt kombinert til en ny. Den første, dynamisk FBA (dFBA), tillater modelleringen av cellulær metabolisme over tid, og av hvordan denne interagerer med sitt miljø. Denne metoden har vist seg nyttig i optimalisering av industrielle forhold for mikrobiell produksjon av biomolekyler, og for å undersøke transientene i cellulær metabolisme. Den andre metoden, internt begrenset FBA (ircFBA) har vist bemerkelsesverdig treffsikkerhet i å forutse metabolsk adferd over en rekke vekstmedia med behov for mindre situasjonsspesifikk kunnskap og justering enn standard FBA.

Disse to metodene ble kombinert til et nytt rammeverk, her kalt dynamisk internt begrenset FBA (dircFBA). Dette ble gjort i et forsøk på å kombinere nytteverdien til dFBA med treffsikkerheten til ircFBA. Metoden, dircFBA, ble applisert på en omfattende genom-skala modell av *E. coli*. Etter testing av den nye modellens forutsigelser mot eksperimentelle data ervervet fra litteraturen, og analyse av resulatene, ble forsøket vurdert som vellykket. Modellens forutsigelser forbedret på de gjort av sine forgjengere, og tillot videre innsikt i dynamikken i cellemetabolisme som interagerer med et endrende miljø.

Videre tilbyr dircFBA et solid rammeverk for fremtidig utvidning, med lovende prospekter for enda bedre forutsigelser i fremtidige utvidninger av metoden.

Contents

Chapter 1

Introduction

Advances in biology have accelerated in recent years, aided by great advances in technology. New tools, methods, and an ever faster-growing knowledge base have propelled the science forward, allowing the acquisition of biological data on an unprecedented scale [27, 10, 19]. However, with more data the task of sorting through it and extracting useful information becomes harder [27, 12], with global stores of human genome data alone expected to enter the exabytes range in the near future [37]. Therefore, some of the most important advances in biology in recent years have been made possible not mainly due to innovations in the lab, but due to innovations in computation. Automated methods are essential in penetrating the dense networks characterizing intracellular communication [10, 19]. Data mining, the automated smart extraction of information, and computer-assisted refinement into usable knowledge, has enabled theory to almost keep up with high-throughput practice [9]. Likewise, modeling of biological systems has led to remarkable breakthroughs in the understanding of how, and by which principles, they work. This includes insights into the importance of transient dynamics in cell signaling [17], the optimization of metabolic networks in response to evolutionary pressure [5], and by which principles genes are conserved [40].

Science, biology being no exception, has traditionally taken a reductionist approach to investigating natural phenomena, i.e. picking things apart into comfortably-sized pieces and dealing with them individually. Over the past few decades, the interest of many scientists has increasingly shifted from this reductionist approach to a systems-level integrative approach [9, 19]. The way the parts are organized and the interplay between them has proven to be, in many cases, as important as the parts themselves. Biological systems have shown to be highly modular and hierarchical in nature, with components grouped into functional elements whose interplay is tightly regulated, and we have only just begun to untangle [9, 17].

A powerful tool for understanding biology on a systems-level has been network analysis, with graph theory having been applied often and fruitfully in the elucidation of biological systems in recent years [2, 28]. Features fundamental to their functioning and origins, such as the robustness stemming from their scale-free nature [3], and the preferential attachment guiding their formation [22], only come clearly to light when taking a network view. In the face of this, extensive work and thought has been put into building a rigorous theoretical infrastructure for the description of biological models and networks, such as the popular systems biology markup language (SBML) for describing biochemical reaction networks [16], and the establishment of minimum quality standards in the construction of biochemical models; "Minimum information requested in the annotation of biochemical models" (MIRIAM) [20].

Another tool which has shown remarkable versatility in its application within systems biology, is optimization [5]. This is perhaps not surprising, as it has proven invaluable in many other fields of engineering and research [31], and evolution can in many ways work as an optimizing process itself. Nonetheless, it has proven invaluable in areas as diverse as more streamlining experimental design, making parameter estimation for large systems effective and efficient, and helping model cellular metabolism by assuming evolution tuned biological mechanisms for optimality [5].

Among the most popular and successful modeling methods developed so far in systems biology is flux balance analysis (FBA); an elegant method for modeling cell metabolism. FBA is based on the topology of the metabolic network, the stoichiometry of the biochemical reactions that make up an organism's reactome, and optimization [32, 33]. Key to the success of this method has been the formulation of whole-cell metabolism as a set of linear equations, subject to optimization of a goal function and certain constraints. When comparing the predictions based on such simple principles with experimental results, the accuracy is remarkable [38]. As such, FBA methods have found application in science and industry for filling gaps in knowledge of biochemical networks [32], optimizing batch conditions for the production of valuable compounds, and aiding design in synthetic biology [33].

The same principles that make FBA relatively simple to implement are also the source of its key weaknesses: the basic implementation does not incorporate kinetic parameters or regulatory effects, and only predict fluxes at steady state [33]. This may in many cases lead to inaccurate predictions and biologically infeasible behavior [33]. Work is being done, however, to bridge the gap; adding kinetic parameters [1], adding regulatory functionality [7], and relaxing the steady-state assumption [25].

Indeed, the FBA formulation is highly amenable to extensions and modifications, and as such, a wide range of related methods and approaches have been devised [21]. Models have been made with added constraints based on kinetic parameters; metabolic modeling with enzyme kinetics (MOMENT) [1], and internally constrained FBA (ircFBA) [35]. Overlaid on the basic FBA framework, this approach is based on the simple facts that 1) enzymes have mass and volume, 2) the catalytic rate of enzymes is limited, and 3) per amount of cell, there can only be a certain amount of enzyme. Thus, internal kinetic constraints impose upon the cell a biologically plausible limitation as to what it can reasonably produce in any given environment, even independently of uptake bounds [1]. These models have increased prediction accuracy on diverse media and have helped explain observed metabolic flux rates by showing how cells can only reach a certain level of productiveness per amount of mass [1, 35]. Despite its successes, the method remains hampered by a relative lack of information available on enzyme kinetic parameters [1].

Dynamic flux balance analysis (dFBA) has also shown promise of great utility [4]. This method consists of solving many FBA problems sequentially, and letting the model interact with a simulated environment at each step. This effectively simulates a cell culture not only capable of consuming nutrients and oxygen and secreting waste products such as ethanol, but also of exhausting those nutrients and then turning to reuptake of earlier waste [38, 26]. As more accurate

FBA models have been developed, dFBA has seen successful use for optimizing industrial batch conditions to significantly increase production yield of valuable substances such as recombinant biopharmaceutical proteins and biofuels [29, 14].

With more sophisticated behavior comes more risk of complications, and as such, dFBA has a few key weaknesses. As the model is solved and re-solved over consecutive timesteps and the solution is integrated, the process is prone to numerical complications that lead to an infeasible (i.e. unsolvable) linear program [15, 14]. FBA problems also often have non-unique solutions, which may also cause problems as consecutive solutions are integrated [15, 14]. Recent work addresses these issues, but implementation remains complicated [15].

In this thesis is presented a merging of dynamic FBA with internally constrained FBA (ircFBA) into a new method called dynamic internally constrained FBA (dircFBA). This merging serves several purposes: it marries obvious utility with increased accuracy and biological plausibility, allows further insights into the transients of metabolism in changing environments, and provides a sound framework for further development and sophistication of the FBA method. As a side effect, it also circumvents the issue of non-unique solutions and most numerical complications by implicitly imposing a kind of parsimonious FBA (minimizing flux values subject to the solution remaining optimal) [32]. As long as the internal kinetic constraint is what's holding the objective (usually growth rate) back, mass devoted to enzymes that are not essential to maintaining the current objective will be reallocated to enzymes that will increase it.

The method developed, dircFBA, takes dFBA and places two layers of global internal kinetic constraints on it: 1) the global fraction of cellular mass devoted to metabolic enzymes, in the style of ircFBA [35] and 2) the global fraction of cellular mass that can be reallocated per unit of time. It thereby places constraints on: 1) how productive the model can be on a given medium, and 2) how quickly the cell can adjust to a changed og changing medium to increase or maintain its productivity.

After being implemented on the iAF1260b *E. coli* model [13], the method is tested in several simulated dynamic environments and compared to experimental results [38], before the consequences of the internal constraints are discussed. The dircFBA method is found to predict cell density and environmental concentrations of nutrients with a high level of accuracy, especially considering the size of the model and the low level of manual tuning. The second layer of global internal constraints, limiting the rate at which the cell can reallocate resources, is not found to impact the results to a large extent, but proves useful at pinpointing where flux changes and cellular reallocation of resources occurs during the simulation run. It seems plausible that further extensions upon the dircFBA framework, especially in the vein of regulatory mechanisms, could improve accuracy even further, and make for a powerful tool both in batch optimization and in probing the transients of cellular metabolism.

Chapter 2

Theory

2.1 Cellular metabolism

Central to metabolism are enzymes: proteins, i.e. biological macromolecules, that catalyze chemical reactions, by lowering the activation energy of highly specific reactions, and thereby increasing the rate at which they happen. Thus they take one or several substrate molecules, and turn them into one or several products far faster than the process would otherwise take. In biochemical terms, both substrate and product are referred to as metabolites. The chemical reactions that transform metabolites are what we call metabolism [30, 39].

And so metabolites take turns being the products and substrates of different enzyme-catalyzed reactions, turning various nutrients into energy and proteins filling a myriad of functions. Although this process occurs at a massive scale, with the number of reactions occurring per second in a cell ranging from 10^6 to 10^9 depending on species and activity, it is far from indecipherable or impossible to describe (Section 2.3).

Enzymes catalyze biochemical reactions by binding the substrate to its active site, forming an enzyme-substrate complex. This complex may break apart back into its constituent components, enzyme and substrate, or the enzyme may induce a conformational change in the substrate, causing it to fuse with another molecule, break apart into several molecules, or otherwise alter its structure, in which case the complex will break apart into enzyme and product instead.

The enzyme is then ready repeat the process, again and again, until it runs out of substrate to bind, or is broken down [30].

Besides allowing reactions to occur at much greater rates than they otherwise would, enzymes enable great control over metabolic processes. They can be activated and deactivated by the addition or removal of chemical groups, such as phosphoryl or methyl groups. These alter the shape of the enzyme's active site, which is where the reaction takes place, or otherwise alter the conformation of the enzyme such that it becomes able or unable to perform its task. Rather than switching them strictly on or off, such addition or removal of chemical groups can make the enzymes work faster or slower. Other attributes in the cell's internal environment, or the environment of one of its compartments, can also affect the speed at which enzymes operate, such as temperature, pH, the availability of "energy currency" molecules such as ATP, and the concentration of substrate [30]. In addition to allowing for tight regulation of its biochemical processes by the cell, it also makes measuring the catalytic rate of an enzyme a difficult task. Measuring the rate at which it catalyzes reactions *in vivo*, that is, inside the organism itself, is near impossible to do accurately using currently available methods [11]. Therefore, such measurements are usually performed *in vitro* instead; that is, in the lab outside their natural biological context.

Since the catalytic rate of enzymes depend on the amount of substrate present in the environment, a simple flat metric for the catalytic rate does not suffice to describe the process, and a model that allows meaningful parameters to be derived from measurements is required. The most common model of enzyme kinetics is called Michaelis-Menten kinetics, which describes an enzyme's reaction rate v using three parameters: the maximum catalytic rate of the enzyme when saturated with substrate (denoted $V_{\max}$), the concentration of substrate (denoted $[S]$), and the concentration of substrate for which the catalytic rate is half the maximum catalytic rate (denoted K_M). [30]

$$v = \frac{d[P]}{dt} = \frac{V_{\max}[S]}{K_M + [S]} \tag{2.1}$$

The Michaelis-Menten equation, seen in Equation 2.1, expresses how these parameters combine to give the reaction rate. The maximum number of substrate molecules that are converted into product molecules per enzyme molecule per second is called turnover number, denoted k_{cat} [30].

2.2 Linear optimization

Optimization has become an indispensable tool in many fields of science and engineering, carried on by a coevolution with computer science. At its most basic, optimization consists of finding the "best" solution to a given problem. What qualifies a solution as being the "best" one is not always obvious however, and determining this objective can be a challenge in and of itself. [31] As such, optimization is better defined as: "given precisely defined problem, a precisely defined set of tools to solve it, and precisely defined objective, what is the best solution?". Of course, "best" solutions need not be good, or unique. As such, any "best" solution is, in technical terms, simply designated as "optimal".

Also known as linear programming, linear optimization[1] is performed by constructing an optimization problem with a linear objective function and linear constraints [31]. Linear optimization is well suited for computation, and these kinds of problems have been studied thoroughly for many decades. Current implementations and hardware allow large problems with hundreds of thousands of variables subject to constraints to be solved swiftly and efficiently. The standard form of a linear program is given in Equation 2.2 [31].

$$\begin{aligned} \text{minimize} \quad & \bar{\mathbf{c}}^T \bar{\mathbf{v}} \\ \text{subject to} \quad & \bar{\bar{\mathbf{S}}}\bar{\mathbf{v}} = \bar{\mathbf{b}} \\ & \bar{\mathbf{v}} \geq \bar{\mathbf{0}} \end{aligned} \tag{2.2}$$

In Equation 2.2, $\bar{\mathbf{v}}$ is a vector containing the variables to be determined, while $\bar{\mathbf{c}}$ is a vector de-

[1] The presentation of linear optimization here leans heavily on the pedagogical approach from the book Numerical Optimization by Nocedal and Wright [31], as this is the primary source of the author's knowledge on the subject.

termining the objective function, a linear weighting of the different variables' *utility*, as it were. The matrix $\bar{\bar{\mathbf{S}}}$ and the vector $\bar{\mathbf{b}}$ impose equality constraints on the variables, while the last line impose the constraint that the variables must take on positive values. While this standard form may appear rigid, various reformulations allow for the solution of different problems, such as the introduction of *slack variables* (usually denoted $\bar{\mathbf{z}}$) to allow for inequality constraints, and the splitting of $\bar{\mathbf{v}}$ into nonnegative and nonpositive parts to allow the variables to take on "negative" values. If a maximization rather than a minimization is desired, negative weights can be added to the objective function [31].

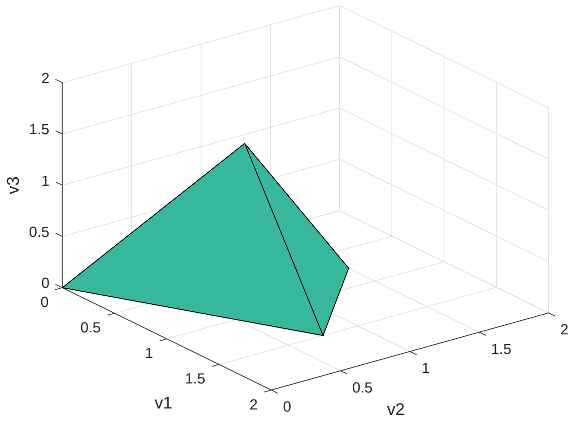

Figure 2.1: **An example of a three-dimensional convex polytope.** Created in Matlab. Plainly, any point within the polytope can be connected to any other point within the polytope, without the connecting line going outside it. This feature generalizes to arbitrary dimensions; as long as the constraints are linear, i.e. form lines, planes or hyperplanes, the bounded space within will form a convex polytope.

Linear programming is popular in large part because it is guaranteed to arrive at a global optimum, as opposed to non-linear optimization, which risks trapped in a local optimum. Adding to that, solution algorithms exist that allow swift arrival at this global optimum. This stems from the fact that the linear constraints guarantee a quality known as a convex solution space. The solution space is also known as the feasible set, i.e. the set of all points that do not violate a constraint. An example of a 3-dimensional convex polytope can be seen in Figure 2.1. A convex space is one in which any point in that space can be connected to any other point in the space by a straight line, without that line passing through any point which does not lie in the space.

This is combined with a linear objective function, which ensures that the optimal point(s) will lie on the surface of the convex solution space. This is illustrated in Figure 2.2. At any point in the solution space, the gradient of the objective function will unambiguously reveal which directions are favorable. Therefore, when a solution algorithm arrives at a point from which any and all favorable directions entail the violation of a constraint, a locally optimal point has been found. Due to the convex nature of the solution space in a linear program, a local optimum can also unambiguously be claimed to be a global optimum [31].

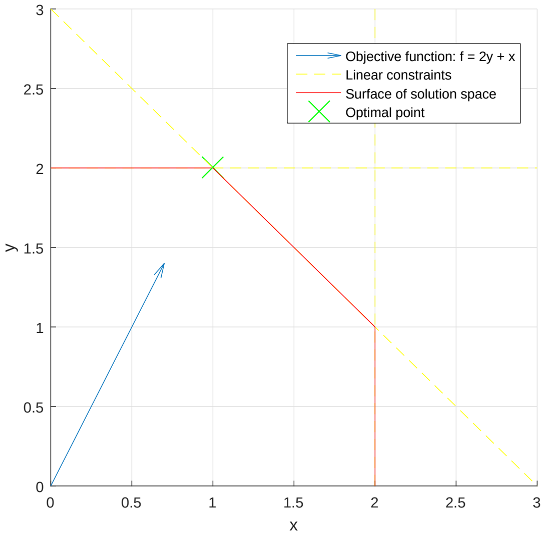

Figure 2.2: **A simple example of a 2-dimensional linear optimization problem.** Created in Matlab. Here, a maximization of the objective function f is sought. As the objective is linear, it points straight forwards in one direction, and does not curve. This ensures that the optimal point must lie on the surface of the convex polytope created by the linear constraints.

Since any sound algorithm whose step-size does not decrease too rapidly is guaranteed to terminate at a global optimum, the best measure of what makes a good algorithm for solving linear programs is usually speed. More sophisticated algorithms exploit certain known fact about linear programs, such as the shape of the solution space, to take shortcuts. This enables far more swift traversal of the solution space, and thus termination at an optimal point, than naive implementations. One example of a popular algorithm for solving linear programs is the simplex method, which bases itself on the fact that the solution to a linear program will lie on the surface of its solution space. A convex polytope has vertices, i.e. "corners" in arbitrary dimensions.

While the surface of the solution space consists of an infinite number of points, as any non-zero-dimensional shape does, it only contains a finite number of vertices. The simplex method iterates through the vertices of the feasible polytope, getting closer to the global optimum with each step. Once no further step can be taken that improves the solution, optimality is achieved, and the algorithm terminates [31].

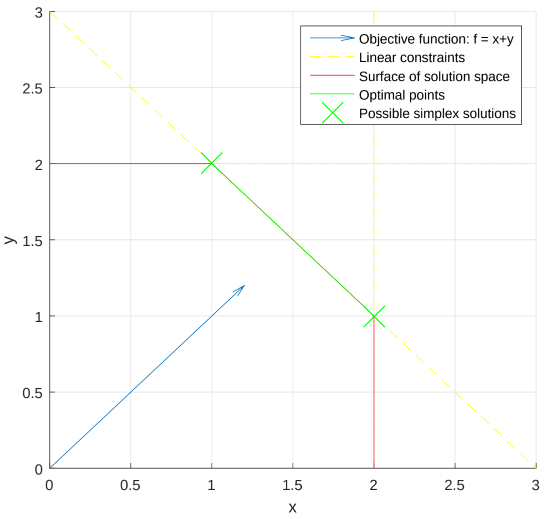

Figure 2.3: **A simple example of a 2-dimensional linear optimization problem without a single unique solution.** Created in Matlab. Here, a maximization of the objective function f is sought. As the objective is linear, it points straight forwards in one direction, and does not curve. However, multiple solutions are possible; there are an infinite number of points along the green line. Marked with green X-es are the optimal vertices, one of which the simplex algorithm would terminate at; of these there are only two.

While solutions to linear programs are relatively easy to find, they are not guaranteed to be unique. Any number of vertices, and consequently the lines between these vertices, might comprise a global optimum, and this range of possible optimal solutions might not always make logical sense when applied to real-world problems. An example of a linear program with no single unique solution can be seen in Figure 2.3. Discrepancies such as this may arise particularly in cases where non-linear phenomena have been linearized to simplify computation. Fortunately, algorithms also exist to find all of the optimal solutions; that is, the breadth of space within which the solution remains optimal. This may in turn help an operator make better decisions when applying the results of the optimization to the real world [31].

In many real-world applications of optimization, knowing the change in the value of the objective function resulting from a change in a constraint is both useful and important. This is called the shadow price. The shadow price is the marginal change in the objective function per unit step of change in the constraint. In other terms, it's basically the directional derivative of the objective function in the solution point, orthogonally onto the constraint in question. Using shadow prices, it is possible to determine which constraints can be relaxed to provide the greatest gain in utility. In an environment with limited resources, this can help an operator determine where to allocate resources to relax constraints in order to gain the greatest benefit [31]. A simple example illustrating the usefulness of shadow prices would be a transport company, which naturally earn money by transporting goods. Management may be looking to invest a surplus, and wondering whether to buy more trucks or hire more drivers. Comparing the shadow prices for the constraints imposed by lack of personnel with the shadow prices imposed by lack of vehicles can help management make a more informed decision.

2.3 Metabolic modeling

Certain creatures, like some bacteria and yeast, are akin to tiny molecular factories. They consume nutrients, such as sugars and nitrogen-rich compounds, and produce waste products, such as ethanol and carbon dioxide, and are constantly building more copies of themselves. As the main business of these creatures mostly consists of balancing tiny internal production budgets, they lend themselves exceedingly well to computer modeling: construct the metabolic network, simulate the environment, and use that to define a solvable problem, to which you try to find the optimal solution. If your model, method and parameters are all correct, along with the assumption about what the organism optimizes for, the result will carry an almost uncanny resemblance to what the organism would do under the same conditions in the lab [32].

Systems biology integrates knowledge of biological systems into predictive mathematical models, in an attempt to root out errors in their understanding, and thus aid in the formulation of new hypotheses. A great variety of models have sprung up in recent years, differing in detail, complexity, and core working principles. Especially popular are constraint-based reconstruction and analysis (COBRA) methods [21]. An illustrative overview of these and their interrelations can be seen in Figure 2.4.

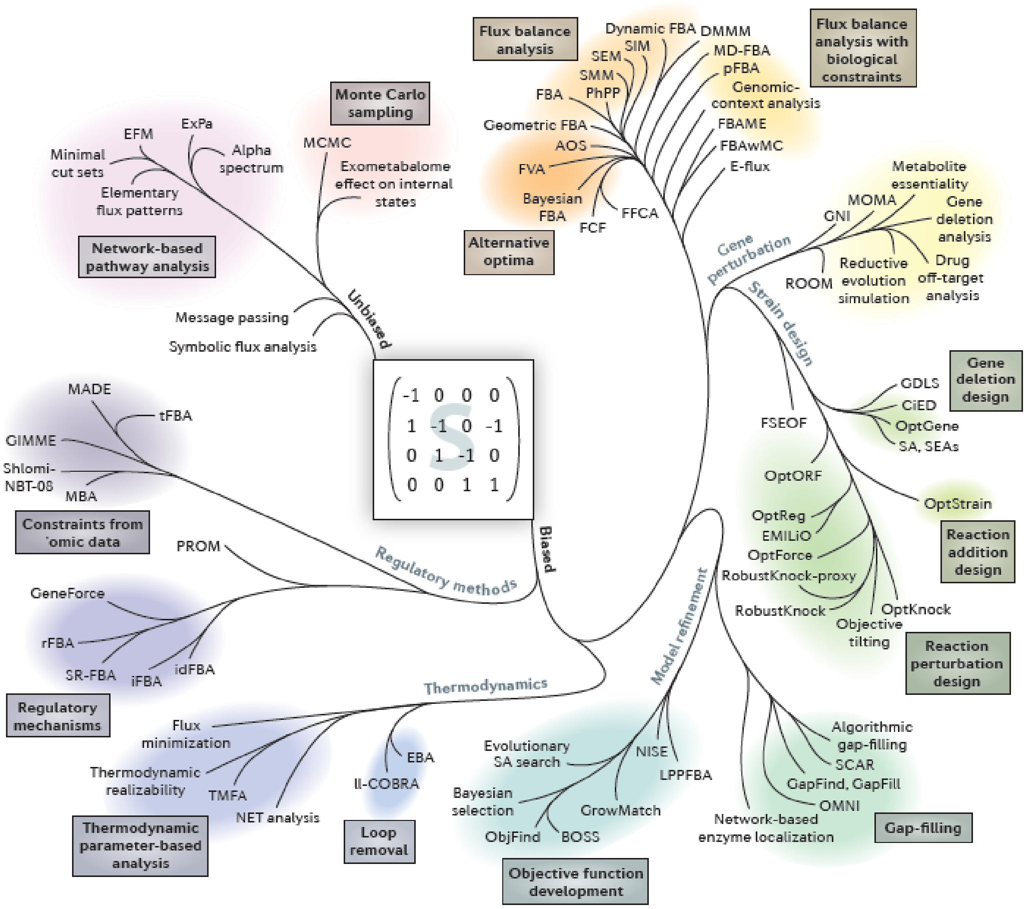

Figure 2.4: **The 'phylogeny' of constraint-based modelling methods.**" [21]. Figure 2 from the paper "Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods" by Lewis et al. [21]. It shows an overview of the various COBRA methods and their relationships, per 2012.

Constraint-based optimization [32] of genome-scale models (GEMs) has shown a remarkably accurate ability to predict cellular behavior at the steady state while adhering to a relatively simple set of rules. A complete genome-scale model is intended to fully describe a metabolic network. The process of building of such a metabolic network is called "metabolic reconstruction". The metabolic network consists of a number of reactions. These reactions consume a certain amount of substrate, creates a corresponding amount of product, may or may not be reversible, and take place at a given compartment of the cell. An environment is also defined for the cell, giving it access to a set of nutrients. Given this information, it is possible to formulate constraints, which limit the flux values that the various reactions can take on. This is combined with a clearly defined goal for the cell, such as the production of as much biomass as possible. All this allows for the clear formulation of a mathematical model. This process allows a COBRA program to determine the flux through the system's reactions and pathways through flux balance analysis (FBA) [34][33].

2.4 Flux Balance Analysis

Flux balance analysis is the most common method for constraint-based analysis of metabolic models. It is intended to simulate steady-state metabolism in a living cell for a given nutritional environment [33], finding the steady-state flux through each biochemical reaction. Key to this simulation is the quasi-steady state assumption (QSSA) which posits that the metabolic network has reached equilibrium with the environment, and all metabolites within the network are being produced and consumed at an identical rate, so that nothing except biomass is accumulated within the organism.

The simulation is performed by formulating a linear program in which the flux through the various reactions are represented as variables, and reaction stoichiometry and uptake bounds are represented as equality and inequality constraints, respectively. The objective function is usually chosen to be the maximization of a "biomass dummy function", which takes in metabolic precursors required for cell growth and proliferation, such as lipids for making more cell membrane. When the FBA problem is formulated in such a way (as seen in Equation 2.3), the (reasonable) assumption is made that evolution has optimized cell metabolism for maximization of growth rate. The various flux values are given as millimoles per gram of dry weight per hour; "$[\frac{mmol}{gDW\ h}]$", and the biomass function is formulated so that the flux through it corresponds directly to growth rate in terms of "number of divisions per hour".

$$\begin{aligned} \text{maximize} \quad & \overline{\mathbf{c}}^T \overline{\mathbf{v}} \\ \text{subject to} \quad & \overline{\overline{\mathbf{S}}}\overline{\mathbf{v}} = \overline{\mathbf{b}}(= \overline{\mathbf{0}}) \\ & \overline{\mathbf{lb}} \leq \overline{\mathbf{v}} \leq \overline{\mathbf{ub}} \end{aligned} \tag{2.3}$$

(Above, the " = ", " ≤ ", and ≥ operators are meant in an element-wise fashion.)

More explicitly, in its implementation the problem will have the following form (Equation 2.4):

$$\begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \dots & \dots & \dots & \dots \\ S_{m1} & S_{m2} & \dots & S_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad where \quad \begin{bmatrix} lb_1 \\ lb_2 \\ \vdots \\ lb_n \end{bmatrix} \leq \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \leq \begin{bmatrix} ub_1 \\ ub_2 \\ \vdots \\ ub_n \end{bmatrix} \tag{2.4}$$

Where n is the number of reactions, and m is the number of metabolites. The columns of the $\overline{\overline{\mathbf{S}}}$ matrix will give the stoichiometrics of the corresponding element in the $\overline{\mathbf{v}}$ vector (that is, the corresponding reaction flux) with a positive number on a given row signifying that the corresponding reaction produces that number of the metabolite corresponding to that row, and vice versa for negative numbers. Unless errors have been made in the assembly of the model, these reactions are mass-balanced. The $\overline{\mathbf{b}}$ vector will be all zeros so that every metabolite produced by a reaction must be consumed by another, which ensures steady-state.

A standard FBA solution returns the flux distribution within the cell, including the flux through the biomass function which represents the growth rate. For a given bound on uptake fluxes, the FBA solution will, assuming growth is the objective, return the flux values corresponding to the highest possible growth rate it can achieve in that environment without violating stoichiometry or other imposed bounds. That is, the flux vector $\overline{\mathbf{v}}$, lying in the null-space of the stoichiometric matrix $\overline{\overline{\mathbf{S}}}$ that maximizes the value of $\overline{\mathbf{c}}^T \cdot \overline{\mathbf{v}}$. This is usually equivalent to v_{biomass}, i.e. the flux through the biomass function, i.e. the growth rate [33]. As with linear programs in general, solutions need not be unique. There are multiple ways of dealing with this issue, e.g. running parsimonious FBA (pFBA), which minimizes the sum of fluxes, subject to maintaining the objective at an optimal value. Another is to find the range of different steady-state fluxes for which the solution remains optimal. This is performed through flux variability analysis (FVA). When performing FVA, the objective is bounded from below, only allowing solutions that keep it at least that high. Each flux is then minimized and maximized in turn to find the upper and lower steady-state bound at which they still allow the objective to remain at the set value [33].

2.4.1 Dynamic Flux Balance Analysis

Regular FBA seeks a steady-state solution to the problem of optimal growth. Dynamic flux balance analysis (dFBA) is an attempt to extend this modeling framework to help model how metabolic states change over time and interact with their environment, useful for optimizing batch conditions in industrial bioprocessing [29, 14] and for helping to understand the temporal dynamics of metabolic systems [4].

A single "run" of dFBA consists of solving many FBA problems, meant to represent the metabolic states at different times throughout the run. Therefore, dFBA necessarily has a certain "resolution", in that every solved FBA problem is claimed to represent cellular metabolism for an entire subinterval. The shorter these subintervals are, the finer this resolution becomes. Between each interval, the environment, the cellular metabolism, and other model parameters such as accumulated biomass and the availability of environmental nutrients, interact with one another according to the rules posited in the overall dFBA model, and are updated. [26] While not an absolute requirement for a dFBA formulation, dynamic constraints may be placed on how rapidly the cell can adjust its metabolism in response to changes in the environment [26].

As with FBA, the quasi-steady state assumption (QSSA) is key. The intracellular dynamics of the cell metabolism must act on a rapid timescale compared to the extracellular environmental dynamics, so that the metabolic network can reach equilibrium for a given set of uptake fluxes [4].

The method was first used by Varma and Palsson [38] to predict the behavior of *E. coli* during several batch runs. In a later article, Mahadevan et al. [26] formalized the method, and presented two different approaches to dynamic flux balance analysis, called the *dynamic optimization approach* (DOA) and the *static optimization approach* (SOA) [26].

In both cases, a given time period over which the optimization is meant to occur is determined. A set of initial environmental conditions are then, defined along with a function determining how the cell culture interacts with this environment. Additionally, constraints are imposed upon how fast the different fluxes within the cell are allowed to change. This time period is then divided up into a number of discrete intervals [26].

In their DOA formulation, the entire optimization run is combined into a single non-linear programming problem (NLP), which returns the solution fluxes at every time interval for the whole run [26]. Compared to linear programs, discussed in Section 2.2, non-linear programs are not as easy to solve, and numerical solvers are not necessarily guaranteed to arrive at a global optimum [31].

In the SOA formulation, a linear program (LP) is reformulated and solved much in the way of regular FBA at each distinct time interval, with environmental conditions determined by those of the previous interval and the cells' interaction with them [26].

Dynamic flux balance analysis has seen widespread use and is today part of standard constraint-based modeling toolkits for cellular metabolism [6].

2.4.2 Internally Constrained Flux Balance Analysis

All FBA is based on constrained optimization, but in the standard formulation of FBA, the constraints lie solely in the stoichiometry and the upper bounds on the uptake fluxes. Internally constrained FBA (ircFBA) on the other hand, contains additional internal constraints, in the form of how large a proportion of the cell mass can be occupied by enzymes, and these enzymes' catalytic rates. If optimal enzyme composition and metabolite concentration is assumed, the rates of the different cell fluxes will be equal to the amount of the corresponding enzymes times their maximum catalytic rates. Thus, ircFBA integrates genome-scale metabolic modeling with kinetic parameters by placing kinetic constraints on the FBA problem [1]. In assembling the model, the mass and maximum catalytic rate of the enzyme corresponding to each reaction is gathered and coupled within the model. As FBA works on a "per gram of dry weight"-basis, constraining the proportion of the cell mass available for enzymes is a relatively simple matter through a reformulation of the FBA linear program [35].

$$\begin{aligned} \text{maximize} \quad & \overline{\mathbf{c}}^T \overline{\mathbf{v}} \\ \text{subject to} \quad & \overline{\overline{\mathbf{S}}}\overline{\mathbf{v}} = \overline{\mathbf{b}} \\ & \overline{\mathbf{lb}} \leq \overline{\mathbf{v}} \leq \overline{\mathbf{ub}} \\ & -\overline{\overline{\mathbf{kCat}}} \cdot \overline{\mathbf{g}} \leq \overline{\mathbf{v}} \leq \overline{\overline{\mathbf{kCat}}} \cdot \overline{\mathbf{g}} \\ & \overline{\mathbf{g}}^T \cdot \overline{\mathbf{eM}} \leq M_c \\ & \mathbf{g} \geq \mathbf{0} \end{aligned} \tag{2.5}$$

(Above, ⊙ means "elementwise multiplication", and the " = ", " ≤ ", and ≥ operators are meant in an element-wise fashion.)

Equation 2.5 shows how the reformulation of the FBA problem applies the internal constraints. In it, $\overline{\mathbf{g}}$ is a length n (n being the number of reactions in the original FBA model, i.e. the breadth of the $\overline{\overline{\mathbf{S}}}$ matrix) vector describing the amount per gram of cell devoted to the corresponding enzyme, $\overline{\overline{\mathbf{kCat}}}$ is an n-by-n diagonal matrix describing the catalytic rates of the corresponding enzymes, and the vector $\overline{\mathbf{eM}}$ is a length n vector describing the mass of the corresponding enzymes. M_c is the mass constraint, given as the proportion of its mass the cell can devote to metabolic enzymes. The "corresponding enzyme" above refers to the enzyme catalyzing the reaction which flux is given in the corresponding element of $\overline{\mathbf{v}}$. So the flux of reaction v_i is limited by the amount of enzyme given by g_i times its catalytic rate given by the corresponding turnover number $kCat_i$. This amount of enzyme takes up a proportion of total cell mass equal to g_i times eM_i [35].

Seen below in Equation 2.6 is the concrete implementation used by Røynestad [35]. It is an expansion of the standard FBA formulation seen in Equation 2.4.

$$\begin{bmatrix} \overline{\overline{\mathbf{S}}} & \overline{\overline{\mathbf{0}}} \\ \overline{\overline{\mathbf{eRx}}} & \overline{\overline{\textbf{-kCat}}} \\ \overline{\overline{\mathbf{eRx}}} & \overline{\overline{\mathbf{kCat}}} \\ \overline{\mathbf{0}} & \overline{\mathbf{eM}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{v}} \\ \overline{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{b}} \\ \leq \overline{\mathbf{0}} \\ \geq \overline{\mathbf{0}} \\ \leq M_c \end{bmatrix} \tag{2.6}$$

Equation 2.6, shows a matrix of sub-matrices, and two vectors of sub-vectors. The super-matrix is a $(m+2n+1)$-by$(2n)$ matrix where n is the number of reactions in the original FBA problem, and m is the number of metabolites. In the super-vector following it, the vector $\overline{\mathbf{g}}$ contains the amount of the enzymes corresponding to the reactions that $\overline{\mathbf{v}}$ gives the fluxes of. The $\overline{\overline{\mathbf{eRx}}}$ matrices in the super-matrix ties the values of the $\overline{\mathbf{v}}$ vector to the values of the $\overline{\mathbf{g}}$ vector. They are n-by-n matrices, where n is still the number of reactions in the model. It contains a single element of 1 on each row/column, and could therefore be turned into an n-by-n identity matrix with an appropriate interchanging of rows/columns. The matrix $\overline{\overline{\mathbf{kCat}}}$ contains one non-zero element in each column, which gives the k_{cat} value of the corresponding enzyme. Likewise, the length n vector $\overline{\mathbf{eM}}$ contains the masses of the corresponding enzymes.

The "equality" constraints in this case are not all strict equality constraints, but a mixture of the equality constraints from the original FBA formulation and a set of new (non-strict) inequality constraints, signified by the usage of "≤" and "≥" in front of the entries in the right-hand vector of subvectors.

As can be seen from Equation 2.6, when the S-matrix is multiplied with the vector containing the fluxes $\overline{\mathbf{v}}$ and the enzyme amounts $\overline{\mathbf{g}}$, the $\overline{\overline{\mathbf{eRx}}}$ matrices and inequality constraints will limit the reaction fluxes to the catalytic capacity of the amount of corresponding enzyme present, which in turn will be limited by the proportion of its mass the cell can devote to metabolic enzymes

[1, 35]. This cell mass proportion can vary between organisms, and a mix of investigation into the proportion of cellular protein devoted to metabolism and trial-and-error seems to be the best approach for now [1].

While it requires far more knowledge about the metabolome, i.e. the mass and kinetic parameters, than regular FBA, ircFBA can help predict metabolic behavior in media without detailed measurements of uptake fluxes in those media. Its success in predicting growth rates on a diverse set of media also points to physical constraints on the concentration of enzymes within cells being an important bound on microbial growth rates [1].

Key weaknesses of the method as it currently stands is the relatively low proportion of metabolic enzymes whose mass and/or turnover numbers are known and the uncertainty tied to their *in vivo* catalytic rate versus the *in vitro* measurements. This can be dealt with by, for instance, assuming the unknown values are all equal to the median of the known ones, which introduces its own set of problems, as few of these values will be correct. Methods do exist that help ameliorate or circumvent this issue, however, with one such method being presented below in Section 2.5 [1, 35]. Furthermore, it seems reasonable to assume that the accuracy of such methods will increase as better techniques for measurement and approximation of microbial *in vivo* catalytic rates are developed [11].

2.5 SP algorithm

Many different factors can affect the behavior of enzymes *in vivo* [30]. When seeking to build models that incorporate the catalytic rates of enzymes, it can therefore be difficult to lean on *in vitro* measurement data alone. The quality of assays may vary, and some may have been performed under physiologically unreasonable conditions. While k_{cat} values in the cell and in the lab certainly appear to correlate [11], a relative deviance of an order of magnitude or more can needless to say have a great impact on model predictions. Due to the scarcity of enzyme kinetic data available [11], retrieving known k_{cat} values from the same enzyme in related organisms may be the best option, further compounding the issue of accuracy [11].

One method created to help improve the accuracy of models using kinetic parameters in lieu of accurate data is the SP algorithm, developed by Røynestad and Almaas [35]. This algorithm is based on the titular SP, or shadow prices (briefly touched upon in Section 2.2), found for the various constraints as part of the standard FBA solution. As internally constrained FBA tends to produce lower growth rates than those seen in experimental setups, it seems reasonable to assume that at least some of the kinetic constraints introduced are too strict. The SP algorithm takes in the ircFBA model, the regular FBA model it's derived from, and the experimentally-observed growth rate of the organism in the relevant medium. It then iteratively solves the ircFBA problem, looks at the growth rate and the shadow prices of the kinetic constraints, and uses this to increase the turnover number of a single enzyme at a time. The turnover number to be increased is chosen based on which one will have to be increased the least to increase the objective the most. The goal is to achieve the target growth rate by changing the the set of turnover numbers as little as possible [35].

Chapter 3

Methods and Software

3.1 Matlab

All programming, computation, and plotting in this project was performed within Matlab (version R2015b), a proprietary computing environment and programming language, developed by MathWorks.

3.2 COBRA Toolbox

The COBRA Toolbox is the Matlab implementation of the open-source community-developed code base for COnstraint-Based Reconstruction and Analysis. It provides tools for managing and solving FBA models, and version 2.0.6 was used for solving the FBA problems in this project [36].

3.3 Webplotdigitizer

The experimental data from the article "Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110" by Varma and Palsson [38] was key in assessing the accuracy of the dircFBA model's predictions. It was, however, not listed explicitly except in the form of points on plots, and so needed to be read off of them. Used for this was the Webplotdigitizer software, available in browser-based format online. The model predictions by Mahadevan et al. [26] was also retrieved this way.

Below is listed the citation information from the website. It is listed here as the bibliography format used in this thesis is not well suited for it.

Author: Ankit Rohatgi
Title: WebPlotDigitizer
Website: http://arohatgi.info/WebPlotDigitizer
Version: 3.11
Date: January, 2017
E-Mail: ankitrohatgi@hotmail.com
Location: Austin, Texas, USA

3.4 Retrieval of turnover numbers and enzyme masses

The turnover numbers used in this project were retrieved from the supplementary materials of the article "Prediction of Microbial Growth Rate versus Biomass Yield by a Metabolic Network with Kinetic Parameters" by Adadi et al. [1]. Using the EC numbers also listed for the enzymes here, the masses of the enzymes were retrieved from a dataset on *E. coli* downloaded from BRENDA (The Comprehensive Enzyme Information System) on the 8th of September 2016.

The missing values for turnover numbers and masses were assumed to be equal to the average of the ones available. This accounted to 1933 (80.95%) of the turnover numbers, and 2023 (84.72%) of the mass numbers.

3.5 Preparing a genome-scale model for dFBA

In their original implementation of dFBA, Mahadevan et al. [26] operated with a metabolic network of *E. coli* consisting of 85 reactions and 54 metabolites. This was further simplified into a model containing four reactions and three metabolites (not counting the biomass objective) in order to demonstrate the principles of the method [26]. When attempting to implement this method for the complete iAF1260b *E. coli* model [13], consisting of 2388 reactions (including the biomass dummy function) and 1668 metabolites [13] as part of this project, some additional considerations had to be made even before the internal constraints were added.

The first, and most obvious hurdle, was the sheer size of the model. Finding suitable $\dot{v}_{max}$ values for 2388 reactions is no simple task, and would likely require an extensive literature review followed by rigorous and exhaustive experimentation. Of course, an attempt could be made to find logical "bottlenecks" and only constrain the $\dot{v}_{max}$ of these, but this would still require extensive research, and quite some trial-and-error. However, since internal constraints were to be implemented already, this problem was circumvented by simply setting an upper bound on how large a proportion of its mass the cell could alter per unit of time, called Δg (the implementation is thoroughly explained below, in Section 3.6). This makes the assumption that every enzyme can be produced and degraded at roughly the same rate, which is generally not the case [30]. Still, it reduces the number of tuning variables to one, which arguably makes it better for modeling purposes, at least in the case of this project.

Another hurdle was the ATP maintenance function. Several interesting situations to analyze with dFBA involves an environment starved for particular nutrients, for example due to environmental fluctuations or exhaustive uptake by the model organism. As the ATP maintenance function is an energy-consuming reaction with a positive lower bound, meant to force a certain level of flux through it and simulate the energetic requirements of metabolism and biomass production, it risks making the FBA problem infeasible at certain frames where the organism starves. In some cases, as when the environment is fully depleted, simply setting the growth rate to 0 for the rest of the program's runtime, allowing the cumulative biomass production and composition to literally flatline, produces a satisfactory solution. However, when the organism starves temporarily as it attempts to adjust to a rapidly changing environment, this is not the case. In order to handle these cases, functionality was implemented to find a lower value for the ATP maintenance bound. This was implemented as a binary search going for 10 steps, finding the highest value between zero and the original ATP maintenance requirement that would still allow the problem to be feasible, down to an accuracy of $\frac{1}{2^{10}}$ or "1/1024th" of the original value. Of course, there are other possible ways of finding permissible lower bounds on this reaction, but this one seemed the easiest to implement, is scalable to an arbitrary level of accuracy, and is not obviously slower than any other method that was considered.

3.6 dircFBA

Based on the methods of dFBA (static optimization-based) [38, 26] and ircFBA (Røynestad's implementation) [1, 35], dynamic internally constrained FBA attempts to capture the dynamic behavior of a genome-scale metabolic model with internal constraints; that is, a dynamic model in a changing environment where both its metabolism and the changes in its metabolism are constrained by the efficiency of its enzymes relative to their mass. Used for the project is the iAF1260b model of *E. coli* [13].

3.6.1 Mathematical formulation

The model formulation is based on that presented by Røynestad in his master thesis [35], which in turn is based on the work of Adadi et al. with their MOMENT method [1]. The model has been expanded with additional variables, filling two vectors dubbed $\overline{\Delta\mathbf{g}_+}$ and $\overline{\Delta\mathbf{g}_-}$. They are both of length n, n being the number of reactions present in the original FBA model, and they represent the increase and decrease (respectively) in the amount of corresponding enzyme. These variables are subject to mass constraints in the same way the variables of the $\overline{\mathbf{g}}$ vector are in the ircFBA problem, but these constrain them to a far smaller value, being equal to the amount of enzyme the cell is allowed to replace each hour, multiplied by the fraction an interval makes up of an hour.

At each interval, the solution from the last interval needs to be propagated onwards. This is done by constraining the vector containing enzyme amounts to being equal to the same vector at the last interval, plus the vector containing increases in enzyme amounts, minus the vector containing decreases in enzyme amounts. That is, $\overline{\mathbf{g}}(t) = \overline{\mathbf{g}}(t-1) + \overline{\Delta\mathbf{g}_+}(t-1) - \overline{\Delta\mathbf{g}_-}(t-1)$.

$$
\begin{aligned}
\text{maximize} \quad & \overline{\mathbf{c}}^T\overline{\mathbf{v}} \\
\text{subject to} \quad & \overline{\overline{\mathbf{S}}}\overline{\mathbf{v}} = \overline{\mathbf{b}} \\
& \overline{\mathbf{lb}} \le \overline{\mathbf{v}} \le \overline{\mathbf{ub}} \\
& -\overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\mathbf{g}} - \overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\Delta\mathbf{g}_+} + \overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\Delta\mathbf{g}_-} \le \overline{\mathbf{v}} \le \overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\mathbf{g}} + \overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\Delta\mathbf{g}_+} - \overline{\overline{k_{\mathbf{cat}}}} \cdot \overline{\Delta\mathbf{g}_-} \\
& \overline{\mathbf{g}}^T\overline{\mathbf{eM}} + \overline{\Delta\mathbf{g}_+}^T\overline{\mathbf{eM}} - \overline{\Delta\mathbf{g}_-}^T\overline{\mathbf{eM}} \le M_c \\
& \overline{\Delta\mathbf{g}_+}^T\overline{\mathbf{eM}} \le M_{\Delta c} \\
& \overline{\Delta\mathbf{g}_-}^T\overline{\mathbf{eM}} \le M_{\Delta c} \\
& \overline{\mathbf{g}}(t) = \overline{\mathbf{g}}(t-1) + \overline{\Delta\mathbf{g}_+}(t-1) - \overline{\Delta\mathbf{g}_-}(t-1) \\
& \overline{\mathbf{g}} \ge \overline{\mathbf{0}} \\
& \overline{\Delta\mathbf{g}_+} \ge \overline{\mathbf{0}} \\
& \overline{\Delta\mathbf{g}_-} \ge \overline{\mathbf{0}}
\end{aligned}
\tag{3.1}
$$

(Above, ⊙ means "elementwise multiplication", and the " = ", " ≤ ", and ≥ operators are meant in an element-wise fashion.)

3.6.2 Computational implementation

In the implementation itself, seen in Equation 3.2, elements have been added to the "stoichiometric" matrix tying the new flux variables to the old ones and representing these enzymes' mass, in a way much similar to the way the ircFBA model is constructed. The change in enzymatic composition within the cell is then limited by a similar fractional less-than-or-equal-to constraint, while the vector denoting enzyme amounts is constrained to be equal to the same vector from the last interval, plus the changes found in the last interval.

$$\begin{bmatrix} \overline{\overline{\mathbf{S}}} & \overline{\overline{\mathbf{0}}} & \overline{\overline{\mathbf{0}}} & \overline{\overline{\mathbf{0}}} \\ \overline{\overline{\mathbf{eRx}}} & \overline{\overline{\text{-}k_{\mathbf{cat}}}} & \overline{\overline{\text{-}k_{\mathbf{cat}}}} & \overline{\overline{k_{\mathbf{cat}}}} \\ \overline{\overline{\mathbf{eRx}}} & \overline{\overline{k_{\mathbf{cat}}}} & \overline{\overline{k_{\mathbf{cat}}}} & \overline{\overline{\text{-}k_{\mathbf{cat}}}} \\ \overline{\mathbf{0}} & \overline{\mathbf{eM}} & \overline{\mathbf{eM}} & \overline{\mathbf{\text{-}eM}} \\ \overline{\overline{\mathbf{0}}} & \overline{\overline{\mathbf{eRx}}} & \overline{\overline{\mathbf{0}}} & \overline{\overline{\mathbf{0}}} \\ \overline{\mathbf{0}} & \overline{\mathbf{0}} & \overline{\mathbf{eM}} & \overline{\mathbf{0}} \\ \overline{\mathbf{0}} & \overline{\mathbf{0}} & \overline{\mathbf{0}} & \overline{\mathbf{eM}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{v}}(t) \\ \overline{\mathbf{g}}(t) \\ \overline{\Delta\mathbf{g}_{+}}(t) \\ \overline{\Delta\mathbf{g}_{-}}(t) \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{0}} \\ \leq \overline{\mathbf{0}} \\ \geq \overline{\mathbf{0}} \\ \leq M_c \\ \overline{\mathbf{g}}(t-1) + \overline{\Delta\mathbf{g}_{+}}(t-1) - \overline{\Delta\mathbf{g}_{-}}(t-1) \\ \leq M_{\Delta c} \\ \leq M_{\Delta c} \end{bmatrix} \tag{3.2}$$

On solution, at each time interval, as explained above and shown in Equation 3.2, the new enzyme composition (that is, $\overline{\mathbf{g}} + \overline{\Delta\mathbf{g}_{+}} - \overline{\Delta\mathbf{g}_{-}}$) is checked against the normal mass proportion constraint, while the $\overline{\mathbf{g}}$ vector giving "starting" enzyme composition is constrained to be equal to the sum from the last interval. This essentially updates and locks $\overline{\mathbf{g}}$ at each interval, before finding the change in enzyme composition that will maximize growth. In order to minimize "noise" in the Δg terms when no composition change is needed (noise in the form of $\Delta g_{+,i} = \Delta g_{-,i} > 0$), a tiny negative weighting term was added to the $\overline{\Delta\mathbf{g}_{+}}$ values in the optimization objective.

3.6.3 Environment

The simulation environment is constructed similarly to the static optimization approach used by Mahadevan et al. in [26], with the entire simulation (a 10 hour period by default) run being divided into a number of intervals (10,000 by default), with the problem being re-formulated and re-solved at each interval. The cumulative biomass production is summed up over these intervals, with the concentration of biomass per liter being increased by the current biomass times the growth rate times the duration of the interval. The mathematical formulation is given below in 3.3, where $X(t)$ is the accumulated biomass (dry weight) at time t.

$$X(t) = X(t-1)\overline{\mathbf{c}}^{T}\overline{\mathbf{v}}\Delta t \tag{3.3}$$

The availability of oxygen is modeled. It is consumed by the growing cell population, and restored by being absorbed into the liquid medium at a rate proportional to the differing concentration between the solution O^{sol} the gas phase O^{ambient} (i.e. surrounding air) and the mass transfer coefficient $\text{k}_\text{L}\text{a}$. That gives Equation 3.4 seen below for updating the amount of oxygen in solution.

$$O^{\text{sol}}(t) = O^{\text{sol}}(t-1) - \Delta t \cdot v^{\text{ox}} \cdot X(t-1) + 7.5 \cdot \Delta t (O^{\text{ambient}} - O^{\text{sol}}(t-1)) \tag{3.4}$$

All oxygen in the solution is assumed to be freely available (i.e. no Michaelis-Menten dynamics), as it was assumed in the dFBA article [26]. The upper bound on the oxygen uptake (i.e. the lower bound on the export reaction) is set to be the smallest of the absolute oxygen uptake limit and the amount of oxygen remaining in solution divided by the amount of cell dry weight times the duration of the interval. Shown in Equation 3.5.

$$O^{\text{available}}(t) = \min(v^{\text{ox}}_{\max}, \frac{O^{\text{sol}}(t)}{X(t) \cdot \Delta t}) \tag{3.5}$$

Carbon sources in the environment are also depleted from (and excreted into, in the case of acetate) it as the model organism grows, though micronutrients are assumed to be in abundance. Their uptake is subject to, and therefore constrained by, Michaelis-Menten dynamics. In Equation 3.6, $v_{\max}(t)$ is the upper bound on the uptake reaction at interval t. $V_{\max}$ is the maximum allowed rate of the reaction. K_M is the concentration of substrate at which the reaction rate is half its maximum rate. And $[S]_t$ is the concentration of the relevant metabolite in the solution at time t. To avoid negative nutrient concentrations, a physical impossibility, the minimum is taken between the value suggested by the Michaelis-Menten dynamics and the concentration of nutrient available divided by the biomass (which is treated as gDW/l) times the duration of the time interval.

$$v_{\max}(t) = \min(V_{\max} \frac{[S]_t}{K_M + [S]_t}, \frac{[S]_t}{X(t) \cdot \Delta t}) \tag{3.6}$$

Chapter 4

Results and Discussion

4.1 Implementing and tuning dircFBA for aerobic batch growth on glucose

In this section, the environmental parameters from Varma and Palsson's original experiments are retrieved and estimated. The model then attempts to predict the results of their aerobic batch run on 10.8 mM glucose. The dircFBA model is first run without internal global constraints, then the M_c and $M_{\Delta c}$ constraints are added in turn, and the ensuing results are presented and discussed. Comparisons are made with the predictions of dFBA as separately implemented by Mahadevan et al. [26] and Varma and Palsson [38]. Running dircFBA on a model the size of the iAF1260b *E. coli* model, with 2388 reactions and 1668 metabolites, for 10000 timesteps corresponding to 10 hours total, took a little less than an hour on a low-end personal laptop computer.

4.1.1 Environmental and exchange parameters

Before applying and experimenting with any kind of internal constraints, a starting point for the other model parameters had to be established: the initial population density X_0, the oxygen mass transfer coefficient k_La , and uptake bounds on glucose and oxygen. As Varma and Palsson [38] only provided the initial cell density implicitly in a plot, and this value is too small to be accurately read off of it, it requires some estimations to unveil. The initial cell density used by

Mahadevan et al. [26], i.e. $X_0 = 0.001$ was considered, but did not produce satisfactory results with the other model parameters as stated by Varma and Palsson [38]. Specifically, it appeared too low, as prediction plots are all shifted to the right relative to the experimental ones.

An estimate therefore had to be found for the starting cell density. To achieve this, the original results from the aerobic batch run performed by Varma and Palsson [38] were reverse-engineered. Figure 2 from their paper ("Stoichiometric Flux Balance Models Quantitatively Predict Growth and Metabolic By-Product Secretion in Wild-Type Escherichia coli W3110" [38]) contains a description citing a growth rate of $\mu = 0.68\ \mathrm{h}^{-1}$ on glucose (with an uptake bound of $10.5\ \mathrm{mM\,gDW^{-1}\,h^{-1}}$) under aerobic conditions (uptake bound on oxygen of $15\ \mathrm{mM\,gDW^{-1}\,h^{-1}}$). This coincides well with aerobic growth rates on glucose for *E. coli* cited in other literature [1]. The exponential growth curve $X_0e^{\mu t}$ was fitted to the experimental cell densities in the main glucose-utilizing exponential growth phase, as seen in Figure 4.1. The curve was fitted by minimizing the mean distance though a 100-step binary search though values for X_0 on the interval [0,0.01], yielding an estimate for a starting population density of $X_0 = 0.0043$.

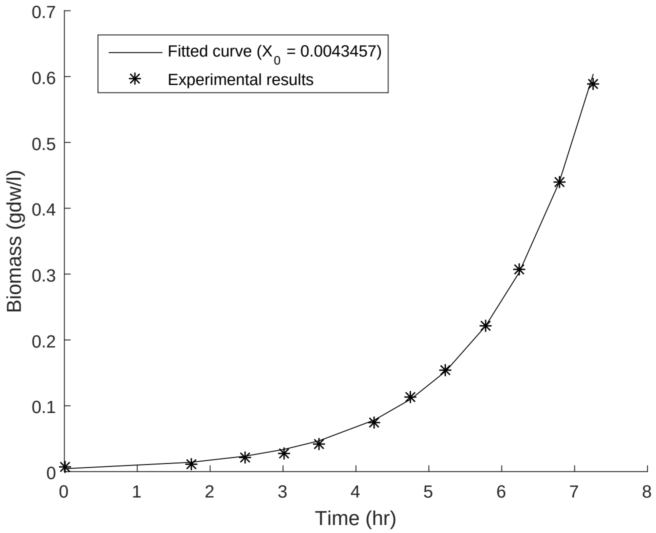

Figure 4.1: **Fitted curve, exponential population growth.** The curve $X_0e^{\mu t}$ is fitted to experimental data from aerobic batch run of Varma and Palsson [38] for a growth rate of $\mu = 0.682$. Seen above is the fitted curve, with a $X_0 \approx 0.0043$.

As stated in Section 3.6.3, a simple environmental model was constructed, similar to that used by Mahadevan et al. [26]. However, their assumed k_La value of 7.5 h^{-1} [26] cannot be correct according to the following claim from Varma and Palsson's article [38]: "A small bubble size that helped keep dissolved oxygen above 50% saturation in aerobic experiments was obtained."; meanwhile, in the dFBA simulation by Mahadevan et al., oxygen levels in solution drop to 0 at around 4 hours in [26]. A new estimate for the k_La parameter therefore had to be found. This was fairly straightforward, as the batch maintains an exponential growth rate appearing to fully utilize glucose and oxygen up to around a population density of 0.7 $gDWl^{-1}$, and the saturation of oxygen lies at around 0.21 mM. From this, a value for the k_La can be approximated according to Equation 4.1 by assuming a stationary point for oxygen in solution at 50% at that cell density. Therefore a k_La of 100 was used in the environmental model.

$$k_La(0.21-\frac{0.21}{2}) = 0.7\cdot 15 \Rightarrow k_La = \frac{0.7\cdot 15}{0.105} = 100 \tag{4.1}$$

For the uptake bound on oxygen, Varma and Palsson [38] and Mahadevan et al. [26] agree, but there is a small discrepancy seen in the bound on glucose uptake. This is a fairly minor difference, with the former determining it to be 10.5 $\text{mmol}\,\text{gDW}^{-1}\,\text{h}^{-1}$ and the latter 10 $\text{mmol}\,\text{gDW}^{-1}\,\text{h}^{-1}$. While both of these are higher than implied in some literature [24], the number from Varma and Palsson [38] was chosen, as their data was considered more reliable as they performed the actual experiment.

With sound estimates for the X_0 and k_La parameters determined, and uptake bounds on glucose and oxygen set, the model was run without internal or dynamic constraints. A significant secretion and later reuptake of pyruvate was observed, but was not mentioned in the original article by Varma and Palsson [38], and pyruvate is not thought to be secreted when feeding on glucose [18]. Therefore pyruvate secretion was turned off in the model before proceeding.

For all the plots comparing model predictions to experimental data for the aerobic batch run from Varma and Palsson [38], both "full-length" and "exponential phase" mean error values are included. The "full-length" mean error value gives the mean error between the prediction and all experimental data points. The "exponential phase" mean error value, however, gives the mean error between the prediction and the experimental data only during the phase of exponential growth. The mean error values listed by Varma and Palsson [38] are all of the "exponential phase"-kind.

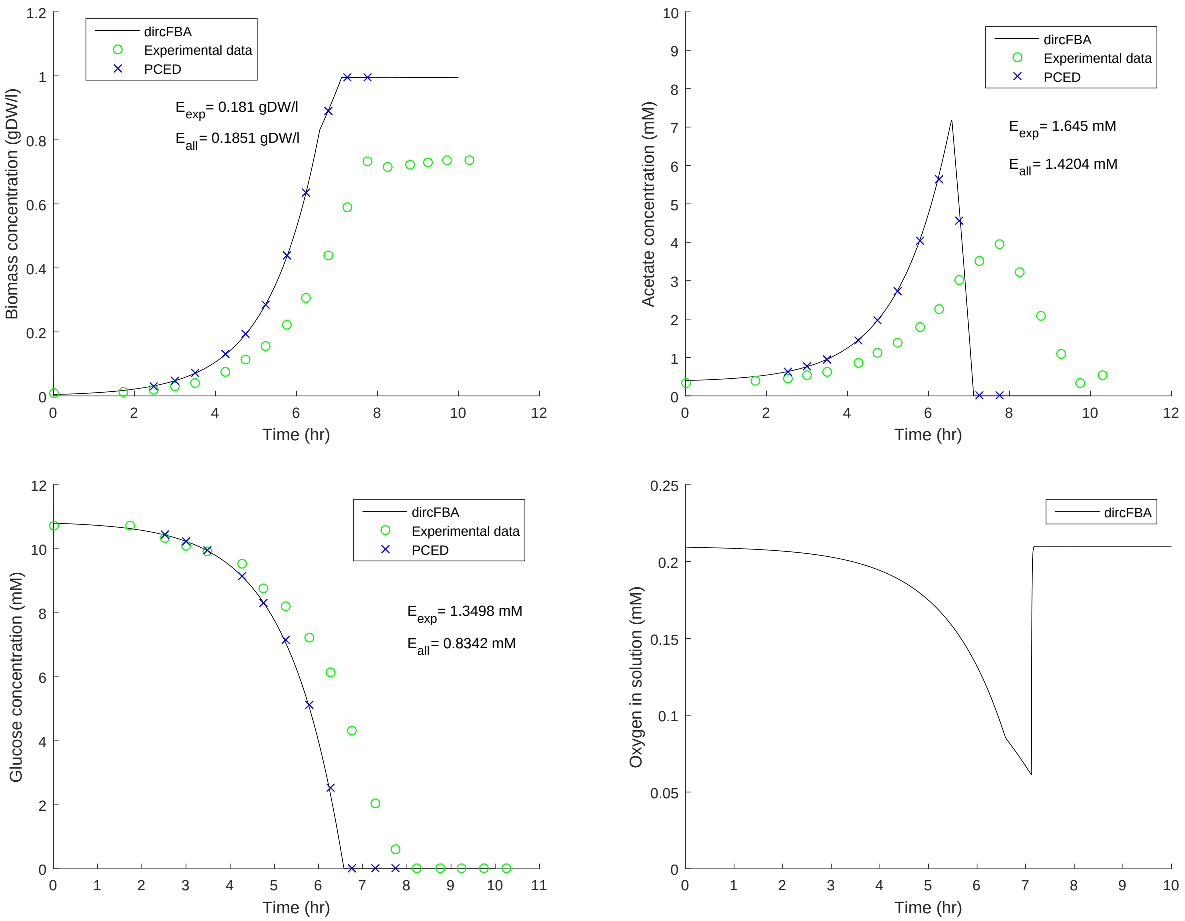

Figure 4.2: **dircFBA without internal constraints or acetate bounds (overestimates rates of acetate exchange).** Predictions of the concentrations of biomass, glucose, acetate, and oxygen in the medium for the dircFBA model. Here run with an uptake bound of 15 and 10.5 mmol gDW^{-1} h^{-1} for oxygen and glucose respectively, without internal constraints ($M_c = 100$ and $M_{\Delta c} = \infty$). Mean error of prediction is given both for "exponential phase" points compared to exponential data (PCED) and full-length run.

Plainly, Figure 4.2 shows a qualitative similarity between predictions and experimental data. The biomass curve, i.e. the cell density, increases much too fast, however, indicating that the growth rate is too high. The environmental acetate concentration spikes too early and then plummets, while glucose is also consumed too rapidly. The oxygen plot is included, showing that the concentration of oxygen dissolved in the simulated environment clearly dips well below 50% for around an hour, which it blatantly was stated not to do in the experiment. The mean error values are included, both for the full-length run and for the "exponential phase" only, but their high values need not be considered to judge this a poor prediction.

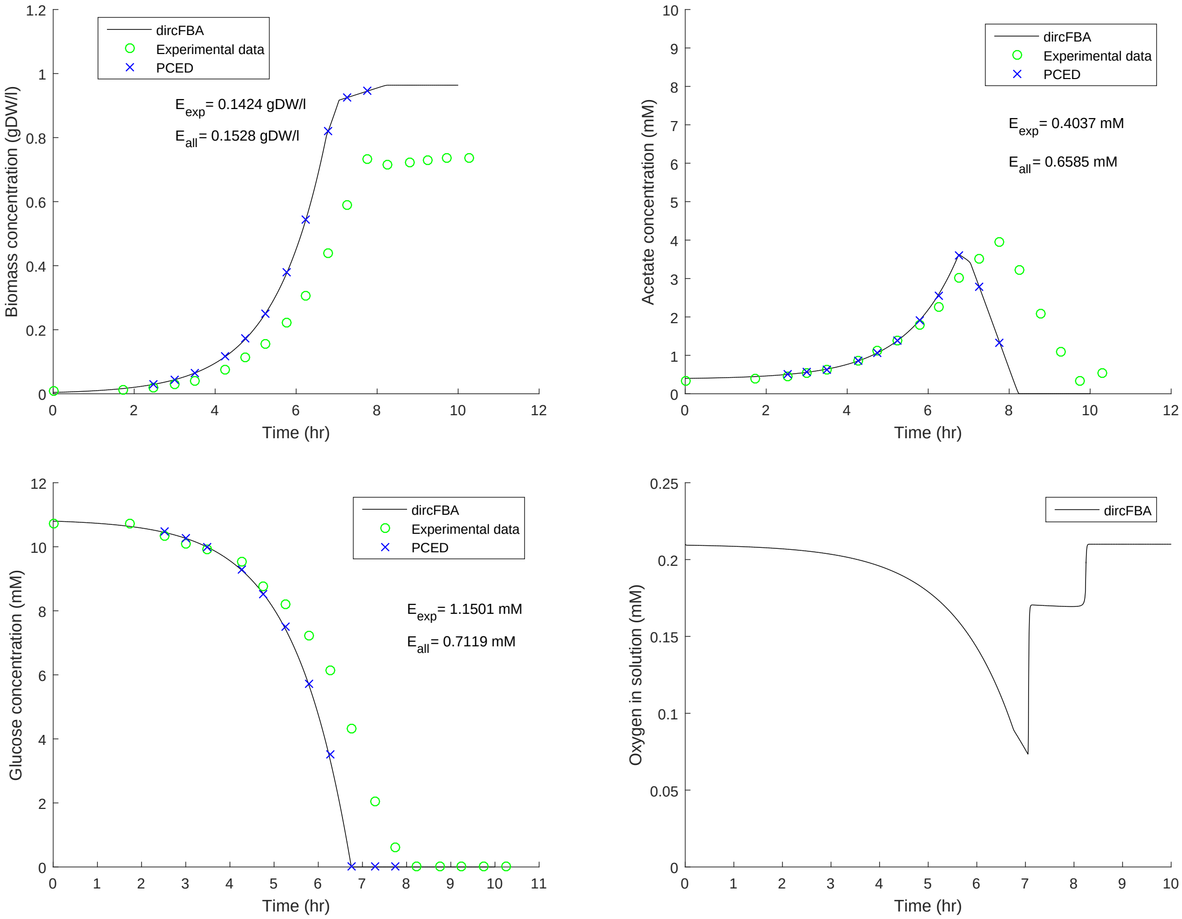

Figure 4.3: **dircFBA without internal constraints but with empirical acetate bounds (reasonably predicts acetate exchange).** Predictions of the concentrations of biomass, glucose, acetate, and oxygen in the medium for the dircFBA model. Here run with an uptake bound of 15 and 10.5 $\mathrm{mmol\,gDW^{-1}\,h^{-1}}$ for oxygen and glucose respectively, and an uptake and secretion bound of 3.1 $\mathrm{mmol\,gDW^{-1}\,h^{-1}}$ on acetate without internal global constraints ($M_c = 100$ and $M_{\Delta c} = \infty$). Mean error of prediction given both for "exponential phase" points compared to exponential data (PCED) and full-length run.

Noting that acetate was both produced and consumed far too rapidly compared to experimental data, a likely bound on these values were retrieved from literature ($\frac{0.183\ \mathrm{g\,gDW^{-1}\,h^{-1}}}{59.04\ \mathrm{g\,mol^{-1}}} \approx 3.1\ \mathrm{mmol\,gDW^{-1}\,h^{-1}}$) [24] and applied to the model. This resulted in a better prediction, which can be seen in Figure 4.3. Here, the growth rate still appears too high, but the acetate concentration in the medium peaks at a height similar to that seen in the experimental data. Glucose is still consumed too rapidly, and oxygen still dips below 50% for short duration. While slightly better than the

4.1.2 Adding internal and dynamic constraints

With the environment and basic FBA parameters set, the global internal constraint M_c, determining the total proportion of cell mass allocated to metabolic enzymes, was applied and tested. It was tested for values ranging from 0.4 to 0.1, and the k_{cat} set was tuned to a growth rate of $\mu = 0.68$ for each value. No dynamic constraint was added yet; i.e. the parameter $M_{\Delta c}$ was not enabled (effectively set to ∞).

The model did not appear particularly sensitive to the internal constraint M_c, as can be seen in Table 4.1; the range of values tested produced quite similar mean error values. The fact that it was present did markedly improve the model predictions, however, and so M_c was set to 0.3, as experimental evidence points to this number [1]. Also seen in Table 4.1 are the $\Delta_{\text{max}}\text{k}_{\text{cat}}$ values, a term here used to describe the largest relative increase in turnover number resulting from the tuning. That is, the k_{cat} value after tuning divided by the k_{cat} value before tuning. Interestingly, in each case, only one k_{cat} value received an increase in the order of magnitude listed for the different $\Delta_{\text{max}}\text{k}_{\text{cat}}$ in the table, and in each case this ketol-acid reductoisomerase. This was not assumed average during the retrieval of kinetic parameters, and thus the value listed in literature for this enzyme may be particularly inaccurate with respect to its catalytic rate *in vivo.* Plots showing the relative changes to turnover numbers can be seen in Appendix B, Figure B.1.

Table 4.1: **Comparison of mean error values for $\mathbf{M_c}$ values.** The mean error values for different values of mass proportion devoted to enzymes, and the biggest relative increase in k_{cat} values required during tuning, denoted $\Delta_{\text{max}}\text{k}_{\text{cat}}$. No dynamic constraints are applied (i.e. $\text{M}_{\Delta\text{c}} = \infty$).

M_c	$\text{E}_{\text{exp}}^{\text{Gluc}}$	$\text{E}_{\text{all}}^{\text{Gluc}}$	$\text{E}_{\text{exp}}^{\text{Biomass}}$	$\text{E}_{\text{all}}^{\text{Biomass}}$	$\text{E}_{\text{exp}}^{\text{Acet}}$	$\text{E}_{\text{all}}^{\text{Acet}}$	$\Delta_{\text{max}}\text{k}_{\text{cat}}$
0.4	0.2817	0.1804	0.0058	0.0106	0.3383	0.4695	67.02
0.3	0.2817	0.1804	0.0058	0.0106	0.3383	0.4696	149.03
0.2	0.2817	0.1803	0.0058	0.0106	0.3382	0.4695	185.94
0.1	0.2816	0.1803	0.0058	0.0105	0.3382	0.4694	473.54

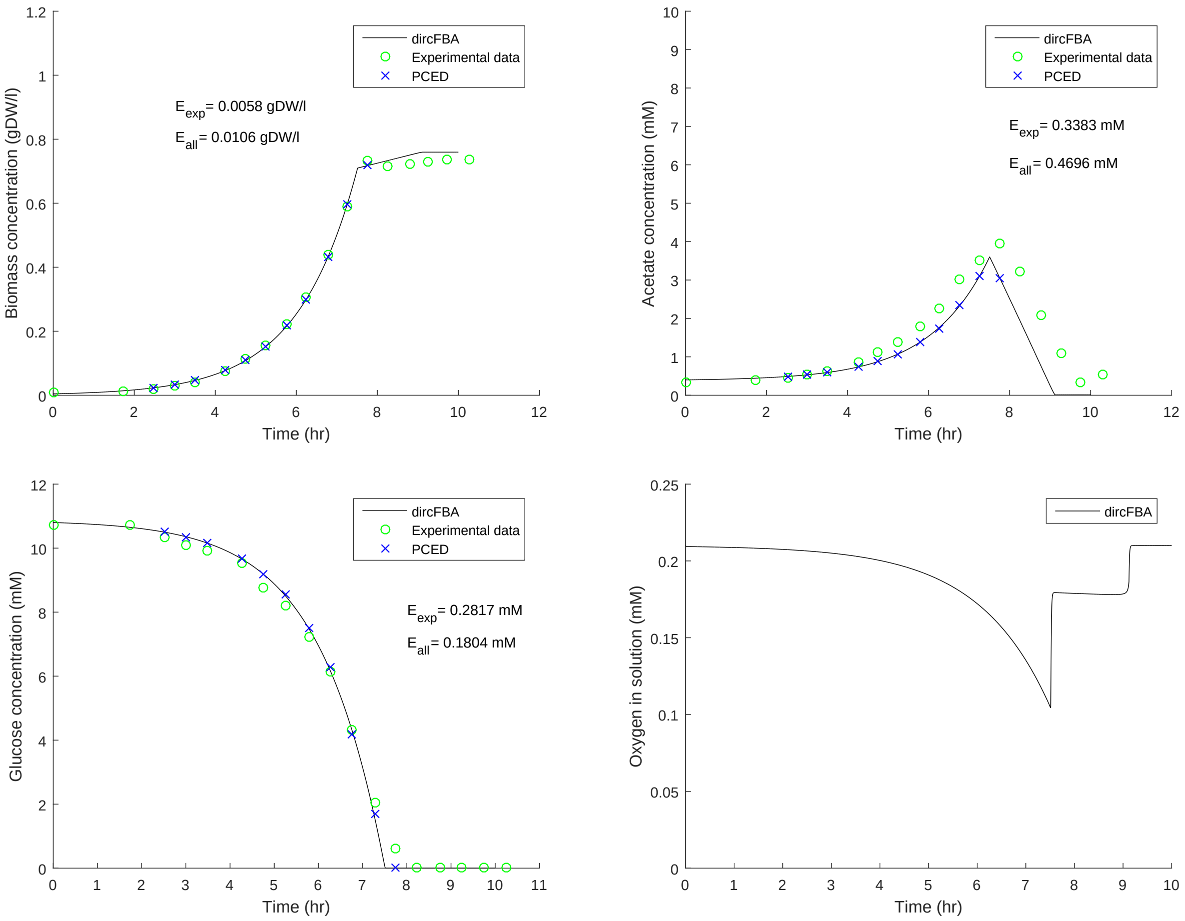

Figure 4.4: **dircFBA run with $\mathbf{M_c}$ = 0.3 and $\mathbf{M_{\Delta c}} = \infty$.** Predictions of the concentrations of biomass, glucose, acetate and oxygen in the medium for the dircFBA model. Here run with an uptake bound of 15 and 10.5 mmol gDW^{-1} h^{-1} for oxygen and glucose respectively, and an uptake and secretion bound of 3.1 mmol gDW^{-1} h^{-1} on acetate. Internal global constraint on total mass proportion devoted to metabolic enzymes, i.e. M_c is set to 0.3, but no dynamic internal constraints are applied ($M_{\Delta c} = \infty$). Mean error of prediction given both for "exponential phase" points compared to exponential data (PCED) and full-length run.

Plots showing dircFBA model predictions with $M_c = 0.3$ and $M_{\Delta c} = \infty$ can be seen in Figure 4.4, and the predictions appear quite good. The biomass barely deviates from experimental measurements during the exponential growth phase on glucose, with a mean error for this region ($E_{exp}^{Biomass}$) of only 0.0058, versus Varma and Palsson's 0.024 [38]. The prediction for terminal biomass is also remarkably close. The prediction of acetate concentration is not quite as good, but remains close to experimental measurements throughout most of the run's length, peaking at a similar place and with a similar height. The mean error during the exponential phase (E_{exp}^{Acet}) is also 0.3383, a marginal improvement on Varma and Palsson's 0.36 [38]. Predictions of glucose concentration appears to follow experimental results nicely, with a mean error of 0.2817 for the exponential growth phase (E_{exp}^{Gluc}); this is not quite as good as Varma and Palsson's model's 0.27 [38]. Finally, solution levels of oxygen do not appear to dip below 50%.

Next, the dynamic constraint, implemented in the same style as the internal constraints (as described in Section 3.6), was turned on. No average value for the speed of protein mass reallocation could be retrieved from literature, and so values ranging over four orders of magnitude were tried, to evaluate how the effects on the model differed. As can be seen in Table 4.2, the effects of the $M_{\Delta c}$ constraint on the glucose mean error are negligible. This is likely due to fact that the glucose uptake remains constrained only by the M_c constraint thoughout the initial period of exponential growth where glucose is utilized. Only when the metabolism switches away from glucose as its primary carbon source does the dynamic constraint come into effect. Smaller values of $M_{\Delta c}$ result in a smaller mean error for the full run for the biomass (E_{all}^{Gluc}), with a constraint of 0.001 h^{-1} resulting in the smallest mean error for biomass for the tested values. Likely, this stems from the fact that it reduces overall biomass production by allowing less efficient utilization of resources, meaning more time passes during which a larger proportion of absorbed nutrients go towards upkeep of non-growth associated maintenance. This in turn means a lower amount of biomass produced overall, and therefore a more "accurate" prediction during the lag phase, where experimentally there is hardly any growth happening at all. The same trend is seen for the acetate mean error, for similar reasons; the chosen bound for acetate uptake appears a bit on the high side, so a slower adjustment to uptake of acetate will lead to a smaller prediction error overall.

Table 4.2: **Comparison of mean error values for values of $M_{\Delta c}$.** The mean error values for different values of dynamic internal constraints, i.e. $\mathbf{M_{\Delta c}}$, and the mean error values for the same plots as given in Varma and Palsson [38]. Global internal constraint on mass fraction occupied by metabolic enzymes, i.e. M_c, is 0.3

$M_{\Delta c}$	E_{exp}^{Gluc}	E_{all}^{Gluc}	$E_{exp}^{Biomass}$	$E_{all}^{Biomass}$	E_{exp}^{Acet}	E_{all}^{Acet}
1	0.2817	0.1804	0.0058	0.0106	0.3382	0.4694
0.1	0.2817	0.1804	0.0058	0.0105	0.3375	0.4699
0.01	0.2817	0.1804	0.0059	0.0104	0.3363	0.4675
0.001	0.2817	0.1804	0.0061	0.0092	0.3343	0.4610
V&P	0.27	N/A	0.024	N/A	0.36	N/A

Therefore, while prediction errors drop monotonically for smaller and smaller values of $M_{\Delta c}$, none of the four values tested and shown in Table 4.2 truly appear to dominate the others. The value of 0.01 h^{-1} was selected for closer inspection and usage for the rest of the runs, as it provided a good scaling of the composition change plots (discussed below and seen in Figure 4.7). The population density and environmental concentrations for the aerobic batch run with $M_c = 03$ and $M_{\Delta c} = 001$ h^{-1} can be seen in Figure 4.5.

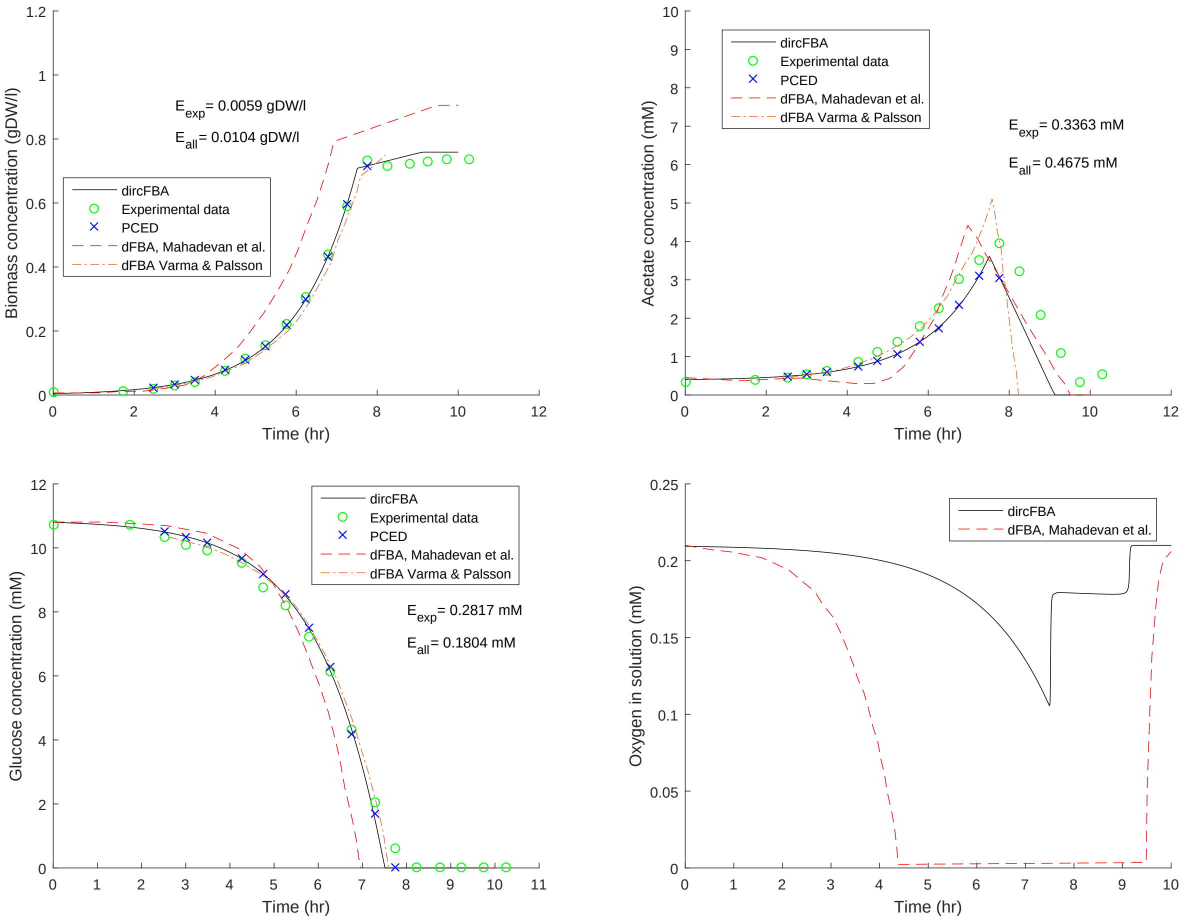

Figure 4.5: **dircFBA run with M_c = 0.3 and $M_{\Delta c}$ = 0.01** h^{-1}**.** Predictions of the concentrations of biomass, glucose, acetate, and oxygen in the medium for the dircFBA model. Here run with an uptake bound of 15 and 10.5 mmol gDW^{-1} h^{-1} for oxygen and glucose respectively, and an uptake and secretion bound of 3.1 mmol gDW^{-1} h^{-1} on acetate. Global internal constraint on total mass proportion devoted to metabolic enzymes, i.e. M_c, is set to 0.3, and dynamic constraints $M_{\Delta c}$ allow replacement of 0.01, that is 1%, per hour. Mean error of prediction given both for "exponential phase" points compared to exponential data (PCED) and full-length run. Included are also the predictions made by dFBA as implemented by Varma and Palsson [38] and Mahadevan et al. [26] in their respective articles.

Simple inspection of Figure 4.5 makes it evident that the implementation of dircFBA here offers a far better prediction than the implementation of dFBA by Mahadevan et al. [26], and a slightly better prediction that the implementation by Varma and Palsson [38]. The prediction by dircFBA matches the experimental results more accurately overall, and is the only method that comes close in predicting terminal biomass concentration. The comparisons with Varma and Palsson [38] in Table 4.2 shows the mean error values for their implementation of dFBA and those for the implementation of dircFBA used in this project. The overall lower mean error values for dircFBA supports the notion that it offers an increase in prediction accuracy.

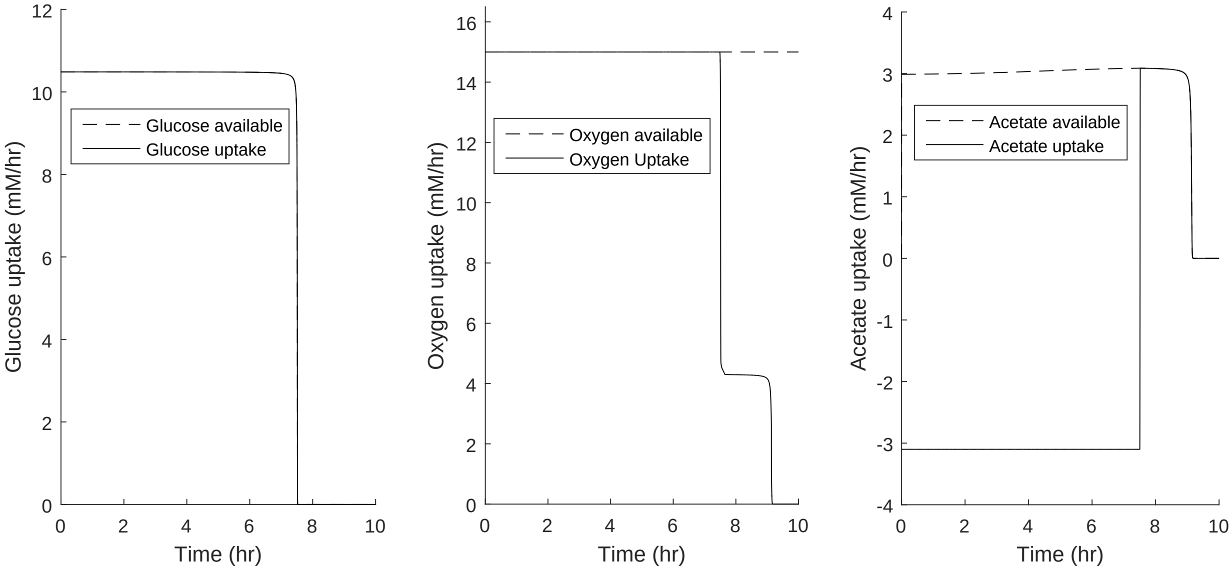

Figure 4.6: **Uptakes predicted by dircFBA with $\mathbf{M_c}$ = 0.3 and $\mathbf{M_{\Delta c}}$ = 0.01** h^{-1} Uptakes of glucose, oxygen, and acetate, with constraints imposed by kinetics and availability marked. Model parameters are the same as in Figure 4.5.

.

Plotted in Figure 4.6 are the uptake rates for glucose, oxygen, and acetate for the dircFBA model with $M_c = 0.3$ and $M_{\Delta c} = 0.01$, along with the bounds on their uptakes. The amount of available glucose can be seen to be growth-limiting throughout the duration of the batch run. The amount of available oxygen is growth-limiting while there is glucose available. Acetate availability is only limiting towards the end of the run, when it becomes the primary source of carbon and energy.

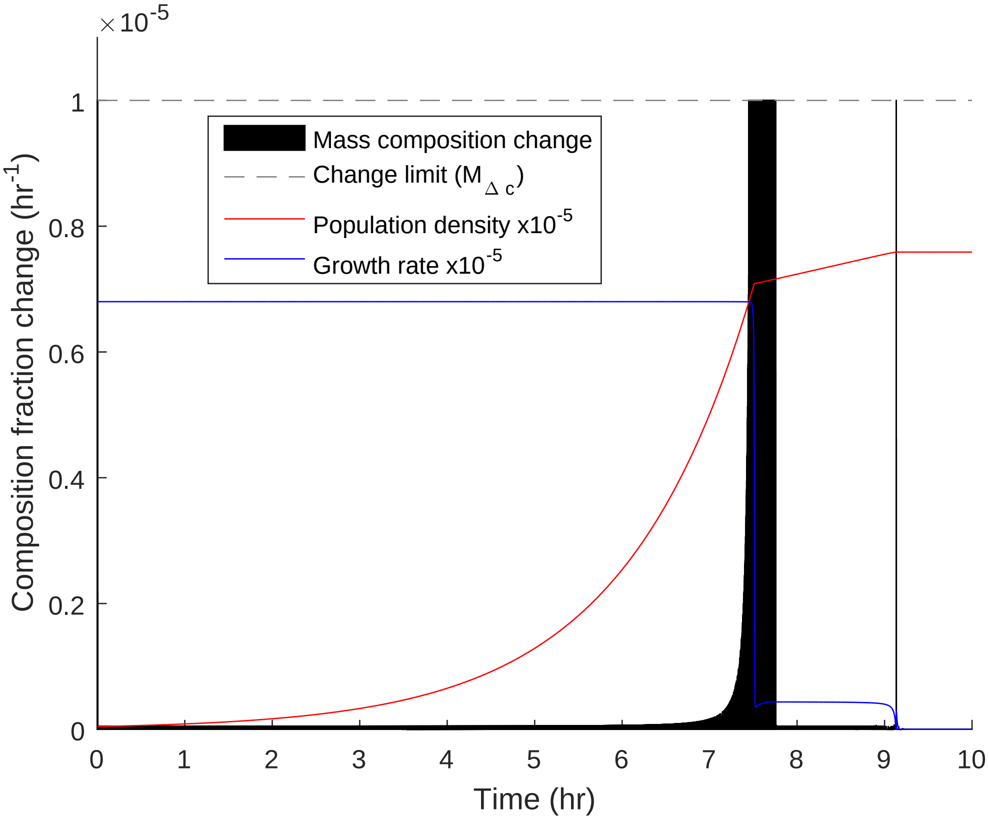

Figure 4.7: **Composition change with $\mathbf{M_{\Delta c}}$ constraint** The fraction of total mass worth of enzyme being replaced at each interval in the dFBA model for the aerobic batch run with $M_c = 0.3$ and $M_{\Delta c} = 0.01$, with the $M_{\Delta c}$ bound seen at the top. Along with the mass composition change is plotted the scaled-down cell density and growth rate to help illustrate at which point during the batch run the enzymatical reallocation takes place, and the $M_{\Delta c}$ constraint becomes limiting. Model parameters are the same as in Figure 4.5.

In order to investigate how the cell must adjust and reallocate its enzyme composition in the face of a changing environment, a bar plot was made showing the fraction of total mass worth of enzyme mass being replaced at each interval, and the constraint placed on this fraction. This can be seen in Figure 4.7 During most of the run, the model appears to be making some low numerical noise. Towards the end of the run, however, the cell model needs to rapidly adjust to the sinking availability of glucose, the rising availability of oxygen and the utilization of acetate, and the model can be seen to be constantly pushing against the constraint on change in mass composition for a duration of about half an hour. The growth rate and cell density are plotted along with the bar plots showing mass reallocation to help illustrate at which points during the growth curve there is significant reallocation activity and the constraint is limiting.

As glucose availability becomes acutely limiting, the growth rate can be seen to take a steep dive, before rising slightly as the metabolic rearrangement allows for utilization of acetate. The growth rate then remains steady until the acetate is exhausted, at which point another spike in enzymatical rearrangement signals a complete flatline in growth rate.

4.1.3 Simple bounding of growth rate

While dircFBA appears to predict the experimental results fairly accurately, a consideration must be made:. The SP algorithm, developed by Røynestad and Almaas [35] and described in Section 2.5, tunes k_{cat} values to permit higher growth rates than the *in vitro* values reported in literature allow. This is its express purpose, and one could potentially conceive of many different algorithms that could be used to achieve similar results, another one of which has also been formulated and implemented in Røynestad's thesis [35]. When a model's k_{cat} set is tuned for a given environment, the upper bound on growth in that environment will reach precisely the value it's tuned for, given that the availability of nutrients can actually support it. This implies that removing internal constraints and simply setting an upper bound on the growth rate μ should provide, at least superficially, similar results. And so indeed it does, as seen in Figure 4.8. This is further illustrated in Table 4.3 by the relatively small difference in mean errors for the tuned dircFBA model without dynamic constraints and the model without internal or dynamic constraint and an upper bound on growth rate μ, dubbed the "μb-model".

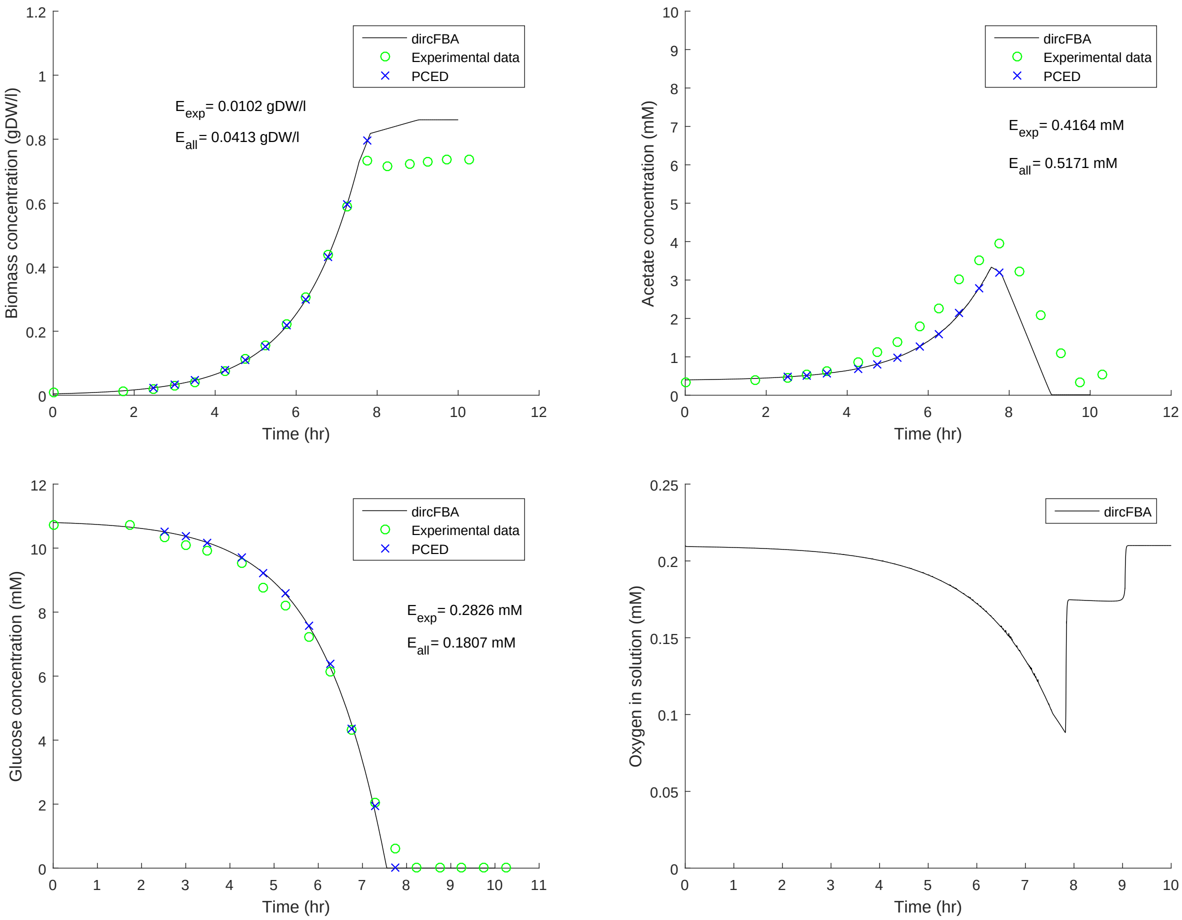

Figure 4.8: **dircFBA run with M_c = 100, $M_{\Delta c} = \infty$, and μb = 0.68** Results for the model run with an uptake bound of 15 and 10.5 mmol gDW^{-1} h^{-1} for oxygen and glucose respectively, and an uptake and secretion bound of 3.1 mmol gDW^{-1} h^{-1} on acetate. There are no dynamic or internal constraints active, but growth rate is bounded at 0.68. Mean error of prediction given both for "exponential phase" points compared to exponential data (PCED) and full-length run.

The uptake of glucose (in Figure 4.8), and the growth curve during exponential growth on glucose, are both very accurate. However, once acetate becomes the primary carbon and energy source, the μb model's ability to predict the experimental results is greatly diminished. Therefore, while it might be tempting to suggest that tuning k_{cat} sets for a given environment is equivalent to bounding the growth rate for that environment, this is not the case. Tuning k_{cat} sets for multiple environments allows several implicit bounds with a sound physical explanation. And the way the internal constraints are implemented can serve several other important functions. Firstly, a k_{cat} set tuned for sufficiently diverse media might be able to make accurate predictions for new combinations of diverse media, which simple upper bounds on growth rates would not. Secondly, the elegant formulation of ircFBA lends itself exceedingly well to adding simple dynamic constraints, in the style of dircFBA, which may become quite useful as more knowledge is gained about rates of protein synthesis and degradation. Thirdly, it is relatively easy to conceive of ways in which the framework presented by ircFBA could allow for the implementation of further elements involving regulation and optimization that could increase prediction accuracy. Fourthly, as mentioned above, and in the paper by Adadi et al. [1], internal global constraints offer a physical and biological explanation for the limit on the upper bound on growth rate.

Table 4.3: **Comparison of mean error values for dircFBA, dircFBA with $M_{\Delta c} = \infty$, dFBA with μb = 0.68, and Varma and Palsson's model.** The mean error values for dircFBA with $M_c = 03$ and $M_{\Delta c} = 001$ h^{-1}, dircFBA with $M_c = 03$ and $M_{\Delta c} = \infty$, dFBA with $\mu b = 0.68$, and Varma and Palsson's model predictions [38]

Model	E^{Gluc}_{exp}	E^{Gluc}_{all}	$E^{Biomass}_{exp}$	$E^{Biomass}_{all}$	E^{Acet}_{exp}	E^{Acet}_{all}
dirc 0.01	0.2817	0.1804	0.0059	0.0104	0.3363	0.4675
irc 0.3	0.2817	0.1804	0.0058	0.0106	0.3383	0.4696
μb	0.2826	0.1807	0.0102	0.0413	0.4164	0.5171
V&P	0.27	N/A	0.024	N/A	0.36	N/A

While the current results and their accuracy are quite similar during part of the batch run for the approaches, methods with global internal kinetic constraints offer a plausible explanation for this upper bound. They also provide a framework for further additions and expansions allowing for more knowledge to be integrated into the model. This in turn can be expected to further increase prediction accuracy in the future. As such, the dircFBA approach is both more accurate and shows more promise for future work than the simple μb approach.

4.1.4 Diauxic growth, lag phase

Out of all the methods (meaning dFBA [26], dircFBA, and Varma and Palsson's original modeling work [38]), none appear to accurately portray the period of arrested growth observed after the exhaustion of glucose in the medium, though dircFBA comes close. More sophisticated models will likely be required to accurately portray the lag phase [8]. Expanding the dircFBA model further with gene-regulatory functionality (implying some kind of built-in delays in response to change), specific rates of production and breakdown for different protein, and energetic and component costs associated with a change in biomass composition could likely markedly improve accuracy in predictions on this kind of behavior, and indeed, metabolic behavior in general.

4.2 Fed-batch runs

Finally, the model is tested on two further sets of conditions and compared with experimental data to see how well it can predict metabolic behavior across different environmental situations. Enzyme mass reallocations are then analyzed to gain insight into the demands these situations place upon the cell.

The experimental data from two of the fed-batch runs from Varma and Palsson's paper were extracted. In one, glucose was added at rates of 0.16 and 0.32 $\mathrm{g\,l^{-1}\,h^{-1}}$ before and after the 2-hour mark, respectively. This will be referred to as the "variably-fed" or VF run. In the other, glucose was added at a constant rate of 0.2 $\mathrm{g\,l^{-1}\,h^{-1}}$, and it will be referred to as the "constantly-fed" or CF run. Once again, explicit information on initial conditions was sparse, but this time, most of the values could be read off the plots with sufficient accuracy for decent simulations to be run. That is, except in the case of the initial biomass concentration in the CF run. As the curve here was less suited for fitting than in the batch growth run, some simple trial-and-error was conducted instead. The readings taken off the plot in Varma and Palsson's article [38] seemed to indicate a value in the area of $X_0 = 0.03$; This appeared too high when the simulation was run, as the plots were shifted to the left. $X_0 = 0.02$ was tried next, but seemed a little too low, as the plots were shifted to the right. $X_0 = 0.025$ seemed to provide decent results, and the model predictions using this value are listed below.

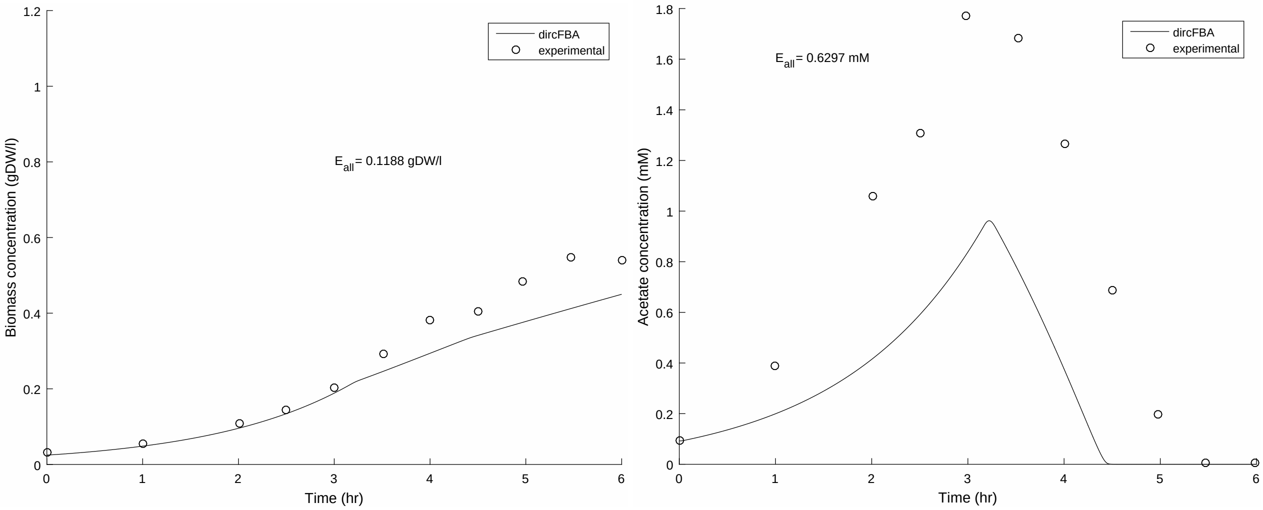

Figure 4.9: **dircFBA predictions for aerobic variably-fed (VF) batch** The internal global constraints are set as $M_c = 0.3$ and $M_{\Delta c} = 0.01$. The initial population X_0 is equal to 0.025 gDWl^{-1}, the initial concentration of glucose at zero, and glucose is being fed into the medium at rates of 0.16 and 0.32 $\mathrm{gl}^{-1}\,\mathrm{h}^{-1}$ before and after the 2-hour mark, respectively. Experimental measurements and mean error values relative to these are shown for the full run.

Seen in Figures 4.9 and 4.10 comparing the model predictions with the experimental data (E-mean values provided), the predictions are at least qualitatively quite similar to the experimental results, though they are not as accurate as those made in the original paper by Varma and Palsson [38]. Whether this stems from some weakness in the model, or is due to some faulty parameter estimate is difficult to tell.

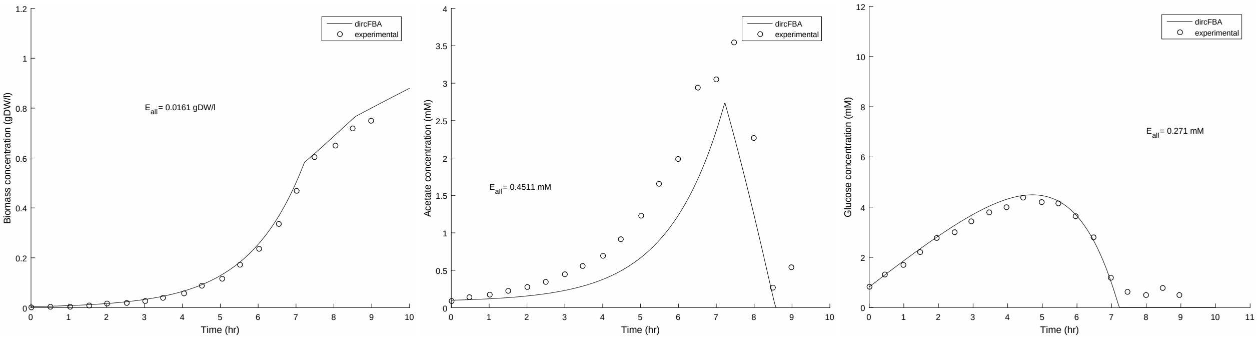

Figure 4.10: **dircFBA predictions for aerobic constantly-fed (CF) batch** The internal global constraints are set as $M_c = 0.3$ and $M_{\Delta c} = 0.01$. The initial concentration of glucose is at 0.82 mM, and glucose is being fed into the medium at a constant rate of 0.2 $\mathrm{gl}^{-1}\,\mathrm{h}^{-1}$. Experimental measurements and mean error values relative to these are shown for the full run.

More interesting are the plots showing how the changes in enzyme composition coincide with changes in growth rate, as this is a more complex environment than the aerobic batch encountered earlier. In Figure 4.11, there are four main stages of enzymatical rearrangement. During the first phase, glucose is extremely sparse, but there is an initial concentration of acetate, and the cell begins feeding on acetate. With an influx of glucose, the cell alters its enzyme composition to accommodate this change in available nutrients, electing to excrete acetate in favor of absorbing glucose. Then comes a second, calm phase, where minor readjustments are made to the enzyme composition. Towards the end of this, likely due to a slight drop in the availability of glucose, the composition change per interval begins to rise, reaching its bound as the third stage sets in, slightly before the growth rate starts to dip. As the availability of glucose drops further, and the cell begins to burn acetate as well, the growth rate steadily declines. When the stored-up acetate is exhausted, the growth rate takes a steep dive, before continuing its steady decline in a fourth phase. At this point, the change in enzyme composition drops to a low-level background noise. This because the composition is already optimized for sub-saturation of glucose, and no further changes can improve the objective. As the growth rate sinks with lower glucose availability, slowing the decline, the growth rate curve can be projected to asymptotically approach zero.

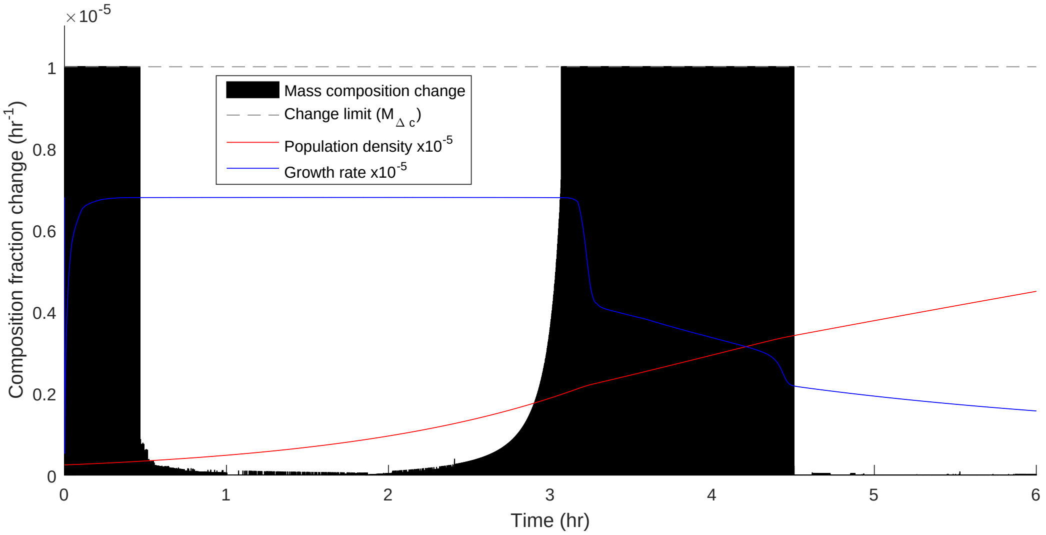

Figure 4.11: **Simulated composition change for VF run with constraints** The fraction of total mass worth of enzyme being replaced at each interval, with the bound seen at the top. Along with the mass composition change is plotted the scaled cell density and growth rate to illustrate at which points during the run the cell composition is altered. Plotted from the same simulation run as Figure 4.9.

Figure 4.12 showing the mass reallocation of enzymes for the CF run tells a similar story to Figure 4.11, showing this for the VF run. Here, the initial population is smaller, and the initial glucose injection is greater, meaning that the population is saturated for glucose from the very beginning. A long phase with a constant high growth rate passes, before the growth rate drops suddenly and the change in enzyme spikes, as glucose becomes in short supply, and acetate is consumed to supplement. A short period with no metabolic rearrangement passes, before a short spike signals the exhaustion of acetate, and the growth rate settles into the same asymptotic decline as seen at the end of the VF simulation in Figure 4.11.

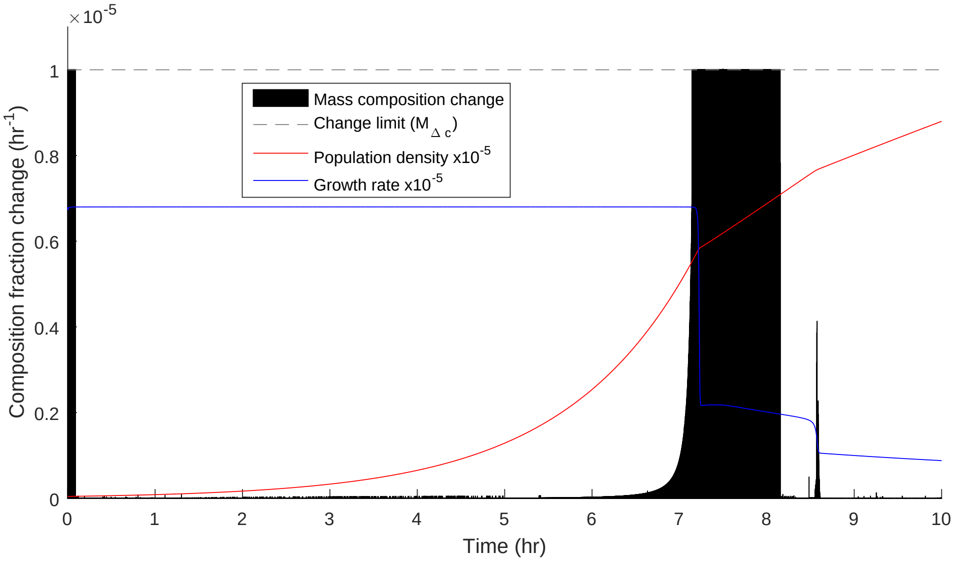

Figure 4.12: **Simulated composition change for CF run with constraints** The bar plot shows the fraction of total mass worth of enzyme being replaced at each interval, with the bound $M_{\Delta c}$ seen at the top. This illustrates how the cell must reallocate enzyme mass to deal with the changing environment. Along with the mass composition change is plotted the scaled-down cell density and growth rate. Plotted from the same simulation run as Figure 4.10.

The prediction accuracy offered here by dircFBA is lower than that of the original model prediction by Mahadevan and Palsson [26], especially for the VF run. However, they are not awful considering that no further tuning was performed between the batch run and the VF and CF runs. Additionally, being able to look at how the changes in enzyme composition interplays with the growth rate and the availability of nutrients provides an easy-to-interpret and intuitive display of how complex and changing environment affects the demands placed upon the cell's metabolism.

Chapter 5

Conclusion and Outlook

Internal global kinetic constraints were successfully merged with dynamic flux balance analysis and implemented for an extensive *E. coli* model. This allowed accurate predictions of time-series data for cell density and medium nutrient concentrations for several different environmental time-series. Considering the size of the model and the low level of manual tuning necessary, the result was considered impressive. Dynamic constraints were successfully applied as well, but did not markedly improve prediction accuracy as the model currently stands. They did however allow the extraction of useful and intuitive insights into the transients of metabolic states in changing environments. This modeling framework was dubbed dircFBA - dynamic internally constrained flux balance analysis.

The model does not predict the lag-phase, nor could it without gene-regulatory functionality, but it does quite accurately estimate terminal cell density for aerobic batch-growth on glucose.

The model treats all enzymes equally at present, assuming a universal speed of synthesis and breakdown of enzymes. This is rather naive, but it is a consequence of lacking knowledge, and not a fault inherent in the modeling approach itself. The framework could easily be extended to allow individual rates of synthesis and breakdown for for each individual enzyme. Moreover, the framework could conceivably be extended with further sophistications for increased prediction fidelity. These include, but are not limited to, gene-regulatory functionality, material and energetic cost of composition change, and better modeling of the availability of nutrients in the medium.

Due to the exponential nature of growth, the model is highly sensitive to initial conditions, such as starting population density. This means highly accurate data about these initial conditions are necessary for accurate simulations. Such initial conditions can be reverse-engineered from time-series data, however, using a mix of mathematical and computational modeling, as illustrated in this project. As more knowledge about enzymes and cell metabolism is gathered, and more sophisticated methods for parameter estimation are developed [23], future refinements and extensions of the dircFBA modeling framework hold great promise both as an investigative and a predictive tool.

Appendix A

Acronyms

COBRA Constraint-based reconstruction and analysis

dFBA Dynamic flux balance analysis

dircFBA Dynamic internally constrained flux balance analysis

DOA Dynamic-optimization approach

FBA Flux balance analysis

ircFBA Internally constrained flux balance analysis

LP Linear program

MOMENT MetabOlic Modeling with ENzyme kineTics

NLP Non-linear program

RMSD Root-mean-square deviation

SOA Static-optimization approach

SP Shadow price

Appendix B

Turnover number tuning results

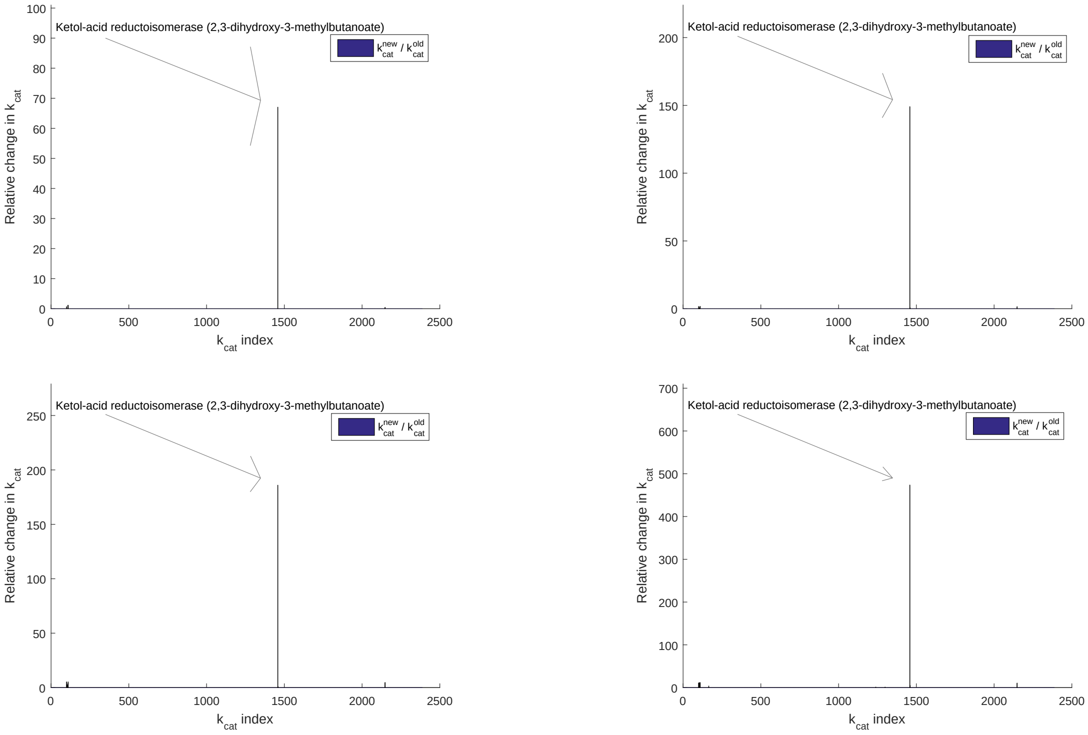

Figure B.1: **Relative changes in turnover numbers** These plots show the relative change in turnover numbers for the different enzymes in the iAF1260b *E. coli* model, after being extended into a dircFBA model according to Section 3.6 and tuned with the SP algorithm [35]. The plots are ordered left-to-right, top-to-bottom, and correspond to metabolic enzyme fractions, i.e. M_c values, of 0.4, 0.3, 0.2, and 0.1. As can be seen, very few changes are made to the k_{cat} set overall, with ketol-acid reductoisomerase being decidedly tuned the most. Note that $k_{cat}^{\text{new}} / k_{cat}^{\text{old}}$ is meant in the sense of "divided by", and that the arrowhead has the same height in terms of the y-axis in each plot.

Bibliography

[1] R. Adadi, B. Volkmer, R. Milo, M. Heinemann, and T. Shlomi. Prediction of microbial growth rate versus biomass yield by a metabolic network with kinetic parameters. *PLoS Comput Biol*, 8(7):e1002575, 2012.

[2] T. Aittokallio and B. Schwikowski. Graph-based methods for analysing networks in cell biology. *Briefings in bioinformatics*, 7(3):243–255, 2006.

[3] R. Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

[4] M. R. Antoniewicz. Dynamic metabolic flux analysis—tools for probing transient states of metabolic networks. *Current opinion in biotechnology*, 24(6):973–978, 2013.

[5] J. R. Banga. Optimization in computational systems biology. *BMC systems biology*, 2(1):47, 2008.

[6] S. A. Becker, A. M. Feist, M. L. Mo, G. Hannum, B. Ø. Palsson, and M. J. Herrgard. Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox. *Nature protocols*, 2(3):727–738, 2007.

[7] A. Chiappino-Pepe, V. Pandey, M. Ataman, and V. Hatzimanikatis. Integration of metabolic, regulatory and signaling networks towards analysis of perturbation and dynamic responses. *Current Opinion in Systems Biology*, 2017.

[8] D. Chu and D. J. Barnes. The lag-phase during diauxic growth is a trade-off between fast adaptation and high growth rate. *Scientific reports*, 6, 2016.

[9] H.-Y. Chuang, M. Hofree, and T. Ideker. A decade of systems biology. *Annual review of cell and developmental biology*, 26:721–744, 2010.

[10] M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429(6987):92, 2004.

[11] D. Davidi, E. Noor, W. Liebermeister, A. Bar-Even, A. Flamholz, K. Tummler, U. Barenholz, M. Goldenfeld, T. Shlomi, and R. Milo. Global characterization of in vivo enzyme catalytic rates and their correspondence to in vitro kcat measurements. *Proceedings of the National Academy of Sciences*, 113(12):3401–3406, 2016.

[12] M. Eisenstein. Big data: the power of petabytes. *Nature*, 527(7576):S2–S4, 2015.

[13] A. M. Feist, D. C. Zielinski, J. D. Orth, J. Schellenberger, M. J. Herrgard, and B. Ø. Palsson. Model-driven evaluation of the production potential for growth-coupled products of escherichia coli. *Metabolic engineering*, 12(3):173–186, 2010.

[14] R. J. Flassig, M. Fachet, K. Höffner, P. I. Barton, and K. Sundmacher. Dynamic flux balance modeling to increase the production of high-value compounds in green microalgae. *Biotechnology for Biofuels*, 9(1):165, 2016.

[15] K. Höffner, S. Harwood, and P. Barton. A reliable simulator for dynamic flux balance analysis. *Biotechnology and bioengineering*, 110(3):792–802, 2013.

[16] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[17] D. Kell. Metabolomics, machine learning and modelling: towards an understanding of the language of cells, 2005.

[18] T. Kodaki, H. MURAKAMI, M. TAGUCHI, K. IZUI, and H. KATSUKI. Stringent control of intermediary metabolism in escherichia coli: pyruvate excretion by cells grown on succinate. *The Journal of Biochemistry*, 90(5):1437–1444, 1981.

[19] B. D. Landry, D. C. Clarke, and M. J. Lee. Studying cellular signal transduction with omic technologies. *Journal of molecular biology*, 427(21):3416–3440, 2015.

[20] N. Le Novère, A. Finney, M. Hucka, U. S. Bhalla, F. Campagne, J. Collado-Vides, E. J. Crampin, M. Halstead, E. Klipp, P. Mendes, et al. Minimum information requested in the annotation of biochemical models (miriam). *Nature biotechnology*, 23(12):1509–1515, 2005.

[21] N. E. Lewis, H. Nagarajan, and B. O. Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, 2012.

[22] S. Light, P. Kraulis, and A. Elofsson. Preferential attachment in the evolution of metabolic networks. *Bmc Genomics*, 6(1):159, 2005.

[23] G. Lillacci and M. Khammash. Parameter estimation and model selection in computational biology. *PLoS Comput Biol*, 6(3):e1000696, 2010.

[24] H. Lin, B. Mathiszik, B. Xu, S.-O. Enfors, and P. Neubauer. Determination of the maximum specific uptake capacities for glucose and oxygen in glucose-limited fed-batch cultivations of escherichia coli. *Biotechnology and bioengineering*, 73(5):347–357, 2001.

[25] M. MacGillivray, A. Ko, E. Gruber, M. Sawyer, E. Almaas, and A. Holder. Robust analysis of fluxes in genome-scale metabolic pathways. *Scientific Reports (Nature Publisher Group)*, 7:1, 2017.

[26] R. Mahadevan, J. S. Edwards, and F. J. Doyle. Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340, 2002.

[27] V. Marx. Biology: The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[28] O. Mason and M. Verwoerd. Graph theory and networks in biology. *IET systems biology*, 1(2):89–119, 2007.

[29] A. L. Meadows, R. Karnik, H. Lam, S. Forestell, and B. Snedecor. Application of dynamic flux balance analysis to an industrial escherichia coli fermentation. *Metabolic engineering*, 12(2):150–160, 2010.

[30] D. L. Nelson, M. M. Cox, and A. L. Lehninger. *Lehninger Principles of Biochemistry*. W. H. Freeman, New York, 2nd edition, 2013.

[31] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[32] E. J. O'Brien, J. M. Monk, and B. O. Palsson. Using genome-scale models to predict biological capabilities. *Cell*, 161(5):971–987, 2015.

[33] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

[34] N. D. Price, J. L. Reed, and B. Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004.

[35] P. Røynestad. Development of an internally constrained flux balance analysis method for saccharomyces cerevisiae. Master's thesis, Norwegian university of science and technology, May 2016.

[36] J. Schellenberger, R. Que, R. M. Fleming, I. Thiele, J. D. Orth, A. M. Feist, D. C. Zielinski, A. Bordbar, N. E. Lewis, S. Rahmanian, et al. Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6(9):1290–1307, 2011.

[37] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson. Big data: astronomical or genomical? *PLoS Biol*, 13(7):e1002195, 2015.

[38] A. Varma and B. O. Palsson. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Applied and environmental microbiology*, 60(10):3724–3731, 1994.

[39] R. Wolfenden and M. J. Snider. The depth of chemical time and the power of enzymes as catalysts. *Accounts of chemical research*, 34(12):938–945, 2001.

[40] S. Wuchty and E. Almaas. Peeling the yeast protein network. *Proteomics*, 5(2):444–449, 2005.