Accelerated Construction of Kinetic Models for Cell Metabolism

A thesis submitted to The University of Manchester for the degree of

Doctor of Philosophy

in the Faculty of Biology, Medicine and Health

2018

Chuan Fu Yap

School of Biological Sciences

Table of Contents

Total Word Count: 25806

List of Tables

List of Figures

List of Abbreviations

2PG	2-Phosphoglycerate
3PG	3-Phosphoglycerate
ADH	Alcohol dehydrogenase
ADH	Alcohol dehydrogenase.
ALD	Aldolase
ATP	Adenosine triphosphate
AcAld	Acetaldehyde
BPG	1,3-Bisphosphoglycerate
CLI	Command Line Interface
DHAP	Dihydroxyacetone phosphate
ENO	Enolase
F16BP	Fructose 1,6-bisphosphate
F6P	Fructose 6-phosphate
G1P	Glucose 1-phosphate
G6P	Glucose 6-phosphate
G6PDH	Glucose 6-phosphate dehydrogenase
GA	Genetic Algorithm
GAP	Glyceraldehyde 3-phosphate
GAPDH	Glyceraldehyde-3-phosphate dehydrogenase
GLCi	Intracellular glucose
GLCo	Extracellular glucose
GLT	Glucose transport
GPH	Glycogen phosphorylase
GPM	Phosphoglycerate mutase
GSY	Glycogen synthase
GUI	Graphical User Interface
HXK	Hexokinase
HXT	Glucose transport
JNA	Java Native Access

JNI	Java Native Interface
NAD	Nicotinamide adenine dinucleotide
NADH	Nicotinamide adenine dinucleotide (reduced form)
ODE	Ordinary Differential Equations
PDC	Pyruvate decarboxylase
PEP	Phosphoenolpyruvate
PFK	Phosphofructokinase
PGI	Phosphoglucose isomerase
PGK	3-Phosphoglycerate kinase
PGM	Phosphoglucomutase
PSO	Particle Swarm Optimization
PYC	Pyruvate carboxylase
PYK	Pyruvate kinase
PYR	Pyruvate
SBML	Systems Biology Markup Language
SOSlib	SBML ODE Solver Library
SS	Scatter Search
T6P	Trehalose 6-phosphate
T6PP	Trehalose phosphatase
T6PS	Trehalose-phosphate synthase
TPI	Triose-phosphate isomerase
UDPG	Urine diphosphate glucose
UDPGP	Urine diphosphate glucose phosphorylase
UTP	Phosphoglyceromutase

Abstract

The University of Manchester, Doctor of Philosophy

Chuan Fu Yap

Accelerated Construction of Kinetic Models for Cell Metabolism

The use of mathematical models is enriching biological research, as it allows biologists to learn how the different components within a biological system interact, leading to a holistic approach to research. This is a result of technology improvements that enable the generation of high-throughput data, and of increased collaboration with mathematicians, physicists and computer scientists. There are various methodologies to model a biological system. A dynamic model allows users to represent quantitative information and follow the temporal changes of the system, which are very important for understanding complex systems.

The construction of kinetic models is often impeded by incomplete information on kinetic data, including the kinetic parameters and rate laws. Additionally, data are frequently collected under different conditions. Previously, the software GRaPe was introduced to address these issues by automatically generating generic kinetic rate equations and estimating kinetic parameters by searching a local solution in the parameter space using only steady state data. This thesis introduces an upgraded version of the software that uses convenience kinetics (allowing for inclusion of regulatory effects to models) and a global solution of equations for parameter estimation, using a genetic algorithm with both time-series and steady state data.

As a proof of concept for the software, the glycolytic network of *Saccharomyces cerevisiae* was modelled using the software, and produced favourable results. Following this, trehalose metabolism of *Saccharomyces cerevisiae* was studied using a model generated with the new tool. It confirmed that the increase of flux during heat stress is caused by the positive feedback on pyruvate kinase. Additionally, the model was able to determine the best enzymes to overexpress in order to increase the yield of trehalose, a commercially valuable product.

This thesis introduces an intuitive software that will serve as a gateway tool for building kinetic models of cell metabolism, aimed at non-expert users that wish to study complex biological systems or to generate rapid prototype of models.

Declaration

I declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://http://www.library.manchester.ac.uk/about/regulations/) and in The University's policy on Presentation of Theses.

Acknowledgements

I would like to express the sincerest gratitude towards my supervisors Dr Jean-Marc Schwartz and Prof Simon Hubbard for their support and guidance throughout the journey of my PhD. Specifically, I would like to thank Dr Schwartz for not only sharing with me his knowledge and experience on modelling in biological research, he also provided guidance on how to formulate good scientific writing, and educating me the importance of prioritizing the right things within the research so as not to fly too close to the sun, this skill has helped kept me on track to complete my PhD. To Prof Hubbard, I would to thank him for broadening my horizon on the field of proteomics.

Secondly, I would like to thank my advisor Prof Magnus Rattray for passing to me his wisdom on being a good scientist. I would also like to thank Dr Andrew Jarnuczak and Dr Manuel Garcia Albornoz for sharing with me the heat stress proteomics that was generated and analysed by them respectively.

I would like to thank the present and past group members of the Schwartz Lab Group, especially Dr Kun Tian and Dr Jamie Soul, and the wonderful office mates that I was fortunate enough to meet at the start of my PhD, they are Dr Nicholas Gresham and Andrew Gilchrist. I am grateful for their patience in answering all the queries I had when I first started my PhD no matter how trivial, as this really helped me settle into the office, a place where I would spend a lot of my time during my PhD.

Finally, I would like to thank my family and friends for the support and encouragement they have provided me over the years.

The Author

I am a graduate from The University of Sheffield with an Integrated Masters (MBiolSci) in Biochemistry. During my undergraduate studies, I have completed three different research projects, one of which I was fortunate enough to work with the late Prof Peter Artymiuk in a project where we hunted for toxins using protein motif search. This project introduced me to the possibilities of bioinformatics, a wonderful use of computer in biological research.

It was during my time in Sheffield where I was introduced to the field of Systems Biology, through Prof David Hornby's module Molecular Systems Biology and Synthetic Biology. This showed me that the marriage of mathematics and biology was possible. Which led me to complete master's research in using mathematical models to study the interactions of macrophages with *Staphylococcus aureus*.

Combination of all the experiences from Sheffield have led me to Manchester to allow me to write this thesis which will hopefully introduce to more biologists the wonders of mathematics, and begin applying them in their research.

Chapter 1: Introduction

1.1 Systems Biology

A system is an assembly of components that interact with one another to complete a job. There is no limit to the scale of a system within the biological context. It could be used to refer to the ecosystem of a given habitat or it could be the various systems in the human body, such as respiratory, nervous and immune system. As all these different parts interact with each other functional emergence can occur; emergence is a property that arises from the interactions in a system but not from individual parts. Hence, it is advantageous for them to be studied with a holistic approach, instead of a reductionist approach.

In the context of systems biology, systems is often referred to as the networks of genes, proteins and metabolites (Hillmer, 2015). A genetic network can be represented as a regulatory network coupled with the transcriptional processes of genes into mRNA. A protein network is often a protein-protein interaction network, depicting the interactions of proteins whether they are stable or transient. The chain of reactions catalysed by enzymes that allow cells to grow and sustain life is represented by the metabolic network, which is an interconnected network of metabolites (Machado et al., 2011). Some would argue the field of systems biology came about from trying to understand the feedback dynamics of those networks (Wolkenhauer and Mesarović, 2005). One thing for certain is that a hallmark of systems biology research is the use of a mathematical or computational model in their studies.

The surge in the field of systems biology was made possible by the recent improvement in high-throughput experimental methods, which generated 'omics' data. These omics data include genomics, transcriptomics, proteomics, metabolomics, fluxomics and lipidomics. These data made the reconstruction of various biological networks possible (Yurkovich and Palsson, 2018). The collaboration between physicists, mathematicians and computer scientists with biologists was another factor in making these fields flourish, as their inputs were valuable in constructing the models used for research. With a model, we can better understand how the system functions, as well as make *in silico* experimentation by simulating the biological outcome of a given system after

perturbation (i.e. making gene knockouts, knockdowns, overexpression studies), to help experimental biologists save time and money by simulating experiments before carrying them out. A model's prediction could also be applied to increase yield of a given compound that could be commercially viable or useful for disease treatment; or models can be built to study conditions that cannot be easily approached through wet lab studies (Hübner et al., 2011).

1.1.1 What exactly is a mathematical model and how do we build one?

A mathematical model is an abstraction of reality, usually represented with equations. As an abstraction, not every detail has to be considered and instead assumptions can be made to simplify the model. Therefore, there is a fine balancing act involved, omittance of certain details might lead to a meaningless model or some would say a 'spherical cow', a term that described an oversimplified model, which deviated significantly from reality (Azeloglu and Iyengar, 2015). The decision to determine what detail to include or omit often comes down to the problem of interest that is being investigated. For example, if a compound of interest is being studied, the compound and its neighbouring compounds in the network would be important, while pathways that channels flux/metabolite towards or away from them can be simplified. Additionally, one of the common assumptions to be considered when building models is if the system being studied is isolated (in a fixed and controlled environment) or coupled with the dynamic external environment (Klipp et al., 2009a).

On top of considering the details and assumptions of a model, depending on the type of model, several other types of information are needed. A commonality would be information on the reaction pathways or network structure of the system being studied. If it is a kinetic model, knowledge of the rate law for each reaction in the pathway, the kinetic parameter values, and if necessary, training and validation data for the parameter estimation process are needed. These types of information can be obtained through measurement in the lab, or be searched in the vast literature that has been accumulated over the years. Another source would be the databases that pools these data together, such as ConsensusPathDB (Kamburov et al., 2011), Reactome (Fabregat et al., 2018) and KEGG (Ogata et al., 1999) that have pathway information; BRENDA (Schomburg et al., 2004), KiMoSys (Costa et al., 2014) and SABIO-RK (Krebs et al., 2007) for information on enzyme kinetics; CeCaFDB (Zhang et al., 2015) for fluxomics data; proteomics data

can be found in the PRIDE Archive (Vizcaíno et al., 2016), PaxDB (Wang et al., 2015) and ProteomicsDB (Schmidt et al., 2018); and for metabolomics data, they can be found on METLIN (Guijas et al., 2018), MetaboLights (Haug et al., 2013), The Metabolomics Workbench (http://www.metabolomicsworkbench.org).

After collecting all the information, the next step would be to determine the modelling formalism to use and to actually construct the model. For a beginner, this task can be very daunting, however with years of research in the field, the community have developed various tools to make the process easier as well as standards that makes sharing of information easier. For example, there are the Systems Biology Markup Language (Hucka et al., 2003) and CellML (Cuellar et al., 2003), both of which are file formats for storage and exchange of computational models of biological networks; Systems Biology Graphical Notation (Novere et al., 2009) is a standard for visualisation of biological networks; and Systems Biology Ontology (Courtot et al., 2011), which is a set of controlled vocabularies commonly found in Systems Biology research in relation to modelling. When it comes to the building and simulation of models, there are software available for that purpose, such as COPASI (Hoops et al., 2006), CellDesigner (Funahashi and Matsuoka, 2008), and Virtual Cell (Schaff et al., 2000).

The remainder of this review will be breaking down some of the modelling formalisms employed by systems biologists in their research, followed by a greater focus on the various rate functions to model enzymatic reactions. Lastly, the process of parameter estimation is discussed along with how some of the algorithms functions.

1.2 Modelling Frameworks

A model can be dynamic (commonly represented with a system of differential equation to follow temporal changes) or static (qualitative models, such as ones built using graph theory to imitate a biological phenomenon). In this section, some of the commonly used modelling frameworks in systems biology will be briefly discussed.

1.2.1 Boolean Networks

The Boolean network formalism was introduced to model gene regulatory networks by Kauffman (Kauffman, 1969). This form of model is a directed graph made up of nodes/vertices of Boolean variables (if the node is 'ON' or 'OFF', in gene regulatory networks it would indicate that the gene is expressed or unexpressed respectively, or in signalling networks that it is activated or not). A state is a binary vector of all the nodes' values at a given time. Thus, a model with n number of nodes will have 2^n possible states (Klipp et al., 2009b). It is also known to some as logic model as it makes use of logic rules such as 'AND', 'OR', and 'NOT' to determine the state of the next node. Each state in a Boolean network has a deterministic output state that is determined by the inputs. This method of modelling is often used to find steady states (in Boolean network terms, it is called 'point attractor' where there is only a single state in the 'attractor' which is the final state(s) of the network given an input) and analyse robustness of the network (Li et al., 2004).

1.2.2 Petri Nets

Petri net was developed by Carl Adam Petri to describe chemical processes (Petri, 1962). It is a graphical and mathematical modelling format has been adapted to study biological processes. A Petri net is a directed bipartite graph (made up of two sets of nodes), one set is called 'places', represented graphically with circles, the other set is called 'transitions', represented graphically with rectangles (Klipp et al., 2009b). In the context of biological models, places represent molecules, and transitions represent reactions. Places can hold zero or more 'tokens' (describing the number of molecules), and they are produced when an incoming transition 'fires' (a reaction takes place). For this to occur, an input place must have sufficient tokens connected to outgoing transitions; when a transition fires, tokens are consumed from the input place. There are extensions to petri nets making it a flexible framework to explore different biological networks, such as metabolic and signalling (Breitling et al., 2008; Koch et al., 2005; Machado et al., 2011). Extensions include 'Timed Petri nets' that makes it possible for transitions to have a time delay in

the firing step (Zuberek, 1991), and 'Hierarchical Petri nets' that introduced modularity, where a whole net can be represented as a place or transition (Bernardinello and Cindio, 1992).

1.2.3 Flux Balance Analysis for Constraint-based models

Constraint-based models can be built using flux balance analysis, which is a mathematical method to determine the flux distribution within a metabolic network (Varma and Palsson, 1994). The first step to building this model is to represent the metabolic network as a stoichiometric matrix (S), of size $m * n$ (m is the number rows representing compounds and n is the number columns representing reactions). The entries in the matrix are their respective stoichiometric coefficients, where consumed metabolites are represented with negative coefficients and metabolites produced are represented with positive coefficients. An important assumption in constraint-based models is that the systems achieves steady-state quickly. Hence, $Sv = 0$ at steady state, where v is the vector of fluxes. We can solve for vector v using linear programming. However, there is no unique solution when there are more reactions than compounds. A solution would be to impose constraints on the fluxes to reduce the solution space. The types of constraints can be thermodynamic, enzyme capacity or upper and lower bounds on the fluxes. Models built using flux balance analysis can be used to predict growth of an organism or rate of production of a compound. This method has been applied to build various genome-scale metabolic model (Duarte et al., 2007; Feist et al., 2007; Forster, 2003; Orth et al., 2011)

1.3 Kinetic Modelling Formalisms for Metabolic Models

The use of kinetic models is preferable over other frameworks as this form is quantitative, allowing us quantify the metabolites in the network; kinetic models are dynamic allowing us to follow the changes over time instead of specific states, and they allow us to input the regulatory effects within the model. These factors also allow us to better understand complex dynamic processes in the system (Link et al., 2014). However, this formalism is often challenging to apply as the amount of information needed is relatively higher than other methods for a network of similar size. This leads to a longer timeline for construction of a model. Additionally, there is an added difficulty in metabolic models when it comes to selecting the rate function for a reaction (Costa et al., 2010). This section will briefly discuss some of the approximate rate laws developed to simplify the mechanisms of enzymatic reactions. Kinetic models are often built using ordinary differential equations (ODE), an example of how this is carried out using convenience rate law formalism is shown in figure 1.1.

1.3.1 Michaelis-Menten Kinetics

Before we dive into approximate rate laws, we should first be acquainted with the enzyme rate law that started it all for metabolic models, Michaelis-Menten rate law (Equation 2). The rate law is formulated based on the enzyme binding and catalysis mechanism of Equation 1, together with the introduction of two assumptions, quasi-equilibrium and quasi-steady state.

$$\mathbf{E} + \mathbf{S} \underset{k_r}{\overset{k_f}{\rightleftharpoons}} \mathbf{ES} \xrightarrow{k_{cat}} \mathbf{E} + \mathbf{P} \quad \text{(Eq. 1)}$$

The first assumption considers a quasi-equilibrium between free enzymes (E in Eq. 1) and the enzyme substrate complex (ES in Eq. 1), such that reversible binding of E+S to ES is faster than the catalysis process of ES into E + P, implying a higher kinetic constant value for k_f and k_r than k_{cat}. The second assumption applies only in the event of the substrate concentration being much higher than the enzyme concentration. The quasi-steady state assumption is that the concentration of the ES complex remains constant throughout the reaction.

$$v = \frac{V_{max} S}{S + K_m} \quad \text{(Eq. 2)}$$

v = Rate of reaction/flux

S = Substrate concentration

V_{max} = Maximal rate

K_m = Michaelis-Menten constant

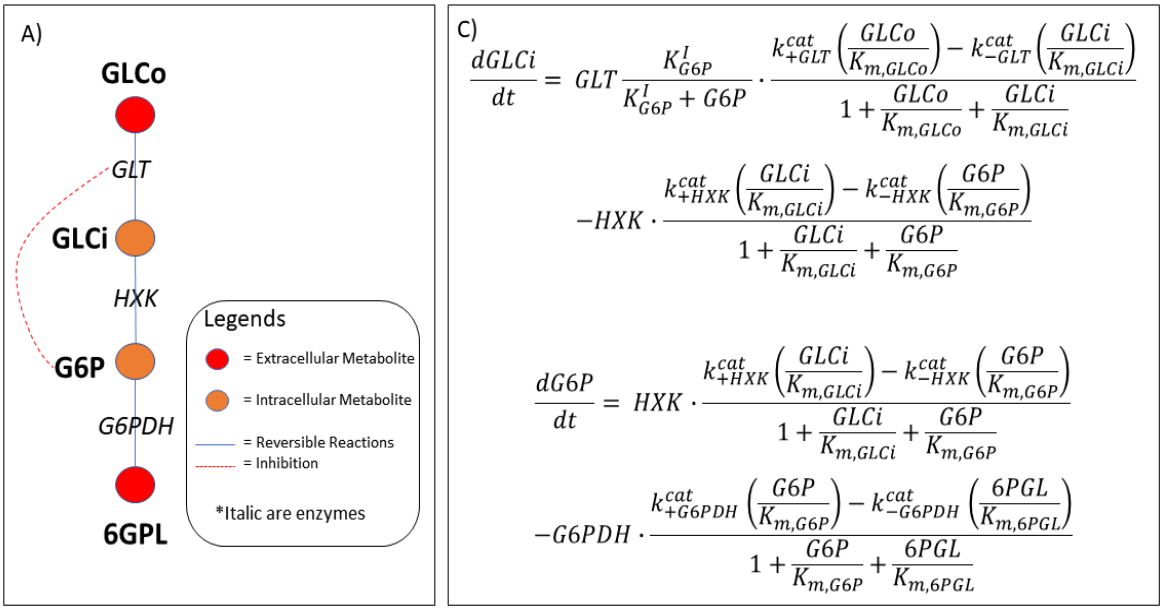

***Figure 1.1*: Modelling with ordinary differential equation (ODE).** A) A simplified reaction pathway of glucose entry into the pentose-phosphate-pathway via 6-phosphogluconolactone. The simplified pathway does not include any cofactors, and is modeled in an isolated manner with extracellular glucose and 6-phosphogluconolactone acting as external metabolites and are set to have constant concentrations. The simplified reaction does include an inhibition on the glucose transport enzyme by glucose-6-phopsphate. GLCo: extracellular glucose, GLCi: intracellular glucose, G6P: glucose 6-phosphate, 6PGL: 6-phosphogluconolactone, HXT: glucose transport, HXK: hexokinase, G6PDH: B) Example of how ODE modelling is done for cell metabolism, where rate of change of metabolite is tracked by following the production of the metabolite (the enzyme reaction that generates the metabolite) deducted from the consumption of the metabolite (the enzyme reaction that breaks down the metabolite). C) Group of equations modelling simplified pathway seen in A, equations used here are using the convenience rate law formalism. As concentrations of extracellular glucose and 6-phosphogluconolactone are constant, they would not be tracked, only concentrations of intracellular glucose and glucose-6-phosphate are tracked.

The maximal rate is the highest possible rate achievable when the enzyme is completely saturated with substrate; in reference to equation 1 it would be k_{cat} multiplied with total enzyme concentration. The Michaelis-Menten constant is the amount of substrate that gives half-maximal rate; in reference to equation 1, it is $\frac{k_r+k_{cat}}{k_f}$. Equation 2 is the simplest form of Michaelis-Menten kinetics, describing single substrate and product catalysis. As the number of substrates and products increase, as well as enzyme activation or inhibition are added, or different binding order for substrates is assumed, the equation expands and requires more parameters. However, not all of the possible reaction mechanisms for enzyme reactions are known. For reactions with unknown rate laws, they can be replaced with the approximate rate laws discussed below.

1.3.2 Generalized Mass Action kinetics

Generalized mass action (Eq 3, Savageau, 1976) is a refined version of mass action kinetics, making it the simplest rate function as it hides the enzyme effects, such as the binding between enzyme and substrate or the saturation effect of having more substrates than enzymes. The kinetic order *f* can take up any real value, it will have positive values if it is a substrate or activator, and negative values for inhibitors.

$$v_i = k_i \prod S_j^{f_{i,j}} \quad \text{(Eq. 3)}$$

v = Rate of reaction/flux

S = Concentration of substrate/effector

k = Turnover rate

f = Kinetic order

1.3.3 Convenience rate law

The convenience rate law (Eq. 4, Liebermeister and Klipp, 2006) is a generalized form of Michaelis-Menten kinetics (Eq. 2) developed to ease the process of parameter estimation. It is able to cover all possible stoichiometries by inserting the value into *n* for the respective compounds.

$$v = E_{tot} \cdot f_{reg} \frac{k_+^{cat} \Pi (\frac{S_i}{K_{m,S_i}})^{n_i} - k_-^{cat} \Pi (\frac{P_j}{K_{m,P_j}})^{n_j}}{\Pi\left(1+\left(\frac{S_i}{K_{m,S_i}}\right)+\cdots+\left(\frac{S_i}{K_{m,S_i}}\right)^{n_i}\right)+\Pi\left(1+\left(\frac{P_j}{K_{m,P_j}}\right)+\cdots+\left(\frac{P_j}{K_{m,P_j}}\right)^{n_j}\right)-1} \quad \text{(Eq. 4)}$$

v = Rate of reaction/flux

f_{reg} = Regulatory prefactor

E_{tot} = Enzyme concentration

S = Substrate concentration

P = Product concentration

$K^{cat}_{+/-}$ = Forward and reverse turnover rate

K_m = Michaelis-Menten constant

n = stoichiometry value for substrate/product

To describe regulatory effects on enzymes, a regulatory prefactor (Eq. 5/6) is included, if there is no regulation it takes a value of 1. The regulatory prefactor are represented in two of the following forms:

$1 + \frac{d}{K^A}$ (Eq. 5) for activator regulatory effect

$\frac{K^I}{K^I + d}$ (Eq. 6) for inhibitory regulatory effect

d = Modifier concentration

K^I = Inhibitor constant

K^A = Activator constant

The kinetic parameters in this formulation are all comparable to the parameters in Michaelis-Menten kinetics that are measured in enzyme assays. K_m, represents the substrate concentration value when the reaction rate at half-maximal velocity. K^I and K^A represent concentration values at which activator or inhibitor gives half-maximal velocity. This rate law assumes random binding order for the substrates and all reactions are reversible. Analogous to Michaelis-Menten kinetics it is a saturable rate law.

1.3.4 Linlog Kinetics

Linlog kinetics (Eq. 7, Visser and Heijnen, 2003) represents flux as a linear combination of logarithmic terms. The ^{0} superscript denotes the reference state of the enzyme, substrate and products. The reference state would usually be wild type steady state.

$$v = v^0 \frac{E}{E^0} \left(1 + \sum \varepsilon^0_{S_i} log\left(\frac{S_i}{S^0_i}\right)\right) \text{ (Eq. 7)}$$

v = Rate of reaction/flux

S = Substrate concentration

E = Enzyme concentration

ε_S = Substrate elasticity

As it requires a reference state, this rate law would only be able to produce good results close to the said state. Additionally, due to the use of logarithmic values, the rate becomes negative for small compound concentration.

1.3.5 Modular rate laws

Modular rate laws (Eq.8, Liebermeister et al., 2010) are a family of five different rate laws developed to ensure thermodynamic accuracy and numerical stability for all the parameters. The five different rate laws are 'common' (Eq. 9), 'direct binding' (Eq. 10), 'simultaneous binding' (Eq. 11), 'power-law' (Eq. 12) and 'force-dependent' (Eq. 13) which would take the form of the following to be inserted into D_r:

$$v = E_{tot} \cdot f_{reg} \frac{k_+^{cat}\Pi(\frac{S_i}{K_{m,S_i}})^{n_i} - k_-^{cat}\Pi(\frac{P_j}{K_{m,P_j}})^{n_j}}{\mathrm{D}_r + D_r^{reg}} \text{ (Eq. 8)}$$

v = Rate of reaction/flux

E_{tot} = Enzyme concentration

f_{reg} = Regulatory prefactor

S = Substrate concentration

P = Product concentration

K_m = Michaelis-Menten constant

$K^{cat}_{+/-}$ = Forward and reverse turnover rate

D_r = Denominator for different rate laws

D_r^{reg} = Specific regulation

n = stoichiometry value for substrate/product

Common modular rate law,

$$\prod\left(1+\left(\frac{S_i}{K_{m,S_i}}\right)\right)^{n_i} + \prod\left(1+\left(\frac{P_j}{K_{m,P_j}}\right)\right)^{n_j} - 1 \text{ (Eq. 9)}$$

Direct binding modular rate law:

$$\prod\left(\frac{S_i}{K_{m,S_i}}\right)^{n_i} + \prod\left(\frac{P_j}{K_{m,P_j}}\right)^{n_j} + 1 \text{ (Eq. 10)}$$

Simultaneous binding modular rate law:

$$\prod\left(1+\left(\frac{S_i}{K_{m,S_i}}\right)\right)^{n_i} . \prod\left(1+\left(\frac{P_j}{K_{m,P_j}}\right)\right)^{n_j} \text{ (Eq. 11)}$$

Power-law modular rate law:

$$1 \text{ (Eq. 12)}$$

Force-dependent modular rate law:

$$\sqrt{\prod\left(\frac{S_i}{K_{m,S_i}}\right)^{n_i} . \prod\left(\frac{P_j}{K_{m,P_j}}\right)^{n_j}} \text{ (Eq. 13)}$$

The regulatory prefactors are represented in the following form:

$$\prod\left(p_{rj}^{A}+\left[1-p_{rj}^{A}\right]\left(\frac{\frac{a}{K_{rj}^{A}}}{1+\frac{a}{K_{rj}^{A}}}\right)\right)^{w_j}\prod\left(p_{rl}^{I}+\left[1-p_{rl}^{I}\right]\left(\frac{1}{1+\frac{i}{K_{rl}^{I}}}\right)\right)^{w_l} \quad \text{(Eq. 14)}$$

$P^{A/I}$ = Relative basal rate

K^{I} = Inhibitor constant

K^{A} = Activator constant

a = Activator concentration

i = Inhibitor concentration

Specific regulation is represented as:

$$\sum\left(\frac{K_{ri}^{A}}{a}\right)^{w_{ri}^{+}}+\sum\left(\frac{i}{K_{ri}^{I}}\right)^{w_{ri}^{-}} \quad \text{(Eq. 15)}$$

The activator's relative basal rates within the regulatory prefactor can vary between value of 0 and 1, which is represented by the ratio of amount activator is absent to saturating levels of activator, while the inhibitor's relative basal rate is defined accordingly. When the basal rates are zero, the regulation achieves complete activation or inhibition, else it would be partial activation or inhibition. The superscripts w in both regulatory functions are regulation numbers which by default is 0 if absent and 1 if present. These rate laws are able to cover any number of stoichiometries and different types of regulation with two different regulatory factors. Like convenience kinetics it has a lower number of kinetic parameters compared to classic Michaelis-Menten kinetics, as it scales in the number of substrates and products. However, at a glance it can be difficult for the uninitiated to choose which one of them to apply to the reactions, as the choices depend on the type of binding the enzyme/substrate would have (direct-binding, simultaneous-binding, etc.).

1.4 Parameter Estimation in Systems Biology

Parameters make up a major portion of kinetic models used to describe biological phenomena. They play a big role in defining model's behaviour as they describe molecular properties. Typical parameters include Michaelis-Menten constants, maximal velocities, biological half-lives, binding constants, molecule concentrations and diffusion rates. These parameters can usually be determined through experimental studies, or be obtained from literature or databases that collect these information such as BRENDA and SABIO-RK (Costa et al., 2014; Krebs et al., 2007; Schomburg et al., 2004). However, not all parameters can be measured experimentally (Sun et al., 2012). To overcome this, we can use computational and/or mathematical methods to estimate these missing parameters. If necessary, parameter estimation can also be used to determine all the kinetic parameters in a model when given enough data for the estimation process. Parameter estimation is argued by some to be the most difficult step in the model building process (Chou and Voit, 2009). This section will discuss what is parameter estimation, some of the algorithms and some readily available tools to perform parameter estimation for computer models of biological systems.

1.4.1 What is Parameter Estimation?

Parameter estimation is also known as inverse problem, model fitting and data fitting (Ashyraliyev et al., 2009; Klipp et al., 2009a; Sun et al., 2012). Parameter estimation can be viewed as an optimization task where the aim is to find a parameter vector (the set of parameters in a model) that produces the least difference between the model and a given set of experimental data using a metric such as root mean squared error or mean absolute error.

Typically, parameter estimation would start with a random set of parameter values (a pre-set range can be given) and the values would change over several iterations according to the rules of a given algorithm. After each change in values, the model's output (i.e. flux values, metabolite concentrations) would be compared with experimental data (also known as fitting data) and scored using the desired metric. In optimization tasks, the scoring function would often be called the objective function, where a given algorithm would aim to find the lowest value (or highest if desired) of the given objective function. The parameter vector that gives the best score (if an error

metric is used the lowest scoring vector is the best) would be deemed the best set of values for the model in a given optimization task.

Two general types of data are used in parameter estimation, they are steady-state and time series. Steady state is a condition where the metabolic system have reached balanced fluxes and constant levels of compound concentrations. A time series is just as the name suggest, a series of data following a property of the biological system such as protein concentration as it changes over a given time frame. For parameter estimation, the more data provided the better, therefore in the case of time series data, higher granularity would lead to better outcome. Alternatively, more data can be generated by altering conditions of the experimental settings or using knockouts of the model species.

One problem that can arise from parameter estimation is the model may only be able to generate outputs that match the fitting data, and unable to reproduce outputs of new data previously not used in the fitting process. This phenomenon is known as overfitting and it is caused by the parameter estimation process forcing the model's output to agree with the given data. When this occurs, the model will not be able to serve its purpose in biological research of predicting new data for hypothesis generation. To overcome this issue, cross-validation can be performed to determine if the model is overfitted. This is done by dividing fitting data into training and test sets (i.e. one set of data collected under one condition for training and another set for test, or in the case of time series data, a later portion of the data can be used as test set), where the training set is used for fitting the model, while the test set is used to assess the model's predictive quality after it is fitted. If the fitted model is able to reproduce the values of the previously unused test set, the model is considered validated. A validated model can be used for reliable predictions in research.

1.4.2 Parameter Estimation Methods

Optimization methods are of two broad classes, which are global and local optimizers. This distinction comes from the phenomenon that in the parameter search space there exist global and local optima (Figure 1.1); local methods are those that converges to a local optimum in neighbourhood of the starting point while global methods ideally search the entire parameter

space for a global (some call it 'true') optimum (Ashyraliyev et al., 2009; Sun et al., 2012). A global optimum is defined as the point where no other parameter vector gives a smaller (if looking for minimum) or bigger (if looking for maximum) value for the objective function. A local optimum is when no other sets of parameter in the vicinity of search space can produce a better objective function (Klipp et al., 2009a).

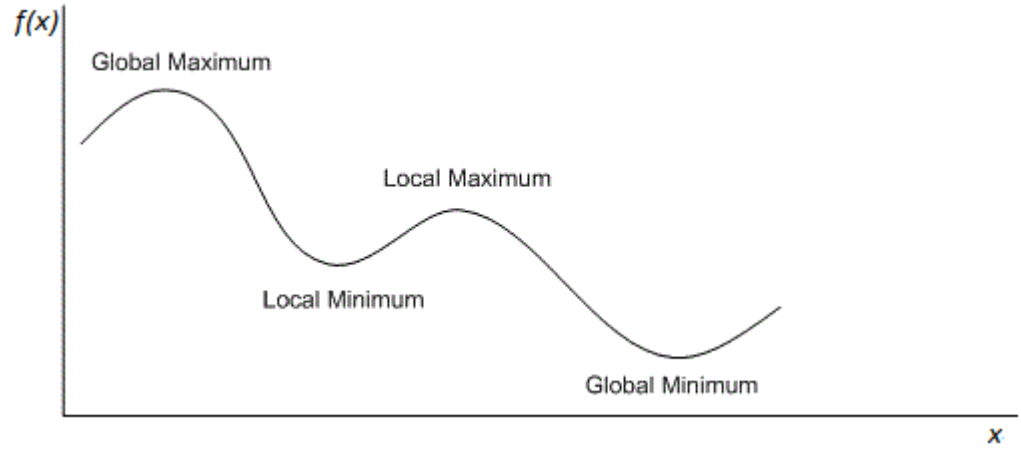

Figure 2.2: **Global vs Local optimization.** x is the parameter vector and f(x) is the objective function

For local optimizers, there are two commonly used categories of methods. Direct-search methods such as the Hooke and Jeeves method (Hooke and Jeeves, 1961), and the Nelder-Mead method (Nelder and Mead, 1965). Direct-search methods search through a limited number of candidates and determine if any of them are better than the current one. The other commonly used category is gradient-based methods, examples include the Levenberg-Marquadt method (Levenberg, 1944) and Newton's method. Gradient descent makes use of the first derivative of the function being solved to determine the direction of descent, and makes a step in that direction, with the aim of improving on the objective function.

Global optimizers are often stochastic search algorithms, as the stochasticity would usually allow it to escape local optima as it searches the parameter space. Examples of such algorithms include simulated annealing (Kirkpatrick et al., 1983), inspired by the physics phenomenon of slow cool down of a solid after heating it up; and other methods inspired by nature such as genetic algorithms (Holland, 1962), ant colony optimization (Colorni et al., 1991), particle swarm optimization (Kennedy and Eberhart, 1995) and scatter search (Glover, 1977).

1.4.2.1 Hooke-Jeeves Method

This method is done is two steps, an exploratory move followed by a pattern move. In the exploratory move, a series of changes is made on the parameter vector, usually one parameter is changed in a positive and negative direction. This is done to collect information on the immediate parameter space, for any changes that brought an improvement on the objective function the value is maintained. After all the parameters have been manipulated and there a decrease/increase on the objective function, the algorithm proceeds with the next step, if not, the exploratory move is repeated with a decrease in step size (in this context decrease magnitude of parameter value manipulation). In the pattern move, information from the previous step is used to determine the best direction to take in the parameter search space and move towards it. These two steps will repeat until the step size have been reduced to a specified value, and an optimum is assumed to have been found.

1.4.2.2 Levenberg-Marquadt Method

This method is an interpolation of two minimization methodologies: gradient descent method and Gauss-Newton method. In gradient descent, also known as steepest descent, the search direction is the opposite of the gradient ('downhill') of the objective function. The Gauss-Newton method minimizes the sum of squares of the objective function by assuming the function is quadratic, and solves for its minimum. Levenberg-Marquadt combines these two methods with a damping factor, which is set to a large value initially and decreases as iteration of the algorithm increases. When the damping factor is high, it takes advantage of the gradient descent method and vice versa. In this manner, it would be using steepest descent when it is far away from the optimum, and the Gauss-Newton method as it approaches the optimum.

1.4.2.3 Genetic Algorithm

Genetic algorithm (GA) is a form of search algorithm inspired by the Darwinian evolutionary principle. In brief, GA 'replicates' the evolutionary process with a population of potential solutions (the parameter vectors) that go through multiple generations of mutations, selection and reproduction. The potential solutions are evaluated at every generation with a fitness function (this is GA terminology for objective function), and the population evolves over generations to produce the best solution.

The first step in GA is to initialise a population (a user determined number, a larger population might result in a better solution, but also increases computation time) of parameter vectors made up of random values, which are traditionally encoded in binary numbers. These vectors are usually referred to as chromosome or individuals in GA terminology. Following this, each individual is scored with a function that gives them a fitness score.

With the individuals scored, they undergo a selection process to determine parents for the mating (reproduction) process. For the selection process, there are several variations users can choose from. All these different options rely on the fitness scores of the individuals, it can be as direct as picking the best individuals in an 'elitist' manner; or rely on element of randomness to avoid the aforementioned 'elitist' selection, such as picking several random individuals and retaining only the best of the chosen ones in a 'tournament' fashion.

After the parents are chosen, the reproduction or recombination step can take place to produce offspring for the next generation. Like the selection step, there are variations for this step. If can be as simple as a single point crossover of the binary encoded vectors, where the bits (binary values) are swapped at a random point. Another variant is uniform crossover where each bit is independently chosen to be exchanged between two parents.

Mutation is another important step in GA to produce new individuals for the next generation. It is done to a random individual, with slight modifications on it. Different approaches exist for the mutation step too. One method consists in flipping the bits at random positions with a given mutation rate. Alternatively, the chosen individuals' entire bits get inverted (i.e. 1 becomes zero and vice versa).

New generations produced by mating and mutation are scored, and the process of selection, mating, mutation repeats for multiple generations until a termination criterion is met. It could be that an optimum is found with one of the individuals, or that the maximum number of generations is reached or the best fitness score have reached a plateau (i.e. no new individual is able to produce a better fitness score for consecutive generations since the best was found).

1.4.2.4 Scatter Search

Scatter search (SS) is also an iterative population based global optimization algorithm. While the general idea is analogous to the previously mentioned GA, it differs in requiring a relatively smaller population size, requires lesser degree of randomness (every step in GA potentially has a random element, while in SS only the first step has randomness involved), and makes use of a local optimization method in one of its steps. Classically, scatter search has a 'five-methods' template, and like GA they are all flexible in their execution. This section will outline a basic approach to each of the methods in SS for parameter estimation.

The five-methods are:

i) *Diversification Generation Method,* this is for creating the initial pool of trial solutions (parameter vectors), which could be random and/or be generated using prior information of good solutions.

ii) *Improvement Method,* this is for enhancing the trial solutions, it can be carried out in any manner the user sees fit, but a common approach in this context is to use one of the local minimization methods for parameter estimation.

iii) *Reference Set Update Method,* this is to generate and maintain a reference set for the subsequent method, the solutions in this set must have a good objective function score and be diverse. Solutions chosen to be in reference set are deleted from the pool of solutions.

iv) *Subset Generation Method,* from the reference set, subsets are created. The direct approach would be to create pairs from all the reference solutions.

v) *Solution Combination Method,* using the subsets solutions are combined to create new solutions for the pool. It can be by directly swapping elements in the vectors or using geometry to search in the neighbourhood space or beyond.

In general, SS is executed in the following manner: the pool of solutions is generated and subsequently improved upon using the desired algorithm (they are also scored using the objective function of choice). From the pool a reference set is created, which is used to create subsets. New solutions are generated from the subsets through the combination method, and they are improved upon before being introduced to the pool. Using sets from the pool, the reference set is updated.

If an optimum is not found in the new set, subset generation, solution combination and improvement methods are repeated until a stopping criterion is reached. The same criteria used by GA can be applied here.

1.4.2.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is another iterative, stochastic and population-based optimization method. This method was inspired by the social behaviour of flock of birds or school of fish. PSO emulates the swarm intelligence of birds when they search for food in a group, the strategy employed is for the swarm to follow the bird closest to the food.

PSO follows the GA and SS methodology of initialising a random set of solutions and scoring them with an objective function, then they go through multiple iterations of improving until an optimum is located. However, unlike GA and SS it does not make use of recombination of the parameters to improve on them.

In PSO, each parameter vector is known as a particle, and information on its position and velocity in the parameter space is retained as well as the best solution. Each particle determines its velocity using positional information of the locally best particle in its neighbourhood. Its position is then updated using the velocity.

The velocity is determined with the following equation:

$$v = v + c1 * \text{random} * (pbest - present) + c2 * \text{random} * (nbest - present)$$

and the position is updated with this equation:

$$present = present + v$$

v = velocity

c1/c2 = learning factors

pbest = best scoring parameter vector for the particle

present = present parameter vector

nbest = parameter vector of the neighbourhood best particle

random = random value between 0 and 1

The process of determining the velocity and updating the position is repeated until a stopping criterion is reached, which is the same as in GA and SS. Learning factor is a parameter that the user has to calibrate. Another user specified parameter in this algorithm is the maximum

velocity, where if the velocity calculated for the particle exceeds maximum velocity, the maximum value is used instead. This is to prevent particle from going too far from the search space.

1.4.3 Software Tools for Parameter Estimation

As model building is being widely adopted for biological research, multiple tools capable of performing parameter estimation for biochemical models have been developed. For example, SBML-PET (Zi, 2011; Zi and Klipp, 2006) is a parameter estimation software dedicated to the SBML format of models. It performs parameter estimation using a Stochastic Ranking Evolution Strategy. In its second iteration of the software, it makes use of a message passing interface protocol to parallelize its estimation process, allowing for an increase in speed. SBMLSimulator (Dörr et al., 2014) is a Java based software with a graphical user interface (GUI). It enables users to simulate models in SBML format, as well as to perform parameter estimation with algorithms provided by the EvA2 (Kronfeld et al., 2010) optimization framework. EvA2 provides a wide variety of algorithms within the evolutionary algorithms' family such as GA, PSO, Differential Evolution, and Population Based Incremental Learning. Another software that packs a variety of parameter estimation tools is COPASI (Hoops et al., 2006), this software provides a mix of local and global methods such as Hooke and Jeeves, Levenberg-Marquadt, evolutionary programming, simulated annealing PSO and GA.

1.5 Aims

The construction of kinetic models of cell metabolism for biologists can be challenging. This is due to lack of expertise, or the lack of data needed to generate a model. Despite improvements in technology and a high volume of biological data being generated, metabolomics and enzyme kinetics data are lacking in comparison to other omics data. Available data for a complete network has often been measured in different conditions and/or by different labs. To overcome heterogenous data, some labs might opt to generate the kinetics data themselves, which is often expensive, labour intensive and time consuming. The combination of these factors presents a gatekeeping effect for biologists who are unfamiliar with the world of modelling and the benefits it can bring to research. To address this issue a software, GRaPe 2.0 has been developed. This is aimed to be a software of intuitive usage and to allow users to build models using heterogenous data (metabolomics, fluxomics and proteomics).

In Chapter 2, the software architecture and the framework of kinetic model construction is described. The framework in brief is as follows. To overcome the incomplete information on enzyme rate laws, a reversible approximate rate law, convenience kinetics, is used to describe all enzyme reactions. To fill in the unknown kinetic parameters needed to complete the model, parameter estimation can be performed through a genetic algorithm with the software.

In Chapter 3, application of the software is demonstrated with a kinetic model of the *Saccharomyces cerevisiae* glycolytic network. This model was able to reproduce the original values for metabolite concentrations and fluxes during steady state, which were used to train the model. Additionally, it was able to reproduce the increased flux values typically observed under heat stress conditions for *S. cerevisiae*, which was a separate set of data that was not used in training the model. This work showcased the software's capability in constructing a dynamic metabolic model that can be validated.

The construction of a model is a first step in systems biology research, and following this process, users should be using the model to generate hypotheses, study biological phenomena that can be difficult to measure experimentally, or use it to find the best metabolic engineering

target to produce a biotechnology compound of interest. In Chapter 4, a model of *S. cerevisiae* trehalose metabolism was built and validated using quantitative omics data (metabolites, fluxes and proteins) and GRaPe 2.0. The completed model was used to study the reason behind the rise in glycolytic flux that occurs during heat stress. Trehalose is a compound with commercial value as it possesses protective properties that can prolong life of other compounds. With this in mind, the model was used to determine the best engineering target to increase yield of trehalose in *S. cerevisiae*. Work in this chapter demonstrates that models generated with GRaPe 2.0 can be easily applied to assist bioengineering research.

In Chapter 5, the main outcomes of this work are assessed and discussed, and directions for future research are presented.

Overall, this project aims to produce a software that serves as a gateway modelling tool for biologists that wish to enter systems biology research.

1.6 References

Adiamah, D. a, Handl, J., and Schwartz, J.-M. (2010). Streamlining the construction of large-scale dynamic models using generic kinetic equations. Bioinformatics *26*, 1324–1331.

Ahnert, K., Demidov, D., and Mulansky, M. (2014). Solving Ordinary Differential Equations on GPUs. In Numerical Computations with GPUs, (Cham: Springer International Publishing), pp. 125–157.

Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J., and Jirstrand, M. (2014). Kinetic models in industrial biotechnology - Improving cell factory performance. Metab. Eng. *24*, 38–60.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. a., and Blom, J.G. (2009). Systems biology: Parameter estimation for biochemical models. FEBS J. *276*, 886–902.

Azeloglu, E.U., and Iyengar, R. (2015). Good practices for building dynamical models in systems biology. Sci. Signal. *8*.

Bartocci, E., and Lió, P. (2016). Computational Modeling, Formal Analysis, and Tools for Systems Biology. PLoS Comput. Biol. *12*, 1–22.

Bernardinello, L., and Cindio, F. (1992). A survey of basic net models and modular net classes. pp. 304–351.

Blank, L.M., Kuepfer, L., Sauer, U., Papin, J., Stelling, J., Price, N., Klamt, S., Schuster, S., Palsson, B., Price, N., et al. (2005). Large-scale 13 C-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast. Genome Biol. *6*, R49.

Bornstein, B.J., Keating, S.M., Jouraku, A., and Hucka, M. (2008). LibSBML: an API Library for SBML. Bioinformatics *24*, 880–881.

Breitling, R., Gilbert, D., Heiner, M., and Orton, R. (2008). A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Brief. Bioinform. *9*, 404–421.

Chelouah, R., and Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. Eur. J. Oper. Res. *148*, 335–348.

Chou, I.C., and Voit, E.O. (2009). Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. *219*, 57–83.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). Distributed Optimization by Ants Colonies. Proc. ECAL - Eur. Conf. Artif. Life, Paris, Fr. 134–142.

Costa, R.S., Machado, D., Rocha, I., and Ferreira, E.C. (2010). Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations. Biosystems *100*, 150–157.

Costa, R.S., Veríssimo, A., and Vinga, S. (2014). KiMoSys: a web-based repository of experimental data for KInetic MOdels of biological SYStems. BMC Syst. Biol. *8*, 85.

Costa, R.S., Hartmann, A., and Vinga, S. (2016). Kinetic modeling of cell metabolism for microbial production. J. Biotechnol. *219*, 126–141.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. Mol. Syst. Biol. *7*.

Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. Simulation *79*, 740–747.

Dörr, A., Keller, R., Zell, A., and Dräger, A. (2014). SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. Computation *2*, 246–257.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst. Biol. *2*, 39.

Dräger, A., Rodriguez, N., Dumousseau, M., Dörr, A., Wrzodek, C., Le Novère, N., Zell, A., and Hucka, M. (2011). JSBML: A flexible java library for working with SBML. Bioinformatics *27*, 2167–2168.

Du, B., Zielinski, D.C., Kavvas, E.S., Dräger, A., Tan, J., Zhang, Z., Ruggiero, K.E., Arzumanyan, G.A., and Palsson, B.O. (2016). Evaluation of rate law approximations in bottom-up kinetic models of metabolism. BMC Syst. Biol. *10*, 40.

Duarte, N.C., Becker, S.A., Jamshidi, N., Thiele, I., Mo, M.L., Vo, T.D., Srivas, R., and Palsson, B.O. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc. Natl. Acad. Sci. *104*, 1777–1782.

Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., et al. (2018). The Reactome Pathway Knowledgebase. Nucleic Acids Res. *46*, D649–D655.

Fast, T., Wall, T., and Chen, L. (2007). Java Native Access.

Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007). A genome-scale metabolic reconstruction for

Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol. Syst. Biol. *3*.

Fonseca, L.L., Sánchez, C., Santos, H., and Voit, E.O. (2011). Complex coordination of multi-scale cellular responses to environmental stress. Mol. Biosyst. *7*, 731–741.

Fonseca, L.L., Chen, P.-W., and Voit, E.O. (2012). Canonical Modeling of the Multi-Scale Regulation of the Heat Stress Response in Yeast. Metabolites *2*, 221–241.

Forster, J. (2003). Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network. Genome Res. *13*, 244–253.

Funahashi, A., and Matsuoka, Y. (2008). CellDesigner 3.5: a versatile modeling tool for biochemical networks. Proc. IEEE *96*, 1254–1265.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell *11*, 4241–4257.

Glover, F. (1977). Heuristics For Integer Programming Using Surrogate Constraints. Decis. Sci. *8*, 156–166.

Guijas, C., Montenegro-Burke, J.R., Domingo-Almenara, X., Palermo, A., Warth, B., Hermann, G., Koellensperger, G., Huan, T., Uritboonthai, W., Aisporna, A.E., et al. (2018). METLIN: A Technology Platform for Identifying Knowns and Unknowns. Anal. Chem. *90*, 3156–3164.

Haug, K., Salek, R.M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., Mahendraker, T., Williams, M., Neumann, S., Rocca-Serra, P., et al. (2013). MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res. *41*, D781–D786.

Heinrich, R., and Rapoport, T.A. (1974). A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. Eur. J. Biochem. *42*, 89–95.

Hillmer, R.A. (2015). Systems Biology for Biologists. PLOS Pathog. *11*, e1004786.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., and Woodward, C.S. (2005). SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. ACM Trans. Math. Softw. *31*, 363–396.

Holland, J.H. (1962). Outline for a Logical Theory of Adaptive Systems. J. ACM *9*, 297–314.

Hooke, R., and Jeeves, T.A. (1961). `` Direct Search" Solution of Numerical and Statistical Problems. J. ACM *8*, 212–229.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI--a COmplex PAthway SImulator. Bioinformatics *22*, 3067–3074.

Hübner, K., Sahle, S., and Kummer, U. (2011). Applications and trends in systems biology in biochemistry. FEBS J. *278*, 2767–2857.

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

Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., et al. (2007). Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations. Science (80-.). *316*, 593–597.

Jamshidi, N., and Palsson, B.Ø. (2008). Formulating genome-scale kinetic models in the post-genome era. Mol. Syst. Biol. *4*, 171.

Jarnuczak, A.F., Albornoz, M.G., Eyers, C.E., Grant, C.M., and Hubbard, S.J. (2018). A quantitative and temporal map of proteostasis during heat shock in *Saccharomyces cerevisiae*. Mol. Omi. *14*, 37–52.

Kacser, H., and Burns, J.A. (1973). The control of flux. Symp. Soc. Exp. Biol. *27*, 65–104.

Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (2011). ConsensusPathDB: toward a more complete picture of cell biology. Nucleic Acids Res. *39*, D712–D717.

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. *22*, 437–467.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, (IEEE), pp. 1942–1948.

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

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

King, E.L., and Altman, C. (1956). A Schematic Method of Deriving the Rate Laws for Enzyme-Catalyzed Reactions. J. Phys. Chem. *60*, 1375–1378.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. Science (80-.). *220*, 671–680.

Kitano, H. (2002). Systems biology: a brief overview. Science *295*, 1662–1664.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009a). Parameter Estimation. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 152–164.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009b). Mathematics. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 449–473.

Koch, I., Junker, B.H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics *21*, 1219–1226.

Krebs, O., Golebiewski, M., Kania, R., and Mir, S. (2007). SABIO-RK: a data warehouse for biochemical reactions and their kinetics. J. Integr. Bioinform.

Kronfeld, M., Planatscher, H., and Zell, A. (2010). The EvA2 optimization framework. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) *6073 LNCS*, 247–250.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least. Q. Appl. Math. *2*, 164–168.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci. *101*, 4781–4786.

Liebermeister, W., and Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor. Biol. Med. Model. *3*, 41.

Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010). Modular rate laws for enzymatic reactions: thermodynamics, elasticities and implementation. Bioinformatics *26*, 1528–1534.

Link, H., Christodoulou, D., and Sauer, U. (2014). Advancing metabolic models with kinetic information. Curr. Opin. Biotechnol. *29*, 8–14.

Machado, D., Costa, R.S., Rocha, M., Ferreira, E.C., Tidor, B., and Rocha, I. (2011). Modeling formalisms in Systems Biology. AMB Express *1*, 45.

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: A native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics *22*, 1406–1407.

Matsuda, F., Kinoshita, S., Nishino, S., Tomita, A., and Shimizu, H. (2017). Targeted proteome analysis of single-gene deletion strains of Saccharomyces cerevisiae lacking enzymes in the central carbon metabolism. PLoS One *12*, e0172742.

Mensonides, F.I.C., Schuurmans, J.M., Teixeira de Mattos, M.J., Hellingwerf, K.J., and Brul, S. (2002). The metabolic response of Saccharomyces cerevisiae to continuous heat stress. Mol. Biol. Reports JT - Mol. Biol. Reports *29*, 103–6 PL–Netherlands PT–Journal Article.

Nelder, J.A., and Mead, R. (1965). A Simplex Method for Function Minimization. Comput. J. *7*, 308–313.

Nocedal, J., and Wright, S. (1999). Numerical Optimization (New York: Springer).

Novere, N. Le, Hucka, M., and Mi, H. (2009). The Systems Biology Graphical Notation. Nat. Biotechnol. *27*, 735–742.

Le Novère, N., Hucka, M., Hoops, S., Keating, S., Sahle, S., and Wilkinson, D. (2008). Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Nat. Preced. 1–38.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *27*, 29–34.

Orth, J.D., Conrad, T.M., Na, J., Lerman, J. a, Nam, H., Feist, A.M., and Palsson, B.Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. Mol. Syst. Biol. *7*, 535.

Paiva, C.L.A., and Panek, A.D. (1996). Biotechnological Applications of the Disaccharide Trehalose. pp. 293–314.

Park, T.-Y., and Froment, G.F. (1998). A hybrid genetic algorithm for the estimation of parameters in detailed kinetic models. Comput. Chem. Eng. *22*, S103–S110.

Parrou, J.L., Teste, M.A., and François, J. (1997). Effects of various types of stress on the metabolism of reserve carbohydrates in Saccharomyces cerevisiae: Genetic evidence for a stress-induced recycling of glycogen and trehalose. Microbiology *143*, 1891–1900.

Petri, C.A. (1962). Kommunikation mit Automaten (Communication with Automata). University of Bonn.

Postmus, J., Canelas, A.B., Bouwman, J., Bakker, B.M., Van Gulik, W., De Mattos, M.J.T., Brul, S., and Smits, G.J. (2008). Quantitative analysis of the high temperature-induced glycolytic flux increase in Saccharomyces cerevisiae reveals dominant metabolic regulation. J. Biol. Chem. *283*,

23524–23532.

Puig-Castellví, F., Alfonso, I., Piña, B., and Tauler, R. (2015). A quantitative 1H NMR approach for evaluating the metabolic response of Saccharomyces cerevisiae to mild heat stress. Metabolomics *11*, 1612–1625.

Rodriguez, N., Donizelli, M., and Le Novère, N. (2007). SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). BMC Bioinformatics *8*, 79.

Roser, B. (1991). Trehalose, a new approach to premium dried foods. Trends Food Sci. Technol. *2*, 166–169.

Savageau, M.A. (1976). Biochemical Systems Analysis (Reading, MA: Addison-Wesley Pub. Co.).

Schaff, J., Slepchenko, B., and Loew, L. (2000). Physiological modeling with virtual cell framework. Methods Enzymol. *321*, 1–23.

Schiraldi, C., Di Lernia, I., and De Rosa, M. (2002). Trehalose production: Exploiting novel approaches. Trends Biotechnol. *20*, 420–425.

Schmidt, T., Samaras, P., Frejno, M., Gessulat, S., Barnert, M., Kienegger, H., Krcmar, H., Schlegl, J., Ehrlich, H.-C., Aiche, S., et al. (2018). ProteomicsDB. Nucleic Acids Res. *46*, D1271–D1281.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. *32*, D431-3.

Smallbone, K., Simeonidis, E., Swainston, N., and Mendes, P. (2010). Towards a genome-scale kinetic model of cellular metabolism. BMC Syst. Biol. *4*, 6.

Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., et al. (2013). A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett. *587*, 2832–2841.

Strassburg, K., Walther, D., Takahashi, H., Kanaya, S., and Kopka, J. (2010). Dynamic transcriptional and metabolic responses in yeast adapting to temperature stress. OMICS *14*, 249–259.

Sun, J., Garibaldi, J.M., and Hodgman, C. (2012). Parameter estimation using meta-heuristics in systems biology: a comprehensive review. IEEE/ACM Trans. Comput. Biol. Bioinform. *9*, 185–202.

Varma, A., and Palsson, B.O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. Bio/Technology *12*, 994–998.

Viant, M.R., Kurland, I.J., Jones, M.R., and Dunn, W.B. (2017). How close are we to complete

annotation of metabolomes? Curr. Opin. Chem. Biol. *36*, 64–69.

Visser, D., and Heijnen, J.J. (2003). Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab. Eng. *5*, 164–176.

Vizcaíno, J.A., Csordas, A., Del-Toro, N., Dianes, J.A., Griss, J., Lavidas, I., Mayer, G., Perez-Riverol, Y., Reisinger, F., Ternent, T., et al. (2016). 2016 update of the PRIDE database and its related tools. Nucleic Acids Res. *44*, D447–D456.

Voit, E.O. (2003). Biochemical and genomic regulation of the trehalose cycle in yeast: Review of observations and canonical model analysis. J. Theor. Biol. *223*, 55–78.

Wang, M., Herrmann, C.J., Simonovic, M., Szklarczyk, D., and von Mering, C. (2015). Version 4.0 of PaxDb: Protein abundance data, integrated across model organisms, tissues, and cell-lines. Proteomics *15*, 3163–3168.

Wolkenhauer, O., and Mesarović, M. (2005). Feedback dynamics and cell function: Why systems biology is called Systems Biology. Mol. Biosyst. *1*, 14.

Yen, J., Liao, J.C., Lee, B.L.B., and Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. Syst. Man Cybern. Part B Cybern. a Publ. IEEE Syst. Man Cybern. Soc. *28*, 173–191.

Yurkovich, J.T., and Palsson, B.O. (2018). Quantitative -omic data empowers bottom-up systems biology. Curr. Opin. Biotechnol. *51*, 130–136.

Zhang, Z., Shen, T., Rui, B., Zhou, W., Zhou, X., Shang, C., Xin, C., Liu, X., Li, G., Jiang, J., et al. (2015). CeCaFDB: a curated database for the documentation, visualization and comparative analysis of central carbon metabolic flux distributions explored by 13C-fluxomics. Nucleic Acids Res. *43*, D549–D557.

Zi, Z. (2011). SBML-PET-MPI: A parallel parameter estimation tool for Systems Biology Markup Language based models. Bioinformatics *27*, 1028–1029.

Zi, Z., and Klipp, E. (2006). SBML-PET: A Systems Biology Markup Language-based parameter estimation tool. Bioinformatics *22*, 2704–2705.

Zuberek, W.M. (1991). Timed Petri nets definitions, properties, and applications. Microelectron. Reliab. *31*, 627–644.

Chapter 2: GRaPe 2.0: Gateway for Building Kinetic Models of Cell Metabolism

2.1 Abstract

As number of biological data in public domain grows it should allow for more computational model of biological networks to be built. Size of models should also ideally be increasing in size. Unfortunately, this is not often the case for kinetic models of cell metabolism. Several factors prevent this from happening, incomplete kinetic data, from rate laws to parameter values, and available modelling tools have steep learning curve, presenting a barrier for biologists intending to construct models for research.

To address these challenges, GRaPe 2.0 was developed. GRaPe 2.0 is a software tool that does all difficult model building process in the background automatically, requiring only the basic information from the users about the network. Information include, metabolites, enzymes and reactions of the network. To fill in the missing parameter values, users can provide measured metabolomics or fluxomics data, with it the software can run a genetic algorithm to estimate the parameters. This software will serve as a gateway modelling tool for biologists.

2.2 Introduction

Over the past decade, applications of computer and mathematics in aiding biological research have been growing, especially when it comes to studying biochemical networks. This interdisciplinary research spawned a new field, systems biology, which tackles biological problems from a holistic approach. As the field grows, so does the number of techniques and software suites developed to study biological problems (Bartocci and Lió, 2016).

For the construction of kinetic models alone, there are various software developed for different stages of model building. For example SBMLeditor is a tool for creating and editing models down to its minute details (Rodriguez et al., 2007), from determining the substrates, reactions, rate laws and every value that is associated with the respective entities. There are also tools that support the estimation of missing kinetic parameters in models such as SBML-PET (Zi and Klipp, 2006) and SBMLSimulator, which allows for parameter estimation and simulation of models (Dörr et al., 2014). CellDesigner lets the users create, visualise and simulate models (Funahashi and Matsuoka, 2008), as well as supports plugins for additional functions, such as SBMLsqueezer that helps users select rate laws for the reactions (Dräger et al., 2008). COPASI allows for creation, parameter estimation and simulation of models (Hoops et al., 2006). On top of software tools aimed at kinetic models, some biologists opt for more control over the entire modelling process by using programming languages such as MATLAB and Python.

The variety of software also comes with a varying level of expertise requirements, such as the need to have programming background to use the software effectively. Some tools might require an extensive reading of the manual or some form of training before being able to dive into model building, despite having a graphical user interface. Additionally, the majority of the software grants users very detailed control over every part of the model, requiring their input at every step of the process. This would increase the complexity of the model building process and the time needed for its completion as the model gets bigger, as well as increase the probability of introducing errors in the models. All of these hurdles could result in an unwanted gatekeeping effect for biologists who want to join the modelling community if they fail to secure bioinformatics collaborators.

On top of software tools, another element needed for building models is the information necessary for the model. While there are various databases that pool together data needed for kinetic models, such as BRENDA (Schomburg et al., 2004), SABIO-RK (Krebs et al., 2007), and KEGG (Ogata et al., 1999), there are still either missing data or data from heterogenous conditions. The majority of the missing information could be the kinetic rate laws for reactions or kinetic parameters.

The combination of incomplete data and unfamiliar tools can be intimidating and confusing for beginners who wants to incorporate modelling into their research. To address this issue, GRaPe (Adiamah et al., 2010) was previously developed, which aimed to simplify the process of building kinetic models for cell metabolism. It did this by automatically generating rate equations in the form of generic Michaelis-Menten equations (King and Altman, 1956) and estimated the missing kinetic parameters using either a genetic algorithm (Holland, 1962) or Levenberg–Marquardt method (Nocedal and Wright, 1999).

Here, I present an enhanced version of GRaPe that improves upon the earlier version as well as introducing features not found previously. GRaPe 2.0 with its ease of use and the allowance for heterogenous data integration aims to be a gateway modelling tool for biologists interested in metabolic modelling.

2.3 Software Features

The following is a summary of the main features in GRaPe 2.0:

1. Import and export of models in Systems Biology Markup Language (SBML) format (Hucka et al., 2003), a commonly accepted standard for the storage of biochemical reaction models. Newly built models are all saved in SBML format, which can then be imported by other software that supports this format if desired.
2. Model construction through a polished and intuitive graphical user interface (Figure 2.2) or command line interface (new feature) that parses through tab separated text file. All that is necessary for the model construction are the metabolites, reactions involved and the enzymes that catalyses them, and if known, the initial concentrations of the metabolite and enzymes. As

models are built following the SBML standard, users are allowed to set boundary conditions for input and output metabolites of network if desired.

3. Reactions in the model support up to three substrates and products (previous version was limited to two only). It is also possible to include enzyme modifiers, either activators or inhibitors, or both if necessary for the reaction (new feature).
4. Parameter estimation of all the kinetic parameters in the model with a genetic algorithm (GA) (Holland, 1962) using steady state or time series data (previous version was limited to steady state data). It requires metabolite and/or flux data, even allowing for incomplete information where not every metabolite or flux is present, as well as data from different sources obtained under different conditions. The previous version estimated the kinetic parameters locally (estimating them by individual reaction), the updated version estimates all the parameters from a systemwide approach (entire model is solved and estimated together). Additionally, the previous version only estimated some of the parameters, giving the remaining parameters a value of one, whereas the updated version finds all the kinetic parameters.
5. Parameter estimation function now accommodates the ability to make use of data gathered from multiple conditions; i.e. under different conditions, cells would express a different protein make up, possibly resulting in varying metabolite and flux outputs. GRaPe is now able to make use of this form of information to impose further constraints on the estimation process and create a 'stiffer' model if desired.
6. Through Java's multithreading feature, users can increase the number of CPUs used during parameter estimation step to solve multiple models concurrently and decrease the time needed for this process.

2.4 Methods

2.4.1 Kinetic Rate Law for Reactions

All the enzymatic reactions modelled using GRaPe 2.0 are generated using the convenience kinetics rate law (Liebermeister and Klipp, 2006, Equation 1), a generalized Michaelis Menten rate law. This rate law assumes a reversible random binding order for all substrates. Convenience kinetics is chosen as its simplicity allows for easy addition and removal of substrates and products, with a linear scaling in the number of kinetic parameters involved. Additionally, it is able to model enzyme regulation without major changes to the equation.

Equation 1:

$$v = E_{tot} \cdot f_{reg} \frac{k_+^{cat} \Pi(\frac{S_i}{K_{m,S_i}})^{n_i} - k_-^{cat} \Pi(\frac{P_j}{K_{m,P_j}})^{n_j}}{\Pi\left(1 + \left(\frac{S_i}{K_{m,S_i}}\right) + \cdots + \left(\frac{S_i}{K_{m,S_i}}\right)^{n_i}\right) + \Pi\left(1 + \left(\frac{P_j}{K_{m,P_j}}\right) + \cdots + \left(\frac{P_j}{K_{m,P_j}}\right)^{n_j}\right) - 1}$$

v = Rate of reaction/flux

E_{tot} = Enzyme concentration

f_{reg} = Regulatory prefactor

S = Substrate concentration

P = Product concentration

K_m = Michaelis-Menten constant

$K^{cat}_{+/-}$ = Forward and reverse turnover rate

The regulatory prefactor are represented in two of the following forms:

$1 + \frac{d}{K^A}$ for activator regulatory effect

d = Modifier concentration

K^I = Inhibitor constant

K^A = Activator constant

$\frac{K^I}{K^I + d}$ for inhibitory regulatory effect

2.4.2 Parameter Estimation or Model Training

Kinetic parameters (K_m, K^A, K^I) in the models built using GRaPe 2.0 are estimated from data on the metabolites and/or fluxes that are either at steady state or time series. This set of information (metabolites/fluxes) are known as training data as they are the values the model is expected to produce from simulation after the process of parameter estimation (or training the model). Ideally there should be a separate set of data for testing or validating the model, whereby the model is manipulated in some form to be in a different condition than the one used for training. The model manipulation could be that the enzyme concentrations are changed or 'deleted' by setting the values to zero, and the model's simulation output is compared with the aforementioned testing data. This is done to ensure the model that was trained is not overfitted; overfitting is a phenomenon where the model is trained to only generate output for one condition, this would make the model not usable to test new hypotheses (Klipp et al., 2009a).

Parameter estimation in GRaPe 2.0 is done using a genetic algorithm (Figure 2.1), an optimization algorithm that was inspired by Darwin's evolutionary principles. The system would first initiate a population made up of different sets of kinetic parameters, each making up a model, also known as "chromosome" in the context of GA. From the initial population, the chromosomes would undergo multiple generations of crossovers and mutations to finally produce the model that would best fit to the training data that was given. To determine the best model, each chromosome's fitness is measured by comparing the simulated values against the training values. The quality of fit would be scored using the mean absolute percentage error (MAPE, Equation 2), as it would allow for a wider scale of values to be covered, whereas the commonly used root mean squared error would emphasize the large errors due to squaring.

Equation 2:

$$\Sigma \left| \frac{Observed - Simulated}{Observed} \right| \cdot \frac{100\%}{n}$$

Observed = Training data's value for metabolite and/or flux.

Simulated = Model simulated value for metabolite and/or flux.

n = Total number of metabolites and fluxes in the model.

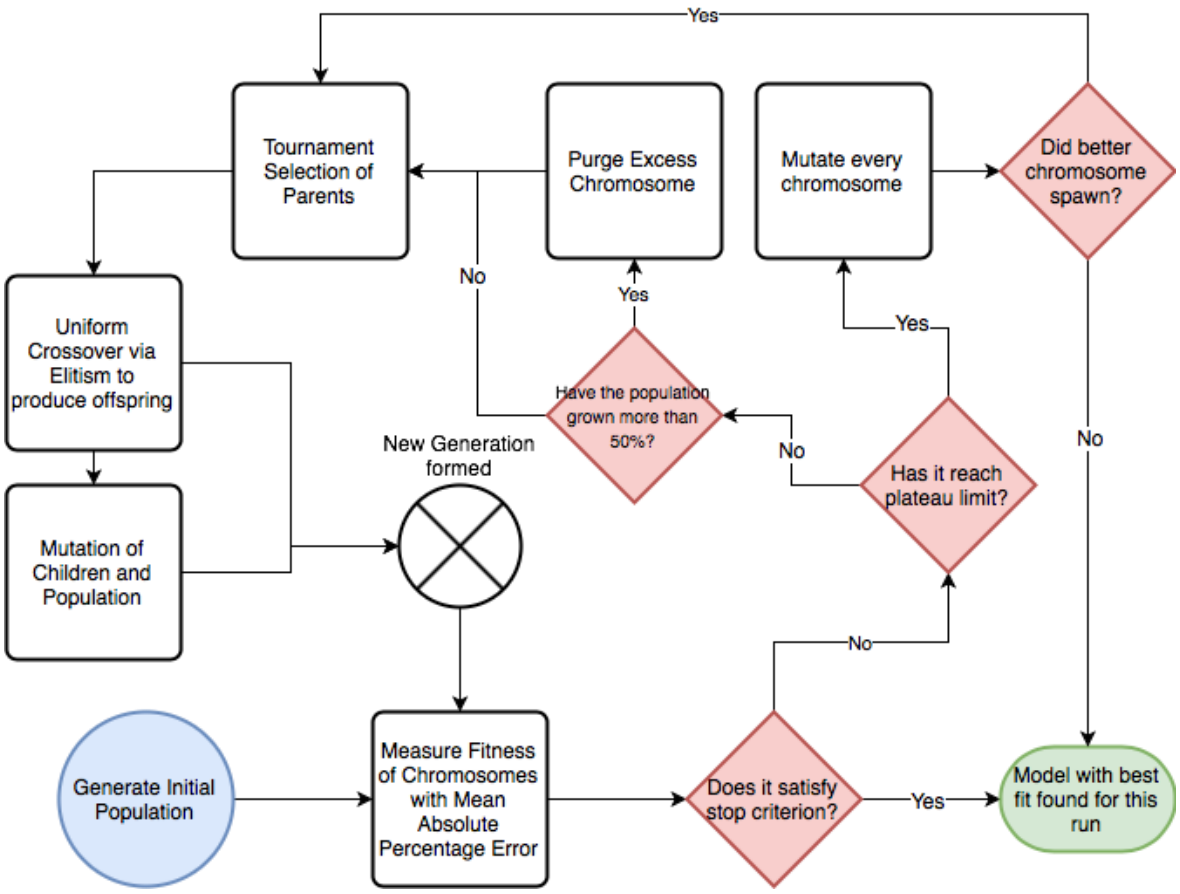

Figure 2.1: **GRaPe 2.0's genetic algorithm decision flowchart.** Diamonds signify decision points. Rectangles signify tasks.

2.4.2.1 Details of the Genetic Algorithm Implemented

1. **Initial Population:** Initial population of parameter values (chromosomes) are generated randomly for the first generation. Unlike most GA, binary encoding is not used for chromosomes, instead every single kinetic parameter in the chromosome is encoded as an exponent for a base value of 10 (decadic logarithm value). The reason for this encoding is that exact values for a kinetic parameter is not a priority, rather the orders of magnitude.
2. **Fitness Evaluation:** The kinetic parameter values of each chromosomes are inserted into the model, and the model is solved. The output of the model is compared to the fitting data, and the fitness value of each chromosome is scored using MAPE.

3. **Crossover:** Next generation is spawned by choosing random chromosomes parents using tournament selection; tournament selection is carried out by sampling population at random several times (depending on the population size) to select fit parents. After parents are selected, crossover is done using uniform crossover to produce offspring for the next generation. In this version of uniform crossover, a fixed ratio of parameters is exchanged instead of exchange at fixed points.

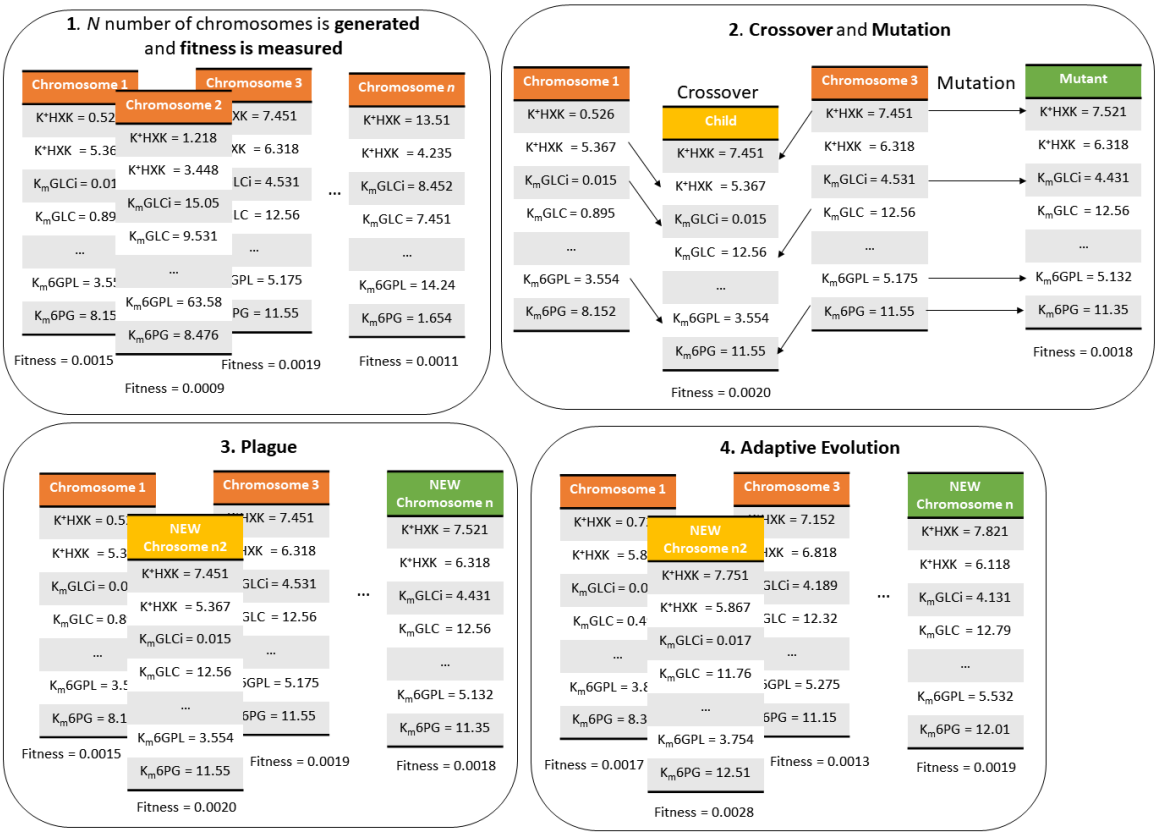

Figure 2.2: **GRaPe 2.0's genetic algorithm's principles visualisation.** Only subset of kinetic parameters from a model of pentose phosphate pathway is displayed. In the mutation step, several parameter values are updated at random. In the plague step, Chromosome 2 and Chromosome *n* are removed from the population as they possess low fitness score. In adaptive evolution, every single chromosome is mutated, and fitness scores are updated.

4. **Mutation:** Following crossover, random chromosomes are chosen to be mutated. Mutation is carried out by randomly increasing or decreasing one or several parameters in the chromosome.

5. **Plague:** A plague function is also written into this GA, whereby after a set amount of generations determined by the user, the population would get reduced to the initial size, purging chromosomes with low fitness score.
6. **Adaptive evolution:** Users can also set a plateau limit, which is the maximum number of generations for when the highest fitness score stays the same. After reaching the plateau limit, there is an adaptive function in the GA where every chromosome is mutated to try and escape this plateau,
7. **Ending criteria:** The ending criteria in a perfect scenario would be when a chromosome is able to perfectly replicate the fitting data as its output, resulting in zero error score. GA would also come to an end after the user determined number of generations is completed. Alternatively, if adaptive evolution fails to improve the fitness value of the fittest individual after the plateau, the search comes to an end, spawning a model from the fittest chromosome.
8. **Strengths of GA implemented:** Use of decadic logarithm value in the kinetic parameter value generation allows greater parameter search space to be covered. Number of random elements in each operator of the GA is increased by choosing at random how each chromosome is processed. The addition adaptive evolution greatly increases the chance of escaping local optima of the parameter search space. This is especially useful in building large scale kinetic models that contain multiple local optima where most parameter estimation methods would be trapped in.

2.5 Implementation

2.5.1 Software Architecture

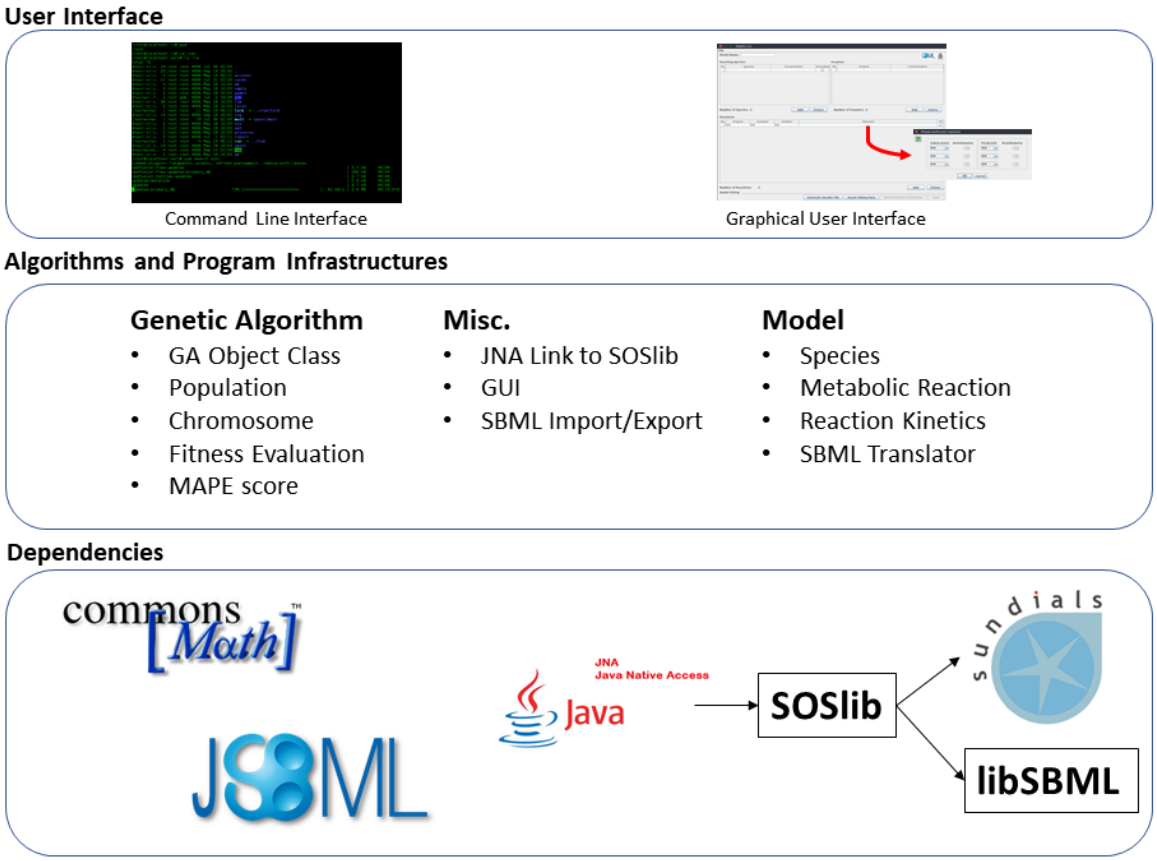

Figure 2.3: **Architecture of GRaPe 2.0.** The software is written in a modular nature, where each main function is written in separate packages in Java, such as the main ones for Model, Genetic Algorithm and Graphical User Interface (GUI). This allows for the software to be used via a command line interface (CLI) if preferred. In order for the software to work, three Java libraries (Apache Common Maths, JSBML, Java Native Access) and three C libraries (SBML Odesolver, SUNDIALS, libSBML) are required.

I built GRaPe 2.0 using mainly in Java programming language (version 8), with small parts written in C. It is written as a modular program, where the interface to the software and the algorithms that power it are all in separate packages. It has been tested in both Linux and MacOS.

This new version gives the user the ability to build models and run parameter estimation through the command line interface (CLI) if needed, which allows the software to be used on a

high-performance computing cluster with ease. To build models through the CLI, users can submit a tab separated file according to the format shown in Figure 3

	A	B	C	D	E	F	G	H
1	#METABOLITES	CONC:	B.C.					
2	GLCo	50	TRUE					
3	GLCi	0.1	FALSE					
4	G6P	3.8	FALSE					
5	F6P	0.74	FALSE					
6	F16P	11.8	FALSE					
7	ATP	4.29	FALSE					
8	ADP	1.29	FALSE					
9								
10	#ENZYMES							
11	GLT	0.002						
12	HXK	0.013						
13	PGI	0.15						
14	PFK	0.16						
15								
16	#REACTION	ACTIVATOI	INHIBITORS					
17	GLT	N/A	G6P	GLCo	=	GLCi		
18	HXK	N/A	T6P	GLCi	ATP	=	G6P	ADP
19	PGI	N/A	N/A	G6P	=	F6P		
20	PFK	N/A	ATP	F6P	ATP	=	F16P	ADP

Figure 2.4: **Example Tab Separated file for building models with GRaPe 2.0 through command line interface**. NOTE: this is an example made in a spreadsheet for clarity, it should be in a text file. The text is read in row by row by the software. A '#' sign denotes start of a new element being read into the system (metabolites, enzymes, reactions). 'B.C' stands for boundary condition, a function in SBML that makes a metabolite's concentration fixed if stated to be 'TRUE' (and vice versa), becoming either an output or input for the system being modelled. The column following either metabolites or enzymes are concentrations for the respective compounds. For the reactions, users must first set an enzyme that catalyses the reaction, and if needed activators and/or inhibitors for the reaction in the proceeding columns. In the following columns, substrates come first, then a '=' sign to separate them from the products, which comes in the next columns. For stoichiometries of reactions, users can add numbers followed by '*' preceding the substrates/products (e.g. 2*G6P)

On top of the CLI, a graphical user interface (GUI) is also built into GRaPe 2.0 for users that prefer interaction through a GUI. The GUI is all written using the Java Swing framework. It is more polished and compact compared to the previous version, making it more intuitive and welcoming for new users.

2.5.2 Dependencies

Models built in GRaPe 2.0 are all in SBML format; for this the software uses JSBML (Dräger et al., 2011) to import and export SBML data files. Within the software itself is an additional translator to allow the data structures to be used in the genetic algorithm. The translator takes advantage of the Systems Biology Ontology (Courtot et al., 2011) to identify every compound's role within a reaction, such as it being an enzyme, substrate, inhibitor, etc.

For the solving of models, GRaPe 2.0 makes use of the SBML ODE Solver Library (SOSlib, Machné et al., 2006), which itself relies on libSBML (Bornstein et al., 2008) for parsing the models and SUNDIALS (Hindmarsh et al., 2005) for performing integrations. To allow for a C library to interface with GRaPe 2.0 that is written in Java, Java Native Access (Fast et al., 2007) is used along with a small C shared library written specifically for this purpose.

The Apache Common Math's library is also used for performing linear algebra.

2.5.3 Model processing

As mentioned earlier, models built using GRaPe 2.0 are made up of three major elements: the metabolites or reacting species, the enzymes and the reactions.

2.5.3.1 Metabolites

Metabolites are the compounds produced and consumed in a metabolic model. This means their values vary as the model is simulated until it reaches a steady state (if there is one). Users are allowed to give the metabolites in the model an initial value, else they default to a value of one. As allowed by the SBML format, users can determine its boundary condition to be true or false; if set to true, its concentration would be fixed independently of internal production or consumption of the metabolite, and vice versa. This would usually mean that the metabolite is an input or output for the system being modelled.

2.5.3.2 Enzymes

Enzymes are the proteins responsible for catalysing metabolic reactions in the model, which are usually constants that can be determined by users through their concentration values. In the event that their values are not known, they can be set to a value of one, which would result in them not having an effect in the equation (a concentration can be left out but an enzyme must still be

named). However, if their values are known, they can be correlated to the state of the system being modelled; for example, under different conditions such as growth rate, it is possible for the cells to express different amounts for certain enzymes. Alternatively, a value of zero can be given to the enzymes, to represent a deletion for the associated reactions.

2.5.3.3 Reactions

Reactions transform metabolites into each other. Users would have to set an enzyme for every reaction; if it is a pseudo reaction, a placeholder enzyme can be used instead. It is also possible to associate activator and/or inhibitor modifiers to reactions. Metabolites form either a substrate or a product in the reaction. A chain of enzyme catalysed reactions should form a complete network, from input metabolite(s) to output metabolite(s).

2.6 References

Adiamah, D. a, Handl, J., and Schwartz, J.-M. (2010). Streamlining the construction of large-scale dynamic models using generic kinetic equations. Bioinformatics *26*, 1324–1331.

Ahnert, K., Demidov, D., and Mulansky, M. (2014). Solving Ordinary Differential Equations on GPUs. In Numerical Computations with GPUs, (Cham: Springer International Publishing), pp. 125–157.

Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J., and Jirstrand, M. (2014). Kinetic models in industrial biotechnology - Improving cell factory performance. Metab. Eng. *24*, 38–60.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. a., and Blom, J.G. (2009). Systems biology: Parameter estimation for biochemical models. FEBS J. *276*, 886–902.

Azeloglu, E.U., and Iyengar, R. (2015). Good practices for building dynamical models in systems biology. Sci. Signal. *8*.

Bartocci, E., and Lió, P. (2016). Computational Modeling, Formal Analysis, and Tools for Systems Biology. PLoS Comput. Biol. *12*, 1–22.

Bernardinello, L., and Cindio, F. (1992). A survey of basic net models and modular net classes. pp. 304–351.

Blank, L.M., Kuepfer, L., Sauer, U., Papin, J., Stelling, J., Price, N., Klamt, S., Schuster, S., Palsson, B., Price, N., et al. (2005). Large-scale 13 C-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast. Genome Biol. *6*, R49.

Bornstein, B.J., Keating, S.M., Jouraku, A., and Hucka, M. (2008). LibSBML: an API Library for SBML. Bioinformatics *24*, 880–881.

Breitling, R., Gilbert, D., Heiner, M., and Orton, R. (2008). A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Brief. Bioinform. *9*, 404–421.

Chelouah, R., and Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. Eur. J. Oper. Res. *148*, 335–348.

Chou, I.C., and Voit, E.O. (2009). Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. *219*, 57–83.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). Distributed Optimization by Ants Colonies. Proc. ECAL - Eur. Conf. Artif. Life, Paris, Fr. 134–142.

Costa, R.S., Machado, D., Rocha, I., and Ferreira, E.C. (2010). Hybrid dynamic modeling of

Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations. Biosystems *100*, 150–157.

Costa, R.S., Veríssimo, A., and Vinga, S. (2014). KiMoSys: a web-based repository of experimental data for KInetic MOdels of biological SYStems. BMC Syst. Biol. *8*, 85.

Costa, R.S., Hartmann, A., and Vinga, S. (2016). Kinetic modeling of cell metabolism for microbial production. J. Biotechnol. *219*, 126–141.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. Mol. Syst. Biol. *7*.

Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. Simulation *79*, 740–747.

Dörr, A., Keller, R., Zell, A., and Dräger, A. (2014). SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. Computation *2*, 246–257.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst. Biol. *2*, 39.

Dräger, A., Rodriguez, N., Dumousseau, M., Dörr, A., Wrzodek, C., Le Novère, N., Zell, A., and Hucka, M. (2011). JSBML: A flexible java library for working with SBML. Bioinformatics *27*, 2167–2168.

Du, B., Zielinski, D.C., Kavvas, E.S., Dräger, A., Tan, J., Zhang, Z., Ruggiero, K.E., Arzumanyan, G.A., and Palsson, B.O. (2016). Evaluation of rate law approximations in bottom-up kinetic models of metabolism. BMC Syst. Biol. *10*, 40.

Duarte, N.C., Becker, S.A., Jamshidi, N., Thiele, I., Mo, M.L., Vo, T.D., Srivas, R., and Palsson, B.O. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc. Natl. Acad. Sci. *104*, 1777–1782.

Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., et al. (2018). The Reactome Pathway Knowledgebase. Nucleic Acids Res. *46*, D649–D655.

Fast, T., Wall, T., and Chen, L. (2007). Java Native Access.

Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007). A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol.

Syst. Biol. *3*.

Fonseca, L.L., Sánchez, C., Santos, H., and Voit, E.O. (2011). Complex coordination of multi-scale cellular responses to environmental stress. Mol. Biosyst. *7*, 731–741.

Fonseca, L.L., Chen, P.-W., and Voit, E.O. (2012). Canonical Modeling of the Multi-Scale Regulation of the Heat Stress Response in Yeast. Metabolites *2*, 221–241.

Forster, J. (2003). Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network. Genome Res. *13*, 244–253.

Funahashi, A., and Matsuoka, Y. (2008). CellDesigner 3.5: a versatile modeling tool for biochemical networks. Proc. IEEE *96*, 1254–1265.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell *11*, 4241–4257.

Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS. Decis. Sci. *8*, 156–166.

Guijas, C., Montenegro-Burke, J.R., Domingo-Almenara, X., Palermo, A., Warth, B., Hermann, G., Koellensperger, G., Huan, T., Uritboonthai, W., Aisporna, A.E., et al. (2018). METLIN: A Technology Platform for Identifying Knowns and Unknowns. Anal. Chem. *90*, 3156–3164.

Haug, K., Salek, R.M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., Mahendraker, T., Williams, M., Neumann, S., Rocca-Serra, P., et al. (2013). MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res. *41*, D781–D786.

Heinrich, R., and Rapoport, T.A. (1974). A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. Eur. J. Biochem. *42*, 89–95.

Hillmer, R.A. (2015). Systems Biology for Biologists. PLOS Pathog. *11*, e1004786.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., and Woodward, C.S. (2005). SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. ACM Trans. Math. Softw. *31*, 363–396.

Holland, J.H. (1962). Outline for a Logical Theory of Adaptive Systems. J. ACM *9*, 297–314.

Hooke, R., and Jeeves, T.A. (1961). `` Direct Search" Solution of Numerical and Statistical Problems. J. ACM *8*, 212–229.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and

Kummer, U. (2006). COPASI--a COmplex PAthway SImulator. Bioinformatics *22*, 3067–3074.

Hübner, K., Sahle, S., and Kummer, U. (2011). Applications and trends in systems biology in biochemistry. FEBS J. *278*, 2767–2857.

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

Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., et al. (2007). Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations. Science (80-.). *316*, 593–597.

Jamshidi, N., and Palsson, B.Ø. (2008). Formulating genome-scale kinetic models in the post-genome era. Mol. Syst. Biol. *4*, 171.

Jarnuczak, A.F., Albornoz, M.G., Eyers, C.E., Grant, C.M., and Hubbard, S.J. (2018). A quantitative and temporal map of proteostasis during heat shock in *Saccharomyces cerevisiae*. Mol. Omi. *14*, 37–52.

Kacser, H., and Burns, J.A. (1973). The control of flux. Symp. Soc. Exp. Biol. *27*, 65–104.

Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (2011). ConsensusPathDB: toward a more complete picture of cell biology. Nucleic Acids Res. *39*, D712–D717.

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. *22*, 437–467.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, (IEEE), pp. 1942–1948.

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

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

King, E.L., and Altman, C. (1956). A Schematic Method of Deriving the Rate Laws for Enzyme-Catalyzed Reactions. J. Phys. Chem. *60*, 1375–1378.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. Science (80-.). *220*, 671–680.

Kitano, H. (2002). Systems biology: a brief overview. Science *295*, 1662–1664.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009a). Parameter Estimation. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 152–164.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009b). Mathematics. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 449–473.

Koch, I., Junker, B.H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics *21*, 1219–1226.

Krebs, O., Golebiewski, M., Kania, R., and Mir, S. (2007). SABIO-RK: a data warehouse for biochemical reactions and their kinetics. J. Integr. Bioinform.

Kronfeld, M., Planatscher, H., and Zell, A. (2010). The EvA2 optimization framework. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) *6073 LNCS*, 247–250.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least. Q. Appl. Math. *2*, 164–168.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci. *101*, 4781–4786.

Liebermeister, W., and Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor. Biol. Med. Model. *3*, 41.

Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010). Modular rate laws for enzymatic reactions: thermodynamics, elasticities and implementation. Bioinformatics *26*, 1528–1534.

Link, H., Christodoulou, D., and Sauer, U. (2014). Advancing metabolic models with kinetic information. Curr. Opin. Biotechnol. *29*, 8–14.

Machado, D., Costa, R.S., Rocha, M., Ferreira, E.C., Tidor, B., and Rocha, I. (2011). Modeling formalisms in Systems Biology. AMB Express *1*, 45.

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: A native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics *22*, 1406–1407.

Matsuda, F., Kinoshita, S., Nishino, S., Tomita, A., and Shimizu, H. (2017). Targeted proteome analysis of single-gene deletion strains of Saccharomyces cerevisiae lacking enzymes in the central

carbon metabolism. PLoS One *12*, e0172742.

Mensonides, F.I.C., Schuurmans, J.M., Teixeira de Mattos, M.J., Hellingwerf, K.J., and Brul, S. (2002). The metabolic response of Saccharomyces cerevisiae to continuous heat stress. Mol. Biol. Reports JT - Mol. Biol. Reports *29*, 103–6 PL–Netherlands PT–Journal Article.

Nelder, J.A., and Mead, R. (1965). A Simplex Method for Function Minimization. Comput. J. *7*, 308–313.

Nocedal, J., and Wright, S. (1999). Numerical Optimization (New York: Springer).

Novere, N. Le, Hucka, M., and Mi, H. (2009). The Systems Biology Graphical Notation. Nat. Biotechnol. *27*, 735–742.

Le Novère, N., Hucka, M., Hoops, S., Keating, S., Sahle, S., and Wilkinson, D. (2008). Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Nat. Preced. 1–38.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *27*, 29–34.

Orth, J.D., Conrad, T.M., Na, J., Lerman, J. a, Nam, H., Feist, A.M., and Palsson, B.Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. Mol. Syst. Biol. *7*, 535.

Paiva, C.L.A., and Panek, A.D. (1996). Biotechnological Applications of the Disaccharide Trehalose. pp. 293–314.

Park, T.-Y., and Froment, G.F. (1998). A hybrid genetic algorithm for the estimation of parameters in detailed kinetic models. Comput. Chem. Eng. *22*, S103–S110.

Parrou, J.L., Teste, M.A., and François, J. (1997). Effects of various types of stress on the metabolism of reserve carbohydrates in Saccharomyces cerevisiae: Genetic evidence for a stress-induced recycling of glycogen and trehalose. Microbiology *143*, 1891–1900.

Petri, C.A. (1962). Kommunikation mit Automaten (Communication with Automata). University of Bonn.

Postmus, J., Canelas, A.B., Bouwman, J., Bakker, B.M., Van Gulik, W., De Mattos, M.J.T., Brul, S., and Smits, G.J. (2008). Quantitative analysis of the high temperature-induced glycolytic flux increase in Saccharomyces cerevisiae reveals dominant metabolic regulation. J. Biol. Chem. *283*, 23524–23532.

Puig-Castellví, F., Alfonso, I., Piña, B., and Tauler, R. (2015). A quantitative 1H NMR approach for

evaluating the metabolic response of Saccharomyces cerevisiae to mild heat stress. Metabolomics *11*, 1612–1625.

Rodriguez, N., Donizelli, M., and Le Novère, N. (2007). SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). BMC Bioinformatics *8*, 79.

Roser, B. (1991). Trehalose, a new approach to premium dried foods. Trends Food Sci. Technol. *2*, 166–169.

Savageau, M.A. (1976). Biochemical Systems Analysis (Reading, MA: Addison-Wesley Pub. Co.).

Schaff, J., Slepchenko, B., and Loew, L. (2000). Physiological modeling with virtual cell framework. Methods Enzymol. *321*, 1–23.

Schiraldi, C., Di Lernia, I., and De Rosa, M. (2002). Trehalose production: Exploiting novel approaches. Trends Biotechnol. *20*, 420–425.

Schmidt, T., Samaras, P., Frejno, M., Gessulat, S., Barnert, M., Kienegger, H., Krcmar, H., Schlegl, J., Ehrlich, H.-C., Aiche, S., et al. (2018). ProteomicsDB. Nucleic Acids Res. *46*, D1271–D1281.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. *32*, D431-3.

Smallbone, K., Simeonidis, E., Swainston, N., and Mendes, P. (2010). Towards a genome-scale kinetic model of cellular metabolism. BMC Syst. Biol. *4*, 6.

Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., et al. (2013). A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett. *587*, 2832–2841.

Strassburg, K., Walther, D., Takahashi, H., Kanaya, S., and Kopka, J. (2010). Dynamic transcriptional and metabolic responses in yeast adapting to temperature stress. OMICS *14*, 249–259.

Sun, J., Garibaldi, J.M., and Hodgman, C. (2012). Parameter estimation using meta-heuristics in systems biology: a comprehensive review. IEEE/ACM Trans. Comput. Biol. Bioinform. *9*, 185–202.

Varma, A., and Palsson, B.O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. Bio/Technology *12*, 994–998.

Viant, M.R., Kurland, I.J., Jones, M.R., and Dunn, W.B. (2017). How close are we to complete annotation of metabolomes? Curr. Opin. Chem. Biol. *36*, 64–69.

Visser, D., and Heijnen, J.J. (2003). Dynamic simulation and metabolic re-design of a branched

pathway using linlog kinetics. Metab. Eng. *5*, 164–176.

Vizcaíno, J.A., Csordas, A., Del-Toro, N., Dianes, J.A., Griss, J., Lavidas, I., Mayer, G., Perez-Riverol, Y., Reisinger, F., Ternent, T., et al. (2016). 2016 update of the PRIDE database and its related tools. Nucleic Acids Res. *44*, D447–D456.

Voit, E.O. (2003). Biochemical and genomic regulation of the trehalose cycle in yeast: Review of observations and canonical model analysis. J. Theor. Biol. *223*, 55–78.

Wang, M., Herrmann, C.J., Simonovic, M., Szklarczyk, D., and von Mering, C. (2015). Version 4.0 of PaxDb: Protein abundance data, integrated across model organisms, tissues, and cell-lines. Proteomics *15*, 3163–3168.

Wolkenhauer, O., and Mesarović, M. (2005). Feedback dynamics and cell function: Why systems biology is called Systems Biology. Mol. Biosyst. *1*, 14.

Yen, J., Liao, J.C., Lee, B.L.B., and Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. Syst. Man Cybern. Part B Cybern. a Publ. IEEE Syst. Man Cybern. Soc. *28*, 173–191.

Yurkovich, J.T., and Palsson, B.O. (2018). Quantitative -omic data empowers bottom-up systems biology. Curr. Opin. Biotechnol. *51*, 130–136.

Zhang, Z., Shen, T., Rui, B., Zhou, W., Zhou, X., Shang, C., Xin, C., Liu, X., Li, G., Jiang, J., et al. (2015). CeCaFDB: a curated database for the documentation, visualization and comparative analysis of central carbon metabolic flux distributions explored by 13C-fluxomics. Nucleic Acids Res. *43*, D549–D557.

Zi, Z. (2011). SBML-PET-MPI: A parallel parameter estimation tool for Systems Biology Markup Language based models. Bioinformatics *27*, 1028–1029.

Zi, Z., and Klipp, E. (2006). SBML-PET: A Systems Biology Markup Language-based parameter estimation tool. Bioinformatics *22*, 2704–2705.

Zuberek, W.M. (1991). Timed Petri nets definitions, properties, and applications. Microelectron. Reliab. *31*, 627–644.

Chapter 3: Software Proof of Concept: Glycolytic Network in *Saccharomyces cerevisiae*

3.1 Abstract

The construction of kinetic models of metabolic pathways has always been hindered by the limited availability of kinetic parameters, in addition to incomplete knowledge of the reaction mechanisms. Strategies have been developed to allow the generation of kinetic models with limited information. Despite this, not many large-scale dynamic and integrative models have been generated. The aim of this research is to streamline the process of generating large-scale metabolic models, while using metabolomics and proteomic data to inform parameter values. As a proof of concept, two different models of yeast glycolysis were being built using metabolite, flux and protein amounts via the software package GRaPe 2.0. We show that our method is capable of generating a dynamic model, which accounts for multiple types of data. The resulting models are performed well on evaluation.

3.2 Introduction

The development of computer models in the field of biology has been increasing continuously (Basta et al., 2014; Blazeck and Alper, 2010; Jamshidi and Palsson, 2008; O'Brien et al., 2013) allowing biological systems to be studied from a quantitative and predictive perspective. A computational model allows us to understand how all the different parts in a system come together, interact and regulate each other, as well as identify emergent properties (Kitano, 2002). There are multiple modelling approaches employed to study different biological systems, such as Boolean rules (Gupta et al., 2007; Saez-Rodriguez et al., 2007), Petri nets (Koch et al., 2005), Bayesian networks (Auliac et al., 2008; Friedman, 2004), constraint-based models (Orth et al., 2011; Varma and Palsson, 1994) and kinetic models (Jahan et al., 2016; Joshi and Palsson, 1989; Khodayari et al., 2014). Kinetic models provide the advantage of studying the quantitative and dynamical behaviours often observed in biology. They are among the most precise and versatile tools to represent complex biological systems, usually built using ordinary differential equations.

However, the construction of kinetic models is often impeded by limited information (Almquist et al., 2014). Whilst metabolic pathway information is readily available in literature and databases like KEGG (Ogata et al., 1999), allocating kinetic equations to all the reactions in a model is often tedious, provided they are available to begin with. One solution proposed to avoid such an issue is to use some generic types of kinetic equations, for example the linlog kinetics for model building (Smallbone et al., 2007, 2010; Visser and Heijnen, 2003). Some models make use of simple mass action kinetics (Chen et al., 2010, 2012); the Generalized Mass Action system is an improvement which allows for greater flexibility by including exponents called kinetic order that accounts for either augmenting or diminishing effects (Alvarez-Vasquez et al., 2005; Fonseca et al., 2011). There are also software packages developed with the idea of simplifying rate law selection during the model building process such as SBMLsqueezer (Dräger et al., 2008, 2015) and BISEN (Vanlier et al., 2009). After accumulating the necessary information, from pathway to reaction kinetics, models can be built, curated and maintained using software such as COPASI (Hoops et al., 2006) and CellDesigner (Funahashi and Matsuoka, 2008).

A major component of kinetic models is the parameters embedded in the rate laws, such as the Michaelis-Menten constants, and the association or dissociation constants. Many of these

are usually unknown and determining them can be challenging. Teusink and colleagues experimentally determined the parameters of the yeast glycolytic system under *in vitro* conditions and demonstrated that this approach is able to reproduce results for *in vivo* conditions (Teusink et al., 2000). However, this approach is very time consuming and expensive for large systems. Another approach is to make use of parameters that are found in the literature for model building (Alvarez-Vasquez et al., 2005) or to make use of databases such as BRENDA (Schomburg et al., 2004) and SABIO-RK (Krebs et al., 2007). However, a problem with this method is that the values are usually collected in various different experimental conditions, cell types, strains and organisms (Costa et al., 2011). Parameter estimation is another method used to acquire the values indirectly. This has been done using metabolomics and/or flux data, with software packages such as SBML-PET (Zi, 2011; Zi and Klipp, 2006), SBMLSimulator (Dörr et al., 2014) and COPASI (Hoops et al., 2006).

The software mentioned above are all designed to assist in various steps of the model construction process, either building the topology, providing help with the setting of the reaction kinetics or estimating the missing parameters, but none of them offers an integrated solution to simplify the entire process. Additionally, some of these tools require the users to have fluency in programming or are in the form of a package attached to commercial software. When a tool has a built-in graphical user interface, the complexity of the model building process scales with the size of the model as user input is required at every step in many aspects. This can lead to errors and long hours in model building. These obstacles would hinder beginners from joining the modelling community.

We previously introduced the GRaPe software (Adiamah et al., 2010) to simplify the construction of kinetic models using automatically generated rate equations, and estimating parameters from flux values under steady state conditions. In this study, we present GRaPe 2.0 that addresses several important limitations of the previous version: GRaPe 2.0 uses convenience kinetics (Liebermeister and Klipp, 2006) to generate rate equations, enabling modelling of reactions containing any number of substrates and products (which was previously limited to only two substrates and products); regulatory interactions such as allosteric inhibitions or activations can be included in the model, a feature not found previously; moreover, several types of omics

data can be used simultaneously in the parameter estimation process; and the estimation is able to handle actual time courses of dynamic values, a function that was previously lacking; the user interface has also been updated to be more intuitive and compact. This updated version aims to be the gateway modelling software for biologists planning to use kinetic models in their research. With the simplified process for model building, it would also serve as a rapid prototyping step in metabolic model construction before moving forward with their project.

A case study using yeast glycolytic models was built to demonstrate the software's capabilities, where temporal omics data was available to help parameterize the model as well as validate it. In this case, the model uses the initial time point's proteomics data to assign enzyme concentrations simulating standard conditions for parameter estimation. Then, the remaining time points' data were used to make predictions on the metabolites and fluxes emulating heat stress conditions. From the case study, we show that our procedure enables the construction of integrated high-quality kinetic models using several types of omics data and greatly accelerates the construction of precise kinetic models of cell metabolism.

3.3 Methods

3.3.1 Enzyme Kinetics

For the construction of models, GRaPe 2.0 makes use of convenience kinetics, a generalised reversible Michaelis-Menten equation developed by Liebermeister and Klipp (Liebermeister and Klipp, 2006). The idea that specific rate law for all reactions in the system is unnecessary, because their importance would be dissipated in a large system, hence the general rate law for all.

$$v(sub, prod) = E_{total} \cdot f_{reg} \frac{k_{+}^{cat} \prod_i \widetilde{sub_i}^{n_i} - k_{-}^{cat} \prod_j \widetilde{prod_j}^{n_j}}{\prod_i (1 + \widetilde{sub_i} + \cdots + \widetilde{sub_i}^{n_i}) + \prod_j (1 + \widetilde{prod_j} + \cdots + \widetilde{prod_j}^{n_j}) - 1} \quad (1)$$

Where *sub* is the substrate concentration; *prod* is the product concentration; E_{total} is the enzyme concentration; f_{reg} is a pre-factor to account for activation (using 1 + d/k^A), *d* is the activator concentration; k^A is the activation constant) or inhibition (using k^I / $k^I + d$, *d* is the inhibitor concentration, k^I is the inhibition constant); $k_{+/-}^{cat}$ are the forward and reverse turnover

rates; $\widetilde{sub}$ = sub/k_{sub}^{M}; $\widetilde{prod}$ = prod/k_{prod}^{M}; $k_{sub/prod}^{M}$ are the Michaelis-Menten constants for either substrate or product; *n* is the stoichiometric coefficient for the reaction.

Convenience kinetics assumes that all reactions in models are reversible and in random binding order. It also allows for easy inclusion of enzymatic modifiers in the equation with the f_{reg} prefactor, and has a lower number of kinetic parameters per equation compared to the original Michaelis-Menten equations. All that is needed for the generation of rate equations is knowledge of the proteins and metabolites involved and their stoichiometry along with their initial quantities.

3.3.2 Genetic Algorithm

After the rate equations are generated, the model can be trained to fit experimental data containing either metabolite concentrations, flux values or both. This is done by estimating the kinetic parameters in the model. This process is quicker than experimentally measuring every single value needed in the model, hence saving a lot of time in the modelling process and speeding up development of large-scale kinetic models.

The information needed for parameter estimation is either steady state or time course data for the desired model type. The parameter estimation method of choice in GRaPe 2.0 is a genetic algorithm, an evolutionary machine learning approach that utilizes Darwinian evolutionary principles (Holland, 1962). In this context, a population is evolved over multiple generations and produces an individual containing a single set of parameters for the model that can fit the input data as closely as possible. The algorithm uses tournament selection for parents, where several "tournaments" among few individuals are run, to select the fittest individuals for cross-over; and uniform crossover, where a fixed ratio of parameters are exchanged between parents at random, rather than fixed points, to form a new individual. GRaPe 2.0 allows the user to determine how frequent the least fit individuals are removed from the population (plague function). Fitness of an individual is measured by the mean absolute percentage errors between training and simulated data. Each parameter value is encoded as decadic logarithm value unlike the classic binary storage method. Mutation is then done by randomly increasing or decreasing one or several parameters in the individual by a small percentage.

In order to determine the metabolite concentrations at steady state, the non-linear systems of equations are solved using the Newton-Raphson algorithm, which is complemented by the SBML ODE Solver Library (SOSlib) (Machné et al., 2006). SOSlib is also used for determining the objective function when fitting time through data.

The model in this study was estimated with an initial population size of 100, plague frequency of every 5 generations, mutation rate of 10% of population at the time, and for the process to stop after the fitness value plateaus for 11 generations.

3.3.3 Software Architecture, Data Structures, and Dependencies

GRaPe 2.0 was designed with the intention of being platform independent, hence the core of the software was written in the Java programming language. GRaPe has a graphical user interface generated using Java Swing and implemented as a separate module from the rest of the software. The other core modules are an SBML converter to import/export models to Systems Biology Markup Language (SBML) file format. The reading and writing of SBML files are done by the JSBML library (Dräger et al., 2011). There is a Reaction module within the software for generating reactions using information from the Species module that stores substrates, products and enzymes data. The Steady State module is dedicated to solving the system of equations using the Newton-Raphson algorithm, utilizing the Apache Common Maths library for matrix manipulation. This module is called by the Genetic Algorithm module when fitting steady state data during the parameter estimation step. When the Genetic Algorithm performs an estimation using time series data it calls the ODE module which in turn calls SOSlib using Java Native Interface (JNI) powered by the Java Native Access (JNA) library.

3.4 Results

We present two models of S. cerevisiae glycolysis that have been built using GRaPe with automatically generated kinetic equations, without knowledge of the kinetics involved. The first model (Model 1) includes enzymatic reactions without regulatory influence; the second (Model 2) adds allosteric regulatory effects using modifiers.

3.4.1 *Saccharomyces cerevisiae* Glycolytic Pathway Case Study

Both glycolytic pathway models contain 22 metabolites, of which 6 are external metabolites (meaning their concentration is externally regulated and unaffected by changes in the system),

and 18 enzymes and reactions (Figure. 3.1). The difference between Model 1 and Model 2 consists of the inclusion of enzyme modifiers. The modifiers used are: inhibition of hexokinase by glucose-6-phosphate, inhibition of phosphofructokinase by adenosine triphosphate (ATP) and activation of pyruvate kinase by fructose 1, 6-bisphosphate.

The models are based on those developed by van Eunen et al. (van Eunen et al., 2012). Changes from the original van Eunen model are: trehalose is now represented as an external metabolite and does not get broken down to glucose; succinate production is now branched off from acetaldehyde instead of from pyruvate; an additional reaction has been added, which is the production of ATP from ADP for the conservation of ATP within the system. In Model 2, the modifiers were kept the same as in the original van Eunen model.

In total, Model 1 has 93 parameters and Model 2 has 96 parameters. Both models were generated and stored in the standard Systems Biology Markup Language Format (Hucka et al., 2003).

3.4.2 Omics Data Integration

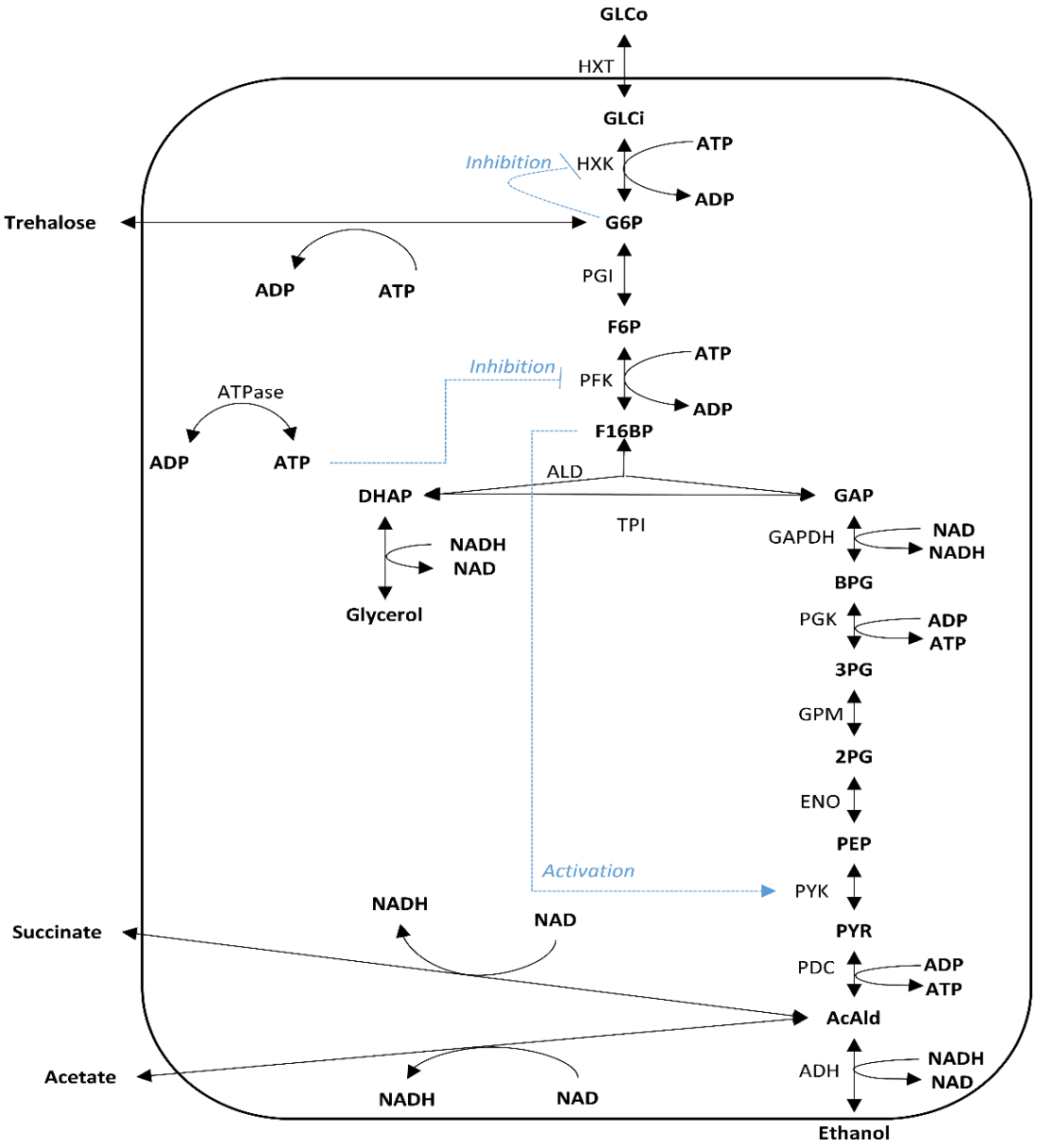

Figure 3.1: **Glycolysis model used in this study**. Everything outside the cell boundary is set to boundary condition 'True' in the SBML file. Metabolites are in bold whilst enzymes are in regular font. The blue lines indicate enzyme modifiers. GLCo: extracellular glucose, GLCi: intracellular glucose, G6P: glucose 6-phosphate, F6P: fructose 6-phosphate, F16BP: fructose 1,6-bisphosphate, DHAP: dihydroxyacetone phosphate, GAP: glyceraldehyde 3-phosphate, BPG: 1,3-bisphosphoglycerate, 3PG: 3-phosphoglycerate, 2PG: 2-phosphoglycerate, PEP: phosphoenolpyruvate, PYR: pyruvate, AcAld: acetaldehyde, HXT: glucose transport, HXK: hexokinase, PGI: phosphoglucose isomerase, PFK: phosphofructokinase, ALD: aldolase, TPI: triose-phosphate isomerase, GAPDH:

glyceraldehyde-3-phosphate dehydrogenase, PGK: 3-phosphoglycerate kinase, GPM: phosphoglycerate mutase, ENO: enolase, PYK: pyruvate kinase, PDC: pyruvate decarboxylase, ADH: alcohol dehydrogenase.

3.4.3 Model Evaluation

For the parameter estimation process, we made use of three types of 'omics' data for input. First, the *in vivo* metabolite concentrations collected by Smallbone et al. (Smallbone et al., 2013); since only 9 of the 22 metabolites were measured, unmeasured values were taken from their model's output. Second, the flux values generated by Smallbone et al.'s model were used. As a third constraint in the estimation process, we used enzyme concentrations determined from a recent label-free, mass spectrometry-based proteomics study (Jarnuczak et al., 2018, PRIDE accession: PXD006262), where yeast protein abundances were measured over a 240-minute heat stress at seven time points; of the seven time points, only the initial time point was used for parameter estimation to emulate standard conditions for the model. Absolute abundances were estimated via linear regression (Rosenberger et al., 2014) using targeted proteomic data from a QconCAT study as internal standards (Lawless et al., 2016), and are provided in Appendix B. The abundances are then converted to mmol by using 5 fl as the cytoplasmic volume used by Smallbone et al. Additionally, the different isozymes involved in the same reaction were summed up for each reaction. The fitted models produced a normalised root mean squared error value of 0.0613 and 0.0314 respectively for Model 1 and Model 2.

Metabolite name	Metabolite concentration (mM)	Model 1	Model 2
Cellular Glucose*	6.277	6.169	5.6754
ATP	3.95147	3.909	5.569
Glucose 6-phosphate*	0.774	0.00728	0.005505
ADP	1.73	1.671	0.01087
Fructose-6-phosphate*	0.235	0.238	0.2438
Fructose 1,6-bisphosphate*	4.58	4.686	4.6301
Dihydroacetone Phosphate*	1.16	1.029	1.1702
Glyceraldehyde Phosphate*	0.316	0.311	0.3139
NADH	0.179	0.1638	0.1872
NAD	1.412	1.6296	1.60609
1,3-Bisphosphoglycerate	0.00165	0.00161	0.001624
3-Phosphoglycerate	0.455	6.29E-06	0.4655
2-Phosphoglycerate*	0.083	0.005238	0.08281
Phosphoenolpyruvate*	0.611	1.35E-05	0.63887
Pyruvate*	2.11	2.0088	1.953
Acetaldehyde	3.43	0.00154	4.67E-04

Table 3.1: **Metabolite concentrations for training data and simulated results** at steady state for both models all values rounded up to the closest 4 s. f. *These are *in vivo* measurements of the metabolites.

Reaction name	Flux (mmol s^{-1})	Model 1	Model 2
Glucose Transport	1.998	2.127	1.934
Hexokinase	1.998	2.127	1.934
Glucose 6-phosphate isomerase	1.887	1.903	1.934
Trehalose Synthase	0.1107	0.112	-2.05E-06
Phosphofructosekinase	1.887	1.903	1.934
Aldolase	1.887	1.903	1.934
Glycerol 3-phosphate dehydrogenase	0.05358	0.053549	0.0539909
Triosephosphate isomerase	1.833	1.8498	1.8802
Glyceraldehye 3-phosphate dehydrogenase	3.721	3.75316	3.81443
Phosphoglycerate kinase	3.721	3.75316	3.81443
Phosphoglycerate mutase	3. 721	3.75316	3.81443
Enolase	3. 721	3.75316	3.81443
Pyruvate Kinase	3. 721	3.75316	3.81443
ATPase	3.502	3.3635	3.76045
Pyruvate Decarboxylase	3. 721	3.75316	3.81443
Succinate Dehydrogenase	0	9.83E-09	0.013514
Alcohol Dehydrogenase	3.694	3.72638	3.80095
Aldehyde Dehydrogenase	0.02679	0.026774	-3.28E-05

Table 3.2: **Flux values for training data and simulated results** at steady state for both models all values rounded up to the closest 4 s. f.

3.4.4 Metabolites Prediction under Heat Stress using Proteomics Data

We used both completed models to make predictions of yeast glycolytic metabolite concentrations and fluxes under heat stress conditions. We inserted the remaining unseen quantitative proteomic

data from the heat stress time course experiment into the model (this is done by inserting the measured enzyme values into the model, instead of the original values measured at time-point zero) to simulate an elevated temperature at their respective time points and validate the model. The models were then run in COPASI, with each time-point's enzyme values inserted into the model and run up to steady state. Time course solving was also performed using CellDesigner, where the enzyme values were varied over time (not shown), but the results did not differ from the solving at steady state; results from COPASI alone are shown for consistency in this work. Simulation results are shown in Figure. 3.2-3.5.

In Model 1, the prediction indicates there is a drop in metabolite and flux levels. As for Model 2, a rise in flux was predicted and minimal changes were observed on the majority of the metabolites. The noticeable changes are a drop in glucose concentration and an initial drop for fructose-6-phosphate, before going back up to slightly higher than initial values by the end of heat stress.

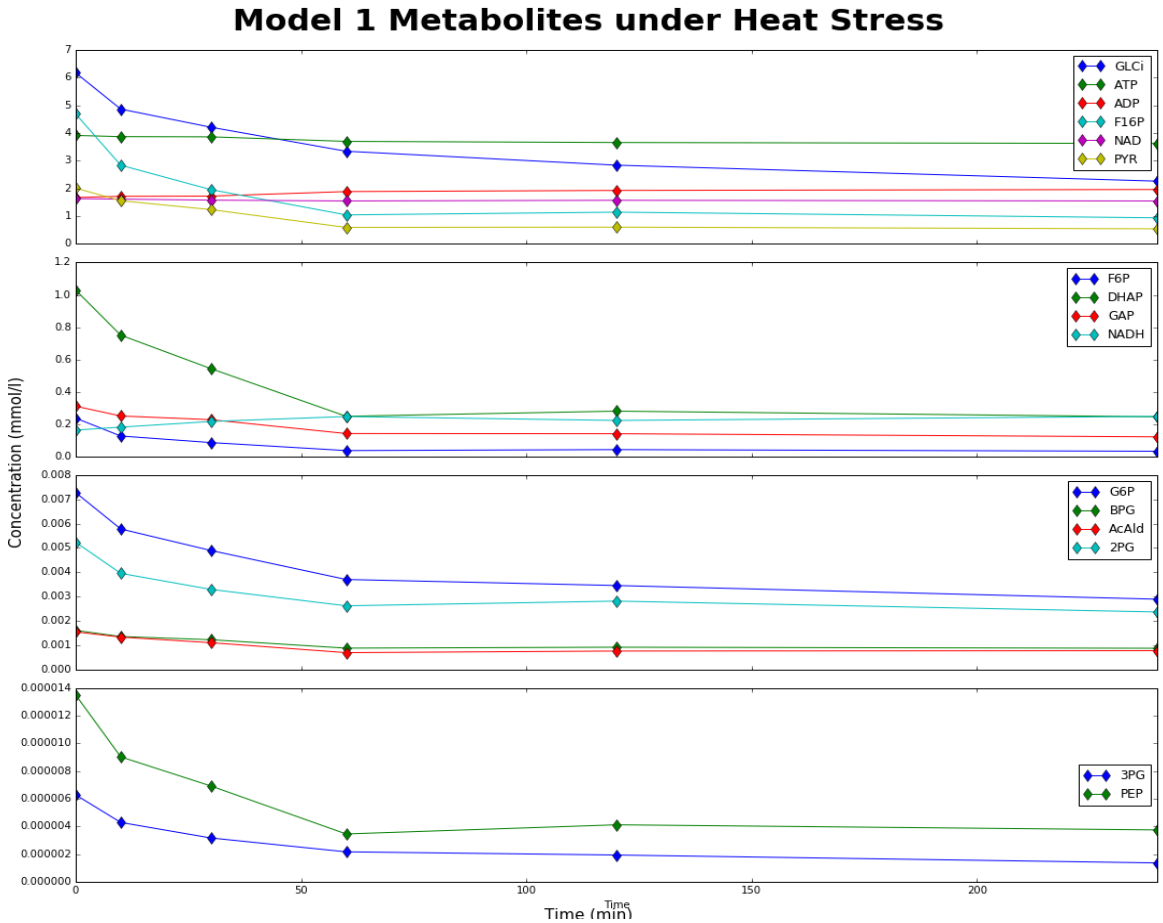

Figure 3.2: **Time course prediction results on yeast glycolytic metabolites for Model 1.** The heat stress last for 240 minutes

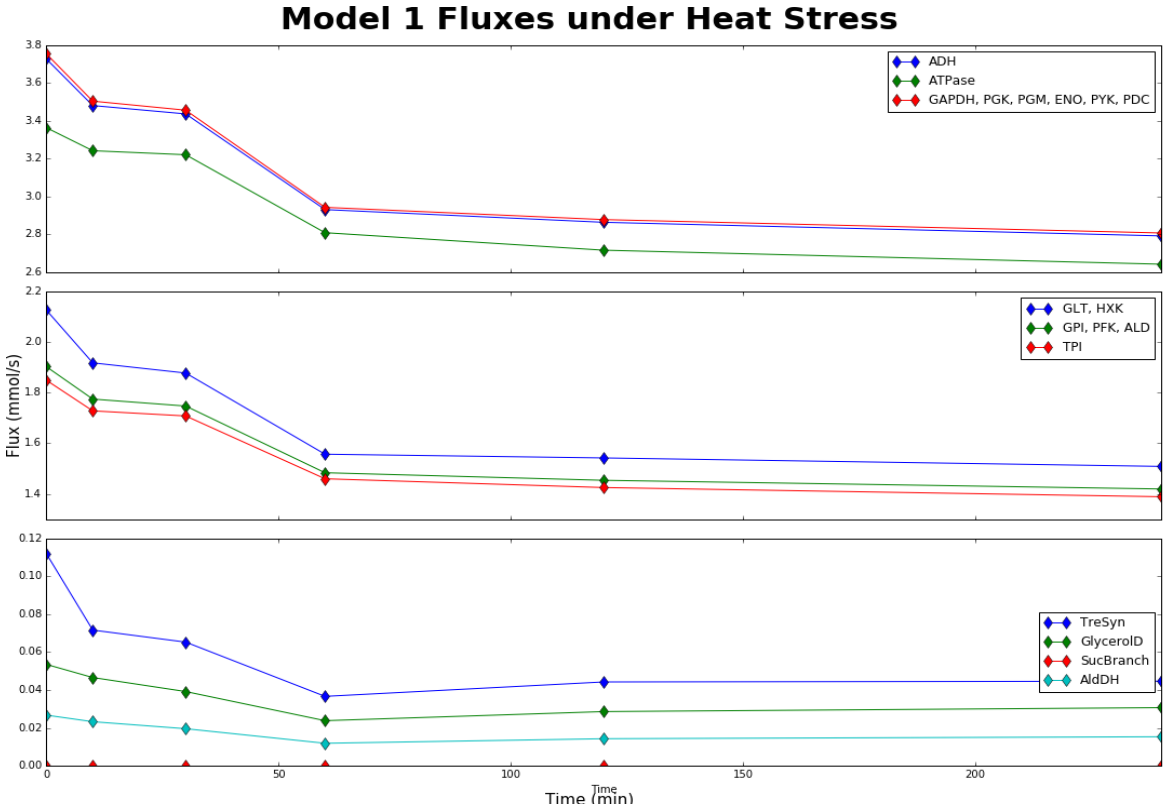

Figure 3.3: **Time course prediction results on yeast glycolytic fluxes for Model 1**. The heat stress last for 240 minutes.

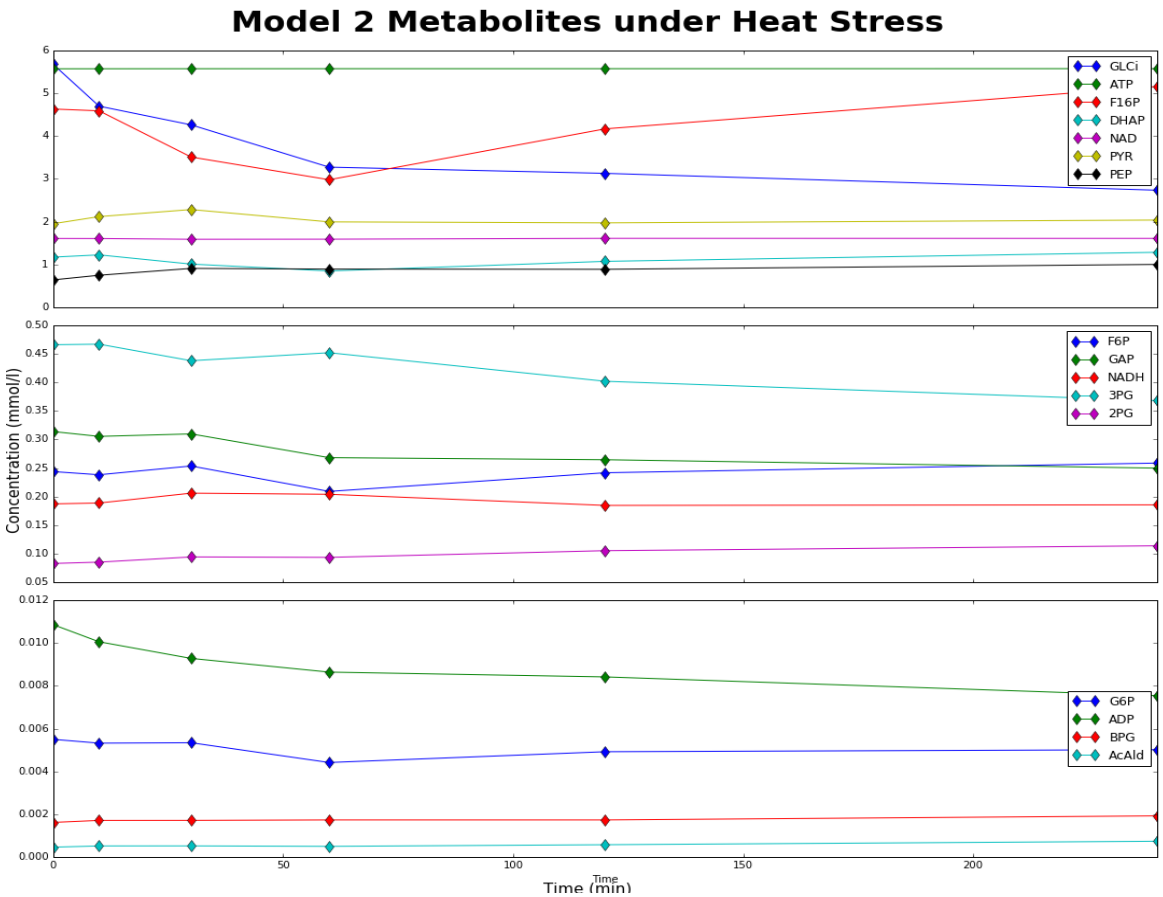

Figure 3.4: **Time course prediction results on yeast glycolytic metabolites for Model 2**.

The heat stress last for 240 minutes.

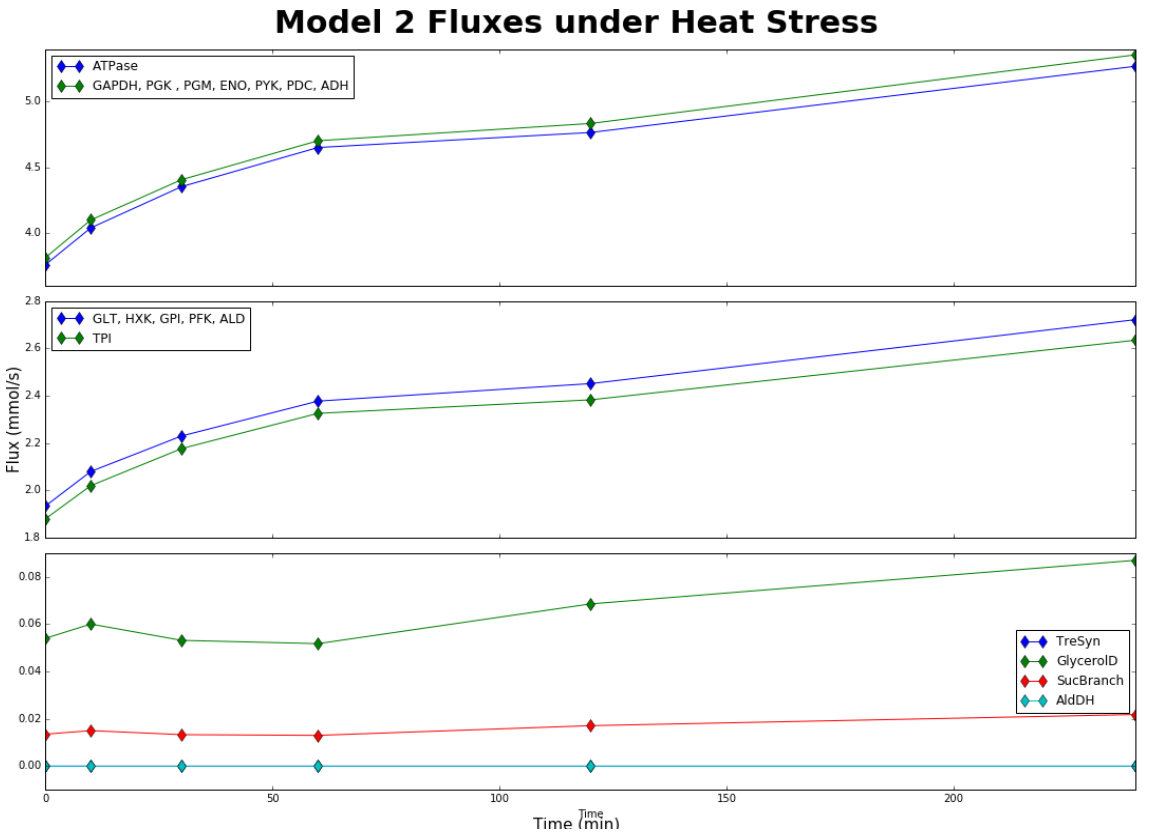

Figure 3.5: **Time course prediction results on yeast glycolytic fluxes for Model 2.** The heat stress last for 240 minutes.

3.5 Discussion

This study demonstrates that the modelling approach employed by GRaPe 2.0 is able to successfully integrate omics data alongside all the components involved in a system, such as metabolite quantities, fluxes and changes through a time course. At the same time GRaPe 2.0 simplifies the process for any would-be user seeking to manage the complex process of building an omic-integrated kinetic model and estimating its kinetic parameters, by integrating these tasks into a straightforward software suite.

The kinetic models obtained from the parameter estimation were relatively accurate, producing good solutions with lower normalised root mean squared errors than comparable models (0.312 for Smallbone's model, Smallbone et al., 2013). This was all done using experimentally measured data available in the public domain instead of using artificially generated data. This choice was made to avoid introducing a bias in the parameter values estimated, as the goal is to eventually explore a novel biological problem using the heat stress protein dataset. For the glycolytic test system evaluated here, the best fit is found for the flux values; the concentration fit was seen to be more heterogeneous with most metabolites reasonably fitting well until it reached the glyceraldehyde phosphate break down branch. Notably, in Model 2, it was predicted that the flux for trehalose production would be negative (Table 3.2); this was also observed by Teusink et al. (Teusink et al., 2000) for the first 20 minutes in their experiments (in pH 7 and at 30^{o}C) and by van Eunen et al. across four different conditions (non-starved cells with dilution rate D=0.1 h^{-1} for glucose; nitrogen starved for 4 hours with dilution rate D=0.1 h^{-1} for glucose; non-starved cells with dilution rate D=0.35 h^{-1} for glucose; 4 hours nitrogen-starved cells with dilution rate D=0.35 h^{-1} for glucose, all under 30^{o}C). This would indicate that inclusion of enzyme modifiers substantially improves modelling of this aspect of the biological system.

Additionally, altered trehalose metabolism is a significant hallmark of heat stress in yeast, and a fundamental aim for the construction of kinetic models is for them to be capable of describing *in vivo* cell metabolism and to predict changes of behaviour under different conditions. Heat stress refers to the situation when yeast is moved to a temperature of around 37^{o}C from 30^{o}C (unlike more extreme heat shock, which is around 45^{o}C). It has been known that when yeast undergoes heat stress, several changes occur within the cell to allow it to adapt to the heat. One

of the changes is increased trehalose production; trehalose is a small molecule that provides a myriad of protective functions for the cell, such as protection from denaturation and adaptation to stress in synergy with chaperones (Piper, 1993). During heat stress, intracellular glucose dropped significantly in both models (Figure. 3.2 & 3.4), as glucose is the energy source needed to adapt to this change in physiological condition (Thammavongsa et al., 2013, Vilaprinyo et al., 2006). In chorus, the protein needed for the synthesis of trehalose increased by 220% (Appendix B: Table 1 & 2), which should also result in increased flux towards trehalose.

In model 2 the flux towards trehalose was negative and further decreased during heat stress, while flux in all other pathways increased. A negative flux means that the reaction is going in the reverse direction (in this case, trehalose is broken down). This could be the result of an oversimplification of the trehalose synthesis pathway. The complete trehalose synthesis pathway includes conversion of glucose 6-phosphate to trehalose 6-phosphate or glucose 1-phosphate, which is then converted to urine diphosphate glucose. Urine diphosphate glucose can be converted to either glycogen or trehalose 6-phosphate, a trehalose precursor. Additionally, there is regulation in this pathway, where glucose 6-phosphate activates production of trehalose 6-phosphate, which at the same time is inhibited by glucose molecules. This negative flux could also be the result of not accounting for the effects of temperature on enzyme activity. It was determined that an increase in temperature would result in increased activity for enzymes involved in the synthesis of trehalose, and at the same time in decreased activity for trehalase, the enzyme responsible for trehalose degradation (Neves and François, 1992).

When compared to measured fluxes under heat stress (Mensonides et al., 2002; Postmus et al., 2008), Model 2 was able to predict an increase in glycolytic flux, validating the success of this model. While Model 1 predicted the opposite, along with a dip in metabolite levels, suggesting overfitting has occurred for this model as it has failed to replicate the metabolic phenomenon during heat stress. Results of this modelling study therefore strongly suggest that metabolic regulation is important for the construction of accurate and informative kinetic models of cell metabolism, as well as that it plays an important role in the cellular stress response.

3.6 Conclusion

GRaPe 2.0 can accelerate the construction of kinetic models using just reaction pathways and training data of steady state or time courses. This technique can be employed for the creation of large-scale kinetic models of cell metabolism and be used to make predictions under varying conditions. All of these can then be translated towards finding bioengineering targets. The ease of use and swift model building process should enable more biologists to join in the modelling community.

3.7 References

Adiamah, D. a, Handl, J., and Schwartz, J.-M. (2010). Streamlining the construction of large-scale dynamic models using generic kinetic equations. Bioinformatics 26, 1324–1331.

Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J., and Jirstrand, M. (2014). Kinetic models in industrial biotechnology - Improving cell factory performance. Metab. Eng. 24, 38–60.

Alvarez-Vasquez, F., Sims, K.J., Cowart, L.A., Okamoto, Y., Voit, E.O., and Hannun, Y.A. (2005). Simulation and validation of modelled sphingolipid metabolism in Saccharomyces cerevisiae. Nature 433, 425–430.

Auliac, C., Frouin, V., Gidrol, X., and d'Alche-Buc, F. (2008). Evolutionary approaches for the reverse-engineering of gene regulatory networks: a study on a biologically realistic dataset. BMC Bioinformatics 9, 91.

Basta, H. a., Sgro, J.-Y., and Palmenberg, A.C. (2014). Modeling of the human rhinovirus C capsid suggests a novel topography with insights on receptor preference and immunogenicity. Virology 448, 176–184.

Blazeck, J., and Alper, H. (2010). Systems metabolic engineering: genome-scale models and beyond. Biotechnol. J. 5, 647–659.

Chen, D.W.-C., Lynch, J.T., Demonacos, C., Krstic-Demonacos, M., and Schwartz, J.-M. (2010). Quantitative analysis and modeling of glucocorticoid-controlled gene expression R eseaRch a Rticle. Pharmacogenomics 11, 1545–1560.

Chen, D.W.C., Krstic-Demonacos, M., and Schwartz, J.M. (2012). Modeling the mechanism of GR/c-Jun/Erg crosstalk in apoptosis of acute lymphoblastic leukemia. Front. Physiol. 3 NOV, 1–16.

Costa, R.S., Machado, D., Rocha, I., and Ferreira, E.C. (2011). Critical perspective on the consequences of the limited availability of kinetic data in metabolic dynamic modelling. IET Syst. Biol. 5, 157–163.

Dörr, A., Keller, R., Zell, A., and Dräger, A. (2014). SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. Computation 2, 246–257.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst. Biol. 2, 39.

Dräger, A., Rodriguez, N., Dumousseau, M., Dörr, A., Wrzodek, C., Le Novère, N., Zell, A., and Hucka, M. (2011). JSBML: A flexible java library for working with SBML. Bioinformatics 27, 2167–2168.

Dräger, A., Zielinski, D.C., Keller, R., Rall, M., Eichner, J., Palsson, B.O., and Zell, A. (2015). SBMLsqueezer 2: context-sensitive creation of kinetic equations in biochemical networks. BMC Syst. Biol. 9, 68.

van Eunen, K., Kiewiet, J.A.L., Westerhoff, H. V., and Bakker, B.M. (2012). Testing biochemistry revisited: How in vivo metabolism can be understood from in vitro enzyme kinetics. PLoS Comput. Biol. 8.

Fonseca, L.L., Sánchez, C., Santos, H., and Voit, E.O. (2011). Complex coordination of multi-scale cellular responses to environmental stress. Mol. Biosyst. 7, 731–741.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. Science 303, 799–805.

Funahashi, A., and Matsuoka, Y. (2008). CellDesigner 3.5: a versatile modeling tool for biochemical networks. Proc. IEEE 96, 1254–1265.

Gupta, S., Bisht, S.S., Kukreti, R., Jain, S., and Brahmachari, S.K. (2007). Boolean network analysis of a neurotransmitter signaling pathway. J. Theor. Biol. 244, 463–469.

Holland, J.H. (1962). Outline for a Logical Theory of Adaptive Systems. J. ACM 9, 297–314.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI--a COmplex PAthway SImulator. Bioinformatics 22, 3067–3074.

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

Jahan, N., Maeda, K., Matsuoka, Y., Sugimoto, Y., and Kurata, H. (2016). Development of an accurate kinetic model for the central carbon metabolism of Escherichia coli. Microb. Cell Fact. 15, 112.

Jamshidi, N., and Palsson, B.Ø. (2008). Formulating genome-scale kinetic models in the post-genome era. Mol. Syst. Biol. 4, 171.

Jarnuczak, A.F., Albornoz, M.G., Eyers, C.E., Grant, C.M., and Hubbard, S.J. (2018). A quantitative and temporal map of proteostasis during heat shock in Saccharomyces cerevisiae. Mol. Omi. 14, 37–52.

Joshi, A., and Palsson, B.O. (1989). Metabolic dynamics in the human red cell: Part I—A comprehensive kinetic model. J. Theor. Biol. 141, 515–528.

Khodayari, A., Zomorrodi, A.R., Liao, J.C., and Maranas, C.D. (2014). A kinetic model of Escherichia coli core metabolism satisfying multiple sets of mutant flux data. Metab. Eng. 25, 50–62.

Kitano, H. (2002). Systems biology: a brief overview. Science 295, 1662–1664.

Koch, I., Junker, B.H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics 21, 1219–1226.

Krebs, O., Golebiewski, M., Kania, R., and Mir, S. (2007). SABIO-RK: a data warehouse for biochemical reactions and their kinetics. J. Integr. Bioinform.

Lawless, C., Holman, S.W., Brownridge, P., Lanthaler, K., Harman, V.M., Watkins, R., Hammond, D.E., Miller, R.L., Sims, P.F.G., Grant, C.M., et al. (2016). Direct and Absolute Quantification of over 1800 Yeast Proteins via Selected Reaction Monitoring * □. 1309–1322.

Liebermeister, W., and Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor. Biol. Med. Model. 3, 41.

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: A native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics 22, 1406–1407.

Mensonides, F.I.C., Schuurmans, J.M., Teixeira de Mattos, M.J., Hellingwerf, K.J., and Brul, S. (2002). The metabolic response of Saccharomyces cerevisiae to continuous heat stress. Mol. Biol. Reports JT - Mol. Biol. Reports 29, 103–6 PL–Netherlands PT–Journal Article.

Neves, M.J., and François, J. (1992). On the mechanism by which a heat shock induces trehalose accumulation in Saccharomyces cerevisiae. Biochem. J. 288, 859–864.

O'Brien, E., Lerman, J., and Chang, R. (2013). Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction. Mol. Syst. Biol. 9, 1–13.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. 27, 29–34.

Orth, J.D., Conrad, T.M., Na, J., Lerman, J. a, Nam, H., Feist, A.M., and Palsson, B.Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. Mol. Syst. Biol. 7, 535.

Piper, P.W. (1993). Molecular events associated with acquisition of heat tolerance by the yeast Saccharomyces cerevisiae. FEMS Microbiol. Rev. 11, 339–355.

Postmus, J., Canelas, A.B., Bouwman, J., Bakker, B.M., Van Gulik, W., De Mattos, M.J.T., Brul, S., and Smits, G.J. (2008). Quantitative analysis of the high temperature-induced glycolytic flux increase in Saccharomyces cerevisiae reveals dominant metabolic regulation. J. Biol. Chem. 283, 23524–23532.

Rosenberger, G., Ludwig, C., Röst, H.L., Aebersold, R., and Malmström, L. (2014). ALFQ: An R-package for estimating absolute protein quantities from label-free LC-MS/MS proteomics data. Bioinformatics 30, 2511–2513.

Saez-Rodriguez, J., Simeoni, L., Lindquist, J.A., Hemenway, R., Bommhardt, U., Arndt, B., Haus, U.U., Weismantel, R., Gilles, E.D., Klamt, S., et al. (2007). A logical model provides insights into T cell receptor signaling. PLoS Comput. Biol. 3, 1580–1590.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. 32, D431-3.

Smallbone, K., Simeonidis, E., Broomhead, D.S., and Kell, D.B. (2007). Something from nothing: bridging the gap between constraint-based and kinetic modelling. FEBS J. 274, 5576–5585.

Smallbone, K., Simeonidis, E., Swainston, N., and Mendes, P. (2010). Towards a genome-scale kinetic model of cellular metabolism. BMC Syst. Biol. 4, 6.

Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., et al. (2013). A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett. 587, 2832–2841.

Teusink, B., Passarge, J., Reijenga, C., Esgalhado, E., Weijden, C.C., Schepper, M., Walsh, M., Bakker, B.M., Dam, K., Westerhoff, H., et al. (2000). Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. Eur. J. Biochem. 267, 5313–5329.

Thammavongsa, V., Missiakas, D.M., and Schneewind, O. (2013). Staphylococcus aureus degrades neutrophil extracellular traps to promote immune cell death. Science 342, 863–866.

Vanlier, J., Wu, F., Qi, F., Vinnakota, K.C., Han, Y., Dash, R.K., Yang, F., and Beard, D.A. (2009). BISEN: Biochemical simulation environment. Bioinformatics 25, 836–837.

Varma, A., and Palsson, B.O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. Bio/Technology 12, 994–998.

Vilaprinyo, E., Alves, R., and Sorribas, A. (2006). Use of physiological constraints to identify quantitative design principles for gene expression in yeast adaptation to heat shock. BMC Bioinformatics 7, 184.

Visser, D., and Heijnen, J.J. (2003). Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab. Eng. 5, 164–176.

Zi, Z. (2011). SBML-PET-MPI: A parallel parameter estimation tool for Systems Biology Markup Language based models. Bioinformatics 27, 1028–1029.

Zi, Z., and Klipp, E. (2006). SBML-PET: A Systems Biology Markup Language-based parameter estimation tool. Bioinformatics 22, 2704–2705.

Chapter 4: Kinetic Model Parameterization with Quantitative Proteomics: Case Study with Trehalose Metabolism in *Saccharomyces cerevisiae*

4.1 Abstract

GRaPe 2.0 is a software built to accelerate construction of kinetic models of cell metabolism. It uses a combination of convenience kinetics and a genetic algorithm to achieve rapid model generation for hypothesis testing. The software was used to build a kinetic model of the trehalose metabolism in *Saccharomyces cerevisiae*. Model was parameterized with quantitative omics under standard conditions, and validated using data under heat stress conditions. The completed model was used to investigate factors related to the rise in flux during heat stress and the production of trehalose, a compound valued for its protective properties. The model found that feedforward activation of pyruvate kinase by fructose 1,6-bisphosphate during heat stress contributes to the increase in metabolic flux. Model was also able to demonstrate that overexpression of enzymes involved in production and degradation of trehalose can lead to higher trehalose yield in the cell.

4.2 Introduction

The development of mathematical models for aiding in metabolic engineering is not an uncommon activity for biologists nowadays, with various frameworks and approaches available for building the models (Almquist et al., 2014; Costa et al., 2016). However kinetic models are often limited to small or medium sized models due to the amount of information needed to construct them, ranging from rate laws for the reactions to the kinetic parameters. In order to circumvent this, GRaPe 2.0 was developed (Yap *et al* 2018, submitted). It uses a generalized Michaelis-Menten equation in the form of convenience kinetics (Liebermeister and Klipp, 2006) and parameter estimation to fill in the missing kinetic parameters. Kinetic parameters, which forms a vital part of kinetic models are constant values in the equations. They can be measured experimentally in the laboratory, and some are stored in databases such as BRENDA (Schomburg et al., 2004) and SABIO-RK (Krebs et al., 2007). GRaPe 2.0 provides users the option to circumvent the process of determining individual kinetic parameters by wet lab experiments.

In this study, a kinetic model consisting of trehalose metabolism in *Saccharomyces cerevisiae* as well as the upper portion of glycolysis was developed (Figure 4.1), using only convenience kinetics to form the rate equations. This model was fitted to steady state data of metabolites and fluxes. On top of that, the model uses quantitative proteomics data collected under two different conditions, for training and validation. Protein data from standard conditions were used to parameterize the model, while separate data collected under 37^{o}C were used to help simulate the biological condition of heat stress and validate the model. Not to be confused with the more extreme heat shock, heat stress is at a more physiological temperature of approximately 37^{o}C. Trehalose was a focus of this study as it is often associated with baker yeast during heat stress, as the microbe would accumulate high concentration of this protective molecule for survival (Mensonides et al., 2002; Parrou et al., 1997; Strassburg et al., 2010). Additionally, owing to its protective properties, trehalose has commercial value as it is used in various industries from pharmaceuticals to food and cosmetics (Paiva and Panek, 1996; Roser, 1991; Schiraldi et al., 2002). Its production in the industry have been relying on the use of enzymes from extremophiles expressed in other microbes (Schiraldi et al., 2002).

Another phenomenon observed during heat stress adaptation in *S. cerevisiae* is the increase in glycolytic flux (Mensonides et al., 2002; Postmus et al., 2008). Postmus et al. set out to investigate the possible factors that could contribute to the increase of flux, including gene expression, enzyme activity, protein expression, and metabolites profile. Interestingly, their data indicated minimal changes in gene and protein expression, but extensive changes in metabolic profile. Therefore, they attributed the changes in flux to the augmentations in the metabolic environment of the enzymes, such as the close to 10-fold increase of fructose 1,6-bisphosphate and 20-fold increase of pyruvate. Additionally, they postulated that the maintenance of the high flux is a result of the feedback activation of phosphofructokinase by fructose 2,6-bisphosphate and feedforward activation on pyruvate kinase by fructose 1,6-bisphosphate.

To investigate if the regulatory effects in the glycolytic system play a role in increasing the flux during heat stress, a kinetic model of trehalose metabolism in *S. cerevisiae* was constructed. This model should also help to determine the best way to produce trehalose without the need for temperature increase. The model was built using metabolomics, fluxomics and proteomics data collected under standard conditions, and subsequently validated using metabolomics and proteomics data collected under heat stress condition. The completed model was subjected to *in silico* regulation analysis and overexpression study. It was found that activation of pyruvate kinase does contribute slightly to the control of flux in the trehalose cycle and glycolysis, and that regulatory effects on enzymes involved in glucose entry play a significant role in affecting flux within the system. The model additionally predicted that the overexpression of enzymes directly involved in the production and degradation of trehalose would lead to an increase in its concentration in the system.

4.3 Methods

4.3.1 Model Construction

The model was built using GRaPe 2.0 (Yap et al. 2018, submitted), which uses convenience kinetics (Liebermeister and Klipp, 2006) to build the rate equations. GRaPe 2.0 is a tool for building kinetic models for cell metabolism, requiring input from the users on the substrates and products involved in the metabolic reactions and the enzymes that catalyse the reactions, which are then translated into rate equations automatically. The convenience kinetics assumes that all the reactions are reversible and have a random binding order. Models built in GRaPe use the Systems Biology Markup Language format (Le Novère et al., 2008).

4.3.2 Parameter Estimation

After establishing the system's network of reactions, the kinetic parameters are needed to complete the model. All the kinetic parameters of the model were estimated using a genetic algorithm (Holland, 1962) that uses tournament selection for choosing parents for the reproductive step and plague feature to remove individuals with low fitness values. The fitness of the individuals is measured using mean absolute percentage error between fitting data and simulated data.

The fitting data for the metabolites were obtained from two separate studies, one on *S. cerevisiae* in normal conditions measured using mass spectrometry (Smallbone et al., 2013) and another measured using nuclear magnetic resonance (Puig-Castellví et al., 2015). Although the work done by Puig-Castellví et al. used arbitrary units for the metabolites, the values of shared metabolites were not too dissimilar from those measured by Smallbone et al, so they were treated as mM. The fluxes were obtained from data from the reference strain measured by Blank and colleagues (Blank et al., 2005). All of the data used for parameter estimation were values measured during steady state.

The protein concentrations used in the model (for both parameter estimation and validation) were obtained from a quantitative and temporal study on *S. cerevisiae*'s proteins undergoing heat stress (Jarnuczak et al., 2018).

4.3.3 Manipulation and Simulation of Model

The validated model was modified using the JSBML library (Dräger et al., 2011) and Java code to make the changes needed for regulation analysis and overexpression model interrogation; the changes include altering concentration values of the enzymes, changes to the equations and kinetic parameters involved, and in the case of regulation analysis removal of activator and inhibitor constants. The simulation of the models up to steady state was done using the SBML ODE Solver Library (SOSLib, Machné et al., 2006).

4.4 Results

A kinetic model focusing on trehalose metabolism parameterized with quantitative metabolite concentrations, flux, and protein data collected before heat stress was applied to the cells, is presented here. The model was validated using a separate set of quantitative proteomics from the same study, but collected from cells undergoing heat stress, allowing the model to simulate heat stress. The validated model was used to study the possible causes of flux increase during heat stress as well as ways of increasing trehalose production in *S. cerevisiae* without the need for an increase in temperature.

4.4.1 Model of Trehalose Cycle & Upper Glycolysis

The completed model in this study included *S. cerevisiae*'s upper portion of glycolysis, and trehalose metabolism (Figure 4.1). Metabolism of trehalose is a cycle that branches off from glucose-6-phosphate of glycolysis and re-enters glycolysis as trehalose is broken down into two glucose molecules. In total the model contains 23 metabolites (6 external metabolites, boundary condition true in SBML), 20 enzymes and reactions. Within the reaction network there are 3 activation regulatory effects, 4 inhibitory effects (Voit, 2003). All these summed up to 106 kinetic parameters in the model that are fitted using omics data.

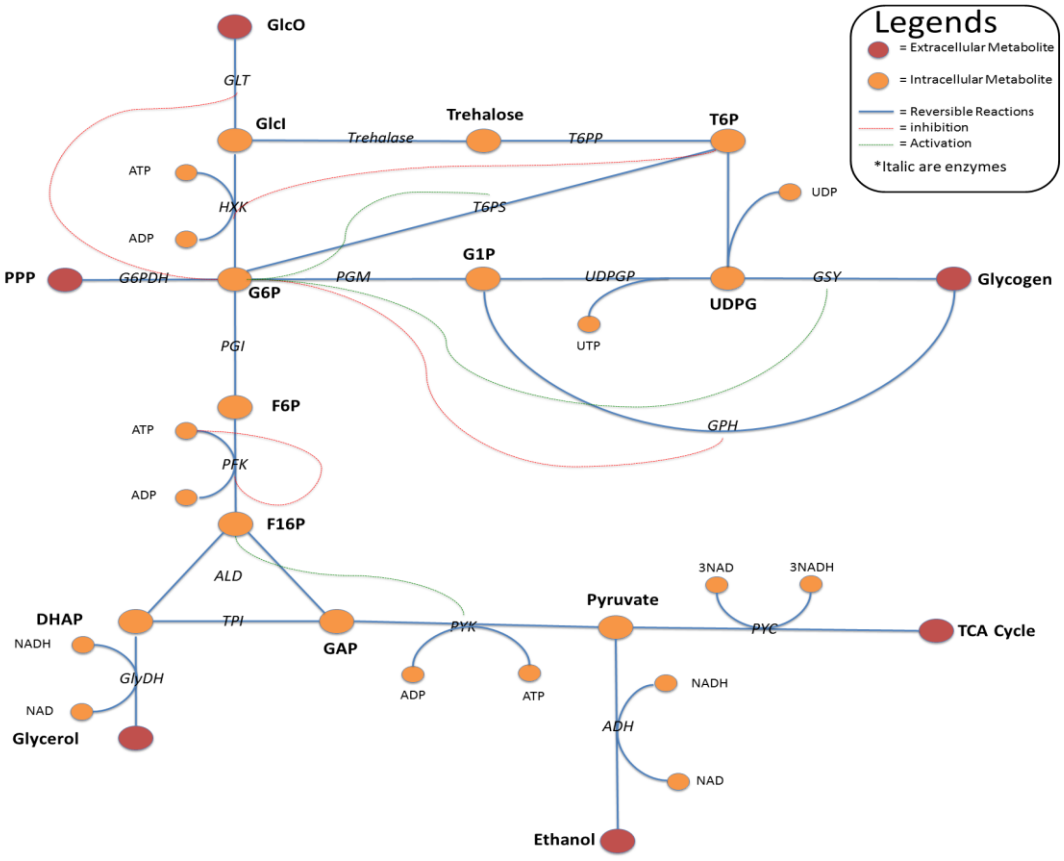

***Figure 4.3: Trehalose metabolism model**, with lower glycolysis simplified to a singular reaction from glyceraldehyde 3-phosphate to pyruvate, as well as skipping production of acetaldehyde to ethanol. GlcO: extracellular glucose, GlcI: intracellular glucose, G6P: glucose 6-phosphate, F6P: fructose 6-phosphate, F16BP: fructose 1,6-bisphosphate, DHAP: dihydroxyacetone phosphate, GAP: glyceraldehyde 3-phosphate, G1P: glucose 1-phosphate , UDPG: urine diphosphate glucose, T6P: trehalose 6-phosphate, GLT: glucose transport, HXK: hexokinase, PGI: phosphoglucose isomerase, PFK: phosphofructokinase, ALD: aldolase, TPI: triose-phosphate isomerase, PYK: pyruvate kinase, PYC: pyruvate carboxylase, ADH: alcohol dehydrogenase, PGM: phosphoglucomutase, UDPGP: UDPG phosphorylase, GSY: glycogen synthase, GPH: glycogen phosphorylase, T6PP: trehalose phosphatase, T6PS: trehalose-phosphate synthase, G6PDH: glucose 6-phosphate dehydrogenase.*

4.4.2 Model Fitting and Validation

Metabolites (mmol l-1)	Fitting Data	30°C	37°C
GlcI*	6.277	6.39	1.76
G6P*	0.774	0.74	0.14
F6P*	0.235	0.23	1.19
F16P*	4.583	4.52	7.81
DHAP*	1.162	1.15	0.99
GAP*	0.316	0.31	0.36
Pyruvate*	2.107	2.01	4.23
G1P		10.34	5.24
UDPG		0.01	0.01
T6P		0.05	0.07
Trehalose**	0.33	0.33	1.91
ATP**	3.2	5.58	5.57
ADP		0.004	0.01
UDP		0.81	0.97
UTP		0.58	0.42
NAD		1.57	1.62
NADH		0.22	0.17

Fluxes (mmol hr-1)	Fitting Data	30°C	37°C
GLT	16.7	15.52	21.71
HXK	16.7	15.93	22.29
PGI	14.2	13.23	21.44
PFK	14.2	13.23	21.44
ALD	15.2	13.23	21.44
TPI	13.5	11.68	10.87
G6PDH	1.8	1.75	0.12
GlycerolDH	1.7	1.55	10.57
PYK	28.6	24.91	32.30
ADH	23.6	18.30	21.58
PYC	5.0	6.62	10.72
PGM	0.8	0.75	0.44
UDPGP		0.85	1.41
GSY		0.65	1.12
GPH	0.10	0.10	0.96
T6PS	0.20	0.20	0.29
T6PP	0.20	0.20	0.29
Trehalase	0.20	0.20	0.29
ATPSyn		5.10	12.83
UDPtoUTP		0.85	1.41

Table 4.1: **Metabolites and fluxes in the trehalose metabolism model**. Fitting data were the values used to generate the model at standard conditions (30°C). Values in the 30°C and 37°C columns are values generated by the model when the model is using enzyme concentration from the respective temperatures. Blanks means the values were not fitted as they were not available. *metabolites uses data from Smallbone et al. (Smallbone et al., 2013). **metabolites (along with ethanol, glycerol and TCA cycle represented by their by-product succinate and citrate, for the external metabolites) uses data from (Puig-Castellví et al., 2015). Flux data are from (Blank et al., 2005).

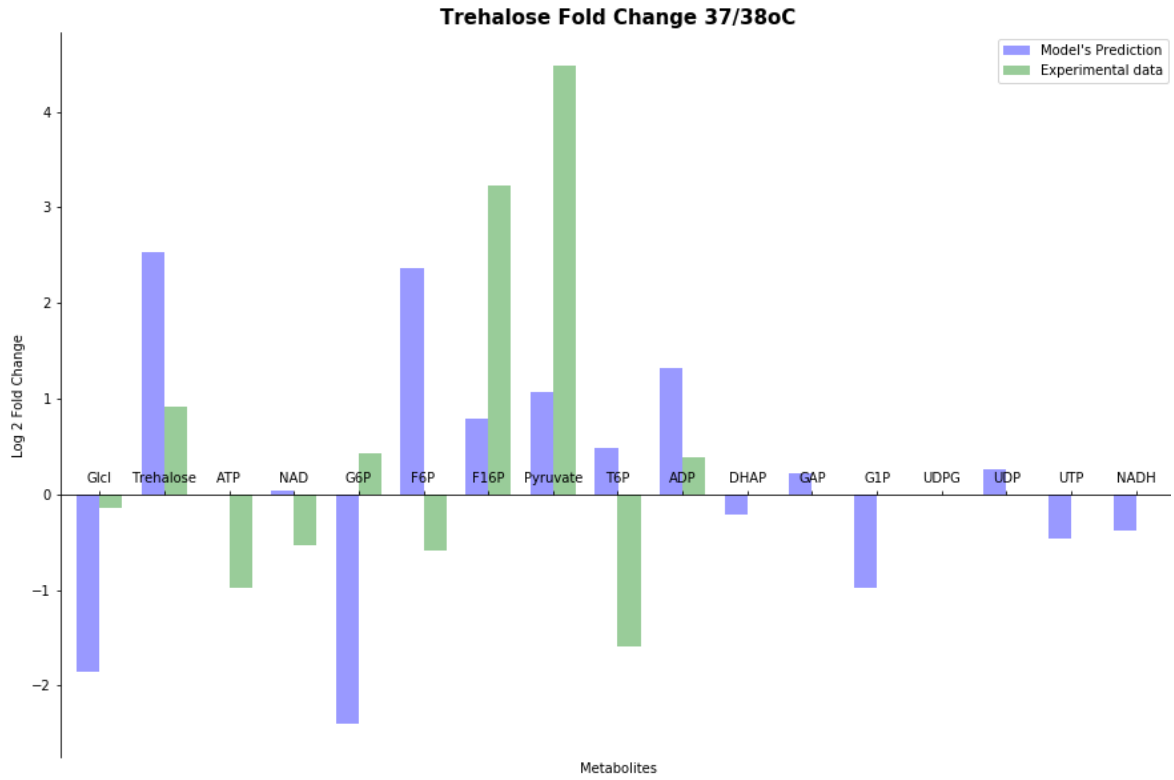

Figure 4.2: **Log2 fold changes of metabolites from 30ºC to 37ºC generated by the trehalose metabolism model**. With comparison from data measured experimentally by Puig-Castellví et al. (Puig-Castellví et al., 2015) for intracellular glucose (GlcI), trehalose adenosine triphosphate (ATP), and nicotinamide adenine dinucleotide (NAD) at 37ºC and Postmus et al. (Postmus et al., 2008) for glucose-6-phosphate (G6P), fructose-6-phosphate (F6P), fructose-1,6-bisphosphate (F16P), trehalose-6-phosphate (T6P), and adenosine diphosphate (ADP) 38ºC. Remaining metabolites had no reference experimental data for comparison.

The model was fitted with heterogeneous data for metabolites and fluxes values (sources discussed in Methods 4.2.2) to obtain the kinetic parameters. Following the parameter estimation step, the enzyme concentrations in the model were all replaced with those measured under heat stress condition to simulate the condition. A single source for protein values was used; values measured under 30°C were used for fitting, while those measured under 37°C were used for validation. The model was able to successfully replicate the metabolic responses observed in heat stress for *S. cerevisiae* (Fonseca et al., 2011; Mensonides et al., 2002; Postmus et al., 2008; Puig-Castellví et al., 2015; Voit, 2003), namely increase in overall fluxes in the network and high accumulation of trehalose (Table 4.1).

The root-mean-square error of the model's simulated values and fitting data at 30°C is 1.56, with the majority of the deviations contributed by fluxes (Table 4.1, right). The differences between simulated and experimentally measured values for fluxes at 30°C are all below 13% except for alcohol dehydrogenase with a deviation of 22%. For the metabolites, only ATP deviated significantly (Table 4.1, left). Qualitatively, changes in the direction for the metabolites are in the right direction for a majority of the metabolites, with the exception of trehalose-6-phosphate (T6P), glucose 6-phosphate (G6P) and fructose 6-phosphate (F6P) (Figure 4.2).

4.4.3 Regulation Analysis

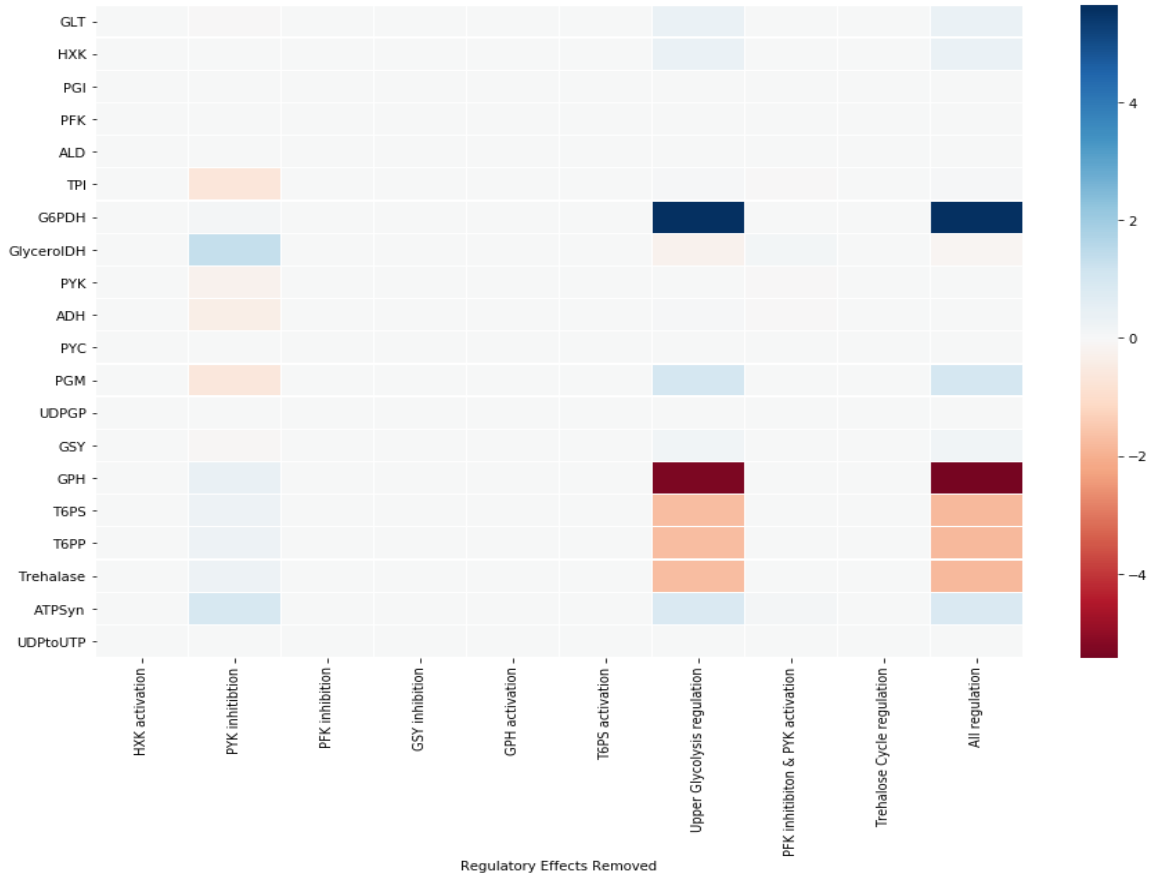

Figure 4.3: **Heat map showing fold changes for fluxes between the original model and a series of modified models where modifiers in reactions are removed**. Each column represents one modified model, each row represents one reaction flux. All these results were generated with the models undergoing heat stress.

In order to determine if modifiers play a role in increasing the overall fluxes during heat stress, regulation analysis was performed on the validated model. This was done by removing individual or groups of enzyme modifiers in the network of reactions while maintaining everything else and simulating the model until it reached a steady state. Data for removal of glucose transport inhibition is not shown because its removal leads to an unstable model that is unable to achieve steady state.

Of all the reactions investigated, the removal of modifiers involved in the upper glycolysis caused significant changes in fluxes relative to the original model's flux values when protein concentrations collected under 37°C were used (Figure 4.3). When there is no inhibition in upper

glycolysis, the trehalose cycle's fluxes see a drop, along with a massive decrease in the reaction where glycogen is broken down into glucose-1-phosphate, catalysed by glycogen phosphorylase (GPH), and with a huge increase in flux towards the pentose phosphate pathway. When the activation of pyruvate kinase is removed, there is a slight increase in flux within the trehalose cycle, but with drops observed in the entry to the trehalose cycle and lower glycolysis.

4.4.4 Theoretical Overexpression and its Yields

To determine the best target for metabolic engineering in *S. cerevisiae* in order to achieve higher trehalose production, theoretical overexpressions (doubling the enzymes' concentration) were performed with the model. In order to narrow down the search of enzymes to overexpress, metabolic control analysis (Heinrich and Rapoport, 1974; Kacser and Burns, 1973) was performed. It suggested that UDP-glucose phosphorylase and trehalose-6-phosphate synthase increase would result in higher trehalose accumulation. That was proved incorrect for UDP-glucose phosphorylase (Figure 4.4), but true for trehalose-6-phosphate synthase. However, overexpression for the trio of enzymes involved in trehalose production and degradation of trehalose delivered the best results in trehalose accumulation with 10000% increase in yield (Figure 4.4).

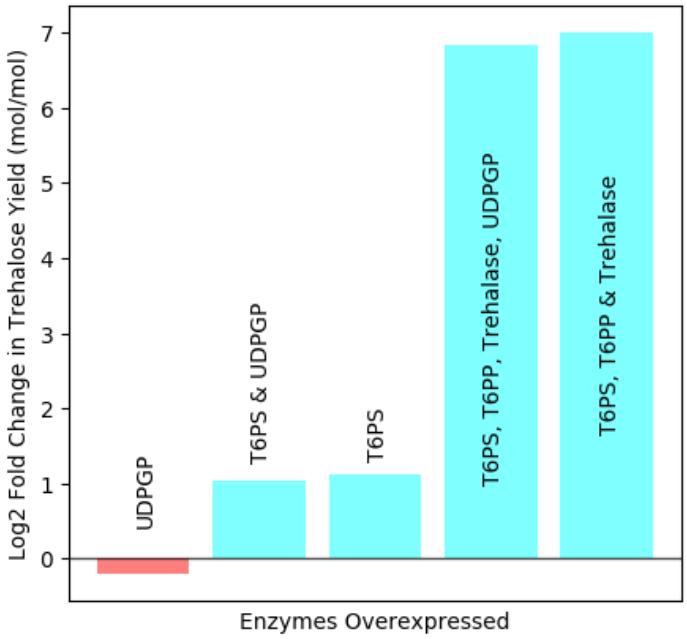

Figure 4.4: **Theoretical percentage increase in mol trehalose per mol glucose**. Enzymes here are overexpressed two-fold

4.5 Discussion

The development of kinetic models studying trehalose metabolism in *S. cerevisiae* is not a novel venture (Fonseca et al., 2011; Voit, 2003). Voit's work was used to determine metabolic regulation involved in the trehalose metabolism, while Fonseca and colleagues developed a model using time series of metabolic profiles to predict changes in protein levels.

Our model used quantitative protein data to inform kinetic parameters as well as to emulate heat stress. The use of proteomics over transcriptomics is the better approach for heat stress simulation as it was found that transcription changes are mostly transient for stress adaptations in *S. cerevisiae* (Gasch et al., 2000) but proteins remain longer in steady state to adapt to environmental stress (Jarnuczak et al., 2018).

The combination of convenience kinetics with heterogeneous sources of fitting data have been proven an effective modeling method here, as it produced a model with a good fit to the training data. The model is also capable of reproducing the metabolic phenomenon commonly observed during heat stress adaptation (Table 4.1). When *S. cerevisiae* is challenged with high temperature, it is commonly observed that fluxes would increase (Mensonides et al., 2002; Postmus et al., 2008) and trehalose would accumulate (Fonseca et al., 2011; Parrou et al., 1997; Puig-Castellví et al., 2015). When protein amounts collected during heat stress were input, the model was able to reproduce the increase in flux generated during heat stress, albeit to a lesser degree than those experimentally measured (Table 4.1). Additionally, it reproduced increased glucose consumption, which is indicated by the drop in glucose level and increased flux in glucose transport (Mensonides et al., 2002), as the cell would need to spend more energy to cope with the stress. Minimal changes in trehalose 6-phosphate were also predicted by the model, which is expected as this compound is toxic for the cell in high concentration (Fonseca et al., 2012). It is worth noting though that predictions for glucose 6-phosphate, fructose 6-phosphate and trehalose-6-phosphate were in the opposite direction of those measured by Postmus et al. The increase of flux as a result of heat stress is at least 5 fold. (Postmus et al., 2008), however our model only predicted a 40% increase in flux for most of the reactions. This discrepancy might be the result of difference in the experimental conditions between the Postmus et al. study, which measured the

flux values, and the Jarnuczak et al. study, which determined the protein concentrations used in this study. Postmus et al. used naïve cells grown to stationary phase to measure fluxes, while Jarnuzack et al. used heat adapted cells to measure protein concentrations, which were generated by moving them to 37C during the mid-exponential growth phase. This results in different genomic changes in the cells to better adapt to the increased temperature (Fonseca et al., 2011).

The model was additionally used to test the hypothesis that regulatory effects in the metabolic network contributes to flux increase. While our model did not include the activation of phosphofructokinase by fructose 2,6-bisphosphate, it included the feedforward activation on pyruvate kinase by fructose 1,6-bisphosphate. The removal of this activation on pyruvate kinase resulted in a slight drop of flux in lower glycolysis (Figure 4.3), confirming the hypothesis *in silico*. However, this also resulted in a slight increase in flux in the trehalose cycle, channeling flux from glycogen instead of upper glycolysis. The regulatory interactions that have the highest impact on fluxes of glycolysis and trehalose cycle are in upper glycolysis. When all three inhibitions on the upper glycolysis (glucose transport, hexokinase and phosphofructokinase) were removed, this resulted in a drop in flux within the trehalose cycle; instead, a higher flux was channeled towards the pentose phosphate pathway.

The prediction made in the enzyme overexpression investigation leads to a high increase in percentage yield of trehalose production (Figure 4.4), which was also observed experimentally. In the work done by Fonseca et al (Fonseca et al., 2011) it was shown that heat adapted yeast cells have trehalose concentrations that increased from 4mM up to 100mM.

4.6 Conclusion

Our work shows that the use of convenience kinetic for metabolic modelling is a good solution for rapid prototyping of metabolic models for hypothesis generation. This methodology allows the use a single set of multi-omics data for parameter estimation. Subsequently, the model's enzyme concentrations can be updated with quantitative proteomics collected from a different condition such as heat stress, which then allows the model to simulate this new condition. The output can then be compared with a different set of multi-omics data, constituting the validation step of the model. Following this approach, we were able to make *in silico* predictions to engineer a strain of

S. cerevisiae that accumulates trehalose without an increase in temperature. This has potential industrial applications as trehalose is a valuable commercial product for its protective properties.

4.7 References

Adiamah, D. a, Handl, J., and Schwartz, J.-M. (2010). Streamlining the construction of large-scale dynamic models using generic kinetic equations. Bioinformatics *26*, 1324–1331.

Ahnert, K., Demidov, D., and Mulansky, M. (2014). Solving Ordinary Differential Equations on GPUs. In Numerical Computations with GPUs, (Cham: Springer International Publishing), pp. 125–157.

Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J., and Jirstrand, M. (2014). Kinetic models in industrial biotechnology - Improving cell factory performance. Metab. Eng. *24*, 38–60.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. a., and Blom, J.G. (2009). Systems biology: Parameter estimation for biochemical models. FEBS J. *276*, 886–902.

Azeloglu, E.U., and Iyengar, R. (2015). Good practices for building dynamical models in systems biology. Sci. Signal. *8*.

Bartocci, E., and Lió, P. (2016). Computational Modeling, Formal Analysis, and Tools for Systems Biology. PLoS Comput. Biol. *12*, 1–22.

Bernardinello, L., and Cindio, F. (1992). A survey of basic net models and modular net classes. pp. 304–351.

Blank, L.M., Kuepfer, L., Sauer, U., Papin, J., Stelling, J., Price, N., Klamt, S., Schuster, S., Palsson, B., Price, N., et al. (2005). Large-scale 13 C-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast. Genome Biol. *6*, R49.

Bornstein, B.J., Keating, S.M., Jouraku, A., and Hucka, M. (2008). LibSBML: an API Library for SBML. Bioinformatics *24*, 880–881.

Breitling, R., Gilbert, D., Heiner, M., and Orton, R. (2008). A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Brief. Bioinform. *9*, 404–421.

Chelouah, R., and Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. Eur. J. Oper. Res. *148*, 335–348.

Chou, I.C., and Voit, E.O. (2009). Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. *219*, 57–83.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). Distributed Optimization by Ants Colonies. Proc. ECAL - Eur. Conf. Artif. Life, Paris, Fr. 134–142.

Costa, R.S., Machado, D., Rocha, I., and Ferreira, E.C. (2010). Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations. Biosystems *100*, 150–157.

Costa, R.S., Veríssimo, A., and Vinga, S. (2014). KiMoSys: a web-based repository of experimental data for KInetic MOdels of biological SYStems. BMC Syst. Biol. *8*, 85.

Costa, R.S., Hartmann, A., and Vinga, S. (2016). Kinetic modeling of cell metabolism for microbial production. J. Biotechnol. *219*, 126–141.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. Mol. Syst. Biol. *7*.

Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. Simulation *79*, 740–747.

Dörr, A., Keller, R., Zell, A., and Dräger, A. (2014). SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. Computation *2*, 246–257.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst. Biol. *2*, 39.

Dräger, A., Rodriguez, N., Dumousseau, M., Dörr, A., Wrzodek, C., Le Novère, N., Zell, A., and Hucka, M. (2011). JSBML: A flexible java library for working with SBML. Bioinformatics *27*, 2167–2168.

Du, B., Zielinski, D.C., Kavvas, E.S., Dräger, A., Tan, J., Zhang, Z., Ruggiero, K.E., Arzumanyan, G.A., and Palsson, B.O. (2016). Evaluation of rate law approximations in bottom-up kinetic models of metabolism. BMC Syst. Biol. *10*, 40.

Duarte, N.C., Becker, S.A., Jamshidi, N., Thiele, I., Mo, M.L., Vo, T.D., Srivas, R., and Palsson, B.O. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc. Natl. Acad. Sci. *104*, 1777–1782.

Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., et al. (2018). The Reactome Pathway Knowledgebase. Nucleic Acids Res. *46*, D649–D655.

Fast, T., Wall, T., and Chen, L. (2007). Java Native Access.

Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007). A genome-scale metabolic reconstruction for

Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol. Syst. Biol. *3*.

Fonseca, L.L., Sánchez, C., Santos, H., and Voit, E.O. (2011). Complex coordination of multi-scale cellular responses to environmental stress. Mol. Biosyst. *7*, 731–741.

Fonseca, L.L., Chen, P.-W., and Voit, E.O. (2012). Canonical Modeling of the Multi-Scale Regulation of the Heat Stress Response in Yeast. Metabolites *2*, 221–241.

Forster, J. (2003). Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network. Genome Res. *13*, 244–253.

Funahashi, A., and Matsuoka, Y. (2008). CellDesigner 3.5: a versatile modeling tool for biochemical networks. Proc. IEEE *96*, 1254–1265.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell *11*, 4241–4257.

Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS. Decis. Sci. *8*, 156–166.

Guijas, C., Montenegro-Burke, J.R., Domingo-Almenara, X., Palermo, A., Warth, B., Hermann, G., Koellensperger, G., Huan, T., Uritboonthai, W., Aisporna, A.E., et al. (2018). METLIN: A Technology Platform for Identifying Knowns and Unknowns. Anal. Chem. *90*, 3156–3164.

Haug, K., Salek, R.M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., Mahendraker, T., Williams, M., Neumann, S., Rocca-Serra, P., et al. (2013). MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res. *41*, D781–D786.

Heinrich, R., and Rapoport, T.A. (1974). A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. Eur. J. Biochem. *42*, 89–95.

Hillmer, R.A. (2015). Systems Biology for Biologists. PLOS Pathog. *11*, e1004786.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., and Woodward, C.S. (2005). SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. ACM Trans. Math. Softw. *31*, 363–396.

Holland, J.H. (1962). Outline for a Logical Theory of Adaptive Systems. J. ACM *9*, 297–314.

Hooke, R., and Jeeves, T.A. (1961). `` Direct Search" Solution of Numerical and Statistical Problems. J. ACM *8*, 212–229.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI--a COmplex PAthway SImulator. Bioinformatics *22*, 3067–3074.

Hübner, K., Sahle, S., and Kummer, U. (2011). Applications and trends in systems biology in biochemistry. FEBS J. *278*, 2767–2857.

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

Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., et al. (2007). Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations. Science (80-.). *316*, 593–597.

Jamshidi, N., and Palsson, B.Ø. (2008). Formulating genome-scale kinetic models in the post-genome era. Mol. Syst. Biol. *4*, 171.

Jarnuczak, A.F., Albornoz, M.G., Eyers, C.E., Grant, C.M., and Hubbard, S.J. (2018). A quantitative and temporal map of proteostasis during heat shock in *Saccharomyces cerevisiae*. Mol. Omi. *14*, 37–52.

Kacser, H., and Burns, J.A. (1973). The control of flux. Symp. Soc. Exp. Biol. *27*, 65–104.

Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (2011). ConsensusPathDB: toward a more complete picture of cell biology. Nucleic Acids Res. *39*, D712–D717.

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. *22*, 437–467.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, (IEEE), pp. 1942–1948.

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

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

King, E.L., and Altman, C. (1956). A Schematic Method of Deriving the Rate Laws for Enzyme-Catalyzed Reactions. J. Phys. Chem. *60*, 1375–1378.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. Science (80-.). *220*, 671–680.

Kitano, H. (2002). Systems biology: a brief overview. Science *295*, 1662–1664.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009a). Parameter Estimation. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 152–164.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009b). Mathematics. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 449–473.

Koch, I., Junker, B.H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics *21*, 1219–1226.

Krebs, O., Golebiewski, M., Kania, R., and Mir, S. (2007). SABIO-RK: a data warehouse for biochemical reactions and their kinetics. J. Integr. Bioinform.

Kronfeld, M., Planatscher, H., and Zell, A. (2010). The EvA2 optimization framework. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) *6073 LNCS*, 247–250.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least. Q. Appl. Math. *2*, 164–168.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci. *101*, 4781–4786.

Liebermeister, W., and Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor. Biol. Med. Model. *3*, 41.

Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010). Modular rate laws for enzymatic reactions: thermodynamics, elasticities and implementation. Bioinformatics *26*, 1528–1534.

Link, H., Christodoulou, D., and Sauer, U. (2014). Advancing metabolic models with kinetic information. Curr. Opin. Biotechnol. *29*, 8–14.

Machado, D., Costa, R.S., Rocha, M., Ferreira, E.C., Tidor, B., and Rocha, I. (2011). Modeling formalisms in Systems Biology. AMB Express *1*, 45.

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: A native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics *22*, 1406–1407.

Matsuda, F., Kinoshita, S., Nishino, S., Tomita, A., and Shimizu, H. (2017). Targeted proteome analysis of single-gene deletion strains of Saccharomyces cerevisiae lacking enzymes in the central carbon metabolism. PLoS One *12*, e0172742.

Mensonides, F.I.C., Schuurmans, J.M., Teixeira de Mattos, M.J., Hellingwerf, K.J., and Brul, S. (2002). The metabolic response of Saccharomyces cerevisiae to continuous heat stress. Mol. Biol. Reports JT - Mol. Biol. Reports *29*, 103–6 PL–Netherlands PT–Journal Article.

Nelder, J.A., and Mead, R. (1965). A Simplex Method for Function Minimization. Comput. J. *7*, 308–313.

Nocedal, J., and Wright, S. (1999). Numerical Optimization (New York: Springer).

Novere, N. Le, Hucka, M., and Mi, H. (2009). The Systems Biology Graphical Notation. Nat. Biotechnol. *27*, 735–742.

Le Novère, N., Hucka, M., Hoops, S., Keating, S., Sahle, S., and Wilkinson, D. (2008). Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Nat. Preced. 1–38.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *27*, 29–34.

Orth, J.D., Conrad, T.M., Na, J., Lerman, J. a, Nam, H., Feist, A.M., and Palsson, B.Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. Mol. Syst. Biol. *7*, 535.

Paiva, C.L.A., and Panek, A.D. (1996). Biotechnological Applications of the Disaccharide Trehalose. pp. 293–314.

Park, T.-Y., and Froment, G.F. (1998). A hybrid genetic algorithm for the estimation of parameters in detailed kinetic models. Comput. Chem. Eng. *22*, S103–S110.

Parrou, J.L., Teste, M.A., and François, J. (1997). Effects of various types of stress on the metabolism of reserve carbohydrates in Saccharomyces cerevisiae: Genetic evidence for a stress-induced recycling of glycogen and trehalose. Microbiology *143*, 1891–1900.

Petri, C.A. (1962). Kommunikation mit Automaten (Communication with Automata). University of Bonn.

Postmus, J., Canelas, A.B., Bouwman, J., Bakker, B.M., Van Gulik, W., De Mattos, M.J.T., Brul, S., and Smits, G.J. (2008). Quantitative analysis of the high temperature-induced glycolytic flux increase in Saccharomyces cerevisiae reveals dominant metabolic regulation. J. Biol. Chem. *283*,

23524–23532.

Puig-Castellví, F., Alfonso, I., Piña, B., and Tauler, R. (2015). A quantitative 1H NMR approach for evaluating the metabolic response of Saccharomyces cerevisiae to mild heat stress. Metabolomics *11*, 1612–1625.

Rodriguez, N., Donizelli, M., and Le Novère, N. (2007). SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). BMC Bioinformatics *8*, 79.

Roser, B. (1991). Trehalose, a new approach to premium dried foods. Trends Food Sci. Technol. *2*, 166–169.

Savageau, M.A. (1976). Biochemical Systems Analysis (Reading, MA: Addison-Wesley Pub. Co.).

Schaff, J., Slepchenko, B., and Loew, L. (2000). Physiological modeling with virtual cell framework. Methods Enzymol. *321*, 1–23.

Schiraldi, C., Di Lernia, I., and De Rosa, M. (2002). Trehalose production: Exploiting novel approaches. Trends Biotechnol. *20*, 420–425.

Schmidt, T., Samaras, P., Frejno, M., Gessulat, S., Barnert, M., Kienegger, H., Krcmar, H., Schlegl, J., Ehrlich, H.-C., Aiche, S., et al. (2018). ProteomicsDB. Nucleic Acids Res. *46*, D1271–D1281.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. *32*, D431-3.

Smallbone, K., Simeonidis, E., Swainston, N., and Mendes, P. (2010). Towards a genome-scale kinetic model of cellular metabolism. BMC Syst. Biol. *4*, 6.

Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., et al. (2013). A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett. *587*, 2832–2841.

Strassburg, K., Walther, D., Takahashi, H., Kanaya, S., and Kopka, J. (2010). Dynamic transcriptional and metabolic responses in yeast adapting to temperature stress. OMICS *14*, 249–259.

Sun, J., Garibaldi, J.M., and Hodgman, C. (2012). Parameter estimation using meta-heuristics in systems biology: a comprehensive review. IEEE/ACM Trans. Comput. Biol. Bioinform. *9*, 185–202.

Varma, A., and Palsson, B.O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. Bio/Technology *12*, 994–998.

Viant, M.R., Kurland, I.J., Jones, M.R., and Dunn, W.B. (2017). How close are we to complete

annotation of metabolomes? Curr. Opin. Chem. Biol. *36*, 64–69.

Visser, D., and Heijnen, J.J. (2003). Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab. Eng. *5*, 164–176.

Vizcaíno, J.A., Csordas, A., Del-Toro, N., Dianes, J.A., Griss, J., Lavidas, I., Mayer, G., Perez-Riverol, Y., Reisinger, F., Ternent, T., et al. (2016). 2016 update of the PRIDE database and its related tools. Nucleic Acids Res. *44*, D447–D456.

Voit, E.O. (2003). Biochemical and genomic regulation of the trehalose cycle in yeast: Review of observations and canonical model analysis. J. Theor. Biol. *223*, 55–78.

Wang, M., Herrmann, C.J., Simonovic, M., Szklarczyk, D., and von Mering, C. (2015). Version 4.0 of PaxDb: Protein abundance data, integrated across model organisms, tissues, and cell-lines. Proteomics *15*, 3163–3168.

Wolkenhauer, O., and Mesarović, M. (2005). Feedback dynamics and cell function: Why systems biology is called Systems Biology. Mol. Biosyst. *1*, 14.

Yen, J., Liao, J.C., Lee, B.L.B., and Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. Syst. Man Cybern. Part B Cybern. a Publ. IEEE Syst. Man Cybern. Soc. *28*, 173–191.

Yurkovich, J.T., and Palsson, B.O. (2018). Quantitative -omic data empowers bottom-up systems biology. Curr. Opin. Biotechnol. *51*, 130–136.

Zhang, Z., Shen, T., Rui, B., Zhou, W., Zhou, X., Shang, C., Xin, C., Liu, X., Li, G., Jiang, J., et al. (2015). CeCaFDB: a curated database for the documentation, visualization and comparative analysis of central carbon metabolic flux distributions explored by 13C-fluxomics. Nucleic Acids Res. *43*, D549–D557.

Zi, Z. (2011). SBML-PET-MPI: A parallel parameter estimation tool for Systems Biology Markup Language based models. Bioinformatics *27*, 1028–1029.

Zi, Z., and Klipp, E. (2006). SBML-PET: A Systems Biology Markup Language-based parameter estimation tool. Bioinformatics *22*, 2704–2705.

Zuberek, W.M. (1991). Timed Petri nets definitions, properties, and applications. Microelectron. Reliab. *31*, 627–644.

Chapter 5: Discussion, Future Work and Conclusion

5.1 Discussion

The field of systems biology has spawned out of the explosion of 'omics' data made possible with the massive improvement in high-throughput experiment methods (Jamshidi and Palsson, 2008; Kitano, 2002). Within this growing field, various modelling techniques to study biology at a system level were introduced, and as a result multiple models were generated. Not all methods provide the same level of understanding of the system though; there are static models that provide the qualitative information on the final state of a system and there are kinetic models that inform the users how the state is reached dynamically and quantitatively.

Outputs of kinetic models are advantageous relative to static models as they allow users to understand the dynamic processes and the impact of regulatory effects in a biological system, which is typically not possible with static models (Link et al., 2014). Despite the advantages they offer, dynamic models that study cell metabolism are not favoured by biologists (Smallbone et al., 2010). The development of kinetic metabolic models is impeded by the incomprehensiveness of metabolic data in comparison to other omics data (Viant et al., 2017); kinetic data such as parameters and rate laws are incomplete; metabolomics data are missing or measured under variable conditions. Additionally, not every biologist is equipped with know-how on building kinetic models as it can be overwhelming to dive into a new computational technique. While software tools for the process of model building exist, they often require a certain level of expertise before users can take advantage of the features they offer.

To encourage more biologists to utilise dynamic models in their research, GRaPe 2.0 was developed. GRaPe 2.0 is a software that simplifies the kinetic model building process for users that wish to study cell metabolism. It offers an intuitive user interface, automated equation generation, and parameter estimation to fill in missing kinetic parameters. On top of its graphical user interface, it offers a format that biologists are familiar with, tab separated spreadsheet to build models. It solves the problem of incomplete rate laws for most of enzymatic reactions by using convenience kinetics (Liebermeister and Klipp, 2006) for all the reactions. Convenience kinetics is

an approximate rate law for describing enzymatic reactions, it is a direct generalisation of the classic Michaelis-Menten kinetics following similar assumptions. As a generalisation, it is reversible, has fewer parameters, is saturable, and allows for inclusion of activators and inhibitors, which is useful as feedback dynamics is an important feature in biological processes to model (Link et al., 2014; Wolkenhauer and Mesarović, 2005). Use of an approximate rate law is a good substitute for enzymatic reactions that are well studied (Costa et al., 2010). Additionally when it is applied to an entire metabolic network, it is still able to produce results close to a true model that uses well studied rate laws and measured kinetic parameters (Du et al., 2016). For the parameter estimation process, a genetic algorithm was implemented in the software. This algorithm was chosen as it is a global optimization algorithm that explores multiple minima, and with its random elements in each step, it typically escapes local minima of the search space better than other algorithms if it is able to maintain the diversity in the group of candidate parameter sets throughout the process.

This framework that combines convenience kinetics and genetic algorithm was applied to study the glycolytic network in *Saccharomyces cerevisiae* with two separate models, one that included enzyme regulatory effects and one that did not. The models' kinetic parameters were fitted using steady state quantitative metabolite, flux and protein data measured under standard conditions. One of the resulting models was successfully validated as it was able to generate increased flux phenomena that occurs during heat stress condition, after the model's protein values were replaced with those measured under heat stress condition. The validated model was one that included feedback regulations in it; a separate fitted model that did not include enzyme modifiers was unable to produce the increased flux. This work here demonstrates the importance of enzyme regulatory effects in kinetic models as well as the application of GRaPe 2.0 as a tool for model building, the models generated provide an easy way for users to validate the models.

A computational model is a very useful tool for biologists in research, it can aid in planning experiments by determining the best target to manipulate in a system. Models generated by GRaPe 2.0 can be used for this purpose with ease, as the models built are exported in SBML format, they can be imported by supported tools for manipulation and analysis. An example of this process was shown with a model built to study trehalose metabolism in *S. cerevisiae*. This model was fitted and validated using quantitative metabolomics, fluxomics and proteomics data. The

parameterization of the model was done using protein values collected under 30°C. Subsequently, to simulate heat stress, the protein values were substituted with those collected under 37°C (heat stress condition). During heat stress, *S. cerevisiae*'s glycolytic flux is known to increase. A reason for this increase was speculated to be the feedforward activation of pyruvate kinase by fructose 1,6-bisphosphate. Our model confirmed this hypothesis, as its simulation under heat stress without the activator effect in pyruvate kinase shows a decrease in flux compared to an unmodified model under heat stress. Additionally, to determine the best metabolic engineering target to produce higher amount of trehalose (a commercially valuable product), an overexpression study was carried out. This was done by doubling the protein values in the model for the respective enzymes of interest, and it led to finding specific enzymes for overexpression that can result in higher trehalose yield per glucose. While this work performed with our framework demonstrated the effects of overexpression studies and removal of regulatory effects, it can just be as easily applied to study knockouts or knockdowns in the system by manipulating the enzyme values in the model with the right approach; i.e. setting the initial enzyme values to zero creates a knockout effect; similarly to achieve a knockdown effect, users can use tools like CellDesigner (Funahashi and Matsuoka, 2008) to make enzyme values zero at a given timeframe in a simulation.

5.2 Future work

While this study has shown some of the capabilities and work that can be carried out with our software, there is more that can be done to improve the software as well as more applications that can be carried out. The work that can be done to improve the software and further applications are discussed in this section.

5.2.1 Hybrid Parameter Estimation

One of the assets of the genetic algorithm methodology is its flexibility. This flexibility allows for a myriad of variations in its operators. To further improve on the genetic algorithm in the software, it can be hybridised it with other optimization methods, which is a common practice to reduce computation time or to produce better results (Chelouah and Siarry, 2003; Park and Froment, 1998; Yen et al., 1998). There are several ways to approach this, a direct method consists in running a local optimizer on the set of parameters found by the genetic algorithm to refine the results within the local search space. Alternatively, a local search can be performed on each set of new parameters in the genetic algorithm similar to a scatter search, while still maintaining

operators used in genetic algorithms. However, this approach would be computationally expensive, a balance would be to select a diverse set of parameter sets and running local search methods on them; this combination of global and local search algorithms would theoretically explore most of the minima in the search space. Another approach is to reduce computation time by applying a local search method in each generation of genetic algorithm only on select fittest individuals, increasing the convergence rate.

5.2.2 Use of GPU to solve Ordinary Differential Equations

The main bottleneck in the parameter estimation process is solving the objective function for each parameter set, which mainly comes down to solving the ordinary differential equations (ODE) of the model that are needed for the objective function evaluation. To reduce this bottleneck, a solution is to make use of the computational power of modern GPUs (Ahnert et al., 2014). Currently the ODEs are solved in GRaPe 2.0 using the SUNDIALS library that runs on CPU; replacing it with appropriate libraries that run on GPU such as odeint (www.odeint.com) would allow for a decrease in computation time when running the parameter estimation step.

5.2.3 Central Carbon Metabolism Model in *Saccharomyces cerevisiae*

As a model scales so do the number of parameters within it. This results in a higher volume of information necessary to better constraint the parameter estimation step. Having more data during fitting is important to reduce the number of unidentifiable parameters. For *Escherichia* coli data, there was a landmark paper that generated multiple perturbation data sets for metabolites, RNA and protein levels (Ishii et al., 2007). Data from this paper have been used to build kinetic models of varying sizes (Khodayari and Maranas, 2016; Khodayari et al., 2014). Unfortunately, there isn't a similar level of information available for *Saccharomyces cerevisiae*, although there is quantitative flux and protein data of central carbon metabolism collected for multiple *S. cerevisiae* mutants (Blank et al., 2005; Matsuda et al., 2017). The combination of these two datasets provides enough information for parameterization of a model as well as its validation using the GRaPe 2.0 software.

5.2.4 Extended Model Upstream of the Metabolic Model

Models built here have been done with the assumption that the system is enclosed, isolated from other dynamic processes. However, the reality is not generally isolated, therefore one method to improve on models studying metabolic systems would be to include interactions with other dynamic systems. One level above the metabolic system would be proteins, which are controlled

by gene expression, which is subsequently controlled by signalling processes. This can be done in a stepwise process by first including a model of protein production and degradation, which is to be coupled with the metabolic model. Following that a model of gene expression can be added, and finally a signalling network on top of it. This requires a multiscale modelling format that uses different formalisms for each level of the model. For example, for signalling networks, agent-based models can be applied, and generalized mass action kinetics can be applied to both gene expression and protein models.

5.3 Conclusion

Computational models are an extremely useful tool for research, they can help direct experiments, saving time and money for researchers. However, the adoption rate of kinetic models is low despite the increasing amount of biological data found in the public domain. To address this, GRaPe 2.0 was introduced, a tool that simplifies and accelerates the construction of kinetic models for cell metabolism. It does this with an intuitive user interface, automated approximate rate equation generation, and parameter estimation to fill in the gaps of missing and heterogenous data. Its effectiveness was demonstrated with two separate investigations, one as a proof of concept of the framework, another on the its application as a model in aiding biological research. GRaPe 2.0's ease of use that hides away the complex modelling processes will serve as a gateway tool for scientists that want to make use of kinetic models in their research.

5.4 References

Adiamah, D. a, Handl, J., and Schwartz, J.-M. (2010). Streamlining the construction of large-scale dynamic models using generic kinetic equations. Bioinformatics *26*, 1324–1331.

Ahnert, K., Demidov, D., and Mulansky, M. (2014). Solving Ordinary Differential Equations on GPUs. In Numerical Computations with GPUs, (Cham: Springer International Publishing), pp. 125–157.

Almquist, J., Cvijovic, M., Hatzimanikatis, V., Nielsen, J., and Jirstrand, M. (2014). Kinetic models in industrial biotechnology - Improving cell factory performance. Metab. Eng. *24*, 38–60.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. a., and Blom, J.G. (2009). Systems biology: Parameter estimation for biochemical models. FEBS J. *276*, 886–902.

Azeloglu, E.U., and Iyengar, R. (2015). Good practices for building dynamical models in systems biology. Sci. Signal. *8*.

Bartocci, E., and Lió, P. (2016). Computational Modeling, Formal Analysis, and Tools for Systems Biology. PLoS Comput. Biol. *12*, 1–22.

Bernardinello, L., and Cindio, F. (1992). A survey of basic net models and modular net classes. pp. 304–351.

Blank, L.M., Kuepfer, L., Sauer, U., Papin, J., Stelling, J., Price, N., Klamt, S., Schuster, S., Palsson, B., Price, N., et al. (2005). Large-scale 13 C-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast. Genome Biol. *6*, R49.

Bornstein, B.J., Keating, S.M., Jouraku, A., and Hucka, M. (2008). LibSBML: an API Library for SBML. Bioinformatics *24*, 880–881.

Breitling, R., Gilbert, D., Heiner, M., and Orton, R. (2008). A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Brief. Bioinform. *9*, 404–421.

Chelouah, R., and Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. Eur. J. Oper. Res. *148*, 335–348.

Chou, I.C., and Voit, E.O. (2009). Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. *219*, 57–83.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). Distributed Optimization by Ants Colonies. Proc. ECAL - Eur. Conf. Artif. Life, Paris, Fr. 134–142.

Costa, R.S., Machado, D., Rocha, I., and Ferreira, E.C. (2010). Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations. Biosystems *100*, 150–157.

Costa, R.S., Veríssimo, A., and Vinga, S. (2014). KiMoSys: a web-based repository of experimental data for KInetic MOdels of biological SYStems. BMC Syst. Biol. *8*, 85.

Costa, R.S., Hartmann, A., and Vinga, S. (2016). Kinetic modeling of cell metabolism for microbial production. J. Biotechnol. *219*, 126–141.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. Mol. Syst. Biol. *7*.

Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. Simulation *79*, 740–747.

Dörr, A., Keller, R., Zell, A., and Dräger, A. (2014). SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. Computation *2*, 246–257.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst. Biol. *2*, 39.

Dräger, A., Rodriguez, N., Dumousseau, M., Dörr, A., Wrzodek, C., Le Novère, N., Zell, A., and Hucka, M. (2011). JSBML: A flexible java library for working with SBML. Bioinformatics *27*, 2167–2168.

Du, B., Zielinski, D.C., Kavvas, E.S., Dräger, A., Tan, J., Zhang, Z., Ruggiero, K.E., Arzumanyan, G.A., and Palsson, B.O. (2016). Evaluation of rate law approximations in bottom-up kinetic models of metabolism. BMC Syst. Biol. *10*, 40.

Duarte, N.C., Becker, S.A., Jamshidi, N., Thiele, I., Mo, M.L., Vo, T.D., Srivas, R., and Palsson, B.O. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proc. Natl. Acad. Sci. *104*, 1777–1782.

Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., et al. (2018). The Reactome Pathway Knowledgebase. Nucleic Acids Res. *46*, D649–D655.

Fast, T., Wall, T., and Chen, L. (2007). Java Native Access.

Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007). A genome-scale metabolic reconstruction for

Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol. Syst. Biol. *3*.

Fonseca, L.L., Sánchez, C., Santos, H., and Voit, E.O. (2011). Complex coordination of multi-scale cellular responses to environmental stress. Mol. Biosyst. *7*, 731–741.

Fonseca, L.L., Chen, P.-W., and Voit, E.O. (2012). Canonical Modeling of the Multi-Scale Regulation of the Heat Stress Response in Yeast. Metabolites *2*, 221–241.

Forster, J. (2003). Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network. Genome Res. *13*, 244–253.

Funahashi, A., and Matsuoka, Y. (2008). CellDesigner 3.5: a versatile modeling tool for biochemical networks. Proc. IEEE *96*, 1254–1265.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell *11*, 4241–4257.

Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS. Decis. Sci. *8*, 156–166.

Guijas, C., Montenegro-Burke, J.R., Domingo-Almenara, X., Palermo, A., Warth, B., Hermann, G., Koellensperger, G., Huan, T., Uritboonthai, W., Aisporna, A.E., et al. (2018). METLIN: A Technology Platform for Identifying Knowns and Unknowns. Anal. Chem. *90*, 3156–3164.

Haug, K., Salek, R.M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., Mahendraker, T., Williams, M., Neumann, S., Rocca-Serra, P., et al. (2013). MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res. *41*, D781–D786.

Heinrich, R., and Rapoport, T.A. (1974). A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. Eur. J. Biochem. *42*, 89–95.

Hillmer, R.A. (2015). Systems Biology for Biologists. PLOS Pathog. *11*, e1004786.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., and Woodward, C.S. (2005). SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. ACM Trans. Math. Softw. *31*, 363–396.

Holland, J.H. (1962). Outline for a Logical Theory of Adaptive Systems. J. ACM *9*, 297–314.

Hooke, R., and Jeeves, T.A. (1961). `` Direct Search" Solution of Numerical and Statistical Problems. J. ACM *8*, 212–229.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI--a COmplex PAthway SImulator. Bioinformatics *22*, 3067–3074.

Hübner, K., Sahle, S., and Kummer, U. (2011). Applications and trends in systems biology in biochemistry. FEBS J. *278*, 2767–2857.

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

Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., et al. (2007). Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations. Science (80-.). *316*, 593–597.

Jamshidi, N., and Palsson, B.Ø. (2008). Formulating genome-scale kinetic models in the post-genome era. Mol. Syst. Biol. *4*, 171.

Jarnuczak, A.F., Albornoz, M.G., Eyers, C.E., Grant, C.M., and Hubbard, S.J. (2018). A quantitative and temporal map of proteostasis during heat shock in *Saccharomyces cerevisiae*. Mol. Omi. *14*, 37–52.

Kacser, H., and Burns, J.A. (1973). The control of flux. Symp. Soc. Exp. Biol. *27*, 65–104.

Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (2011). ConsensusPathDB: toward a more complete picture of cell biology. Nucleic Acids Res. *39*, D712–D717.

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. *22*, 437–467.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, (IEEE), pp. 1942–1948.

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

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

King, E.L., and Altman, C. (1956). A Schematic Method of Deriving the Rate Laws for Enzyme-Catalyzed Reactions. J. Phys. Chem. *60*, 1375–1378.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. Science (80-.). *220*, 671–680.

Kitano, H. (2002). Systems biology: a brief overview. Science *295*, 1662–1664.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009a). Parameter Estimation. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 152–164.

Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. (2009b). Mathematics. In Systems Biology, (Federal Republic of Germany: Deustsche Nationalbibliothek), pp. 449–473.

Koch, I., Junker, B.H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics *21*, 1219–1226.

Krebs, O., Golebiewski, M., Kania, R., and Mir, S. (2007). SABIO-RK: a data warehouse for biochemical reactions and their kinetics. J. Integr. Bioinform.

Kronfeld, M., Planatscher, H., and Zell, A. (2010). The EvA2 optimization framework. Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) *6073 LNCS*, 247–250.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least. Q. Appl. Math. *2*, 164–168.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci. *101*, 4781–4786.

Liebermeister, W., and Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor. Biol. Med. Model. *3*, 41.

Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010). Modular rate laws for enzymatic reactions: thermodynamics, elasticities and implementation. Bioinformatics *26*, 1528–1534.

Link, H., Christodoulou, D., and Sauer, U. (2014). Advancing metabolic models with kinetic information. Curr. Opin. Biotechnol. *29*, 8–14.

Machado, D., Costa, R.S., Rocha, M., Ferreira, E.C., Tidor, B., and Rocha, I. (2011). Modeling formalisms in Systems Biology. AMB Express *1*, 45.

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: A native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics *22*, 1406–1407.

Matsuda, F., Kinoshita, S., Nishino, S., Tomita, A., and Shimizu, H. (2017). Targeted proteome analysis of single-gene deletion strains of Saccharomyces cerevisiae lacking enzymes in the central carbon metabolism. PLoS One *12*, e0172742.

Mensonides, F.I.C., Schuurmans, J.M., Teixeira de Mattos, M.J., Hellingwerf, K.J., and Brul, S. (2002). The metabolic response of Saccharomyces cerevisiae to continuous heat stress. Mol. Biol. Reports JT - Mol. Biol. Reports *29*, 103–6 PL–Netherlands PT–Journal Article.

Nelder, J.A., and Mead, R. (1965). A Simplex Method for Function Minimization. Comput. J. *7*, 308–313.

Nocedal, J., and Wright, S. (1999). Numerical Optimization (New York: Springer).

Novere, N. Le, Hucka, M., and Mi, H. (2009). The Systems Biology Graphical Notation. Nat. Biotechnol. *27*, 735–742.

Le Novère, N., Hucka, M., Hoops, S., Keating, S., Sahle, S., and Wilkinson, D. (2008). Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Nat. Preced. 1–38.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *27*, 29–34.

Orth, J.D., Conrad, T.M., Na, J., Lerman, J. a, Nam, H., Feist, A.M., and Palsson, B.Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011. Mol. Syst. Biol. *7*, 535.

Paiva, C.L.A., and Panek, A.D. (1996). Biotechnological Applications of the Disaccharide Trehalose. pp. 293–314.

Park, T.-Y., and Froment, G.F. (1998). A hybrid genetic algorithm for the estimation of parameters in detailed kinetic models. Comput. Chem. Eng. *22*, S103–S110.

Parrou, J.L., Teste, M.A., and François, J. (1997). Effects of various types of stress on the metabolism of reserve carbohydrates in Saccharomyces cerevisiae: Genetic evidence for a stress-induced recycling of glycogen and trehalose. Microbiology *143*, 1891–1900.

Petri, C.A. (1962). Kommunikation mit Automaten (Communication with Automata). University of Bonn.

Postmus, J., Canelas, A.B., Bouwman, J., Bakker, B.M., Van Gulik, W., De Mattos, M.J.T., Brul, S., and Smits, G.J. (2008). Quantitative analysis of the high temperature-induced glycolytic flux increase in Saccharomyces cerevisiae reveals dominant metabolic regulation. J. Biol. Chem. *283*,

23524–23532.

Puig-Castellví, F., Alfonso, I., Piña, B., and Tauler, R. (2015). A quantitative 1H NMR approach for evaluating the metabolic response of Saccharomyces cerevisiae to mild heat stress. Metabolomics *11*, 1612–1625.

Rodriguez, N., Donizelli, M., and Le Novère, N. (2007). SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). BMC Bioinformatics *8*, 79.

Roser, B. (1991). Trehalose, a new approach to premium dried foods. Trends Food Sci. Technol. *2*, 166–169.

Savageau, M.A. (1976). Biochemical Systems Analysis (Reading, MA: Addison-Wesley Pub. Co.).

Schaff, J., Slepchenko, B., and Loew, L. (2000). Physiological modeling with virtual cell framework. Methods Enzymol. *321*, 1–23.

Schiraldi, C., Di Lernia, I., and De Rosa, M. (2002). Trehalose production: Exploiting novel approaches. Trends Biotechnol. *20*, 420–425.

Schmidt, T., Samaras, P., Frejno, M., Gessulat, S., Barnert, M., Kienegger, H., Krcmar, H., Schlegl, J., Ehrlich, H.-C., Aiche, S., et al. (2018). ProteomicsDB. Nucleic Acids Res. *46*, D1271–D1281.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. *32*, D431-3.

Smallbone, K., Simeonidis, E., Swainston, N., and Mendes, P. (2010). Towards a genome-scale kinetic model of cellular metabolism. BMC Syst. Biol. *4*, 6.

Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., et al. (2013). A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS Lett. *587*, 2832–2841.

Strassburg, K., Walther, D., Takahashi, H., Kanaya, S., and Kopka, J. (2010). Dynamic transcriptional and metabolic responses in yeast adapting to temperature stress. OMICS *14*, 249–259.

Sun, J., Garibaldi, J.M., and Hodgman, C. (2012). Parameter estimation using meta-heuristics in systems biology: a comprehensive review. IEEE/ACM Trans. Comput. Biol. Bioinform. *9*, 185–202.

Varma, A., and Palsson, B.O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. Bio/Technology *12*, 994–998.

Viant, M.R., Kurland, I.J., Jones, M.R., and Dunn, W.B. (2017). How close are we to complete

annotation of metabolomes? Curr. Opin. Chem. Biol. *36*, 64–69.

Visser, D., and Heijnen, J.J. (2003). Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab. Eng. *5*, 164–176.

Vizcaíno, J.A., Csordas, A., Del-Toro, N., Dianes, J.A., Griss, J., Lavidas, I., Mayer, G., Perez-Riverol, Y., Reisinger, F., Ternent, T., et al. (2016). 2016 update of the PRIDE database and its related tools. Nucleic Acids Res. *44*, D447–D456.

Voit, E.O. (2003). Biochemical and genomic regulation of the trehalose cycle in yeast: Review of observations and canonical model analysis. J. Theor. Biol. *223*, 55–78.

Wang, M., Herrmann, C.J., Simonovic, M., Szklarczyk, D., and von Mering, C. (2015). Version 4.0 of PaxDb: Protein abundance data, integrated across model organisms, tissues, and cell-lines. Proteomics *15*, 3163–3168.

Wolkenhauer, O., and Mesarović, M. (2005). Feedback dynamics and cell function: Why systems biology is called Systems Biology. Mol. Biosyst. *1*, 14.

Yen, J., Liao, J.C., Lee, B.L.B., and Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. Syst. Man Cybern. Part B Cybern. a Publ. IEEE Syst. Man Cybern. Soc. *28*, 173–191.

Yurkovich, J.T., and Palsson, B.O. (2018). Quantitative -omic data empowers bottom-up systems biology. Curr. Opin. Biotechnol. *51*, 130–136.

Zhang, Z., Shen, T., Rui, B., Zhou, W., Zhou, X., Shang, C., Xin, C., Liu, X., Li, G., Jiang, J., et al. (2015). CeCaFDB: a curated database for the documentation, visualization and comparative analysis of central carbon metabolic flux distributions explored by 13C-fluxomics. Nucleic Acids Res. *43*, D549–D557.

Zi, Z. (2011). SBML-PET-MPI: A parallel parameter estimation tool for Systems Biology Markup Language based models. Bioinformatics *27*, 1028–1029.

Zi, Z., and Klipp, E. (2006). SBML-PET: A Systems Biology Markup Language-based parameter estimation tool. Bioinformatics *22*, 2704–2705.

Zuberek, W.M. (1991). Timed Petri nets definitions, properties, and applications. Microelectron. Reliab. *31*, 627–644.

Appendix A: GRaPe 2.0 Source Code

The source code of the software as well as the compiled software is available electronically on the DVD. For the genetic algorithm to run properly, users are required to compile the necessary libraries in the right directory, for users on Linux, the directory would normally be "/usr/local/lib".

It is important that the necessary libraries are needed for the software to run, from the Java libraries to the C libraries (even the shared libraries). For the C shared library, it may be necessary to recompile it. Before the recompilation, it is important to make the changes to configurations.xml and Makefile-Debug.mk within the nbproject folder before compiling, this is to ensure the library know where to the prerequisite libraries are located.

On top of the source code and program, sample Toy Model is available for testing.

Appendix B: Supplementary Data for Chapter 3

	mmol/l	0 minutes	10 minutes	30 minutes	60 minutes	120 minutes	240 minutes
Glucose Transport	E1	0.00206694	0.00184489	0.00179696	0.00147943	0.00145959	0.00142133
Hexokinase	E2	0.01304457	0.01489337	0.01683076	0.01753665	0.02043178	0.02505539
Glucose phosphate isomerase	E3	0.01503592	0.01635902	0.01846131	0.01995669	0.02105563	0.02436216
Trehalose synthase	E4	0.00256135	0.00269138	0.00344178	0.00386471	0.00554248	0.00819933
Phosphofructose kinase	E5	0.01599541	0.01747348	0.01750436	0.02116473	0.02018877	0.02182754
Aldolase	E6	0.01319835	0.01455592	0.01714946	0.01582394	0.01443258	0.01458597
Glycerol phosphate dehydrogenase	E7	0.01753296	0.01861877	0.01774078	0.02029782	0.0243373	0.02638715
Triose phosphate Isomerase	E8	0.00992828	0.01027433	0.01275739	0.01496328	0.0128581	0.01239997
Glyceraldehyde-3-phosphate dehydrogenase	E9	0.119079	0.13343662	0.15598313	0.19034017	0.17734509	0.21143739
Phosphoglycerate kinase	E10	0.02042608	0.02250027	0.02455869	0.02879076	0.02691297	0.02753256
Phosphoglycero mutase	E11	0.00619993	0.00664331	0.00760563	0.00786648	0.00906558	0.01094958
Enolase	E12	0.03434487	0.0416871	0.04894869	0.05174404	0.04726277	0.05471189
Pyruvate Kinase	E13	0.12397888	0.13756031	0.14720304	0.14310765	0.11722709	0.11684261
Pyruvate Decarboxylase	E15	0.03369734	0.03480904	0.03612466	0.04109938	0.04248587	0.04632455
Alcohol Dehydrogenase	E17	0.05953937	0.05647566	0.05488353	0.06172011	0.06153019	0.05305157
Aldehyde Dehydrogenase	E18	0.05235646	0.05356178	0.05589662	0.05557215	0.05980883	0.06398089

Supplementary Table 1: Time course data of glycolytic proteins.

These data are gathered under heat stress condition over 7 different time points, this table uses mmol/l unit for the proteins measured.

	cpc	0 minutes	10 minutes	30 minutes	60 minutes	120 minutes	240 minutes
Glucose Transport	E1	6223.6956	5555.09047	5410.76095	4454.67649	4394.92214	4279.7213
Hexokinase	E2	39278.1199	44844.997	50678.6052	52804.1002	61521.5279	75443.5345
Glucose phosphate isomerase	E3	45274.2252	49258.1543	55588.2926	60090.9874	63399.9947	73356.1842
Trehalose synthase	E4	7712.39513	8103.93096	10363.4294	11636.9255	16688.8036	24688.762
Phosphofructose kinase	E5	48163.3098	52613.8857	52706.8544	63728.4848	60789.8126	65724.2505
Aldolase	E6	39741.1615	43828.905	51638.2212	47646.9975	43457.5103	43919.3976
Glycerol phosphate dehydrogenase	E7	52792.9901	56062.4353	53418.7263	61118.1556	73281.3128	79453.5635
Triose phosphate Isomerase	E8	29894.7497	30936.7386	38413.393	45055.4956	38716.642	37337.176
Glyceraldehyde-3-phosphate dehydrogenas	E9	358555.251	401787.065	469676.202	573127.646	533998.566	636652.88
Phosphoglycerate kinase	E10	61504.355	67749.9032	73947.9497	86691.0006	81036.8372	82902.4792
Phosphoglycero mutase	E11	104208.704	98386.4461	114551.852	119190.079	107105.729	93286.5719
Enolase	E12	103414.823	125522.788	147387.952	155804.944	142311.53	164741.365
Pyruvate Kinase	E13	373309.127	414203.767	443238.734	430907.199	352979.034	351821.329
Pyruvate Decarboxylase	E15	101465.052	104812.473	108773.904	123753.137	127927.935	139486.468
Alcohol Dehydrogenase	E17	179277.246	170052.204	165258.169	185843.592	185271.727	159742.016
Aldehyde Dehydrogenase	E18	157648.994	161278.303	168308.659	167331.645	180088.588	192650.981

Supplementary Table 2: Time course data of glycolytic proteins.

These data are gathered under heat stress condition over 7 different time points, this table uses counts per cell (cpc) unit for the proteins measured.

The resulting models of this chapter are available electronically on the CD.

Appendix C: Supplementary Data for Chapter 4

Mmol/l	30°C	37°C
GLT	0.002067	0.001421
HXK	0.013045	0.025055
PGI	0.015036	0.024362
PFK	0.015995	0.021828
ALD	0.013198	0.014586
TPI	0.009928	0.0124
G6PDH	0.004896	0.006393
Glycerol	0.017533	0.026387
PYK	0.123979	0.116843
ADH	0.063677	0.061078
PYC	0.015471	0.012254
PGM	0.0062	0.01095
UDPGP	0.002838	0.006523
GSY	0.002264	0.004246
GPH	0.003608	0.00636
T6PS	0.002561	0.008199
T6PP	0.001947	0.004378
Trehalase	0.002518	0.004924
ATP Synthesis	1	1
UDP conversion to UTP	1	1

Supplementary Table 3: Data for protein involved in Trehalose Metabolism.

They are gathered under two different conditions, 30°C and 37°C, recorded as mmol/l.

The resulting model of this chapter are available electronically on the CD.