Algorithms for Improving the Predictive Power of Flux Balance Analysis

Dissertation
zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Heinrich-Heine Universitaet Duesseldorf

von
Abdelmoneim Mahmoud Amer Desouki
M.Sc., Computer Science
geboren am 09. 10. 1975
in Beheira, Egypt

Heinrich-Heine University
2016

Referent:	Prof. Dr. Martin Lecher
Korreferent:	Prof. Dr. Oliver Ebenhöh
Tag der mündlichen Prüfung	16.Juni.2016

To my Parents, my wife, Mariam , Omar and Ahmed.

Contents

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABBREVIATIONS

ATM-FBA	Automatic Thresholding to Minimize deviation between Flux State and Gene Expression State
CBM	Constraint-based Modeling
ccFBA	Cost-Constrained Flux-Balance-Analysis
COBRA	COnstraint-Based Reconstruction and Analysis
EFM	Elementary Flux Modes
FBA	Flux Balance Analysis
FBAwMC	FBA with molecular crowding
FCA	Flux Coupling Analysis
FECorr	Flux Expression Correlation
FVA	Flux Variability Analysis
GIMME	Gene Inactivity Moderated by Metabolism and Expression
GPR	Gene-Protein-Reaction
GSM	genome-scale metabolic network
LMOMA	Linear Minimization of Metabolic Adjustment
LP	Linear Problem
MADE	Metabolic Adjustment by Differential Expression
MILP	Mixed integer linear problem
MOMA	Minimization of Metabolic Adjustment
MOMENT	MetabOlic Modeling with ENzyme kineTics
MTC	Minimum Total Cost
MTF	Minimum Total Flux
pFBA	Parsimonious Flux Balance Analysis
QP	Quadratic problem
RBA	Resource Balance Analysis
SBML	Systems Biology Markup Language
SyBil	Systems Biology library
TCA	Tricarboxylic Acid

ACKNOWLEDGMENTS

"الحمد لله رب العالمين"

First and foremost I want to thank Allah for enlightening my way.

I would like to thank my family for everything they have been doing for me, specially my great parents and my great wife.

I am very grateful and thankful to Professor Dr. Martin Lercher for his constructive supervision, his experienced guidance and unfailing advice.

I wish to express my special thanks and gratitude to Dr. Gabriel Gelius-Deitrich for his sincere support and constant encouragement. We had long productive discussions and he helped me a lot to solve technical problems.

I would like to thank Professor Dr. Amin Shoukry in Computer Science department, Faculty of Engineering, Alexandria university, Egypt for his continuous support and guidance.

I would like to thank Professor Dr. Hamed Nassar the previous dean of the faculty of Computers in Ismailia and all the staff for their support and encouragement, special thanks to Dr. Ahmad Magdi Ghanim for his help in introducing me to the Bioinformatics. Thanks to Dr. Hassan Elmahdi and Dr Mostafa Haragy.

I would like to thank DAAD (Deutscher Akademischer Austauschdienst) for their financial support and giving me the opportunity to conduct my research in Germany. They proved that their role is much more than just giving money.

I wish to thank the staff of the Bioinformatics group Heinrich-Heine university, Dusseldorf, Germany as well as my colleagues for their support during my studies. I would like to mention here Anja, Christian, Thomas, Bastian, David, Ulrich, Jonathan, Janina, Daniel, and Deya.

I want to thank everyone who contributed, in some way, in this work.

Finally, thanks to my friends who kept encouraging and offering me help, special thanks to Mohamed Tahoun and Tarek Gaber.

Dusseldorf, 2016
Abdelmoneim Desouki

ABSTRACT

Constraint-based metabolic modeling methods such as Flux Balance Analysis (FBA) are routinely used to predict metabolic phenotypes, e.g., growth rates, ATP yield, or the fitness of gene knockouts. While powerful, FBA has some important limitations. For example, FBA solutions are not unique and can contain thermodynamically infeasible cycles. FBA ignores gene regulation, and thus FBA solutions may not be compatible with experimentally determined gene expression states. Crucially, FBA ignores important biological constraints, such as limits imposed by the finite cell volume on protein counts, a phenomenon often termed molecular crowding. In this thesis, I introduce three different ways to improve Flux Balance Analysis predictions.

The first improvement eliminates thermodynamically infeasible cycles. It is based on a fast postprocessing step for constraint-based solutions. The algorithm, termed *CycleFreeFlux*, removes internal cycles from any given flux distribution $\boldsymbol{v}^{(0)}$ without disturbing other fluxes not involved in the cycles. It works by minimizing the sum of absolute fluxes $||\boldsymbol{v}||_1$ while (i) conserving the exchange fluxes and (ii) using the fluxes of the original solution to bound the new flux distribution. This strategy reduces internal fluxes until at least one reaction of every possible internal cycle is inactive, a necessary and sufficient condition for the thermodynamic feasibility of a flux distribution. If alternative representations of the input flux distribution in terms of elementary flux modes exist that differ in their inclusion of internal cycles, then *CycleFreeFlux* is biased towards solutions that maintain the direction given by $\boldsymbol{v}^{(0)}$ and towards solutions with lower total flux $||\boldsymbol{v}||_1$. My method requires only one additional linear optimization, making it computationally very efficient compared to alternative strategies.

The second improvement of FBA, termed ccFBA (for capacity-constrained flux balance analysis), provides a framework to convert any complete FBA model into a model for metabolic modeling with enzyme kinetics (MOMENT) that accounts for molecular crowding. I provide an improved implementation of a molecular crowding model for *E. coli* and the first such implementation for the yeast *Saccharomyces cerevisiae*. ccFBA is an extension to *sybil*, a library for efficient constraint-based modeling in the R environment for statistical computing. ccFBA improves the original implementation of MOMENT by partitioning multifunctional enzymes between the different reactions that they catalyze. Although the improved *E. coli* implementation includes kinetic constants for 117 additional reactions, predicted *E. coli* growth rates across different carbon sources still show much less variation than observed experimentally; this discrepancy is likely due to the condition-dependent expression of proteins in preparation for environmental changes, an important but as yet poorly understood element of microbial metabolism.

Finally, I introduce three novel methods that use transcriptomic and/or proteomic data to predict metabolic fluxes on a genome scale. The first of these methods is called FECorr. FECorr fits piecewise linear functions to the experimentally observed relationship between gene expression and possible flux ranges determined from simulations. To do this, it utilizes gene expression from multiple experiments. The flux distributions predicted from these functions show better agreement with measured metabolic fluxes than all other gene

expression methods compared in a previous benchmark study. The second method I introduce in the final part of the thesis is called ATM-FBA. It automatically identifies optimal thresholds to distinguish active from non-active genes and reactions. ATM-FBA also performs slightly better than previously published gene expression methods. The third new method is termed eFBA-gene. Similar to other methods, it uses a constant threshold as input and formulates a mixed-integer linear programming (MILP) problem with the objective to minimize the discrepancy between expression data and predicted flux distributions. However, in contrast to other expression-based methods, eFBA-gene scores the agreement between simulated fluxes and expression data not on a per-reaction basis, but on a per-gene basis. For some combinations of gene expression threshold and flux threshold, eFBA-gene also outperforms other gene expression methods.

ZUSAMMENFASSUNG

Beschränkungs-basierte Methoden zu metabolischen Modellierung (insbesondere die Flux-Balance-Analyse, FBA) werden routinemäßig für die Vorhersage metabolischer Phänotypen eingesetzt, etwa zur Berechnung von Wachstumsraten, der ATP-Ausbeute oder der Fitness von Gen-Knockouts. Trotz der Leistungsfähigkeit der FBA hat diese Methode einige wichtige Anwendungsgrenzen. FBA-Lösungen sind nicht eindeutig bestimmt und können thermodynamisch unmögliche interne Zyklen beinhalten. FBA ignoriert die Regulation von Genen, weshalb FBA-Lösungen eventuell experimentell bestimmten Genexpressions-Zuständen widersprechen. Entscheidend für die Limitierungen der FBA ist die Tatsache, dass diese Methode wichtige biologische Beschränkungen ignoriert. Insbesondere beschränkt das begrenzte Volumen der Zelle die Menge an Proteinen, die gleichzeitig exprimiert werden können, ein Phänomen, das häufig als molekulare Verdrängung (molecular crowding) bezeichnet wird. In dieser Dissertation führe ich drei verschiedene Arten ein, mit Hilfe derer die Flux-Balance-Analyse verbessert werden kann.

Die erste Verbesserung eliminiert thermodynamisch unmögliche Zyklen. Sie basiert auf einem schnellen Nachverarbeitungsschritt für beschränkungs-basierte Lösungen. Der Algorithmus CycleFreeFlux entfernt interne Zyklen von einer beliebigen Flussverteilung v(0), ohne diejenigen Flüsse zu verändern, die nicht an Zyklen beteiligt sind. Das Verfahren arbeitet über die Minimierung der Summe der Flussbeträge $||v||_1$. Sie benötigt lediglich einen linearen Optimierungsschritt und ist damit im Vergleich zu alternativen Strategien ausgesprochen effizient in Bezug auf seine Laufzeit.

Die zweite Verbesserung der FBA, *ccFBA* (für capacity-constrained FBA, Kapazitäts-beschränkte FBA), stellt einen Rahmen für die Konvertierung eines beliebigen vollständigen FBA-Modells in ein Model für metabolische Modellierung unter Berücksichtigung der Enzymkinetik (MOMENT) zur Verfügung, welches die Beschränkung durch molekulare Verdrängung berücksichtigt. Ich stelle eine verbesserte Implementierung für das Bakterium *E. coli* sowie die erste Implementierung überhaupt für die Hefe *Saccharomyces cerevisiae* zur Verfügung. *ccFBA* ist eine Erweiterung von *sybil*, einer Bibliothek für effiziente beschränkungs-basierte Modellierung in der Umgebung für statistische Berechnungen *R*. *ccFBA* verbessert die ursprüngliche Version von MOMENT, indem sie multifunktionale Enzyme zwischen den verschiedenen von ihnen katalysierten Reaktionen aufteilt. Obwohl die verbesserte Implementierung für *E. coli* kinetische Konstanten für 117 zusätzliche Enzyme enthält, zeigen die vorhergesagten Wachstumsraten auf verschiedenen Kohlenstoffquellen dennoch eine sehr viel geringere Variation als die entsprechenden experimentellen Werte; diese Diskrepanz ist vermutlich auf die Umgebungs-abhängige Expression von Proteinen in Vorbereitung auf Umgebungsveränderungen zurück zu führen, ein wichtiges, aber noch weitgehend unverstandenes Phänomen des mikrobiellen Metabolismus.

Schließlich führe ich noch drei neuartige Methoden ein, die Transkriptom- oder Proteomdaten verwenden, um metabolische Flüsse auf Genomskala vorherzusagen. Die erste dieser Methoden bezeichne ich als *FECorr*. *FECorr* fitted stückweise lineare Funktionen an die Beziehung zwischen experimentell beobachteter Genexpression und den aus

Simulationen erhaltenen möglichen Wertebereich des Flusses der entsprechenden Reaktion. Dafür nutzt es Genexpressionsdaten aus verschiedenen Experimenten simultan. Die Flussverteilungen, die so vorhergesagt werden, zeigen eine bessere Übereinstimmung mit gemessenen Flusswerten als alle anderen Methoden, die in einer kürzlich publizierten Vergleichsstudie untersucht wurden. Die zweite von mir eingeführte Methode in diesem Teil der Dissertation wird als *ATM-FBA* bezeichnet. Sie identifiziert automatisch die optimalen Grenzwerte, um aktive von nicht-aktiven Genen und Reaktionen zu unterscheiden. *ATM-FBA* macht ebenfalls etwas bessere Vorhersagen als zuvor publizierte Methoden, die Genexpressiondaten verwenden. Die dritte neue Methode ist *eFBA-gene*. Ähnlich wie andere Methoden benutzt diese einen konstanten Grenzwert als Eingabe und formuliert ein Mixed-Integer lineares Problem (MILP) mit dem Ziel, die Diskrepanz zwischen Expressionsdaten und vorhergesagten Flüssen zu minimieren. Im Gegensatz zu anderen Methoden bewertet *eFBA-gene* jedoch die Übereinstimmung zwischen simulierten Flüssen und Genexpressiondaten nicht pro Reaktion sondern pro Gen. Für bestimmte Kombinationen aus Genexpressions- und Fluss-Grenzwert macht auch *eFBA-Gen* bessere Vorhersagen als vergleichbare Methoden.

PUBLICATIONS

Gelius-Dietrich G, **Desouki AA**, Fritzemeier CJ, Lercher MJ: sybil - Efficient constraint-based modelling in R. BMC Systems Biology 2013, 7:125.

Desouki AA., Jarre F, Gelius-Dietrich G, Lercher MJ: CycleFreeFlux: Efficient removal of thermodynamically infeasible loops from flux distributions. Bioinformatics 2015, *31*(13), pp.2159-2165..

Desouki AA., Gelius-Dietrich G, Lercher MJ: FECorr: An algorithm to improve FBA predictions using transcriptomic data. Metabolic Pathway Analysis (MPA 2015); Braga, Portugal.

Desouki AA., Gelius-Dietrich G, Lercher MJ:. ATM-FBA: Automatic Thresholding to Minimize deviation between Flux State and Gene Expression State. 4th Conference on Constraint-Based Reconstruction and Analysis (COBRA 2015); Heidelberg, Germany.

Chapter 1: Introduction

1.1 Metabolism

Metabolism as studied in this thesis is a set of chemical processes that break down sugars and other nutrients to synthesize energy and the main building blocks (metabolic precursors) of a cell. There are 11 precursors used to build the basic building blocks of the cell [1], namely pyruvate, α-ketoglutarate, oxaloacetate, ribose-5-phosphate, acetyl-CoA, erythrose-4-phosphate, fructose-6-phosphate, glucose-6-phosphate, glyceraldehyde-3-phosphate, phosphoenolpyruvate, and 3-phosphoglycerate. An overview of basic metabolism is shown in Figure 1. Sugar is transported into the cell, where it is first phosphorylated and then converted to hexose monophosphate.The hexose monophosphate is either converted to Pyruvate or used to synthesize carbohydrates (like glycogen). Pyruvate in turn is either oxidized in TCA (Tricarboxylic Acid) cycle to form carbon dioxide or converted to bypoducts via fermentative pathway. Some intermediates in both pathways (glycolysis and TCA) are used as building blocks for macromolecules, that in turn are assembled into different cell structures [1].

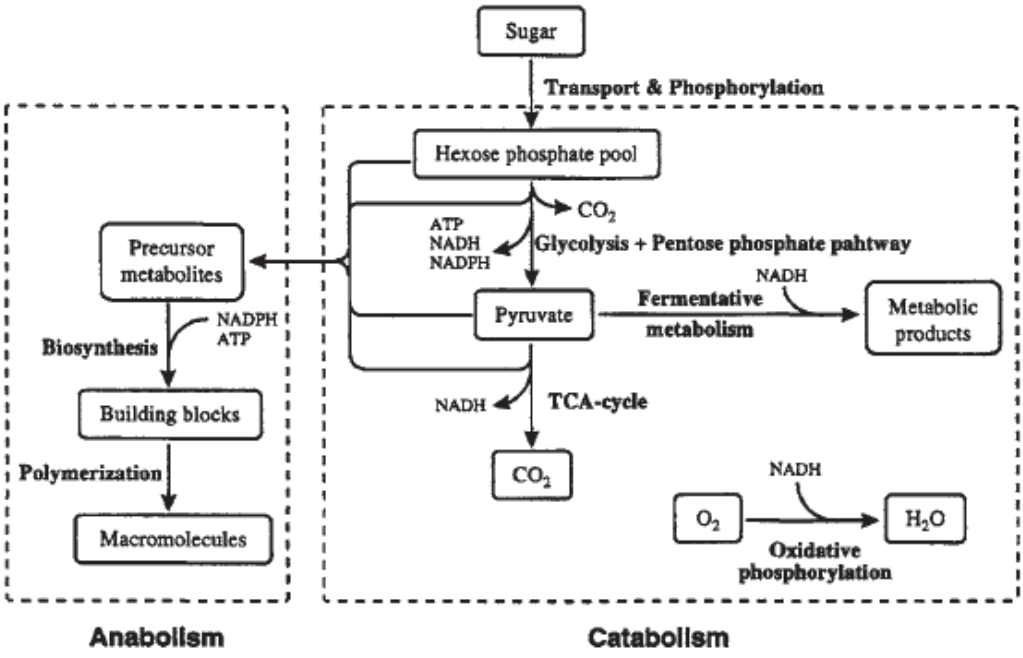

Figure 1 Overall structure of cell synthesis from sugars, reproduced from [1]

Metabolism can be seen as two processes, catabolism and anabolism. In catabolism, sugars are broken down into precursor metabolites. In anabolism macromolecules are built by polymerization of building blocks.

There are some diseases related to metabolic problems, like diabetes, obesity, cancer heart diseases[2]. Some of these metabolic disorders are inherited, while others develop when some organs like liver or pancreas become diseased.

Study of metabolism can be done on four different levels [2]. The first level is the whole cell, in which substrates are considered as inputs, and biomass and by-products are considered as outputs. The second level is the sectors level, in which the metabolic process is seen as two basic sectors, catabolism and anabolism. The third level is pathways, in which pathways and segments of pathways are studied. The fourth level is the reaction level, and this is the finest level of study.

1.2 Metabolic Models

Genome-scale metabolic models (GSM) are the complete set of reactions that a specific cell can perform. They can be constructed using the gene annotation of an organism's genome. The model is then curated using the literature. Sometimes there are some gaps in the model that need to be filled. The GSM represents the current knowledge about a given organism in a structured way. Currently there are models for more than 100 organisms including humans [3]. The standard data format for representing metabolic models is SBML (Systems Biology Markup Language) [4]. The model basically consists of a sparse stoichiometric matrix, with each row representing a metabolite, each column representing a reaction, and each entry is the stoichiometric coefficient of the corresponding metabolite in the given reaction.

Thiele and Palsson [5] proposed a protocol for constructing genome-scale metabolic networks. It consists of four stages. In the first stage a draft reconstruction is generated from annotation of the genome of the organism of interest, using biochemical databases like KEGG [6]. Then in the second stage, the reconstruction is refined and curated. In the third stage the reconstruction is converted into a mathematical format and condition-specific models are defined. The fourth stage is dedicated to network verification, evaluation and validation. The third stage can mostly be automated. Among the tools that can be used to reconstruct metabolic models are rBioNet [7] and SimPheny [8]; the first is free but the latter is commercial.

A metabolic model can be seen as a directed, weighted hypergraph, with the nodes being the metabolites and the edges being the reactions. An example of a genome-scale metabolic model is *E. coli's* genome scale metabolic model iAF1260 [9], which contains 2382 reactions , 1668 metabolites and 1260 genes.

There are online databases for reconstructed models such as the BiGG database (http://bigg.ucsd.edu/) and the SEED database (http://www.theseed.org/wiki/Main_Page). Other websites that can be used in model curation and reconstruction are KEGG[6], Metacyc [10], BRENDA [11], and MetaNetX[12].

1.3 Flux Balance Analysis(FBA)

Flux balance analysis is a mathematical approach for analyzing the flow of metabolites through a metabolic network [13]. It is a direct application of linear programming to biological systems that uses the stoichiometric coefficients for each reaction in the system to construct constraints for the optimization. Additionally, this method requires the assumption of a biological steady state, or homeostasis. Imposing this restriction allows the assumption that at any given time, the concentration of a given compound in the metabolic network is constant (i.e. no net production or consumption of any metabolite). Thus, there is no need to measure concentrations. Then, depending on what is being studied, a specific phenotype (such as the growth rate of a bacterium) will be selected and a relevant biological parameter (i.e., the objective function) will be minimized or maximized[14]. In case of unicellular organisms, this can be done usually by incorporating an artificial reaction to simulate Biomass production. This theoretical Biomass reaction mainly contains amino acids, ATP, and nucleotides with proportions required for growth and multiplication[15-18]. In some applications of FBA, the objective will be to maximize the production of a metabolite or a set of metabolites of interest. At known limits on the uptake rate of nutrients, the prediction of growth rates then becomes a linear optimization problem, which can be solved efficiently using linear programming.

Since FBA is mathematically simple and requires no kinetic information about the reactions of the metabolic network, it is well suited for genome-scale metabolic networks.

There are many applications of FBA. It can predict growth rates by adding an artificial reaction to represent biomass production. This reaction can be scaled so that the flux through it is equal to the exponential growth rate(μ)[13]. It also can predict the yield of cofactors like ATP. By restricting the flux through specific reactions to zero, FBA can efficiently predict the effect of gene knockout. Recently, FBA has also been used to find drug targets[19-22], while other investigators used FBA to study the evolution of metabolic systems [23-25]

Mathematically, FBA is the following linear programming (LP) problem:

$$Maximize\ \boldsymbol{c}^T\boldsymbol{v}$$

$$\text{s.t.} \quad S\boldsymbol{v} = 0$$

$$\text{where}\ \boldsymbol{lb} \leq \boldsymbol{v} \leq \boldsymbol{ub}$$

Here, *c* is a vector of constant weights; *S* is the stoichiometric matrix; *v* is a vector of fluxes; and *lb* and *ub* are vectors of fixed lower and upper bounds, respectively; the inequalities must be respected element-wise.

Popular FBA extensions include MTF, MOMA and FVA.

MTF: FBA solutions are not unique and there are frequently multiple optima giving the same objective value. One way to choose from these solutions is the minimization of total flux, or MTF. In MTF, it is postulated that given the available external substrates and given a set of functionally important target fluxes required to accomplish a specific pattern of cellular functions, the sum of the stationary metabolic fluxes has to become a minimum[17, 26]. This can be seen as minimizing the cost of protein synthesis that catalyzes fluxes by assuming that all proteins have the same cost and catalyze reactions at the same rates. MTF adds a second LP to FBA. One important property of MTF is that the resulting flux distribution is always thermodynamically feasible. The MTF strategy is frequently also called parsimonious FBA (pFBA).

FVA: FVA (flux variability analysis) is used to determine the range of possible values of each reaction from the multiple optima. The approach begins with determining the wildtype value of the objective function. From this solution, the range of variability that can exist in each flux in the network due to alternate optimal solutions can be calculated through a series of LP problems wherein the value of the original objective function is fixed and each reaction in the network is maximized and subsequently minimized to determine the feasible range of flux values for each reaction[27].

The mathematical formulation of this approach is described below:

Case 1:

Max v_i
s:t: $\boldsymbol{Sv} = 0$
$\boldsymbol{c}^T\boldsymbol{v} = Z_{obj}$
$\boldsymbol{lb} \leq \boldsymbol{v} \leq \boldsymbol{ub}$ for i = 1 ... n;

Case 2:

Min v_i
s:t: $\boldsymbol{Sv} = 0$
$\boldsymbol{c}^T\boldsymbol{v} = Z_{obj}$
$\boldsymbol{lb} \leq \boldsymbol{v} \leq \boldsymbol{ub}$ for i = 1 ... n;

Here, Z_{obj} is the value of the objective function in the previously solved wildtype FBA problem.

MOMA: many extensions of FBA exist for the prediction of gene knockouts. Minimization of metabolic adjustment (MOMA) is a flux-based analysis, whereby the hypothesis that knockout metabolic fluxes undergo a minimal redistribution with respect to the flux configuration of the wild type is tested. MOMA employs quadratic programming to identify a point in flux space that is closest to the wild-type point in terms of its Euclidean distance and is compatible with the gene deletion constraint.

Comparing MOMA and FBA predictions to experimental flux data for E. coli pyruvate kinase mutant PB25, MOMA was found to display a significantly higher correlation than FBA[28]. A problem with the initial formation and application of MOMA is the redundancy of FBA solutions. The distance of the MOMA to the FBA solution may depend strongly on the chosen FBA solution.

There are many tools that perform FBA calculations. The one most used is COBRA[29] and COBRAPy [30].

COBRA: The program for constraint-based reconstruction and analysis (COBRA) is a free MATLAB toolbox. COBRA allows quantitative predictions of cellular behavior using constraint-based approaches. Specifically, this software allows predictive computations of both steady-state and dynamic optimal growth behavior, the effects of gene deletions, comprehensive robustness analyses, sampling the range of possible cellular metabolic states and the determination of network modules[30].

sybil: is a free R package that performs FBA, MTF, MOMA, FVA, and gene knockout predictions. sybil is computationally more efficient than COBRA and doesn't require any commercial software[31]. Sybil is introduced in Detail below.

1.4 Elementary Flux Modes (EFM)

Elementary flux mode analysis finds minimal functioning units in a metabolic network [32]. A minimal set of reactions that can carry steady-state nonzero flux alone in the network is called an elementary flux mode. There are three types of elementary flux modes. Type 1 starts from an exchange reaction and ends with another. Type 3 represents internal cycles, it contains only internal reactions. Type 2 contains currency metabolites driving the flux.

1.5 Problems in FBA

While powerful, FBA has the following seven problems or limitations.

1. The solution returned by FBA is not unique (existence of multiple optima), that is, there are multiple flux distributions giving the same objective value. This is because in large-scale models the number of variables (reactions) is much more than the number of equations (metabolites) (i.e. the problem is underdetermined)[33].

2. FBA is limited in that it does not take into account the gene regulatory state, as described for example by gene expression data. In effect, the basic approach predicts metabolic capabilities assuming all reactions have the same maximum capacity. Indeed, many of the errors in the prediction of gene knockout phenotype were traced back to the lack of gene regulation in standard FBA models [20].

3. Third, one of the difficulties of FBA predictions is that they frequently include thermodynamically infeasible cycles, i.e., sets of reactions that together carry a flux that has no influence on the exchange reactions of the model [34-38]. These are metabolic "perpetuum mobiles" and do not occur in biological reality. Such internal cycles thus distort predicted flux distributions and should be removed from the predictions. Thermodynamically infeasible cycles affect not only the predictions of FBA, but also those of many other constraint-based analysis methods. In particular, thermodynamically infeasible cycles distort the output of sampling algorithms that aim to characterize the steady-state solution space, and often lead to the assignment of unrealistically high values to the flux ranges that can be carried by individual reactions in flux variability analysis (FVA) [27].

4. FBA ignores important biological constraints, such as the constraint on cell volume and corresponding maximal protein budget. The enzymes needed to catalyze biochemical fluxes need to be produced using cellular resources, and they need to fit into the limited volume of the cell. The cost of protein synthesis consumes much of the energy in the cell and expression of unnecessary protein can limit the growth of microbes [39-41] . When a certain amino acid is available in the environment, the cell is likely to invest in making transporters instead of making enzymes to synthesize that amino acid. 20%–30% of the Escherichia coli cytoplasm is occupied by macromolecules, many of them enzymes, whose cytoplasmic concentration cannot be further increased without drastically affecting protein folding, protein–protein association rates, biochemical reaction kinetics, and transport dynamics within a cell [42].

5. The FBA method requires the definition of suitable upper bounds $lb_i=\beta_i$ on some fluxes in order to obtain a bounded growth rate[43]. In essence it maximizes yield [44].

6. The biological objective is not always possible to find and it can be context-sensitive. Also. it is more difficult in multicellular organisms to define such a cellular objective.

7. Standard FBA was not able to explain some important phenomena related to metabolism like the Crabtree effect [45], Warburg effect[46], overflow metabolism[47], the order of different carbon source consumption, and cross-feeding[48].

The chapters of this thesis address several of these problems and provide implementations of algorithms that at least partially overcome them.

1.6 Sybil

The algorithms developed in this thesis are implemented as extensions of the Sybil package for R [49]. To facilitate the description of these new implementations, it is necessary to first introduce the main functionalities of Sybil.

Sybil is a software tool for flux balance analysis (FBA) and related methods [31]. It is a free R package available from CRAN [50]. It contains different functions to perform several different types of constraint-based analyses. It is more efficient in terms of running time when compared to the most widely used COBRA toolbox. Also it is not dependent on commercial software like COBRA, which is implemented in MATLAB.

Sybil needs to connect to mathematical solvers to solve different types of linear optimization problems. This is done via the R packages glpkAPI[51], cplexAPI[52], clpAPI[53], lpSolveAPI[54], and sybilGUROBI, which implement interfaces to the solvers GLPK[55], IBM ILOG CPLEX[56], COIN-OR Clp[57], lp_Solve[58], and Gurobi[59], respectively. GLPK is a free solver, but it is not able to solve quadratic programming (QP) problems, and its performance in solving mixed-integer linear programming (MILP) problems is much slower than that of CPLEX. Sybil represents the problem in a class *optObj*, while the solution is returned usually in a class *optsol*. The code in these packages is written in C and R. Currently, Sybil forms problems that require LP, MILP, and QP.

New solvers can be easily interfaced to Sybil with the same methodology. The solver can be chosen by setting the parameter *solver* to the name of the package. The default solver can be known by calling the Sybil function SYBIL_SETTINGS("SOLVER").

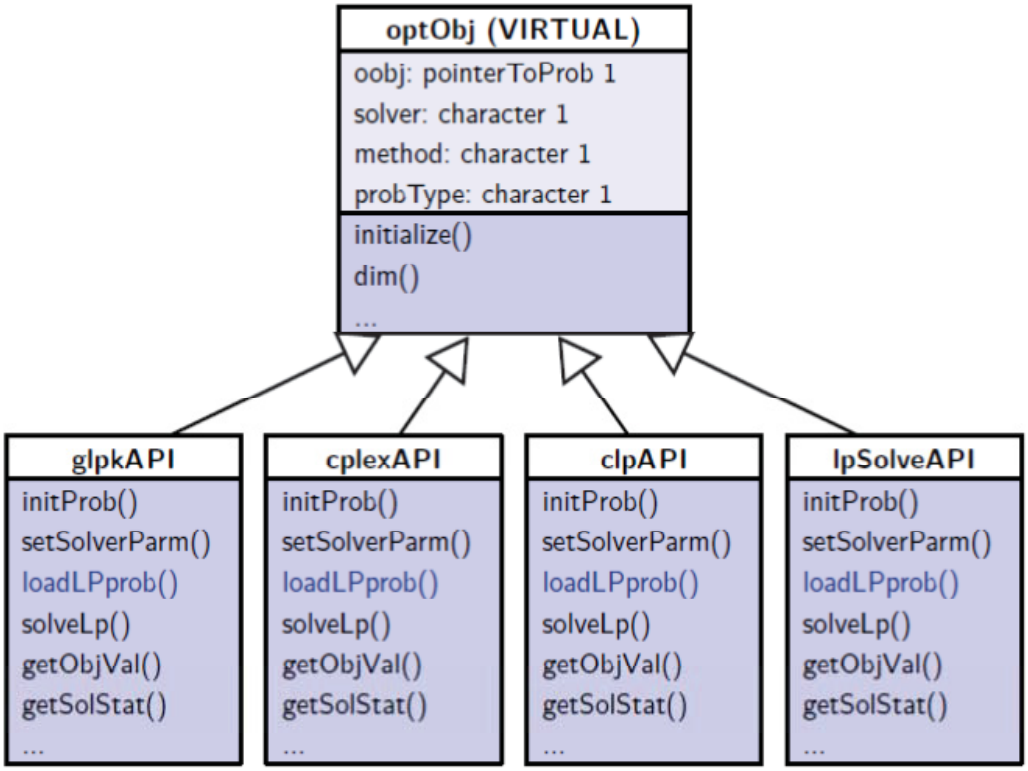

Figure 2 Sybil interface to different solvers

1.6.1 Reading metabolic models

Sybil reads metabolic networks (models) in the form of tab delimited (TSV) files using the function `readTSVmod()`. The model is then represented as a *modelorg* class, used for further analysis. `readTSVmod()` takes the path and the prefix of the files as arguments. The file names are reconstructed from the prefix as:

1- description file <prefix>_desc.tsv: containing the description of the model. The main fields are: name, id, number of genes, number of reactions, number of metabolites, and number of nonzeros.

2- metabolite file <prefix>_met.tsv: containing data for metabolites.

The fields are abbreviation, name, and compartment.

3- reaction file <prefix>_react.tsv: containing data for reactions. The main fields are: abbreviation, name, equation, reversible, lowbnd, uppbnd, obj_coef, rule, and subsystem. The only required field is equation. It should be in the form substrate --> product in case of irreversible reactions and in the form substrate <==> product in case of reversible reactions. The only mandatory file is the reaction file. Example:

```
library(sybil)

mp  <- system.file(package = "sybil", "extdata")
mod <- readTSVmod(prefix = "Ec_core", fpath = mp, quoteChar = "\"")
```

Another way to read models is to use the package sybilSBML[31], which uses LibSBML[60] to read models in SBML[4] - the standard format for representing metabolic models. The function used to perform this is `readSBMLmod()`. It also returns a *modelorg* object.

1.6.2 Running FBA

To run FBA, a model should be loaded in the form of a modelorg object. `readTSVmod()` can be used to get the model. Afterwards the `optimizeProb()` function is used to run FBA as follows. The parameter algorithm is set to "fba"; this is also the default value.

```
data(Ec_core)   # reads the E. coli core metabolic model
optimizeProb(Ec_core,algorithm="fba")
```

These two lines result in the following output:

```
solver:                                  glpkAPI
method:                                  simplex
algorithm:                               fba
number of variables:                     95
number of constraints:                   72
return value of solver:                  solution process was
successful
solution status:                         solution is optimal
value of objective function (fba):       0.873922
value of objective function (model):     0.873922
```

The return value is an object of class optsol_optimizeProb; by default it can be returned as a list by setting the parameter retOptSol to FALSE. To access the fluxes, the function *getFluxDist* can be used.

```
sol=optimizeProb(mod,algorithm="fba")
getFluxDist(sol)
```

Displaying the objective value of the solution:

```
lp_obj(sol)
[1] 0.8739215
```

Also, the function *mod_obj* can be used to get the model objective function; it will be the same value as lp_obj in the case of fba.

Checking the solution status:

```
 checkOptSol(sol)
Return code:
 Code    #       meaning
 0       1       solution process was successful
```

```
Solution status:
 Code    #       meaning
 5       1       solution is optimal
```

Displaying the flux through exchange reactions:

```
data(Ec_core)

# find exchange reactions
ex <- findExchReact(Ec_core)

# run fba
opt <- optimizeProb(Ec_core)

# get flux distribution of exchange fluxes
fd <- getFluxDist(opt, ex)

# display input and output (net flux)
getNetFlux(fd)
```

1.6.3 Running FVA

To run FVA, the function `fluxVar()` is used.

```
data(Ec_core)
fv <- fluxVar(Ec_core)
```

To access the fluxes:

```
nc=length(react_id(Ec_core))
#minimizations
fv_min=lp_obj(fv)[1:nc]
#maximizations
fv_max=lp_obj(fv)[(nc+1):(2*nc)]
```

Plotting the solution:

```
plot(fv)
```

Checking the solution status:

```
table(lp_stat(fv))
```

1.6.4 Running MTF

To run parsimonious FBA or minimum total flux (MTF), the function optimizeProb is used with the algorithm parameter set to "mtf".

```
data(Ec_core)
mtf=optimizeProb(Ec_core,algorithm="mtf")
> mtf
```

```
solver:                                    glpkAPI
method:                                    simplex
algorithm:                                 mtf
number of variables:                       285
number of constraints:                     263
return value of solver:                    solution process was
successful
solution status:                           solution is optimal
value of objective function (mtf):         518.422086
value of objective function (model):       0.873922
```

The lp_obj is the sum of absolute fluxes and mod_obj is the model objective function.

To see the fluxes:

```
getFluxDist(mtf)
```

Checking the solution status:

```
checkOptSol(mtf)
Return code:
 Code    #       meaning
 0       1       solution process was successful

Solution status:
 Code    #       meaning
 5       1       solution is optimal
```

As an extension in Sybil, *costcoeffw* and *costcoefbw* are two parameter vectors that can be used to find a flux distribution with minimum cost; the cost for a given flux is calculated as the flux value multiplied with the predefined weight (given separately for forward and backward fluxes). The default values are all ones, corresponding to MTF. The weight parameters also give the possibility to minimize the total flux for only a set of reactions (for example only enzyme catalyzed reactions) by setting cost to 0 for the reactions not optimized.

1.6.5 Running single gene knockouts

The function `oneGeneDel()` can be used to perform multiple *in silico* single gene knockouts. The function performs *n* optimizations, with *n* being the length of the character vector in the argument `geneList (Default: allGenes(model))`.

```
data(Ec_core)

   # compute phenotypes of genetic perturbations via
   # FBA (default)
Ec_ogd <- oneGeneDel(Ec_core)
compute affected fluxes ... OK
calculating 137 optimizations ...
|            :            |            :            | 100 %
|==================================================| :-)
OK
Done.
> Ec_ogd
```

```
solver:                                    glpkAPI
method:                                    simplex
algorithm:                                 fba
number of variables:                       95
number of constraints:                     72
number of problems to solve:               137
number of successful solution processes:   137
```

List all essential genes:

```
> allGenes(Ec_core)[lp_obj(Ec_ogd)<0.000001]
[1] "b0720" "b2779" "b2415" "b2416" "b1779" "b1136" "b2926"
```

Solution status:

```
> table(lp_stat(Ec_ogd))

  4   5
  2 135
getMeanStatus(4,"glpkAPI")
[1] "no feasible solution exists"
```

Sybil contains also a function `lethal()`, which finds the set of essential genes. Further, the function `doubleGeneDel()` can be used to get all pairwise gene knockouts. More generally, the function `geneDeletion()` can be used to simulate knocking out a set of genes of arbitrary size simultaneously.

1.6.6 Running MOMA

To run MOMA [28], we again use the `optimizeProb()` function. The algorithm parameter is set to "moma", while for linear MOMA it can be set to "lmoma". Wildtype flux can be set using the parameter wtflux; otherwise, an arbitrary FBA solution is used as the wildtype flux. The following is an example of applying MOMA:

```
data(Ec_core)
moma=optimizeProb(Ec_core, algorithm = "moma",solver="cplexAPI");
```

To get the fluxes, the function `getFluxDist()` is used.

1.6.7 Adding a reaction

To add a reaction to a given metabolic network function, `addReact()` is used. The main parameters are the model, reaction id, metabolites, and Scoef, the stoichiometric coefficients of the reaction. This function can also be used to change parameters of a reaction already present in a given metabolic model.

```
data(Ec_core)
newModel=addReact(model,id="TempReact",met=c("2pg[c]","newMet[c]"),Scoe
f=c(-1,1))
> shrinkMatrix(newModel,j="TempReact")
2 x 1 sparse Matrix of class "dgCMatrix"
          TempReact
2pg[c]           -1
newMet[c]         1
```

To add an exchange reaction, the function *addExchReact* is used; to remove a reaction, the function *rmReact* is used.

1.6.7 Extending Sybil

To add a new algorithm to sybil, the sysBiolAlg class can be extended. The details of the new algorithm should be put in the initialize method. At the end of the method, a call to the next method is issued with the parameters of the optimization algorithm. These parameters include the constraint matrix LHS (which will be interpreted as the constraint $LHS\ v\ =\ 0$), the type of the optimization (LP, QP or MILP) , constraint names, and column names. However, in this thesis, I use different strategies to extend sybil (see the next sections).

1.7 Organization of the thesis

The rest of the thesis is organized into four chapters as follows. In chapter two, I introduce cycle free flux, a new, fast algorithm to get thermodynamically feasible flux distributions. This strategy is then applied to get loopless sampling and loopless flux variability calculations. In chapter three, ccFBA is introduced, which implements and extends protein cost or molecular crowding constraints for FBA. In chapter four, I introduce three expression-based methods to predict flux distributions. General conclusions are given in chapter five.

Chapter 2 Loopless Flux

Parts of this chapter are taken verbatim from the publication "CycleFreeFlux: Efficient removal of thermodynamically infeasible loops from flux distributions" [61], for which I was first author and performed all the analyses except the formal proof of the theorem.

2.1 Introduction

One of the difficulties of FBA predictions is that they frequently include thermodynamically infeasible internal cycles, i.e., sets of reactions that together carry a flux that has no influence on the exchange reactions of the model [35-38, 62]. These are metabolic "perpetual motion machines" and do not occur in biological reality. Such internal cycles thus distort predicted flux distributions and should be removed from the predictions. Thermodynamically infeasible cycles affect not only the predictions of FBA, but also those of many other constraint-based analysis methods. In particular, thermodynamically infeasible cycles distort the output of sampling algorithms that aim to characterize the steady-state solution space, and lead to the assignment of unrealistically high values to the flux ranges that can be carried by individual reactions in flux variability analysis (FVA) (Mahadevan and Schilling, 2003).

One frequently used technique to remove thermodynamically infeasible cycles from FBA solutions is to minimize the sum of absolute fluxes (minimization of total flux, MTF, sometimes also called parsimonious FBA, pFBA) [26]. While MTF successfully removes the internal cycles, it does so by severely constraining the predicted flux distributions, and thus MTF results no longer represent the full solution space of the FBA problem. Comparisons to transcriptomic data indicate that at least in some cases, the MTF solution does not adequately reflect real-life biochemical fluxes [20, 63].

Several other approaches for the identification and/or exclusion of thermodynamically infeasible cycles have been proposed [36-38]. The most widely-used method, ll-COBRA [37], is based on the integration of thermodynamic constraints with FBA into a mixed-integer linear problem (MILP). However, this approach is computationally expensive, resulting in runtimes that severely limit its applicability to large-scale studies.

We propose a new algorithm, termed CycleFreeFlux, which removes all thermodynamically infeasible cycles from any given flux distribution with a single linear optimization step; this makes it orders of magnitude faster than previous approaches when applied to sampling or flux variability analyses of genome-scale metabolic networks.

2.2 Characterization of internal cycles

To motivate the CycleFreeFlux algorithm, we first formally characterize thermodynamically infeasible flux distributions. Consider the following standard FBA problem:

$$\begin{aligned} & \max c^T v \\ \text{subject to:} \quad & Sv = 0 \\ & lb \leq v \leq ub \end{aligned} \tag{1}$$

In the following, we will call a flux distribution 'feasible' if it is non-zero and adheres to all constraints in the FBA problem (1) (i.e., if it lies in the solution space of (1)). We call

a flux distribution v' thermodynamically feasible if there exists an assignment of free energies G to the metabolites such that the free energy change caused by each active reaction is strictly negative, i.e., $v'_j \Delta G_j < 0$ for all $j \in Support(v')$, with $\Delta G_j = {s_j}^T G$, where s_j denotes the j-th column of S [37, 64], and $Support(v')$ is the set of indices *i* for which $v'_i \neq 0$.

We will explicitly consider the free energies of external metabolites in order to deal correctly with thermodynamically infeasible flux distributions that involve transport reactions (reactions that shuttle metabolites between internal compartments and the external compartment). For each external metabolite, we add one reversible "exchange" reaction between the external compartment and an additional 'NULL" compartment that is not explicitly modeled. Many genome-scale metabolic reconstructions already define exchange reactions in that way. For any given free energy of a metabolite present in the external compartment, we can set a hypothetical free energy outside the external compartment that drives the exchange reaction in the desired direction. Thus, exchange reactions as defined here are always thermodynamically feasible. When considering thermodynamical feasibility, we only need to examine internal reactions (including transport reactions), and it is hence convenient to partition the stoichiometric matrix into an internal and an exchange part, $S = [S_{int}; S_{ex}]$.

We call a non-zero flux distribution Δv "internal" if, and only if, it fulfills the conditions $S\Delta v = 0, \min\{l_i, 0\} \leq v_i \leq \max\{0, u_i\}$ for all fluxes v_i, and if all its exchange fluxes are zero. Thus, internal flux distributions do not change any internal or external metabolite concentrations; they can be thought of as combinations of internal cycles that collectively neither consume nor produce anything. If the constraints of (1) do not enforce any non-zero fluxes, then internal flux distributions are also feasible.

As free energy is a state variable of metabolites, the free energy changes of the reactions active in Δv must add to zero in steady state:
$\Delta v \Delta G = 0$ [37, 64]. Thus, $\Delta v_k \Delta G_k \geq 0$ for at least one reaction $k \in Support(\Delta v)$. There is hence no thermodynamic driving force for at least one reaction in $Support(\Delta v)$; consequently, all internal flux distributions are thermodynamically infeasible. The following theorem (which was proven by our collaborator Florian Jarre, see [61]) characterizes thermodynamically feasible flux distributions:

Theorem: A flux distribution $v \neq 0$ that is feasible for (1) is thermodynamically feasible if, and only if, there does not exist any internal flux distribution $\Delta v \neq 0$ with $Support(\Delta v) \subset Support(v)$ and $\Delta v_k v_k \geq 0$ for all k.

Proof: We may assume, without loss of generality, that $v_k \neq 0$ for all k. (If not delete the k-th component of v and the k-th column of S.) We may further assume that $v_k > 0$ for all k. (If not replace the k-th component "v_k" by "$-v_k$" and the k-th column "s_k" of S by "$-s_k$".) Then, by definition, v is thermodynamically feasible, if, and only if, there exists a vector of free energies G such that $S_{int}^T G < 0$ (i.e., each component of the vector $S_{int}^T G$ must be negative).

By Gordan's theorem (see, e.g., Theorem 2.2.1 in [65]), this is the case, if, and only if, there does not exist a vector $x \geq 0$ with $x \neq 0$ and $S_{int} x = 0$. Identifying x with Δv the claim follows.

We have thus shown that a feasible flux distribution v is thermodynamically feasible if and only if it cannot be "reduced" by subtracting an internal flux distribution Δv.

2.3 Removing internal cycles from a given flux distribution (CycleFreeFlux)

The theorem provides the motivation for the CycleFreeFlux algorithm: we aim to reduce a given flux distribution $\boldsymbol{v^{(0)}}$ to its thermodynamically feasible part. To achieve this goal, CycleFreeFlux minimizes the sum of absolute fluxes while (i) all exchange fluxes are kept constant, and (ii) no internal flux is allowed to change direction or increase in size. If the input flux distribution is the output of a previous optimization, CycleFreeFlux additionally constrains the value of the objective function $\boldsymbol{c^T v}$ to its optimal value. We assume that there are no lower bounds $lb_i > 0$ and no upper bounds $ub_i < 0$, i.e., all fluxes are allowed to be nonactive; otherwise, these constraints may enforce thermodynamically infeasible fluxes, and then $v^{(0)}$ cannot be reduced to a feasible solution that is also thermodynamically feasible.
We thus solve the following linear optimization problem:

$$\begin{aligned} & \min \textstyle\sum_i |v_i| \\ \text{subject to:} \quad & S\boldsymbol{v} = 0 \\ & 0 \le v_i \le v_i{}^{(0)} \text{ for } i \text{ with } \boldsymbol{v^{(0)}} \ge 0 \\ & v_i{}^{(0)} \le v_i \le 0 \text{ for } i \text{ with } \boldsymbol{v^{(0)}} < 0 \\ & v_j = v_j^{(0)} \text{ for all exchange fluxes } \boldsymbol{v_j} \\ & \boldsymbol{c^T v = c^T v^{(0)}} \end{aligned} \tag{2}$$

As fluxes are not allowed to change directions, we can transform (2) trivially to a linear problem by replacing the sum over the absolute values by two sums over positive and negative fluxes, respectively:

$$\sum_i |v_i| = \sum_{i \text{ with } v_i^{(0)} > 0} v_i - \sum_{i \text{ with } v_i^{(0)} < 0} v_i \,.$$

The flux distribution v resulting from this optimization is ”structurally consistent” with $v^{(0)}$. Here, we define a flux distribution v as structurally consistent with an input flux distribution $v^{(0)}$ if v neither increases (in absolute value) nor inverts any fluxes compared to $v^{(0)}$. CycleFreeFlux outputs a minimal (in terms of the sum of absolute fluxes $\| v \|_1$) flux distribution that is structurally consistent with the input flux distribution $v^{(0)}$.

***Corollary*:** A flux distribution $v \neq 0$ that is feasible for (1) (with $lb \le 0 \le ub$) is thermodynamically feasible if, and only if, the output of the CycleFreeFlux algorithm (2) with input $v^{(0)} = v$ is v itself.

***Proof*:** To simplify the presentation assume again that $v^{(0)} \ge 0$. If CycleFreeFlux returns a flux $v \neq v^{(0)}$, then, by construction, $\Delta v \coloneqq v^{(0)} - v \ge 0$ and $support(\Delta v) \subset support(v^{(0)})$. Thus, by the theorem, $v^{(0)}$ is not thermodynamically feasible. Conversely, if $v^{(0)}$ is not thermodynamically feasible, then, by the theorem, there exists an internal flux distribution $\Delta v \neq 0$ with $support(\Delta v) \subset support(v^{(0)})$. Then, $v^{(0)} - \epsilon \Delta v$ is feasible for (2) for small $\epsilon > 0$, reducing the objective value in (2), so that CycleFreeFlux will not return $v = v^{(0)}$ as output. □

Figure 3 illustrates our strategy on a simple toy model with five reactions, $v_1, \ldots, v_5$. All steady-state flux distributions in this model can be expressed as positive linear combinations of three convex basis vectors, the elementary flux modes e_1, e_2, and e_3. $e_1$and $e_2$involve the exchange fluxes $v_1$and $v_3$and are thus type I extreme pathways. In contrast, the loop e_3 involves only the internal fluxes v_2, v_4, and v_5 and is hence type III (i.e., thermodynamically infeasible).

The fluxes shown in the figure are our input flux distribution, $v^{(0)}$. To apply the CycleFreeFlux algorithm, we constrain the internal reactions v_4 and v_5 to non-negative values ($0 \leq v_4 \leq 1000$ and $0 \leq v_5 \leq 1000$), while the third internal reaction, v_2, is constrained to non-positive values ($-999 \leq v_2 \leq 0$). At the same time, we fix the exchange reactions at their input values, $v_1 = v_3 = 1$. Minimizing the total flux under these conditions reduces the flux through the internal loop down to the point where v_2 becomes zero, and hence the algorithm puts out $v = e_2$ as the thermodynamically feasible part of the input flux distribution $v^{(0)} = e_2 + 999e_3$.

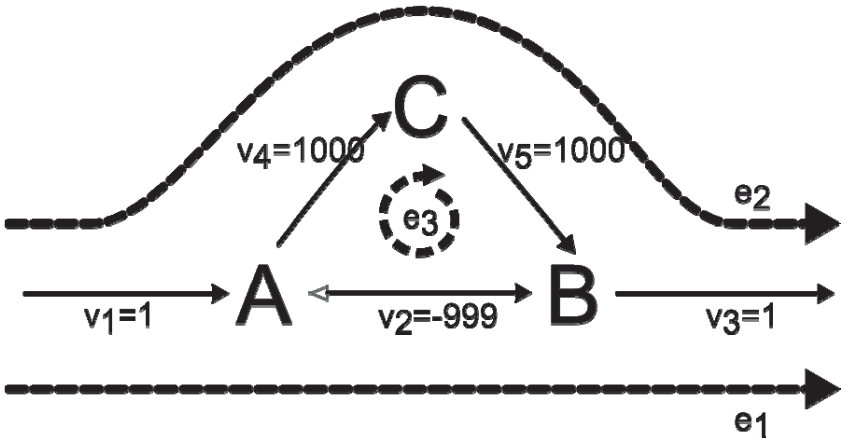

Figure 3 A toy model illustrating our algorithms, consisting of three metabolites (A,B,C) that can be interconverted by three internal reactions (v_2, v_4, v_5). A can be taken up from the environment through the exchange reaction v_1, while B can be secreted through v_3. All fluxes except v_2 are unidirectional, allowed to proceed only from left to right (which is also the nominal flux direction of v_2).The steady-state flux distributions of this model can be expressed as positive linear combinations of the extreme pathways e_1, e_2, e_3. The cycle e_3 involves only internal fluxes (i.e., it is a type *III* extreme pathway), and is thus thermodynamically infeasible

Note that the decomposition of the input flux distribution $v^{(0)}$ in terms of the convex basis vectors is not unique: we can also write $v^{(0)} = e_1 + 1000e_3$. Thus, $v^{(0)} = e_1$ would be an alternative thermodynamically feasible solution consistent with the input flux distribution $v^{(0)}$. This alternative inverts the direction of one internal flux, v_2, and is thus not structurally consistent with $v^{(0)}$.

The preference for structurally consistent solutions causes a corresponding bias if there exist alternative decompositions of $v^{(0)}$ into elementary flux modes that are structurally inconsistent with $v^{(0)}$; in addition, CycleFreeFlux is biased towards solutions with lower total flux $\| v \|_1$. That there always exists a structurally consistent solution is caused by the fact that internal flux distributions are cycles, and hence it is always possible to route the thermodynamically feasible flux through the cycle section that runs in the same direction. The condition of structural consistency with an input flux distribution is what differentiates CycleFreeFlux from MTF.

We can write the thermodynamically feasible output flux distribution $v = \sum_i \alpha_i e_i^{(I)}$ in terms of type I elementary flux modes, and the internal flux distribution $\Delta v = \sum_j \beta_j e_j^{(III)}$ in terms of type III elementary flux modes [34]. This allows us to write

$$v^{(0)} = \sum_i \alpha_i e_i^{(I)} + \sum_j \beta_j e_j^{(III)}$$
$$with\ v = \sum_i \alpha_i e_i^{(I)} \quad (3)$$

i.e., there exists a decomposition of the input flux distribution into type *I* and type *III* elementary flux modes such that the output flux distribution consists exactly of the type *I* contributions.

Note that if the constraints in (1) entail lower flux bounds $l_i > 0$ and/or upper bounds $u_j < 0$, then these enforce nonzero fluxes through the corresponding reactions. Such bounds are rarely used in FBA-type analyses. However, if such bounds are used and are chosen inappropriately, they may result in non-zero activities of thermodynamically infeasible cycles. These will be removed by our algorithm, resulting in an output flux distribution that will not be feasible for the original problem. That such an inconsistency may be present can be detected by comparing the output of the CycleFreeFlux algorithm to the constraints of the original problem to check its feasibility. However, if one is certain that the constraints of (1) do not enforce internal cycles, then corresponding modifications of the constraints in (2) to handle this case are straightforward. This is particularly relevant to allow for a "maintenance energy" term, which enforces a fixed amount of energy consumption in many FBA models.

Futile cycles involve the consumption and/or production of "currency metabolites" such as ATP. Futile cycles are often thermodynamically feasible. They are not internal flux distributions, and do not require special treatment in our formalism.

2.4 Speed

The previous state-of-the-art in obtaining cycle-free FBA solutions was ll-FBA[37]. Because of their different strategies, the output of CycleFreeFlux and ll-FBA cannot be compared directly: ll-FBA [37] directly solves an FBA problem within the subspace of thermodynamically feasible flux distributions; in contrast, CycleFreeFlux takes any given steady-state flux distribution (which may or may not be the result of an FBA calculation) and removes its thermodynamically infeasible contributions.

As the CycleFreeFlux algorithm is a post-processing step consisting of a single linear optimization, it increases the computation time approximately two-fold compared to standard FBA, and is thus very similar to an MTF strategy in terms of run times. Table 1 compares the run time of CycleFreeFBA (standard FBA followed by the CycleFreeFlux algorithm) to that of the ll-FBA algorithm proposed by [37]. For better comparability, both algorithms were implemented in R [49] and are run on the same metabolic networks, using the default environments and biomass reactions supplied by the BIGG database [66]. For the genome-scale networks of *E. coli* [9] and *Saccharomyces cerevisiae* [67], cycleFreeFlux is 400-2500 times faster than the alternative ll-FBA algorithm, which solves a mixed-integer linear problem instead of a standard linear programming problem.

Table 1 Runtime comparison between ll-FBA (Schellenberger et al., 2011) and CycleFreeFBA.

Model	reactions No	Solver	ll-FBA[1]	CycleFreeFlux[1]
Ec_core	95	GLPK	0.08	0.03
Ec_core	95	CPLEX	0.42	0.04
iMM904	1577	GLPK	225	0.53
iMM904	1577	CPLEX	172	0.16
iAF1260	2382	GLPK	1099	0.79
iAF1260	2382	CPLEX	649	0.25

[1] Run times in seconds on a standard laptop with core i7 processor and 8 GB RAM

2.5 Cycle-free sampling

The CycleFreeFlux algorithm can be applied to any given steady-state flux distribution. Thus, it can not only remove internal cycles from FBA solutions, but it can also be directly applied to random samples of the solution space [68]. To remove thermodynamically infeasible cycles from sampled flux distributions, we reduce each sample to its contributions from type I extreme pathways as explained above.

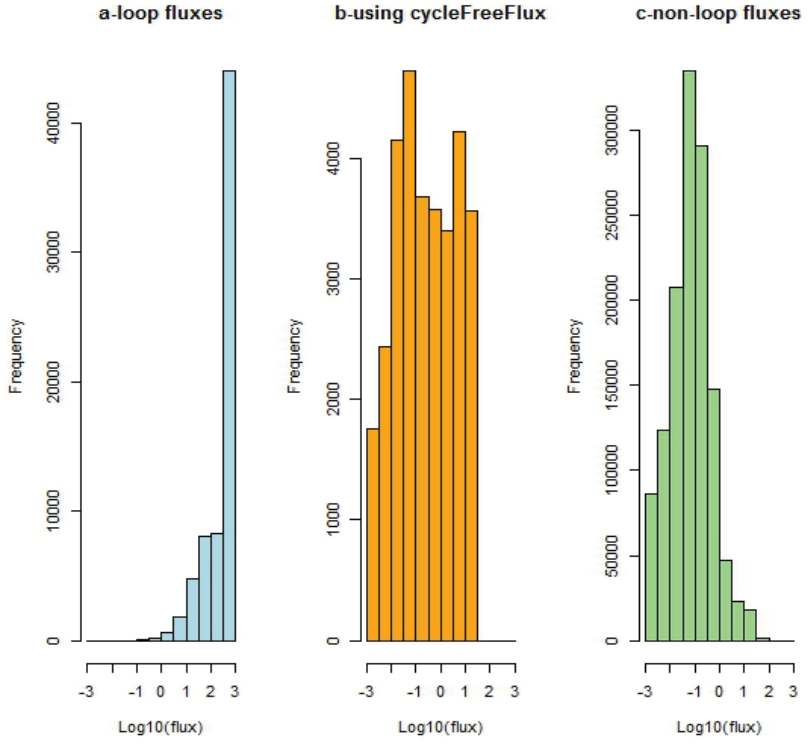

Figure 4 Cycle-free random sampling of the solution space. (a) Histogram of the sampled flux values for fluxes involved in internal cycles. (b) The same fluxes after reduction to the cycle-free solution subspace with the CycleFreeFlux algorithm. (c) Flux values of reactions not involved in active internal cycles. Random samples were taken from the solution space of the E.coli genome-scale metabolic network iAF1260 in a glucose-limited aerobic medium. Flux values affected by involvement in active internal cycles were identified as those that changed through the application of CycleFreeFlux; these are summarized in (a), while all the remaining (unchanged) fluxes are summarized in (c).

Flux samples taken from the two solution spaces including and excluding internal cycles are compared in Figure 4 for the genome-scale metabolic model of *E. coli* [9]. The

sampled fluxes involved in thermodynamically infeasible internal cycles (Figure 4a) are reduced (Figure 4b) to the "normal" range observed for reactions not involved in active loops (Figure 4c; these non-loop fluxes are identified as those that remain unchanged when applying the CycleFreeFlux algorithm).
The sampling of the thermodynamically feasible solution space aims to cover all steady-state solutions with the same finite probability. This aim is not fully achieved through the CycleFreeFlux algorithm, as can be seen from the example in Figure 3. Ideally, the two type I extreme pathways, e_1 and e_2, should be sampled in equal proportions. However, any initial sample that contains a nonzero flux around e_3 will be reduced to a flux distribution that only contains e_2, biasing the samples towards this flux mode. Any point in the solution space that is involved in an internal cycle and simultaneously has a non-unique representation in terms of elementary flux modes may cause a violation of uniform sampling. We note, however, that uniform sampling is required only when volume-related features are investigated. In many applications, the bias introduced by CycleFreeFlux may be preferable to the bias caused by the inclusion of thermodynamically infeasible flux distributions.
ll-COBRA [37] also suggests to remove internal cycles from sampled flux distributions in a post-processing step, minimizing the distance of the flux vectors between the sampled flux vector and the cycle-free subspace. As can be seen from the example in Figure 3, this strategy introduces a bias that is very similar to the one caused by the CycleFreeFlux algorithm. Thus, currently no published method exists that guarantees uniform sampling of the thermodynamically feasible solution space; as CycleFreeFlux increases the time for sampling only about twofold compared to methods that include internal cycles, we consider CycleFreeFlux the method of choice.

2.6 Cycle-free flux variability analysis

Flux variability analysis (FVA) calculates upper and lower bounds for steady-state fluxes through each reaction at the optimal value of the objective function [27]. Thus, FVA results can be used to characterize the space of alternative optimal solutions to an FBA problem. Thermodynamically infeasible internal cycles are unbounded except by *a priori* constraints, leading to artifactual bounds for all reactions involved in such cycles [27]. To exclude internal cycles from FVA, we propose CycleFreeFVA, an extension of the CycleFreeFlux algorithm. Standard FVA performs two linear optimizations for each reaction R_i, one maximization and one minimization [27]. We extend this approach by iteratively performing maximizations (and later minimizations) and removing cycles. To obtain the flux variability for reaction R_i excluding thermodynamically infeasible internal cycles (i.e., based only on type I pathways), we apply the following algorithm. Any constraints added in one iteration are maintained through further iterations, and the flux distribution $v^{(0)}$ is updated accordingly in each iteration:

Repeat: {
find a steady state flux distribution $v^{(0)}$ with maximal flux v_i through R_i ;
apply the cycleFreeFlux algorithm to $v^{(0)}$, resulting in a new flux distribution $v^{(1)}$;
if ($v_i^{(0)} = v_i^{(1)}$) {
exit iteration (as $v_i^{(0)}$ is not affected by internal cycles) ;
} else {
apply the CycleFreeFlux algorithm to $v^{(0)}$ while constraining $v_i = v_i^{(0)}$, resulting in a new flux distribution $v^{(2)}$ that contains one internal cycle involving R_i ;
break the internal cycle: constrain the fluxes v_j that are zero in $v^{(1)}$ but not in $v^{(2)}$ to $v_j = 0$ one at a time if many can go to 0.
}
}

The upper bound for v_i excluding internal cycles is now in $v_i^{(0)}$. In sum, after maximizing the flux through reaction R_i, the algorithm checks if the reaction is involved in internal cycles by applying the CycleFreeFlux algorithm to the resulting flux distribution $v^{(0)}$; if the flux v_i through R_i is reduced through the removal of internal cycles, then the original maximum $v_i^{(0)}$ was indeed influenced by such cycles. We identify one such cycle involving v_i by fixing v_i at its original (cycling) maximum $v_i^{(0)}$ and then removing all other cycles; afterwards, we break this cycle by reducing its flux as little as possible, but as much as necessary. These steps are repeated until all loops involving v_i have been removed. The calculation of the lower bound for v_i excluding internal cycles is performed in the same way, replacing maximizations with minimizations.

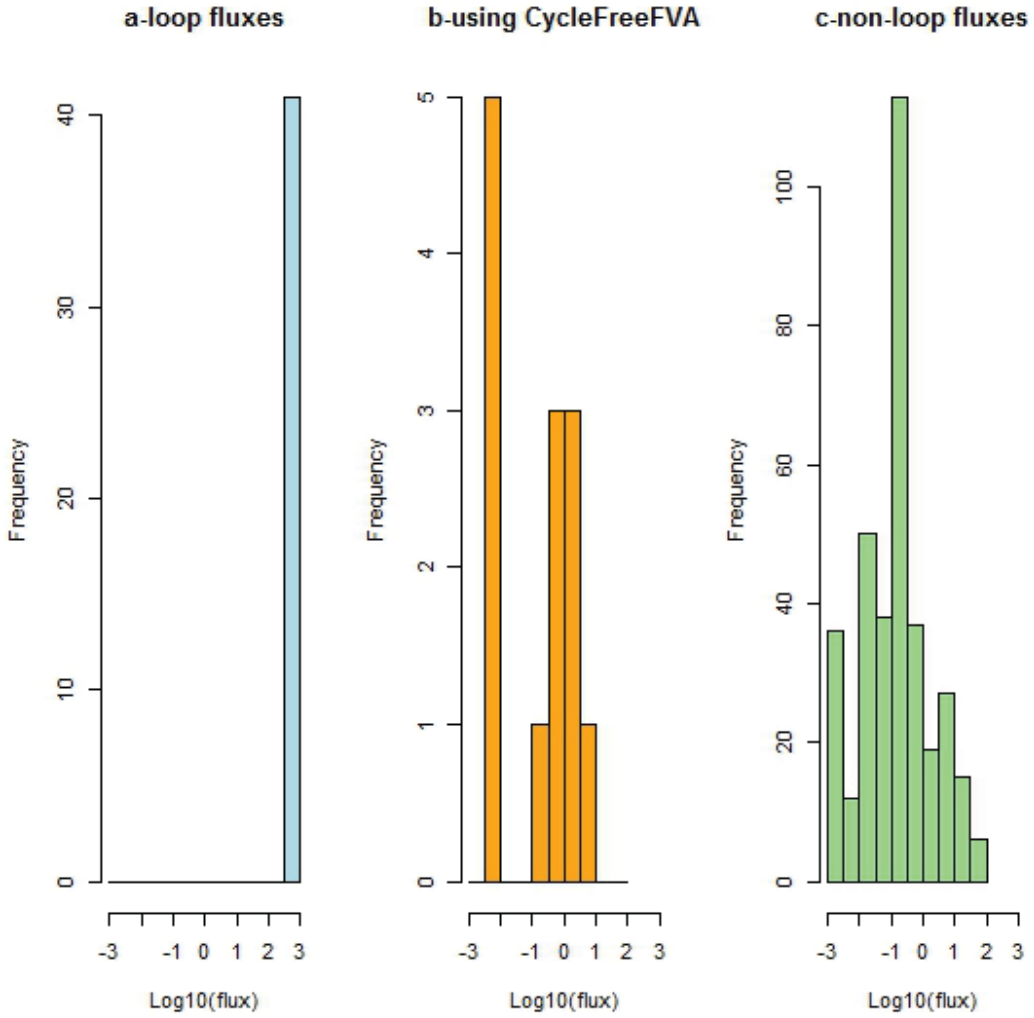

Figure 5 Cycle-free flux variability analysis. (a) Histogram of the maximal flux values that result from involvement in internal cycles. (b) Maximal flux values of the same reactions after excluding internal cycles with the CycleFreeFVA algorithm. (c) Maximal flux values of reactions not affected by internal cycles. Analyses were performed for the E. coli genome-scale metabolic network iAF1260 in glucose-limited aerobic medium. Reactions affected by involvement in internal cycles were identified as those for which the maximum flux differed between standard FVA and CycleFreeFVA; these are summarized in (a) and (b), while the remaining (unaffected) reactions are summarized in (c).

To see how the algorithm works, it is again instructive to consider the toy model in Figure 3. Let us assume that we want to calculate flux variability within the steady-state solution space to an FBA problem with v_3 as the objective function and $v_1 \leq 1$. $v_1 = 1$ and $v_3 = 1$ are then fixed by the boundary conditions, and we need to find the variability of the remaining three reactions. In standard FVA, we would conclude that v_4 can vary within the full range allowed by the *a priori* constraints, say, $0 \leq v_4 \leq 1000$. Now let us apply the above algorithm for v_4. The first flux distribution with maximal flux through v_4 will be the one indicated in Figure 3, $v^{(0)} = e_2 + 999e_3$. Application of the CycleFreeFlux algorithm reduces this to $v^{(1)} = e_2$. v_4 changed its value between $v^{(0)}$ and $v^{(1)}$, indicating that we're not done yet. As the model contains only one internal

cycle, we have $v^{(2)} = v^{(0)}$. The only reaction that is zero in $v^{(1)}$ but not in $v^{(2)} = v^{(0)}$ is v_2; this reaction we now constrain to zero. The maximal value for v_4 is now 1. Exactly the same happens when we consider v_5.
What about v_2? Its maximum, 1, is not affected by the internal cycle e_3. Its minimum is again realized in the flux distribution $v^{(0)} = e_2 + 999e_3$ indicated in Figure 3. Again, we have $v^{(1)} = e_2$ and $v^{(2)} = v^{(0)}$. Reducing the flux through the cycle as little as possible, but as much as necessary, we identify the lower limit $0 \leq v_2$.
We applied the CycleFreeFVA algorithm to the genome-scale model of *E. coli* [9]. Figure 5 shows the distribution of the maximal fluxes for reactions that can be affected by internal cycles with standard FVA (Figure 5a) and with CycleFreeFVA (Figure 5b), and compares those to the distribution for fluxes not affected by internal cycles (Figure 5c). Note that after the publication of our article describing this method [61], we became aware of a very similar strategy for thermodynamically feasible flux variability analysis that was proposed independently by Arne Müller [69].

2.7 Enumeration of internal cycles

To enumerate thermodynamically infeasible cycles, we propose a method based on an extension of the CycleFreeFVA approach (function *enumerateCycles*). We perform CycleFreeFVA for each internal flux v_i, and we simply store the cycles removed in each iteration (i.e., $\Delta v := v^{(2)} - v^{(1)}$). Because the same cycle may be identified for each reaction involved in the cycle, we need to restrict the total set of identified internal cycles to unique reaction sets.
Each set of reactions identified through this algorithm represents a unique internal flux distribution that cannot be reduced while maintaining the flux through one of its reactions (the one used for its identification); thus, it is an ”elementary” internal cycle (type III elementary flux mode).
I used this algorithm to identify thermodynamically infeasible cycles in the genome-scale metabolic model for *E. coli* [9]. The identification of internal cycles in this network took less than five minutes on a standard laptop (core i7 processor and 8GB RAM). Excluding trivial cycles involving only two reactions (i.e., forward and backward reactions of the same stoichiometry), I identified 27 non-trivial thermodynamically infeasible cycles, listed in Table 8. This is identical to the number reported by Wright and Wagner (2008), 27; as the detailed results of [36] are not available, I could not examine if the identified cycles were identical.

Recently, De Martino et al. (2013) reported a lower bound of 189 non-trivial thermodynamically infeasible loops (> 2 reactions each) in the same metabolic network model. 11 of these internal cycles were also identified by our approach; the remaining 178 cycles violate at least one a priori thermodynamic constraint, i.e., at least one reaction in the cycle carries flux in a direction not allowed by the published model [9].

To speed up the enumeration of thermodynamically infeasible cycles, I additionally implemented a method that transforms the model into a reduced irreversible form that only includes reactions involved in cycles (function *getModel_WW*), similar to the strategy proposed in [36]. First, I identified “trivial” cycles, *i.e.*, pairs of reactions that are forward and backward direction of the same stoichiometry. I then performed standard

FVA for all reactions of the network, excluding the reverse direction of the currently studied reaction.
I further implemented an algorithm termed “extended cycle” in *enumerateCycles*, which merges two previously identified and overlapping cycles to find larger cycles directly.
I applied my algorithms to the *S. cerevisiae* model iMM904, detecting 40 thermodynamically infeasible cycles (Table 9).
I applied the preprocessing step (*getModel_WW*) followed by an application of *enumerateCycles* also to the human metabolic network reconstruction Recon1 [70]. This resulted in a list of over 4 million thermodynamically infeasible cycles before I aborted the search.
sybilcycleFreeFlux is an extension to Sybil [31]. As Sybil, sybilcycleFreeFlux is available free of charge from CRAN (http://cran.r-project.org/web/packages/sybilcycleFreeFlux/)

Chapter 3 ccFBA: building MetabOlic Models with ENzyme kineTics (MOMENT) from FBA models in R

3.1 Introduction

As outlined in Chapter 1, FBA ignores important cellular constraints beyond the stoichiometry of biochemical reactions: the enzymes needed to catalyze biochemical fluxes need to be produced using cellular resources, and they need to fit into a finite intracellular space. If the intracellular concentration of enzymes becomes too large, molecular crowding hinders the diffusion of proteins and metabolites necessary for the efficient catalyzation of biochemical reactions.

To account for this molecular crowding, Beg *et al.* [42] incorporated an upper limit on total enzymatic capacity into the FBA framework. They added a constraint on the volume available for enzymes (FBA with molecular crowding, FBAwMC), showing that this extension of FBA improved the prediction of phenotypes for *E. coli*.

Goelzer *et al.* [43] extended this approach by considering metabolic capability, translation capability, and density constraints (RBA). RBA requires the categorization of Metabolites as internal, recycled, or metabolic precursors; further, it assumes that the turnover numbers (K_{cat}) of all enzymes are identical, and does not incorporate gene-protein-reaction (GPR) rules that associated reactions with specific enzymes.

MetabOlic Modeling with ENzyme kineTics (MOMENT) [71] extended FBAwMC by (i) including GPR rules, (ii) using molecular weights to estimate enzyme volume, and (iii) increasing the number of reactions with experimentally determined turnover rates. A recent theoretical study has shown that solutions to such constraint-based models are elementary flux modes [72].

Here, I present an improved general implementation of MOMENT in R, named ccFBA (for cost-constrained flux-balance-analysis). ccFBA is an extension to Sybil [31]. As Sybil, ccFBA is available free of charge from CRAN (http://cran.r-project.org/web/packages/sybilccFBA/)

3.2 Algorithm and Implementation

The MOMENT algorithm was originally implemented in Matlab for *E. coli*, with hard-coded model details and constraints [71]. Thus, it is not straight-forward to modify the existing implementation or to apply it to additional model organisms. In ccFBA, the metabolic model is incorporated as an input parameter, as are the turnover numbers and the molecular weights. In addition to the *E. coli* model originally implemented by [71], I also provide a ccFBA model for the baker's yeast *Saccharomyces cerevisiae* (see below).

However, a major strength of ccFBA is its efficient mechanisms to construct models for additional organisms. Building a new ccFBA model should start from an existing FBA model, such as those available from the BIGG database [73]. To build a new ccFBA model, we need to extend the FBA model by (i) converting the Boolean rules that link genes to reactions (GPR rules) to linear constraints; (ii) adding molecular weights to all proteins covered by GPR rules; (iii) adding turnover rates; and (iv) determining the total constraint on enzyme volume (the budget).

Steps (i)-(iii) are performed by the ccFBA function *cfba_moment*. Step (iv), determining the budget, requires either an estimate of the percentage of cellular mass taken up by enzymes, or a fit of model predictions to experimental data. Molecular weights can either be supplied by the user, or can be calculated from amino acid sequence files through the ccFBA function *calc_MW*.

ccFBA converts GPR rules to constraints with the following algorithm:

$$\text{A or B} \quad \rightarrow \quad v_i \leq k_{cat,i} (n_A + n_B)$$

$$\text{A and B} \quad \rightarrow \quad v_i \leq k_{cat,i} \min(n_A , n_B)$$

Here, A and B are two proteins that catalyze reaction i either independently (OR) or together (AND); v_i is the metabolic flux of this reaction; and $k_{cat,i}$ is the corresponding turnover rate. n_A is the mass fraction of protein copies of A, *i.e.*, the weight of A molecules divided by the total dry weight of the cells (analogous for n_B). These protein mass fractions are treated as auxiliary variables, which are set to the values that maximize the objective function during the optimization step. More complex relationships between genes and reactions are treated by applying the rules recursively. In the case of multiple OR relationships, we directly use the sum of all variables instead of the recursion, which reduces the number of constraints and auxiliary variables. The global constraint on total cellular enzyme volume is then expressed as

$$\sum_A n_A MW_A \leq C \qquad (1)$$

where MW_A denotes the molecular weights of protein encoded by gene A, and C denotes the total weights of proteins. MOMENT incorporates a careful treatment of GPR associations. However, the implementation in [71] does not penalize the usage of the same protein in different reactions. In essence, this means that multifunctional enzymes are assumed to be capable of performing different reactions simultaneously. In a situation where the enzyme is saturated by its substrates (which is a general assumption in MOMENT), this is not biochemically realistic: if an enzyme is involved in two or more alternative reactions, then we have to partition the cellular amount of the enzyme between the different reactions at any given moment. We thus modified the MOMENT algorithm accordingly. Whenever a protein was involved in more than one reaction, we introduced

auxiliary concentration variables x_i for each of these reactions. These x_i replaced the global concentration variable n_A for the protein in the corresponding equation that limits the flux through this reaction based on the enzyme concentration. The sum of the x_i is then equal to the total concentration of protein *A* included in the global enzyme solvent capacity constraint. Below, we refer to this modified model as MOMENT*.

The most elaborate task in building a ccFBA model is collecting the turnover rates from databases (such as BRENDA [11] or SABIO-RK [74]), from primary literature, or through wet lab experiments. The function cfba_moment() converts an existing FBA model into a MOMENT/MOMENT* model. In addition to the FBA model, it expects a file with k_{cat} values as input; missing values, which are the rule rather than the exception also for existing models [42, 43, 71], are replaced by the median of all available k_{cat} values. The function returns a Sybil model structure sysBiolAlg [31]; this can be used for further processing or saved as a linear problem file.

3.2 Results

3.2.1 Escherichia coli

To validate ccFBA against the original MOMENT implementation (kindly provided by the authors of [71]), we built a ccFBA model from the same *E. coli* FBA model, iAF1260 [9], and with the same k_{cat} values and molecular weights. As in the original implementation, we only used k_{cat} values from experiments on *E. coli* enzymes to calculate the median k_{cat}; the full k_{cat} list used for the model also contains values of homologous enzymes from other species. To align the ccFBA model with the original MOMENT implementation, we also set the lower bound of the ATP maintenance reaction to zero. The maximal growth rates calculated with the ccFBA model (without the modification for multifunctional proteins) and the original MOMENT implementation agreed within 0.001 milliMol per gram dry weight per hour (mmol/gDW/h) for each of the 24 carbon sources listed in Table 2.

524 reactions of the iAF1260 model are affected by multifunctional enzymes. The modified model including an explicit consideration of multifunctional enzymes led to substantially reduced growth rate predictions on all 24 carbon sources compared to the original MOMENT implementation (Table 2). This is due to the fact that in the original implementation, once one reaction pays the "price" for inclusion of the enzyme in Eq. (1), all other reactions catalyzed by the same enzyme effectively run "for free". This artifactually allows much more total flux than when explicitly considering multifunctional enzymes. Depending on the carbon source, between 9.1% and 21.7% of the reactions differ in their on/off state between the original MOMENT model and the MOMENT* model, which explicitly considers multifunctional enzymes (Figure 6). Although the MOMENT* model more faithfully reflects biochemical constraints, the correlation of predicted with experimentally measured maximal growth rates was slightly

reduced (Table 2). Note, however, that the predictive power of all model variants examined here is relatively low (Figure 2). Spearman's ρ values for the different model variants listed in Table 2 are not statistically significantly different from each other (the 95% confidence interval for Spearmans' rank correlation between measurements and the original MOMENT model, for example, is 0.086 - 0.736).

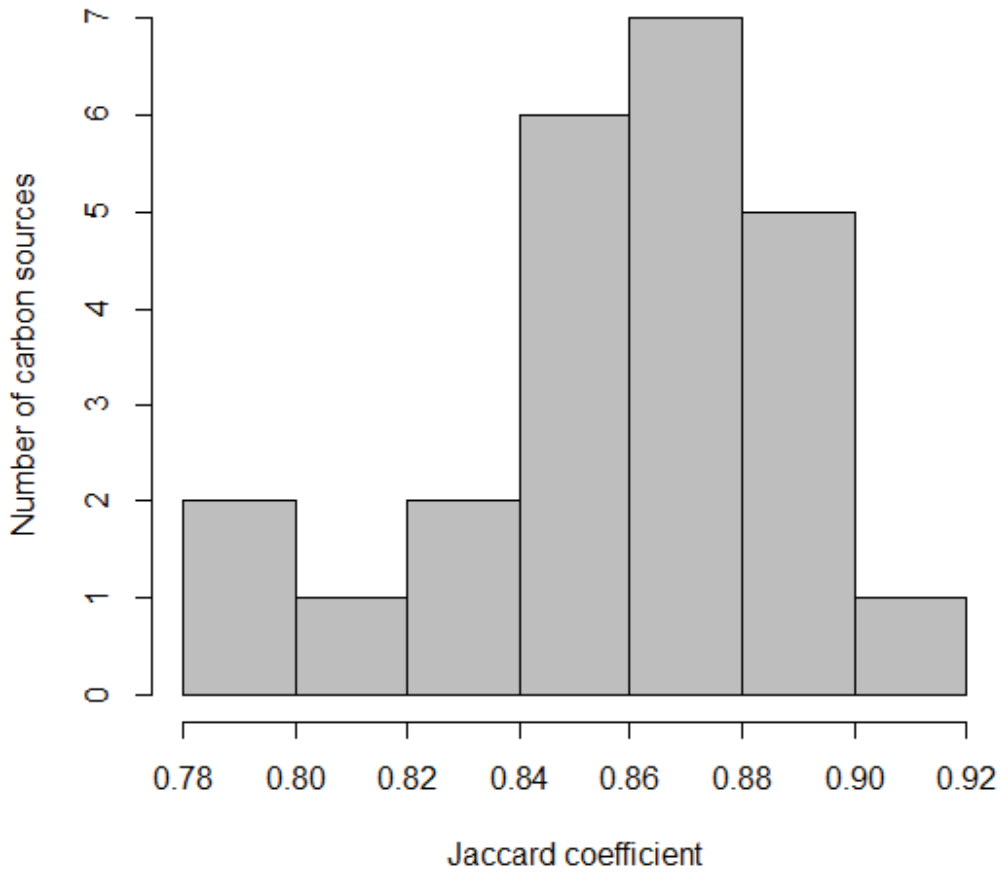

Figure 6 At least 10% of reactions show different activities between the original MOMENT iJO1366 model and the modified model considering multifunctional enzymes. The figures shows a histogram of the distribution of Jaccard coefficients across the environments representing 24 carbon sources examined in Adadi R, Volkmer B, Milo R, Heinemann M and Shlomi T [71]. For each experiment, this coefficient represents the number of reactions that are active (flux > 10^{-6}) in one model but inactive (flux ≤ 10^{-6}) in the other, divided by the number of reactions active in at least one of the models.

The incorporation of cellular constraints on total enzymatic capacity (Eq. (1)) is an important step towards bringing FBA-type models closer to biological reality. The quality of such extended models depends strongly on the reliability and completeness of the information on enzyme turnover rates. We thus extended the k_{cat} list of the original

MOMENT implementation by adding 117 turnover rates obtained from the BRENDA database [11] to the MOMENT* model. Using this extended model led to very similar predictions of maximal growth rates (Table 2), and the correlations between experimental growth rates and predictions are rather similar (ρ =0.412 with 513 k_{cat} values vs. ρ =0.390 with 630 k_{cat} values). The predicted flux distributions were surprisingly different for several carbon sources: the lowest Jaccard coefficient for active reactions shared between the two predictions was 0.841, meaning that almost 16% of reactions changed their activity status when I refined the model through additional kinetic information.

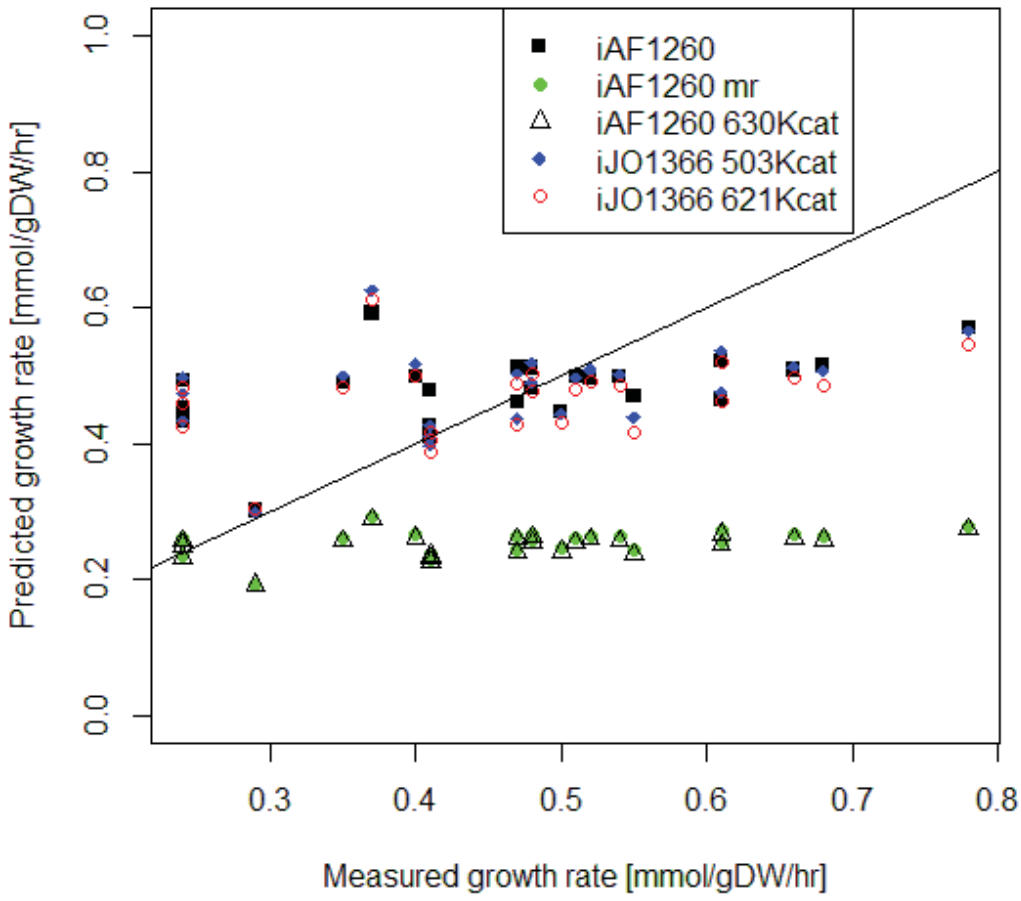

Figure 7 Predicted maximal growth rates for five different metabolic models (y-axis) show much less variation than experimentally measured maximal growth rates (x-axis). See Table 2 for model descriptions and data. The solid black line is the expected identity.

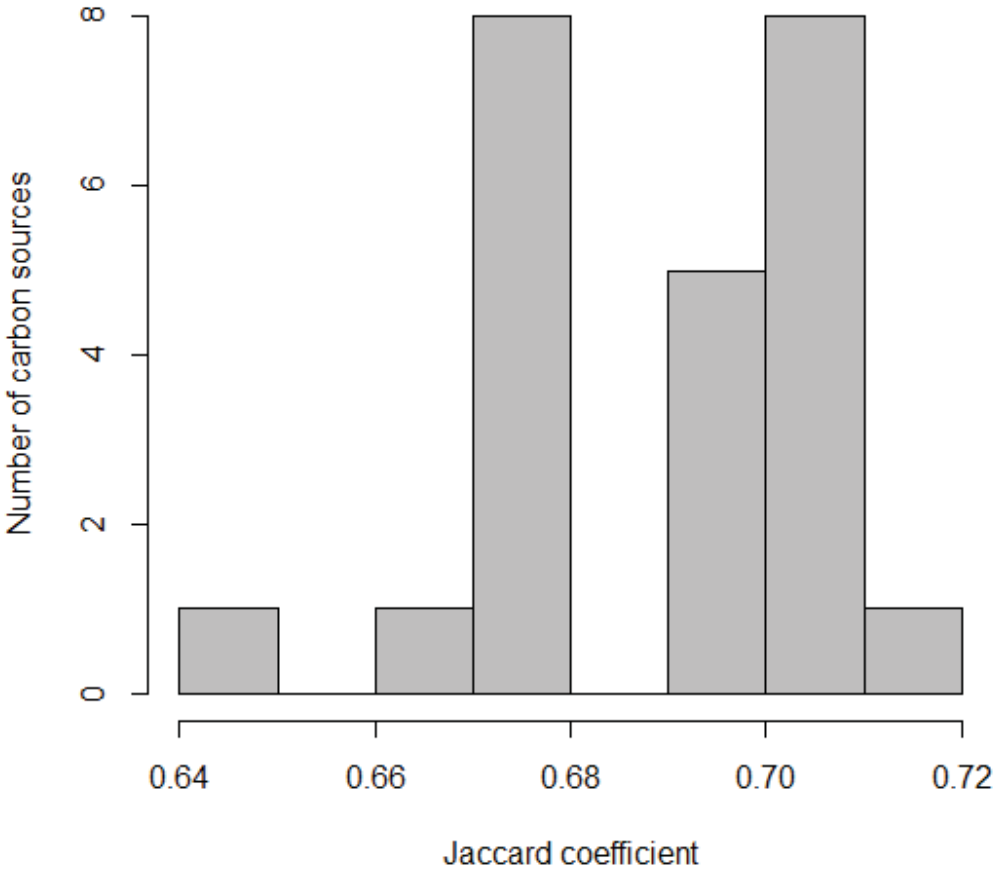

Figure 8 Flux distributions differ strongly between two genome-scale metabolic models for *E. coli,* iJO1366 [75] and iAF1260 [9]. The figures shows a histogram of the distribution of Jaccard coefficients across the environments representing 24 carbon sources examined in [71]. For each experiment, this coefficient represents the number of reactions that are active (flux > 10^{-6}) in one of the two models but inactive (flux ≤ 10^{-6}) in the other, divided by the number of reactions active in at least one of the models.

To update the *E. coli* MOMENT* model, I also used ccFBA to convert the iJO1366 FBA model [75], which contains 2583 reactions and was reconstructed by the same laboratory as the iAF1260 model. I used the same 503 k_{cat} values as in the original MOMENT *E. coli* model. The maximal growth rates predicted with this model were again similar to those obtained with the MOMENT* model based on the older iAF1260 model (see Table 2 for the correlation of predicted growth rates with experimental rates). The predicted fluxes differed strongly between the two *E. coli* models, however: the Jaccard coefficients for active reactions was <0.72 in each case (Figure 8), meaning that more than 28% of reactions changed from active to inactive or vice versa.

Table 2. **Experimentally measured and MOMENT-predicted maximum growth rates (in mmol/gDW/h) for *E. coli* on 24 different carbon sources.**

Carbon source	MOMENT [71] (iAF1260, 513 K_{cat} values)	MOMENT* (iAF1260, 513 K_{cat} values, MFP[1])	MOMENT* [2] (iAF1260, 630 K_{cat} values)	MOMENT* [3] (iJO1366, 503 K_{cat} values)	MOMENT* [4] (iJO1366, 621 K_{cat} values)	WT experimental growth rate (Adadi et al, 2012)
D-Glucose	0.508	0.266	0.260	0.512	0.495	0.66
Glycerol	0.511	0.264	0.259	0.501	0.488	0.47
Acetate	0.301	0.191	0.192	0.299	0.303	0.29
D-Fructose	0.499	0.263	0.258	0.502	0.486	0.54
Pyruvate	0.479	0.239	0.236	0.426	0.417	0.41
D-Galactose	0.492	0.261	0.256	0.498	0.481	0.24
L-Lactate	0.425	0.229	0.225	0.397	0.387	0.41
Maltose	0.496	0.265	0.259	0.509	0.491	0.52
L-Malate	0.469	0.244	0.236	0.438	0.418	0.55
Fumarate	0.462	0.243	0.240	0.437	0.427	0.47
D-Xylose	0.497	0.260	0.255	0.496	0.479	0.51
D-Mannose	0.489	0.262	0.257	0.499	0.483	0.35
Trehalose	0.511	0.268	0.262	0.519	0.502	0.48
D-Mannitol	0.463	0.255	0.251	0.475	0.464	0.61
D-Glucose_6-phosphate	0.570	0.279	0.274	0.565	0.545	0.78
Succinate	0.448	0.245	0.241	0.443	0.431	0.50
D-Glucosamine	0.499	0.266	0.261	0.517	0.500	0.40
D-Sorbitol	0.481	0.259	0.255	0.489	0.476	0.48
D-Gluconate	0.515	0.264	0.258	0.507	0.487	0.68
D-Ribose	0.409	0.233	0.231	0.410	0.405	0.41
Guanosine	0.592	0.292	0.289	0.626	0.612	0.37
L-Alanine	0.434	0.236	0.233	0.433	0.424	0.24
2-Oxoglutarate	0.452	0.254	0.250	0.473	0.459	0.24
N-Acetyl-D-glucosamine	0.521	0.271	0.267	0.536	0.520	0.61
Correlation with experimental growth rates[5]	0.473	0.412	0.390	0.412	0.375	-

[1] MOMENT model as published in [71] with explicit consideration of multifunctional enzymes (MFE)

[2] MOMENT model as published in [71] with MFE and 117 additional turnover rates obtained from BRENDA [11]

[3] MOMENT model based on the iJO1366 FBA model [75] with MFE, using the same 506 Kcat values as in [71]

[4] MOMENT model based on the iJO1366 FBA model [75] with MFE and 118 additional turnover rates obtained from BRENDA [11]

[5] Spearman's rank correlation coefficient ρ between MOMENT predictions and experimental growth rates from [71].

It is striking that experimentally observed growth rates vary much more across carbon sources than the growth rates predicted with each of the model variants tested here. This failure of the models to adequately reflect natural growth rate variation is likely due to one of two possible problems. First, the kinetic information used to calculate the resources required for each individual reaction is incomplete and likely inaccurate; we could include k_{cat} values for only just over a quarter of enzymatic reactions in the iJO1366 model (621 out of 2251). Non-preferred carbon sources are often catabolized by slow enzymatic reactions, so that impossibly high enzyme amounts would be necessary to achieve conversion rates comparable to preferred carbon sources such as glucose. If kinetic information is missing or inaccurate for such a rate-limiting enzyme, growth rate predictions will be artificially inflated.

However, a second problem appears to be responsible at least for the overprediction of growth rates on galactose: an erroneous assumption of maximal growth rate *in vivo*. While galactose can be converted to glucose relatively efficiently, *E. coli* metabolism is transcriptionally regulated so as to not achieve a maximal growth rate [74]. It is currently unclear why this transcriptional response has evolved; only an understanding of the underlying selective forces could allow an improvement of the prediction power of first-principles models such as those examined here. It has been suggested that non-optimal growth rates in *E. coli* are due to the overexpression of enzymes used in the current metabolic state, and by the expression of unused enzymes, possibly to prepare the metabolic system for an expected availability of preferred carbon sources (B. O. Palsson, personal communication)[76]. Thus, it is conceivable that the prediction of maximal growth rates could be significantly improved through knowledge on the cellular volume available for currently active metabolic enzymes.

3.2.2 Saccharomyces cerevisiae

I also used ccFBA to build a MOMENT* model for the baker's yeast *Saccharomyces cerevisiae*, based on the iMM904 FBA model [77]. I obtained k_{cat} values for 577 enzymes collected from the BRENDA database [11], which were compiled in [78, 79], plus 52 manually curated values from BRENDA. Stoichiometries of enzymes were also found from BRENDA for about 250 enzyme complexes. I calculated enzyme molecular weights

from the yeast genome sequence available at NCBI (Saccharomyces cerevisiae S288c). To fix the constraint on total enzyme capacity, I assumed that 27% of the yeast biomass is devoted to enzymes, the same fraction as in *E. coli* [71]. Maximal growth rates on 19 different carbon sources predicted with this model are listed in Table 3.

Table 3 Predicted maximum growth rates for the yeast model on 19 different carbon sources

Carbon source	Growth rate (mMol/gDW/h)
D-Glucose	0.717
Glycerol	0.650
Acetate	0.298
D-Fructose	0.682
Pyruvate	0.499
D-Galactose	0.651
L-Lactate	0.528
Maltose	0.701
L-Malate	0.594
Fumarate	0.581
D-Xylose	0.660
D-Mannose	0.672
Trehalose	0.714
Succinate	0.627
D-Sorbitol	0.720
D-Ribose	0.694
Guanosine	0.932
L-Alanine	0.493
2-Oxoglutarate	0.624

Chapter 4 sybilEFBA: An R package for expression-based FBA

4.1 Introduction

Based on reasonable biological assumptions, flux-balance analysis (FBA) estimates steady-state flux distributions in metabolic networks without knowledge about kinetic parameters. It does this by maximizing a trait relevant for fitness (e.g., biomass yield) under biochemical and environmental constraints. However, solutions are not unique: several distinct metabolic flux distributions may result in the same biomass yield, and it is unclear which of them corresponds to the 'real' biological fluxes. Also FBA doesn't include biological process such as genetic regulation. One strategy to overcome these limitations is to assume a parsimonious use of cellular volumes and to limit metabolism through a constraint on the total cellular enzyme concentration, as explored in the previous chapter. Alternatively, regulatory constraints can be inferred from experimentally determined expression data. Different approaches for integration of gene expression data into constrained-based models have been developed[80, 81]. In this introduction, I present the main ideas of this area.

4.1.1 Existing Expression based methods

4.1.1.1 GIMME(2008)

Gene Inactivity Moderated by Metabolism and Expression(GIMME) [82] requires three inputs: (1) a set of gene expression data; (2) a genome-scale metabolic model reconstruction; and (3) one or more Required Metabolic Functionalities (RMF) that the cell is assumed to achieve. The algorithm uses a threshold on expression data to indicate that reactions are inactive when the corresponding mRNA level is less that the specified threshold. Complex GPR rules are mapped by considering the maximum of two genes when the relation is AND (protein complexes) and the minimum of two genes if the relation is OR (isoenzymes). The algorithm generates a list of reactions in the network that are predicted to be active, and an inconsistency score that quantitatively classifies the disagreement between the gene expression data and the assumed objective function. The method was applied to the *E. coli* network iAF1260 [9] and to gene expression data from three different strains: wildtype, evolved to grow on glycerol, and evolved to grow on lactate [82]. Using a threshold can be problematic because the appropriate threshold value may vary depending on genes, conditions, or organisms [81].

4.1.1.2 E-Flux(2009)

In this method [20], normalized expression levels were used to set upper bounds for fluxes. E-Flux was applied to *Mycobacterium tuberculosis*, the bacterium that causes tuberculosis (TB). E-Flux was used to predict the impact of 75 different drugs, drug combinations, and nutrient conditions on mycolic acid biosynthesis capacity in *M. tuberculosis*, using a public compendium of over 400 expression arrays. The method

successfully predicted seven of eight known inhibitors of mycolic acid and some additional inhibitors that can be used to discover new drugs [20].

4.1.1.2 MADE(2011)

Metabolic Adjustment by Differential Expression (MADE) [83] avoids arbitrary thresholding by using a time series of expression measurements at time points *i*. MADE finds a sequence of binary expression states $\{x_1,x_2,...,x_n\}$, $x_j \in \{0,1\}$ such that the differences between successive states $(x_{i+1}-x_i)$ most closely match the corresponding differences in the expression levels $d_{i\rightarrow i+1}$. MADE uses the statistical significance of the differences to create the most probable approximation. MADE was used to generate a series of models that reflect the metabolic adjustments seen in the transition from fermentative to glycerol-based respiration in *Saccharomyces cerevisiae*. The calculated gene states match 98.7% of possible changes in expression, and the resulting models capture functional characteristics of the metabolic shift [83]. MADE can also be applied to a set of multiple expression measurements that are not part of a time series.

4.1.1.3 tFBA(2011)

The basic assumption of tFBA [63] is that if the activity of a gene drastically changes from one condition to the other, the flux through the reaction controlled by that gene will change accordingly. Up/down constraints are used and the algorithm is allowed to violate them to account for posttranscriptional regulation. One big LP is formulated to find the flux state under a set of conditions simultaneously. The authors were able to predict the fluxes of nine conditions for *Saccharomyces cerevisiae.* The problem becomes very complex and computationally expensive when the number of conditions is large [63].

4.1.1.4 Lee(2012)

Lee and coworkers [84] used absolute gene expression data to form a new objective function which is the absolute difference between the expression level and the corresponding flux, weighted by the inverse of the measurement error. The authors were able to predict measured fluxes with a good correlation under two conditions in *Saccharomyces cerevisiae,* using the yeast model 5 [85] in their simulations. These results were better than those of GIMME and standard FBA. As in other approaches, the mapping from GPR to reactions was made by converting AND to the minimum and OR to the sum of the involved enzymes [84].

4.1.2 Overview of chapter 4

The listed existing methods for expression-based FBA have several shortcomings. Most importantly, they do not explicitly account for differences in the relationships between reaction rates and protein (let alone mRNA) expression levels. Below, I propose two different ways to deal with this issue. In 4.2, I develop FECorr, a method that determines a linear relationship between fluxes and mRNA levels for each reaction. In 4.3, I propose ATM-FBA, a method based on the automatic identification of optimal thresholds, which

effectively assigns individual thresholds for each reaction by scaling mRNA expression levels with k_{cat} . Another potential problem of existing expression-based methods is that they score discrepancies between reaction and gene activities based on the number of conflicting reactions rather than the number of conflicting genes. In 4.4, I propose eFBA_gene, a method that focuses on gene discrepancies.

Kim et al [81] suggested four criteria to compare different methods that use gene expression data to enhance FBA predictions. Table 4 shows these features for the three methods proposed here as well as the previous methods outlined above.

Table 4 Four essential features of eFBA methods.

Method	Requirements for multiple transcription datasets as input	Requirement for a threshold to define a gene's high/low	Requirement for a priori assumption of an appropriate objective function	Validation of predicted fluxes directly against measured intracellular fluxes
E-Flux	No	No	Yes	No
Lee	No	No	No	No
GIMME	No	Yes	Yes	No
MADE	Yes	No	Yes	No
tFBA	Yes	No	Yes	No
FECorr	Yes	No	Adjustable	Yes
eFBA_gene	No	Yes	Adjustable	Yes
ATM-FBA	No	No	Adjustable	Yes

4.2 FECorr Algorithm

We propose an FBA variant that uses additional gene expression data to select among alternative flux distributions (Figure 9). In contrast to a variants of the FBA scheme that considers ON/OFF expression status (e.g., [86]), we account for expression data quantitatively. To obtain a scale linking flux values to gene expression levels (as measured, e.g., by microarray experiments), we start with a flux variability analysis

(FVA)[27] for each reaction under a range of assayed conditions. We then find a best piece-wise linear fit between the resulting flux ranges (which encompass fluxes that all support maximal biomass yield) and measured expression levels across conditions. In a given condition, this relationship is then used to assess the agreement of flux distributions predicted from quantitative gene expression data with all flux distributions consistent with maximal biomass yield (FBA solutions). The FBA solution with minimal distance to expression-predicted fluxes is considered to be close to the biologically realized flux distribution. To account for metabolic states with non-optimal yield, we can constrain our solution space to all flux distributions consistent with a certain minimal percentage of maximal yield instead of requiring 100% of the maximal possible yield. The individual stages of the algorithm are described in detail below.

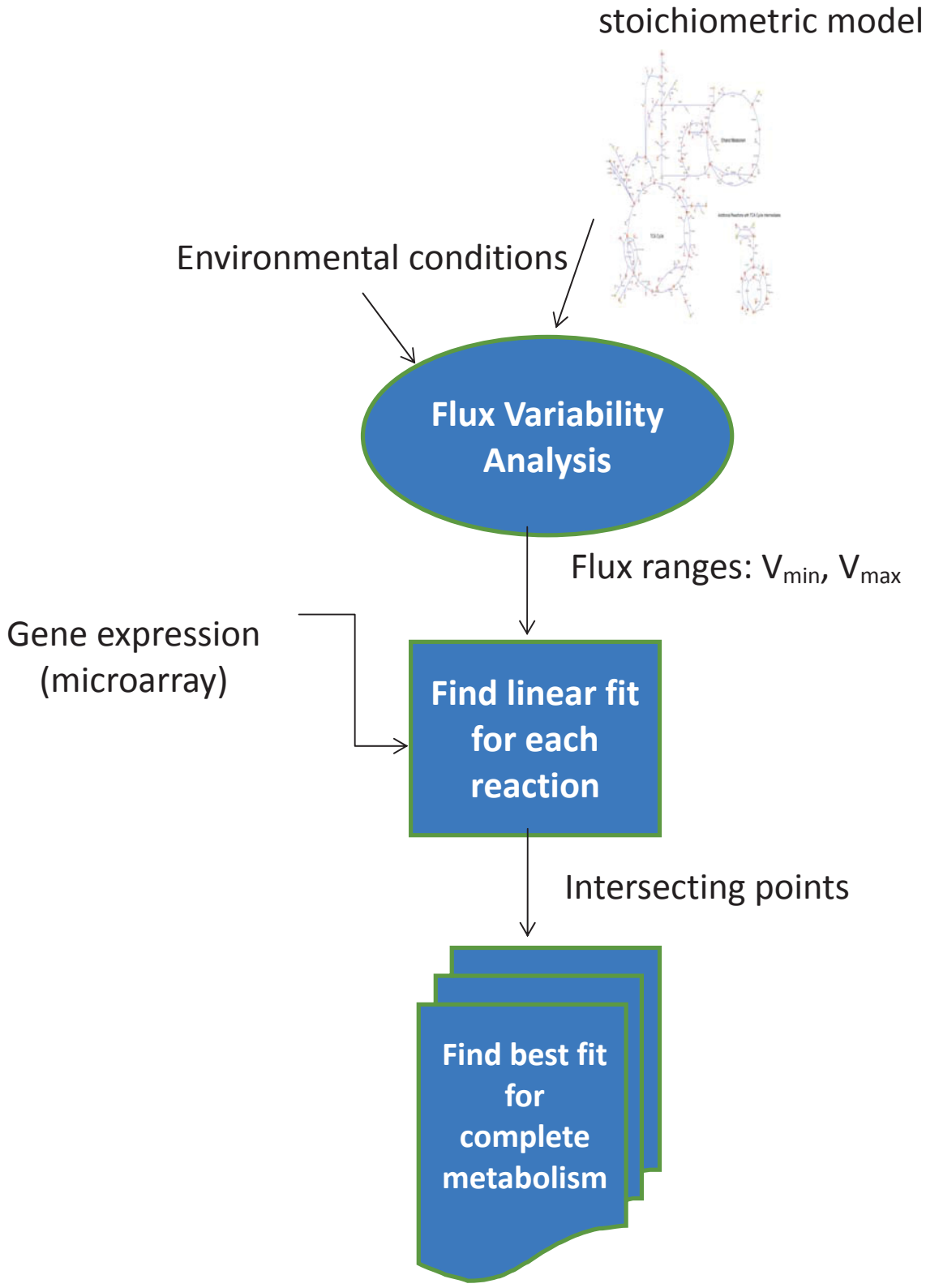

Figure 9 Overview of FECorr Algorithm

4.2.1 Run FVA:

For each experimentally assayed nutrient condition, run FVA to get ranges of possible fluxes under the given environmental condition(s), with the additional constraint that biomass production is at least a certain fraction *b* of its maximal rate, $Z \geq bZ^{max}$, where the default value is *b*=1. Blocked reactions will be excluded.

4.2.2 Fit a piecewise linear function:

As shown in figure 10, for each reaction, we search a piecewise linear function that relates flux values to gene expression levels across environments while respecting the previously determined FVA ranges where possible. Especially in microarray experiments, low but nonzero values of reported expression levels often occur even if the mRNA is not expressed; similarly, low values in RNA-seq data may correspond from spurious transcription rather than to biologically meaningful protein expression. Thus, the proposed relationship between measured expression level and flux consist of two parts: all expression levels up to a point *E0* are considered to correspond to zero flux, while we assume a linear relationship between flux and expression from *E0* onwards (Figure 10). This function thus has two parameters: *E0* and the slope of the line starting at *E0*. In this description, I assume for simplicity that each reaction is catalyzed by only one enzyme; below, I explain how to adapt this procedure for more complicated GPR rules.

The cost function used for fitting this function is :

$$f(x) = \begin{cases} 0 & V_{min} \leq x \leq V_{max} \\ (x - V_{min})^2 & x \leq V_{min} \\ (x - V_{max})^2 & x \geq V_{max} \end{cases}$$

For reversible reactions, there will be three cases: if both V_{min} and V_{max} are positive, then the range will be $[V_{min},V_{max}]$; if both are negative, we will consider the reverse direction of the reaction and set the range to $[-V_{max},-V_{min}]$; finally, if the signs of V_{min} and V_{max} are different, the range will be [0, max(Vmax,-Vmin)]. We used the R function *nlm* to find the equation of the best fitting line according to our cost function, using the following parameter settings: ndigit =15, gradtol = 1e-10, and steptol = 1e-10. The convergence time is very short: the function converges mostly within 10 iterations. As a starting solution we use the *lm* function with the V_{min} points as input.

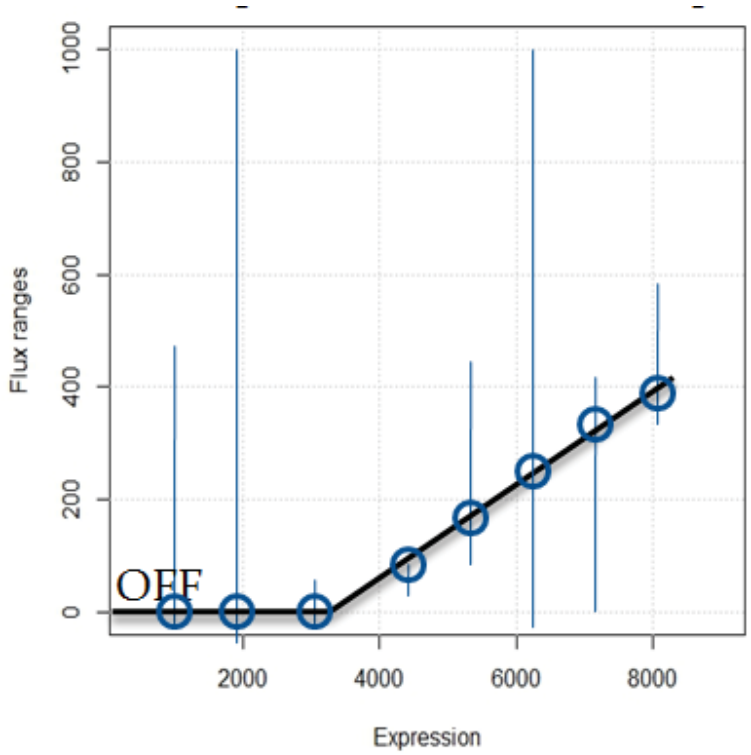

Figure 10 Fitting a piecewise linear function to FVA ranges. Each vertical line marks the FVA range of this reaction in one assayed nutrient condition; the position of the line on the x-axis marks the experimentally determined expression level of a protein associated with this reaction. The circles mark the intersection points between fitted line and FVA ranges

4.2.3 Find closest flux distribution:

According to FBA, each point within the FVA range of a given reaction is equally valid as part of a condition-dependent metabolic state. We propose that approximations to the biologically realized flux value can be read off the piecewise linear function fitted to these ranges across conditions. Thus, we use these points (the circles in Figure 10) to find the flux distribution that most closely matches the experimentally determined expression levels.

For each condition, the predicted flux values in this condition from all reactions are considered together; I then find a point in the solution space of the FBA problem that is closest to these points while producing the desired amount of biomass. This solution is the flux distribution predicted by FECorr.

Formally, the linear problem solved is as follows:

$$\boldsymbol{Min}\ |\boldsymbol{v} - \boldsymbol{f\_fitted}|$$

$$\boldsymbol{Sv} = \boldsymbol{0}$$

$$\boldsymbol{lb} \leq \boldsymbol{v} \leq \boldsymbol{ub}$$

$$\boldsymbol{c^T v} = \boldsymbol{bZ_{obj}}$$

where f_fitted is the set of predicted fluxes for each individual reaction from the piecewise linear function (the circles in Figure 10), S is the stoichiometric matrix, lb is the vector of lower bounds, ub is the vector of upper bounds, Z_{obj} is the value of the objective function in the FBA solution, b is the desired fraction of this value to be achieved by the solution (default b=1), and c is the vector of objective coefficients. The problem is converted into an LP problem by adding two auxiliary variables for each reaction.

4.2.4 Mapping gene expression to reactions

The above description assumes that each reaction is catalyzed by only one enzyme. To account for more complicated GPR rules, the function *gene2Rule*() maps gene expression values to reactions. Gene expression levels are summarized across all genes that feature in a given GPR association; the piece-wise linear function introduced above is then fitted to this aggregated reaction expression level.

The rules used to account for complex GPR associations in FEcorr are as follows: for one to one relations, we directly use the flux value predicted from the expression level as outlined above. For protein complexes (GPR relations with only ANDs), the aggregated reaction expression level is assumed to be the minimum of all individual gene expressions. For isoenzymes (relations with only ORs), the aggregated reaction expression level is taken to be the sum of all gene expression levels of the constituting genes. For multifunctional enzymes, the resulting expression is divided by the number of reactions catalyzed by this enzyme; however, this option can be disabled because sometimes the number of repetitions is big and this makes the expression value very small. For complex GPR relations that include ANDs and ORs, the rules are applied recursively. To facilitate these recursions, GPR rules are expected in the canonical form termed Sum-of-products (SoP; in the language of logical circuits: AND to OR).

4.2.5 Applying FECorr

To test FECorr, I used a compendium of yeast (*S. cerevisiae*) experiments consisting of 170 microarrays under 55 conditions that vary 9 environmental parameters (Aeration type(2 types), Carbon source(5), Nitrogen source(7), Sulfur source(2), Limiting element(6), Growth rate(4), Temperature(2), pH(3), Protocol(2)) [87]. For the calculations, I used the metabolic model *iMM904* [67]. I used 11 cultivation conditions (Table 5), as the other conditions reflect environmental changes that cannot be modeled

using FBA (temperature, pH, etc.). The 11 conditions used differ in aeration type, limiting element, and carbon source.

Table 5 The conditions used in simulations

Abbreviation	Aeration	Limiting Nutrient	Carbon Source
AE-AC-C lim	Aerobic	Carbon	Acetate
AE-Etoh-C lim	Aerobic	Carbon	Ethanol
AE-GLC-C lim	Aerobic	Carbon	Glucose
AE-GLC-P lim	Aerobic	Phosphate	Glucose
AE-GLC-S lim	Aerobic	Sulfur	Glucose
AE-GAL-C lim	Aerobic	Carbon	Galactose
ANAE-GLC-C lim	Anaerobic	Carbon	Glucose
ANAE-GLC-N lim	Anaerobic	Nitrogen	Glucose
ANAE-GLC-P lim	Anaerobic	Phosphate	Glucose
ANAE-GLC-S lim	Anaerobic	Sulfur	Glucose
AE-GLC-N lim	Aerobic	Nitrogen	Glucose

Figure 11 illustrates the effect of FECorr for a single reaction, R_TPI, which is catalyzed by the product of a single gene, *YDR050C*. For the FBA solution returned by CPLEX, the Pearson correlation between fluxes and expression levels across conditions was R^2=22%.

With FECorr, this correlation was improved to R^2=80%.

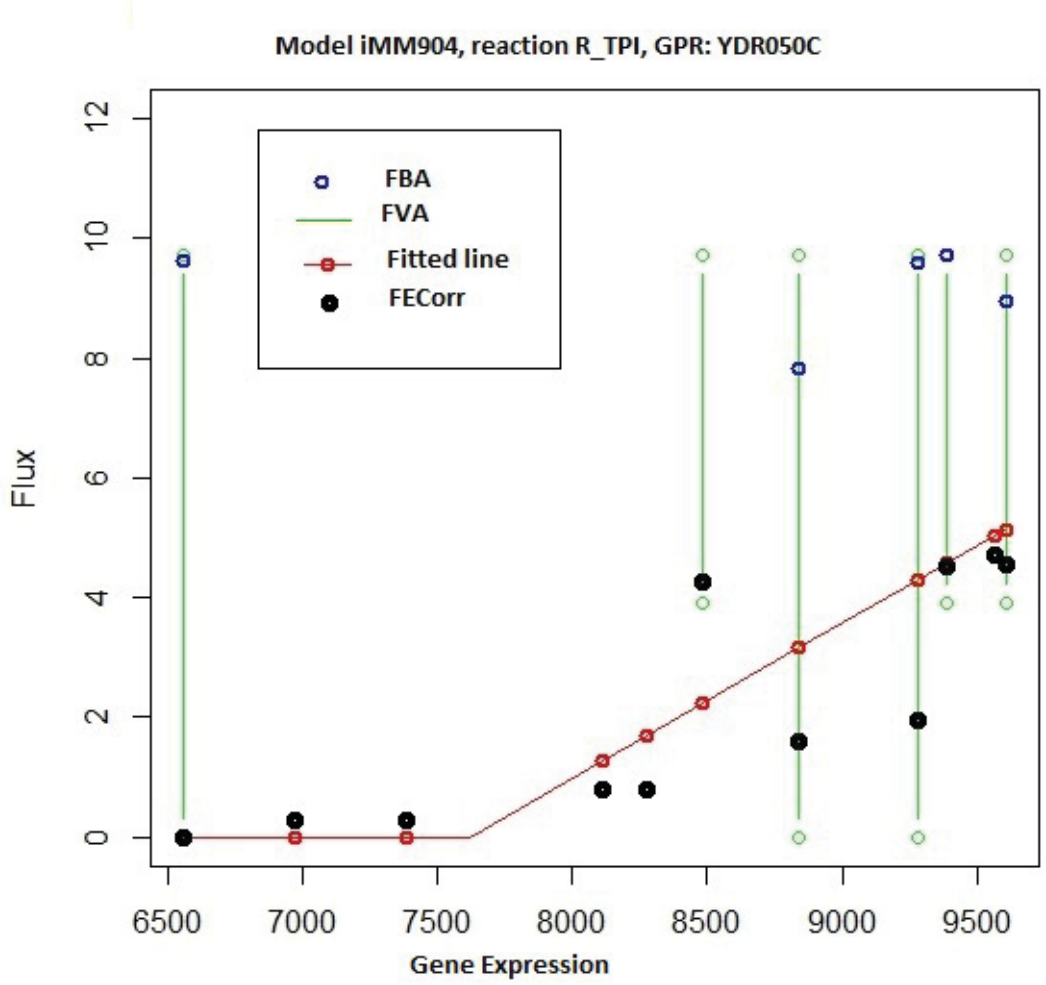

Figure 11 FECorr improves the correlation between flux and expression level for the R_TPI reaction from R^2=22% for FBA (blue circles) to R^2=80% for FECorr (black dots). The green bars show the flux variability in each condition. Red dots are the predictions from the piece-wise linear model.

Machado et al [80] evaluated different expression-based methods. They used in their comparison the normalized error, defined as the normalized Euclidean distance between experimentally determined and predicted flux vectors:

$$e = \frac{|| \boldsymbol{v}^{exp} - \boldsymbol{v}^{sim} ||}{||\boldsymbol{v}^{exp}||}$$

Here, $\boldsymbol{v}^{exp}$ is the vector of flux measurements, while the components of $\boldsymbol{v}^{sim}$ are the predicted (simulated) flux values. For a series of evaluations for the same method across different experiments, the average error is given by the mean of the error across experiments. Machado et al. applied their measure to data from three different experimental studies: those of Holm *et al.* from 2010 [88], Ishii *et al.* from 2007 [89], and Rintala *et al.* from 2009 [90].

The experimental setup of Holm et al [88] consisted of *E. coli* strains growing aerobically in batch cultures. The dataset contains measurements of genome scale gene expression using microarray analysis, as well as flux measurements from metabolic flux analysis (MFA), derived using ^{13}C-labelled metabolites . The dataset contains three wild-type *E. coli* (REF) as well as two over-expression mutants, one for NADH oxidase (nox) and one for the F1-ATPase atpAGD (ATP).

I used the same model iAF1260 [9] as used in Ref. [80]; as done for all methods, I applied the measured uptake of glucose consumption in each condition as a constraint and set all default uptake reactions in the model to ≥-1000. As shown in Figure 12, I found that application of FECorr results in a smaller average error (0.378) than all gene expression-based methods examined in Ref. [80], and performs slightly better than pFBA (which gives 0.385).

When I additionally constrained all measured uptake and excretion rates for each method, I found a similar picture (Figure 12B): FECorr predictions resulted in a somewhat smaller error (0.36) than pFBA (0.42); only the method of Lee *et al.* [84] resulted in smaller normalized errors than these two methods.

The experimental set up of Ishii et al[89] consists of E.coli strains growing aerobically in batch cultures. The dataset contains measurements of gene expression, protein level and metabolic fluxes under 5 growth conditions and 24 mutants.

The experimental set up of Rintala et al [90] consists of Saccharomyces cerevisiae growing in 5 different oxygen levels.

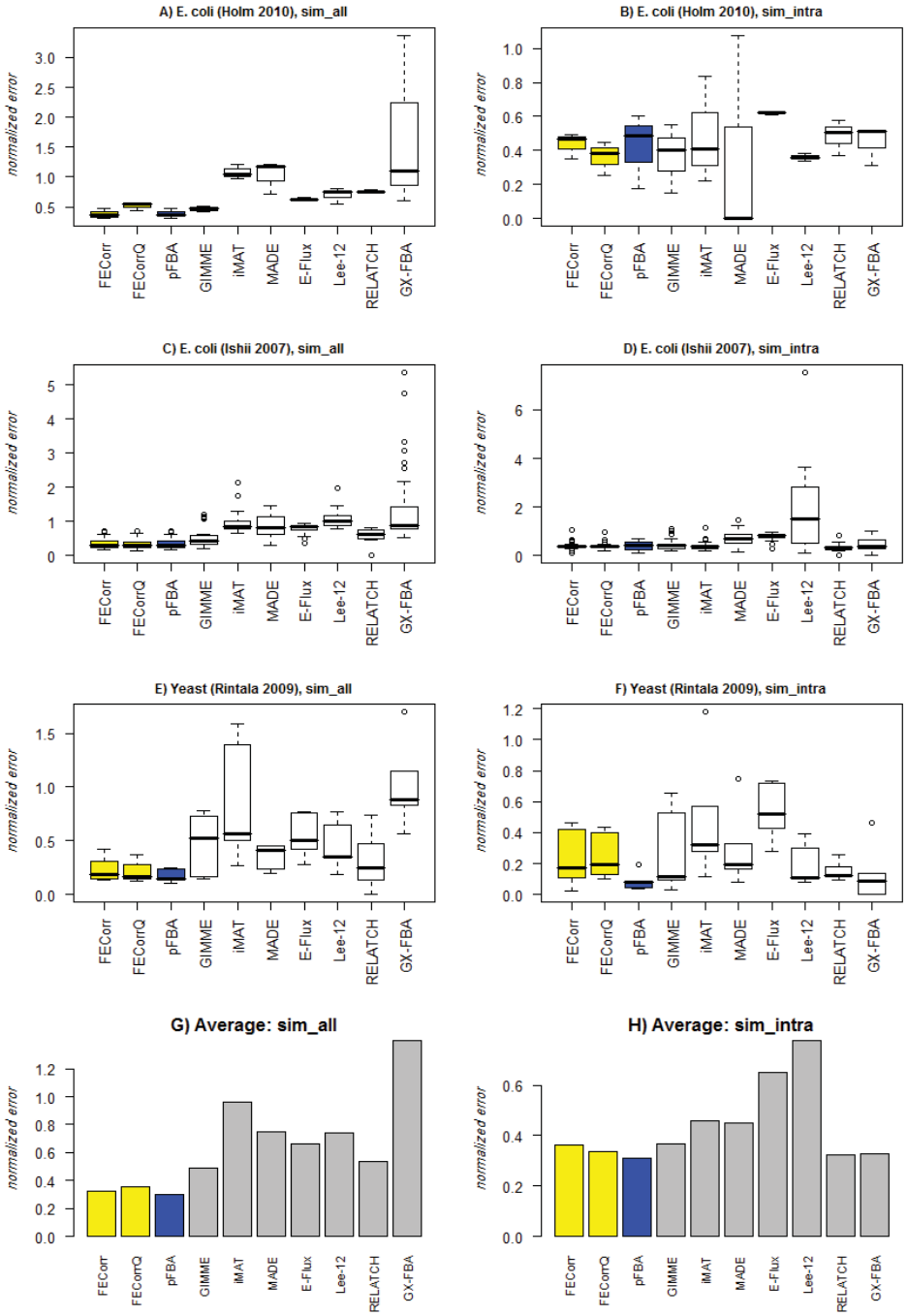

Figure 12 Boxplots of normalized prediction errors of different expression-based methods for the 3 datasets analyzed in Ref. [80]. (A). Data from Holm *et al.* [88], constraining only the glucose uptake rate to the measured value. (B) same data, but additionally constraining all other measured uptake and excretion rates. (C) Data from Ishii *et al.* [89], constraining only the glucose uptake rate to the measured value. (D) same data, but additionally constraining all other measured uptake and excretion rates. (E) Data from Rintala *et al.* [90], constraining only the glucose uptake rate to the measured value. (F) same data, but additionally constraining all other measured

uptake and excretion rates. (G) Average normalized prediction errors across the three datasets, constraining only the glucose uptake rate to the measured value. (H) Average normalized prediction errors when additionally constraining all other measured uptake and excretion rates.

As shown in Figure 12G and Figure 12H, FECorr performed substantially better on average than all other expression-based methods assayed by Machado *et al.* [80] when only constraining the glucose uptake rate. When constraining all measured uptake and excretion rates, two alternative methods (RELATCH [91], mean normalized prediction error E= 0.324, and GX-FBA [92], E= 0.329) performed slightly better than FECorr (E= 0.3299); however, the necessary experimental data is only rarely available. Strikingly, pFBA performs slightly better than all expression-based methods in both situations (E= 0.3022 when constraining only glucose uptake; E= 0.3095 when constraining all exchange fluxes).

Indeed, one of the main conclusions of Machado *et al.* [80] was that pFBA, which does not use any expression information, predicted the measured fluxes with a smaller normalized error than all expression-based methods. However, it is noteworthy that pFBA was the only method that was given explicit information on the knocked-out reactions, which were constrained to zero flux in pFBA [80]. This additional information given to just one of the assayed algorithms may have biased the comparison towards pFBA. Furthermore, the calculations of flux values in MFA rely on a metabolic model that is interpreted with pFBA [89]. This may bias experimental errors in a direction favorable for pFBA predictions.

4.3 Function ATMFBA

4.3.1 Scaling expression levels by k_{cat}

The expected relationship between flux values and enzyme expression level depends on the reaction kinetics. In particular, the maximal flux of a reaction is given by the enzyme turnover number k_{cat}multiplied with the enzyme expression level n_A, $v \leq k_{cat} n_A$. Thus, I also scale gene expression levels by the corresponding k_{cat} values. This is equivalent to having a separate expression level threshold for each gene.

Scaling the measured gene expression by k_{cat} improves the correlation between measured fluxes and gene expression across all reactions for each of the *E. coli* growth rates assayed in [89] (Figure 13A). It also substantially improves the correlation between fluxes and gene expression for individual reactions across conditions (Figure 13B). This is because a given concentration of enzymes with high k_{cat} values can catalyze high fluxes and vice versa. Figure 13B shows that Pearson correlation between flux and mRNA expression across the 5 conditions is >0.8 for almost all assayed reactions. This also suggests that eFBA methods that use information from more than one condition may give better predictions.

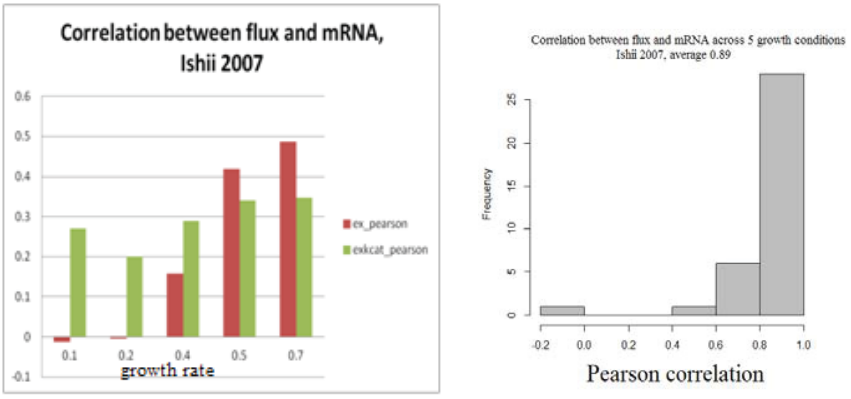

Figure 13 Correlations between measured fluxes and mRNA expression for *E. coli* [89] are improved by scaling with k_{cat}. (A) Correlation across different reactions for five different growth rates. (B) Histogram of the distribution of Pearson correlation coefficients calculated for individual reactions across the five growth rates.

4.3.2 Optimization of activity thresholds

Some methods use predefined thresholds to indicate if a reaction is active or not in the model (e.g., iMAT[93] uses a flux threshold of 1 mM/gDW/h) and another threshold to indicate if a given gene is active or not in the experiment (gene activity threshold). Gene activity thresholds are typically given as quantiles of the expression level distribution; sometimes, two thresholds with an intervening "undetermined" region are used [82]. Here, I introduce a method that finds the best thresholds, defined as those that minimize the deviation between GPR state and reaction state.

The optimization problem is formulated as a MILP problem, with the thresholds as additional variables, gene expression threshold *Tg* and flux threshold *Tf*. These two variables can be bounded to a predefined range given as input to the simulation. GPR rules are modeled as linear constraints with binary variables, where OR is modeled as a sum and AND is modeled as a minimum of two expressions.

The minimized objective function is the discrepancy between the GPR state and the flux state. There are two types of discrepancy. The first is when the GPR state is ON (gene expression is high) while the reaction rate is low (<*Tf*); this is called "unused expression discrepancy", and has an associated penalty *unusedExprPenalty*. The other type of discrepancy is when the reaction must carry significant flux while the measured expression of the GPR is lower than *Tg*. This is assigned a penalty *InsufficientExprPenalty*. As the unused expression discrepancy may be due to post-transcription regulation, I set *InsufficientExprPenalty* > *unusedExprPenalty*.

Error! Reference source not found. Figure 14 gives an overview over the ATM-FBA algorithm. The optimization problem is formulated as:

$$Minimize \sum [\ (GPRst < rxnSt) * InsufficentExprPenalty + (GPRst > rxnSt) * UnusedExprPenalty\ ]$$

$$Sv = 0$$
$$geneSt = (geneExpr <= Tg)$$
$$GPRst = f(geneSt)$$
$$rxnSt = (v <= Tf)$$

Because of overpredictions in reaction activities due to inaccuracies in expression measurements and posttranslational regulation, the solutions to this problem may contain spurious thermodynamically infeasible cycles. Thus, I apply the cyclefreeFBA algorithm to post-process the predicted flux distribution. Finally, I apply pFBA to the reactions with unknown expression state, i.e., I minimize the sum of absolute fluxes through these reactions while maintaining the post-processed flux prediction for reactions with known expression state.

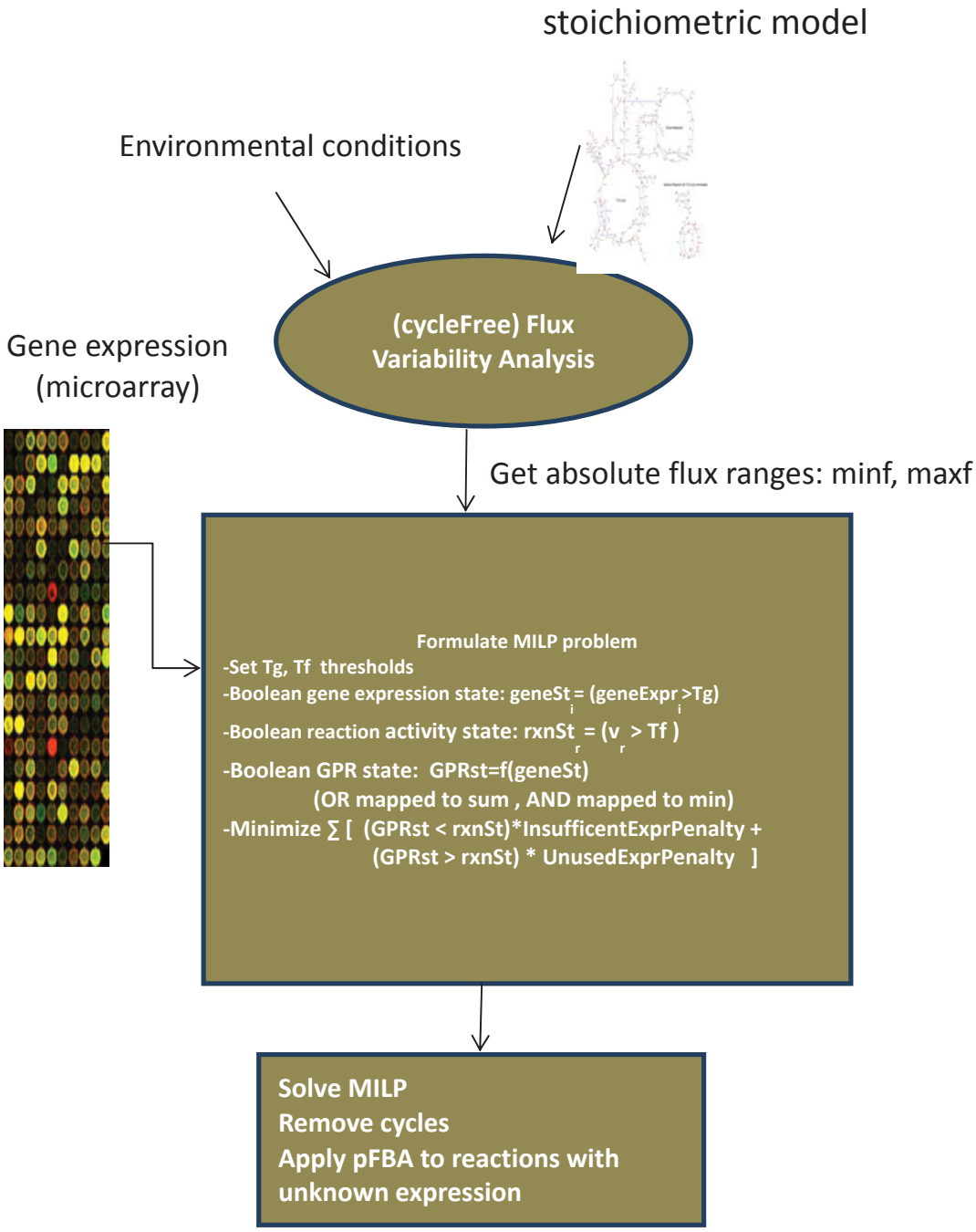

Figure 14 Overview of ATM-FBA

The problem is solved with the CPLEX solver in about 15 minutes on a standard laptop (with core i7 processor and 8 GB RAM) running Windows 7 for the *E. coli* genome-scale model iAF1260 [9].

4.3.3 Results

Comparing ATM-FBA with methods benchmarked in [80] shows that ATM-FBA performed slightly better than other expression based methods (**Error! Reference source not found.**); benchmarking was performed as detailed in Section 4.2.

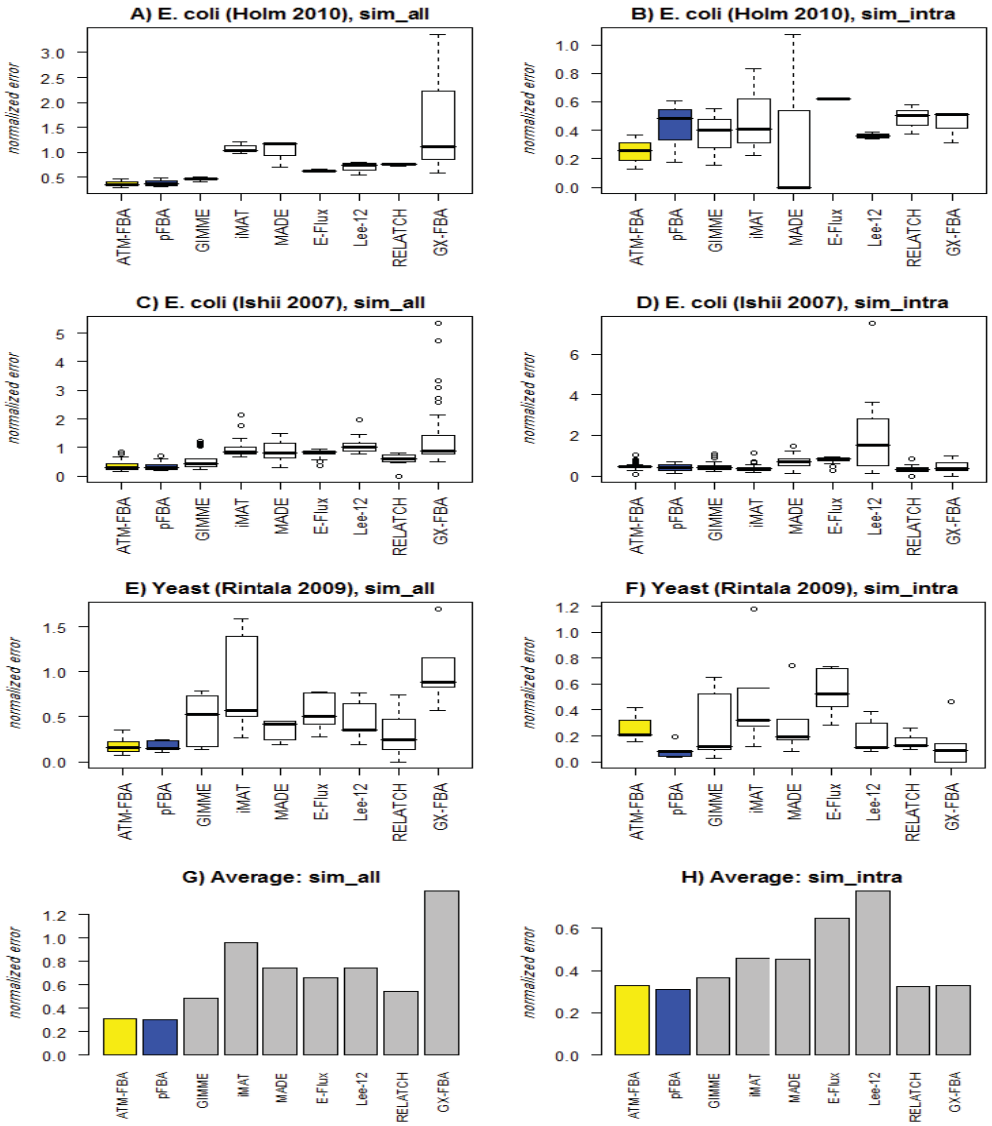

Figure 15 boxplots of normalized prediction errors of different expression-based methods for the 3 datasets analyzed in Ref. [80]. (A). Data from Holm *et al.* [88], constraining only the glucose uptake rate to the measured value. (B) same data, but additionally constraining all other measured uptake and excretion rates. (C) Data from Ishii *et al.* [89], constraining only the glucose uptake rate to the measured value. (D) same data, but additionally constraining all other measured uptake and excretion rates. (E) Data from Rintala *et al.* [90], constraining only the glucose uptake rate to the measured value. (F) same data, but additionally constraining all other measured uptake and excretion rates. (G) Average normalized prediction errors across the three datasets, constraining only the glucose uptake rate to the measured value. (H) Average normalized prediction errors when additionally constraining all other measured uptake and excretion rates.

Figure 16 shows measured excretion rates of different metabolites in comparison to predictions with the expression-based methods and pFBA. Both ATM-FBA and pFBA massively overpredict the excretion of acetate. They both slightly over-predict CO_2 excretion. ATM-FBA also wrongly predicts excretion of formate.

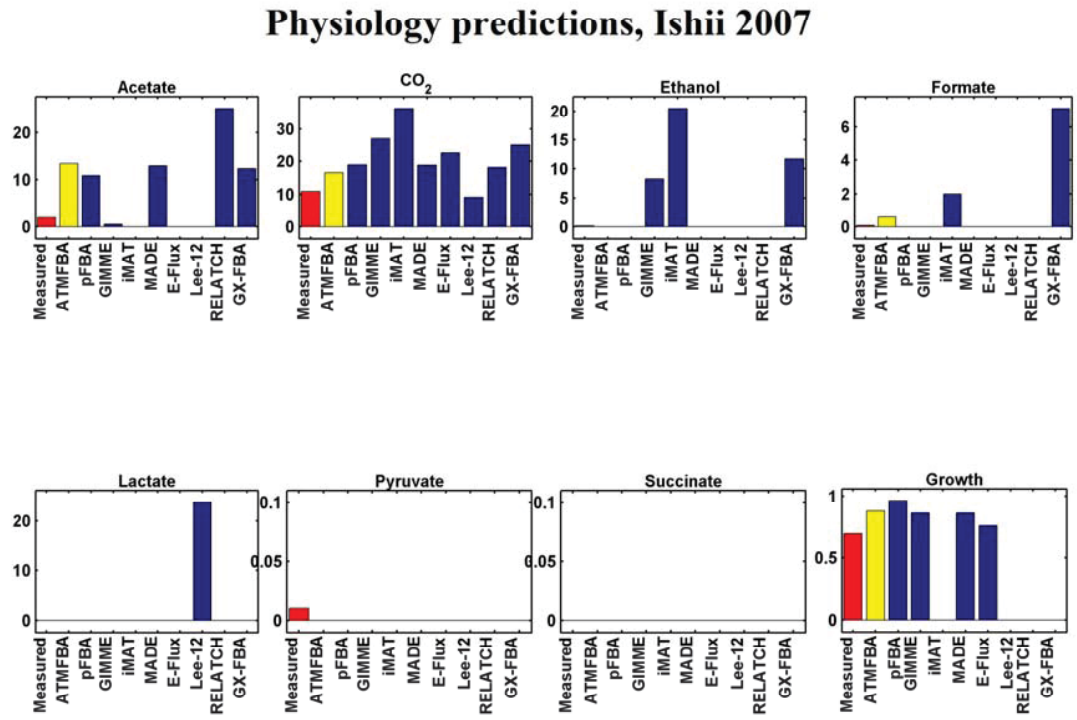

Figure 16 Measured (red) and predicted excretion rates of metabolites following [80].

4.4 eFBA-gene

The last of the three methods I propose in this thesis aims to optimize the agreement between individual gene mRNA levels with predictions rather than optimizing the agreement between reaction states and GPR states derived from the mRNA data. If all GPR-rules are 1-to-1 correspondences between genes and reactions, both approaches are mathematically identical. However, if enzymes catalyze multiple reactions or if a reaction is catalyzed by an enzyme complex, the results should differ from each other. As

expression-based methods aim to minimize the discrepancy between expression data and predictions, and as expression data is collected per gene, it seems appropriate to also focus the optimization on genes.

Thus, the aim of eFBA-gene is to select the solution that minimizes the conflict between the gene states implied by the resulting flux distribution and the measured gene expression. I consider three values of measured gene states: 1 for expressed; 0 for not expressed; and -1 for "don't care" (see below). The problem is formulated as a MILP problem with the following objective:

$$\text{Minimize } \sum_g (expr(g) \neq flux(g))$$

where g is a gene in the model; $expr(g) = 1$ if g is expressed , else 0; and $flux(g) = 0$ if all reactions catalyzed by g have no flux (more precisely: if the absolute fluxes of all reactions catalyzed by g are below the pre-set threshold T_f), else $flux(g) = 1$.

To formulate a new eFBA-gene problem, one needs to follow these steps:

1- add n new integer variables f_i, where n is the number of reactions to be considered. f_i, indicates if the corresponding reaction is carrying a significant flux (1) or is inactive (0).

2- add ng new integer variables (g_i) to represent predicted gene states, where ng is the number of genes in the model (g_i=1 if the gene is predicted to be inactive, g_i=0 if it is predicted to be active)

3- add new constraint for biomass (ensure a minimal fraction of the maximum biomass obtained from wildtype FBA)

4- Add identifying constraints for f_i's as follows:

for irreversible reactions add the following two constraints (with M a large number, *e.g.*, $M = 1000$):

Constraint 1: $v_i - Mf_i \leq T_f$

Constraint 2: $T_f\ f_i - v_i\ \leq 0$

For reversible reactions: include additional binary variable y_i and add the following four constraints:

Constraint 1: $v_i - Mf_i \leq T_f$

Constraint 2: $T_f\ f_i - v_i - My_i\ \leq 0$

Constraint 3: $-v_i - Mf_i\ \leq T_f$

Constraint 4: $T_f\ f_i + v_i - M(1 - y_i)\ \leq 0$

5- Add linear constraints to represent Boolean GPR rules (see below for details).

6- set new objective function if expression(gene *i*)=0 then add $+g_i$ to objective function
else if expression(gene *i*)=1 add $-g_i$; if expression(g_i)=-1 ignore g_i.

The problem and the constraint matrix:

minimize: $\sum(2ei-1)*g_i$

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

$$c^T\boldsymbol{v} \geq Z * pct\_obj$$

identifying constraints

Linear constraints of Boolean GPR

$$lb \leq \boldsymbol{v} \leq ub$$

where ei =0 if gene i is NOT expressed and 1 if gene i is expressed and 0.5 if ignored, Z^* is wildtype objective value, pct_obj is the percent of biomass to be achieved, and ub, lb are the wildtype problem upper bounds and lower bounds respectively.

To make the problem smaller and more efficient, we can select a subset of reactions for which gene states are optimized, ignoring all other reactions. FVA is calculated for the set of chosen reactions (or, if no subset has been selected, for all reactions with GPR rules). The reactions with fixed state flux are excluded (note that this strategy cannot reduce the problem size unless a fixed biomass production is demanded)

Then the Boolean GPR rules are converted to linear constraints in a similar way to [94]. I assume that the GPR rules are in disjunctive normal form(DNF) as SUM of PRODUCT (AND to OR), without NOT, one level. The following rules are used:

A= B AND C $\Leftrightarrow$ $0 \leq b+c-2a \leq 1$

In general, if

$$y = x_1 \wedge x_2 \wedge \ldots \wedge x_n$$

(straight ANDs), the linear constraint will be:

$$0 \leq -n\,y + \sum_i x_i \leq n - 1$$

And if

$$y = x_1 \vee x_2 \vee \ldots \vee x_n$$

(straight ORs), the linear constraint will be:

$$0 \leq ny - \sum_i x_i \leq n - 1$$

For complex rules, auxiliary integer variable are added for each term. For example, if the GPR rule is

(g1 AND g2) OR (g3 AND g4),

then two auxiliary variables are added:

Aux1= g1 AND g2

Aux2= g3 AND g4 ,

and then the result is

Aux1 OR Aux2.

To illustrate the difference between the eFBA-gene strategy and the alternative of considering reaction states, consider the three reactions (R1, R2, and R3) in Table 6. The two solutions Solution1 and Solution2 will be favored differently by the two schemes. In a reaction-centered approach, Solution1 (with 1 conflict) will be favored over Solution2 (with 2 conflicts). In eFBA-gene, Solution2 will be favored instead, as it affects more reactions, but fewer genes.

Table 6 Difference between eFBA_gene and the alternative of considering reaction states objective functions

Reaction	Rule	Solution 1	Solution 2	Gene State
R1	g1	ON	OFF	Present
R2	g2	ON	OFF	Present
R3	g3 & g4 & g5	OFF	ON	All Present
Conflicts(rxn)		**1**	2	
Conflicts(gene)		3	**2**	

Application to the Holm dataset

I applied eFBA_gene to the dataset of Holm [88] described above. Figure 17 shows a histogram of the gene expression levels. I tried different values for the gene expression threshold *Tg*. The minimum value of gene expression in this dataset is 2.23 and the maximum is *15.25*. Thus, at Tg≤2.23, all genes are considered to be expressed (ON), while at Tg≥15.25, all genes are considered not expressed (OFF). Note that considering all genes to be not expressed, e.g., by setting *Tg* =16 can be a good approximation, as it results in a solution with a minimal number of active reactions; this is very similar to requiring a minimum total flux, as done in pFBA.

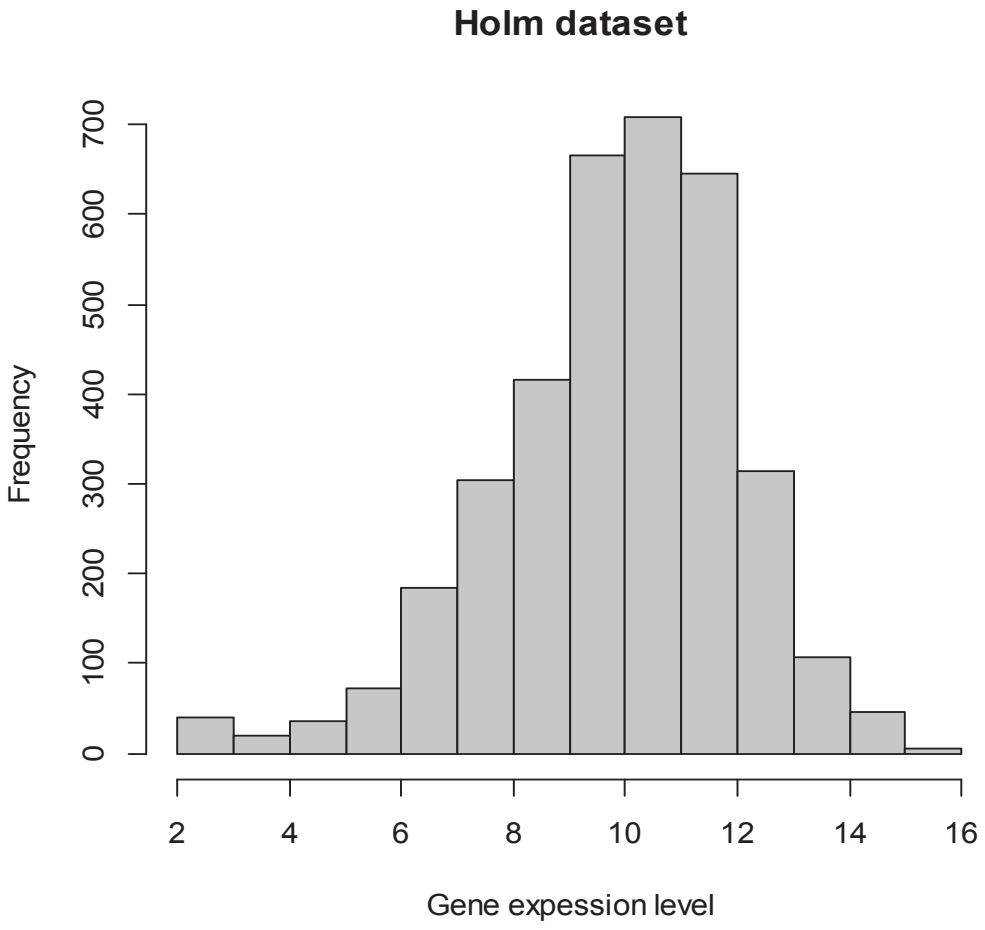

Figure 17 Histogram of gene expression levels from Holm dataset.

The second parameter that needs to be set for the application of eFBA-gene is the flux threshold (*Tf*) at which a reaction is considered to be active in a given simulation. Table 7 shows that the combination of values that gives the minimum normalized error for this dataset are Tg=12 and *Tf*=0.001. As expected, considering all genes to be not expressed

(Tg=16) also gives a good approximation. Comparing the results for the best parameter combination to those of other expression-based methods (see Figure 12 above), I find that eFBA-gene can deliver better results than pFBA and all other expression based methods that were applied to this dataset.

The results shows also that considering all genes to be not expressed(Tg=16) can be a good approximation as it is analogous to solution with minimum cost or minimum total flux. The best result below is better than pFBA and all expression based methods that were applied to this dataset.

Shlomi et al 2008 [95] used flux threshold (Tf) of 1.

Table 7 Applying eFBA_gene with different parameter values, normalized error of predicted and measured fluxes from Holm dataset

Tg\Tf	0.0001	0.001	0.01	0.1	1	% genes ON
2	0.308	6.472	6.142	14.683	26.512	100.00
3	0.371	8.825	0.524	6.337	0.495	98.88
4	0.697	8.824	0.524	6.337	0.495	98.32
5	6.310	0.317	14.681	0.277	4.456	97.31
6	0.402	0.296	8.848	6.130	4.457	95.26
7	12.252	0.279	0.286	0.288	12.992	90.07
8	12.157	6.143	0.288	0.294	4.454	81.51
9	0.280	0.301	0.277	12.155	6.139	69.84
10	12.119	0.273	0.26955	0.271	12.118	51.15
11	0.273	0.273	0.273	0.273	0.304	31.29
12	0.301	**0.269505**	0.297	0.270	0.310	13.19
13	0.301	0.269512	0.26956	0.270	0.289	4.38
14	0.297	0.297	0.297	0.296	0.291	1.40
15	0.297	0.297	0.297	0.297	0.289	0.14
16	0.297	0.297	0.297	0.296	0.290	0.00

Chapter 5 Discussion, Conclusion and future work

In this thesis, I proposed, implemented, and tested a broad range of algorithms for the improvement of constraint-based analysis of metabolic networks. These algorithms deal with three main topics: (1) calculating thermodynamically feasible FBA solutions; (2) applying cost/capacity concepts to FBA; and (3) using gene or protein expression data to improve FBA predictions. For each of these topics, I have implemented the proposed algorithms in an R package.

Each of the proposed algorithms represents a first step into a novel direction, and a number of further steps and improvements are conceivable. In this section, I will summarize my results, discuss the new methods in relationship to previous work, and outline the most important suggestions for next steps.

5.1 loopless FBA

5.1.1 Alternative methods to calculate thermodynamically feasible solutions

The CycleFreeFlux algorithm described in Chapter 2, as well as the two major alternative strategies, MTF [26] and ll-FBA [37], each put out one thermodynamically feasible flux distribution. However, these algorithms are not directly comparable, as they differ in both their goals and their inputs. MTF delivers very special flux distributions, which minimize the total sum of absolute fluxes $\| \boldsymbol{v} \|_1$, constrained only by the optimal value of the objective function. In contrast, CycleFreeFlux takes any given steady-state flux distribution (which may or may not be the result of an FBA calculation) and reduces it to its loopless contribution. Finally, ll-FBA solves an FBA problem such that the resulting flux distribution already lies in the thermodynamically feasible subspace. Given that the MTF flux distribution will often be biologically more realistic than many alternative solutions to an FBA problem [17], MTF (parsimonious FBA) should be the method of choice when just one thermodynamically feasible solution to an FBA problem is required. In contrast, CycleFreeFlux and ll-COBRA provide means to characterize the full space of thermodynamically feasible (optimal) steady-state solutions.

5.1.2 Runtime comparisons

CycleFreeFlux requires one additional linear optimization and is thus comparable in speed to MTF. In contrast, ll-FBA requires solving a MILP problem and thus has a much longer (and in practice unpredictable) runtime. The CycleFreeFlux functions for the generation of thermodynamically feasible random flux samples and flux variability analyses are thus orders of magnitude faster than previously available algorithms. The only exception is an independently developed strategy for thermodynamically feasible FVA that uses similar concepts to CycleFreeFVA [69]; we only became aware of this work after the publication of the CycleFreeFlux paper [61].

5.1.3 Biases introduced by CycleFreeFlux

CycleFreeFlux solutions are biased towards solutions that run in the same direction as internal cycles with which they overlap and towards solutions with lower total flux $\| \boldsymbol{v} \|_1$ whenever alternative decompositions into elementary modes exist; in many applications, these biases will not be important. It is noteworthy that other methods to calculate thermodynamically feasible solutions, such as pFBA and ll-COBRA, also introduce similar biases.

5.1.4 Conclusion

The CycleFreeFlux algorithm and its extensions allow the reliable identification and exclusion of thermodynamically infeasible internal cycles from different types of constraint-based analyses. Thus, it may improve predictions in any application that requires the estimation of flux ranges.

For example, O'Brien et al [96] used FVA to predict biological capabilities, and the use of CycleFreeFlux would improve such predictions. CycleFreeFlux could also be applied to Flux Coupling Analysis (FCA)[97]: this could improve results, as the existence of internal cycles can mask some coupling relations.

Among further possible applications of CycleFreeFlux is calculating ranges of fluxes for FECorr. Internal cycles would give thermodynamically infeasible ranges for the reactions involved, potentially resulting in overestimates of the flux-expression relationship. Another important application may be the estimation of maximal fluxes, e.g., to develop improved methods to predict flux distributions in knockout mutants: it was recently suggested to penalize relative rather than absolute flux ranges in MOMA (Guido Przygoda, personal communication); for reactions that are inactive in the wild-type, a maximal flux can be used instead. Here again, internal cycles could lead to overestimates of maximal fluxes.

We implemented functions for cycleFreeFlux calculations as an extension package for sybil [31], which are freely available from CRAN (http://cran.r-project.org/web/packages/sybilcycleFreeFlux/).

5.1.5 Future work

Currently, cycleFreeFVA is about 10 times slower than standard FVA as performed by Sybil. The performance of cycleFreeFVA can be further improved by using advanced features of linear solvers, such as calculating the basis of the linear problem once and using it in all the simulations, as the constraint matrix does not change between simulations. This can save the overhead of rebuilding the problem thousands of times. One way to implement this would be to perform the simulations using Sybil's lower level class sysBiolAlg instead of the modelorg class.

Currently, the enumerateCycle function is not capable of identifying all internal cycles in the *Homo sapiens* reconstruction Recon1 [70]. Progress may be achieved by adding a preprocessing step as proposed by Wright and Wagner [36], which reduced the model to a submodel that contains only reactions in nontrivial loops.

5.2 Cost-constrained FBA

5.2.1 A general framework to incorporate solvent capacity constraints into FBA

One of the major strengths of FBA is that it can predict many phenotypes correctly without requiring knowledge on enzyme kinetics. However, there are a range of metabolic phenomena that cannot be modeled in this framework, such as the Warburg effect observed in many cancer cell lines [46, 98] and the Crabtree effect in yeast cells grown on abundant glucose [45], or the evolution of crossfeeding in originally monoclonal bacterial populations [48]. Such phenomena are likely a result of compromises in proteome allocation due to the limited solvent capacity of the cell [99, 100]; their explanation requires the inclusion of enzyme kinetics and cellular volume (or concentration) constraints into FBA.

ccFBA (capacity-constrained flux balance analysis) allows to convert any complete FBA model into a model that incorporates enzyme kinetics and cellular constraints on total enzymatic capacity. Building on the algorithmic developments in [42, 43, 71], ccFBA facilitates the application of refined constraint-based analysis methods to metabolic systems beyond *E. coli*. The sybilccFBA package for sybil [31] is freely available from CRAN (http://cran.r-project.org/web/packages/sybilccFBA/).

A recent study demonstrated experimentally that overflow metabolism is indeed a consequence of proteome allocation constraints, but argued that this is due to optimal investment into protein production rather than constraints on protein volume such as those discussed in this thesis [101]. However, the reasoning leading to this conclusion appears flawed. Cellular volume is irrelevant for the calculation of growth rates as long as intracellular concentrations and biomass composition remain constant: a cell of half the size has half the amount of enzymes, which need to produce half the amount of biomass. Thus, the constraints used in MOMENT* and other approaches that account for the limited cellular volume are really constraints on concentrations, accounting for the limited solvent capacity of the cell [99, 100]. Furthermore, protein allocation constraints do not predict a maximal growth rate at unlimited nutrient concentrations, which contradicts experimental evidence.

5.2.2 Comparison of ccFBA to other algorithms that include solvent capacity constraints

The first application of amolecular crowding constraint to FBA, FBAwMC, did not include explicit GPR rules, in contrast to MOMENT and MOMENT*. ccFBA includes appropriate constraints for multifunctional enzymes, while this is not modeled correctly in MOMENT [71].

An alternative implementation of molecular crowding constraints is found in the ME (Metabolism-Expression) models of the Palsson group at UCSD [82, 102]. These models explicitly account for the translation and transcription machinery depending on the protein expression necessary to catalyze the active reactions. While powerful, ME models are much more complex, require more parameters, and need more time both for their reconstruction and for their solutions than the methods developed in this thesis.

5.2.3 Shortcomings of ccFBA and related approaches

A comparison between predicted and experimentally measured growth rates shows that all current *E. coli* models fail to predict the true breadth of utility of different carbon sources; this is also true for more complex models such as that in [102], unless the space available for enzymatic reactions and their activity are adjusted based on transcriptomic or proteomic data for each carbon source [76]. This indicates that not only reliable kinetic information is needed for many more reactions than currently available, but that we also need a better understanding of the selective forces that lead to the under-utilization of certain carbon sources such as galactose [103]. I expect that the prediction power of ccFBA models will grow substantially once reliable genome-wide estimates of k_{cat} become available, and once we achieve a quantitative understanding of the bet-hedging strategies that may underlie the under-utilization of unpreferred carbon sources.

5.2.4 Future work

The current model accounts for the limited intracellular solvent capacity. However, a similar limit exists for the protein density in cellular membranes, constraining the maximal density of transporters on the cellular surface [104]. E.g., at low substrate levels, *E. coli* invests a large amount of cellular resources into transporters, which is not the case at high substrate concentrations. Thus, extensions of MOMENT* should include the cost and the density constraints on transporters.

A major limitation of all methods that aim to incorporate protein production costs and/or solvent capacity constraints is the limited availability of kcat values. Some improvement may be achieved by not substituting the median kcat for all unknown values, but to instead use statistical techniques such as machine learning to predict kcat from protein and reaction properties, such as amino acid composition [105] and EC-numbers; first

steps in this direction have been performed [106], but further work is needed before such methods can be applied to genome-scale models.

The limited solvent capacity does not only constrain protein concentrations, but simultaneously constrains metabolite concentrations (Hugo Dourado, personal communication). Including full Michaelis-Menten kinetics and additional variables for metabolite concentrations may thus further improve the predictive capabilities of MOMENT* and related approaches.

So far, I only explored using an enzyme capacity constraint in FBA. However, the same constraint of course also affects other types of constraint-based algorithms. Thus, ccFBA should be extended to update a wide range of popular constraint-based analyses, such as MOMA [28] and FVA [27]. Furthermore, the efficient application of ccFBA would benefit from algorithmic optimizations such as implemented in the oneGeneDel() function of sybil [31].

Of particular practical importance might be a tool for curating cost-constrained models, targeted at researchers that build organism-specific models. For example, the metabolic network could be imported from the SBML format and stored in a local database. Each user might have an account to keep track of his or her projects. The tool will allow to run simulations using sybil and its derivatives. The tool also will store parameters of the network (k_{cat}, MW, Km,...). Models can be exported to JSON [107], SBML [4], or as CSV lists.

A major limitation to the application of cost-constrained models is the availability and reliability of kinetic parameters. Systematic efforts are necessary to curate curate the cost parameters for genome scale models such as the yeast model (iTO977). One would track the Kcat, Km, complex stoichiometry, and number of active sites from original publications, starting from a local database downloaded from BRENDA (2012) and/or SABIO [74] repositories. The parameters can be curated using sybilccFBA to calculate the minimal required protein expression in different growth conditions and checking if the corresponding enzymatic volume is realistic; for several reactions, I found that data obtained from BRENDA resulted in unrealistically large volumes required for individual reactions (data not shown). The resulting curated dataset will not only be useful for ccFBA, but can also be employed for other types of cost-constrained models such as ME-models [102].

Obvious avenues of further research are the application of sybilccFBA to make quantitative predictions for the wide range of phenomena that cannot be addressed with FBA, such as overflow metabolism (including the Crabtree and Warburg effects) and the evolution of crossfeeding in *E. coli.* All current cost-constrained methods make predictions for the growth of *E. coli* on Galactose and Oxoglutarate that differ

substantially from observed growth rates [103]; it would be highly interesting to examine if these discrepancies can be removed by correcting kinetic parameters in the model or by including metabolites into the solvent capacity constraint. Finally, ccFBA could be extended into a version of dynamic FBA [108] (implemented in sybilDynFBA) to simulate the progressive consumption of multiple sugars and compare the corresponding predictions with gene expression time series (Beg et al 2007).

5.3 Expression-based FBA

In recent years, many algorithms were proposed to achieve better FBA predictions using gene expression or other types of omics data. I proposed three methods to include transcriptomic or proteomic data to get better FBA predictions and implemented them in the sybilEFBA package for sybil [31], which is also available on CRAN (http://cran.r-project.org/web/packages/sybileFBA/).

5.3.1 FECorr: deriving quantitative flux-expression level estimates from data across experiments

In the first method, FECorr, I used quantitative expression data from multiple experiments across different conditions. The method tries to fit a linear relation between FBA-predicted fluxes and the corresponding gene expression data. I applied the method to a dataset that had previously been used to compare different expression based methods [80]. The results showed that my method gave a slight improvement over the previous methods, although, surprisingly, it was still not better than pFBA. It is known that gene expression data is noisy, and this may be one of the reasons why all methods based on such data perform poorly. Other problems in gene expression data include the necessary normalizations and the different sequence features of genes that affect the expression signals. Expression-based methods may give better results if proteomics data were used, but quantitative proteomic data is not widely available. Finally, mapping gene expression data on individual reactions is not an easy task because of complex relationships between genes and reactions, due to the existence of isoenzymes and multifunctional enzymes, as well as the formation of protein complexes.

5.3.2 ATM-FBA: automatic thresholding for reaction and protein activities

The second novel method is ATM-FBA. In this method the thresholds for gene expression and flux are variables that are optimized in a MILP problem. The method was applied to the benchmark dataset in Machado et al [80]. The solution is much slower than FECorr, which can be solved by linear programming. To increase the speed of ATM-FBA, a flux variability calculation is applied in a preprocessing step to exclude fixed reactions from the MILP and hence make the problem much smaller.

In other expression-based methods, thresholds to distinguish active from inactive enzymes and reactions are arbitrarily set by the user. This not only makes the solutions

dependent on subjective decisions, but also leads to sub-optimal predictions. This may explain at least in part the superior performance of ATM-FBA.

Genes with large k_{cat} will be effective when expressed at low amounts, while genes with small k_{cat} will have to be expressed at high levels to get significant flux. If we assume that all enzymes are saturated, then fluxes should equal enzyme expression level multiplied by $k_{cat.}$ Thus, scaling gene expression by k_{cat} values makes expression levels for different genes comparable.

5.3.3 eFBA-gene: reconciling gene rather than reaction activities with expression data

The third novel method introduced in this thesis that utilizes gene expression data is eFBA-gene. It differs from previous methods, including those proposed in this thesis, by penalizing disagreement between experiment and predictions based on individual genes rather than based on individual reactions. To this end, the GPR rules were converted to linear constraints in a MILP problem with the objective to find the point in the solution space that is closest to the input gene expression data in terms of gene states. When the method was applied to the benchmark dataset, it gave results that were better than pFBA and other gene expression methods in terms of the normalized error between predicted and measured fluxes when optimal values were chosen for the gene and reaction activity thresholds. However, choosing the appropriate thresholds may be difficult. The predicted fluxes range from 1e-6 to about 100, and it appears thus unlikely that using a single threshold for all reactions is appropriate.

Furthermore, the distinction between expressed and non-expressed genes may be too coarse. Many proteins may be expressed at low levels even when the corresponding reactions are inactive [76], e.g., because of post-transcriptional regulation or as a bet-hedging strategy in case preferred carbon sources become available. In some species, including *Saccharomyces cerevisiae*, it is hard to find genes that are not expressed at least at low levels [63].

5.3.4 Alternative strategies to utilize expression data in FBA predictions

Compared to other methods, FECorr utilizes information from multiple conditions, comparing the expression of the same gene under different conditions. This approach controls for the fact that different enzymes have different kinetics, and thus the relationship between gene expression and flux varies between them. The disadvantage of FECorr is that it cannot be applied to single conditions. However, my comparison between different expression-based methods shows that when expression data from multiple conditions is available, FECorr performs better than alternative methods and should thus be preferred.

In contrast, ATM-FBA combined with a scaling of gene expression by k_{cat} can be applied to single conditions and also copes with the different kinetics of enzymes. It uses binary gene expression states and is not quantitative. This can be perceived as a shortcoming; however, in my experience disagreement between expression and reaction activity is mostly binary (i.e., a reaction is predicted to be active although the enzyme is not expressed or vice versa), and thus a binary expression state already contains most of the important information. ATM-FBA performed better than previous methods based on gene expression in single conditions, likely both because of the expression level scaling and because of the automated selection of optimal cutoffs. The correlation between mRNA expression and enzyme abundance typically has an R^2 of only around 50%. Thus, protein expression levels should show a much stronger correlation to reaction rates than mRNA data. Accordingly, I expect that ATM-FBA will perform even better when utilizing proteomics data.

Scaling gene expression by k_{cat} as a preprocessing step essentially results in gene-specific cutoffs for expression. I propose that this strategy may improve the accuracy of all existing methods that use expression data to predict reaction activities, except those that already use reaction-specific parameters (determined, e.g., from expression data across multiple conditions).

Possibly the most surprising result of the method comparisons is the excellent performance of pFBA, which often gave the smallest error in the benchmarked dataset. However, pFBA cannot always be applied. In the benchmark study, pFBA results were based on inactivating the reactions associated with the knocked-out gene. Similar strategies are not possible if the aim is to predict flux distributions across different tissues of multicellular organisms or in tumor cells. Here, expression-based methods - in particular those described in this work - are likely to be far superior to pFBA. pFBA is based on the assumption that flux distribution with lower total flux require lower amounts of enzymes. Thus, pFBA corresponds to MOMENT* under the assumption that all reactions have the same k_{cat} and all enzymes have the same molecular weight. It thus appears likely that using MOMENT* instead of pFBA may lead to even better results for the prediction of gene knockout flux distributions.

5.3.5 Future work

The algorithms in this section were described for and applied to transcriptomic data obtained with microarray experiments. However, the algorithms can be applied equally to RNA-Seq data and proteomics data. Proteomics data in particular may provide better results, as reaction activities will correlate more strongly with protein abundance than mRNA abundance, and as the correlation between mRNA and protein expression levels is weak and not well understood.

It is also conceivable to use metabolomics data to improve the predictions of expression-based algorithms. If a metabolite is found to be present in the cell, then at least one reaction producing it must be active. Thus, metabolomics data can be integrated using the same mathematical frameworks as used in ATM-FBA and eFBA_gene.

Posttranscriptional regulation of enzymes may distort the predictions of expression-based methods, as mRNA and proteins may be present for an enzyme, but the enzyme may be unable to catalyze the associated reaction. One way to account for posttranscriptionally regulated enzymes would be to treat them separately, such that their abundance is only used to infer an upper bound for the reaction activity in FECorr, or that the inactivity of the associated reaction is not penalized even when the protein is expressed in ATM-FBA and eFBA-gene.

The ideas implemented in ATM-FBA and eFBA-gene could be merged, such that ATM-FBA penalizes disagreement between predictions and experiments based on individual genes rather than reactions.

Besides these general additions, there are also specific algorithmic changes for the individual methods proposed here. In FECorr, one could apply the minimum disjunction algorithm suggested by [109] to avoid problems that may appear from direct substitution of AND with minimum and OR with sum in evaluating the GPR rules (e.g., to correctly parse GPRs of the form $((A \wedge B) \vee (A \wedge C))$). In ATM-FBA, we could use two thresholds for gene expression data, such that intermediate expression levels are considered uninformative, such as done in the GIMME methods [82]. Both ATM-FBA and eFBA-gene might benefit from reaction-specific thresholds that could be based on a fixed percentage of the maximal flux according to CycleFreeFVA. eFBA-gene could further be improved through gene-specific thresholds implemented by scaling gene expression data with kcat, as done in ATM-FBA. The same scaling might also improve other expression-based methods [20, 63, 82-84].

On a more technical side, the functions `findMDCFlux()` and `eFBA_gene`() in the sybilEFBA package use a switch statement to select different solvers, which was the original style of extending Sybil. However, Sybil now uses the `sysBiolAlg` class for this purpose, and thus new solvers added to Sybil will not be recognized by the functions of sybilEFBA. Thus, the corresponding functions should be modified to use the `sysBiolAlg` class. More generally, it would be beneficial to also implement the methods introduced in this thesis in other popular frameworks for constraint-based analysis, such as the COBRA [30] and COBRApy [29] frameworks.

The data used for the benchmarking of different expression-based models is only of limited utility for this purpose for two reasons. First, they are based on microarray data, which cannot be converted unambiguously to gene expression status or protein

abundance. Second, it is based on gene knockout experiments. The maximization of biomass production, which is employed in all expression-based methods, is only biologically relevant under the assumption that natural selection had enough time to optimize the metabolic network usage under the assayed conditions. This is not the case for gene knockouts, and the relationship between gene expression and reaction activity may thus be strongly distorted in these experiments. More suitable benchmarking data would instead be based on wild-type growth on growth media typical for the evolution of the strains under study.

The methods proposed in this section are not only useful for the prediction of metabolic network usage in microbial growth experiments. An important application will be the identification of context-specific metabolic sub-models for the cells of different tissues in multicellular organisms based on gene expression data from these tissues [82].

Appendix A

Table 8 non-trivial loops in iAF1260 model

sn	reactions in loop	loop length
1	R_ABUTt2pp,R_GLUABUTt7pp,R_GLUt2rpp	3
2	R_ACCOAL,R_PPAKr,R_PPCSCT,R_PTA2,R_SUCOAS	5
3	R_ACKr,R_ACS,R_ADK1,R_PPKr,R_PTAr	5
4	R_ACt2rpp,R_ACt4pp,R_NAt3pp	3
5	R_ACt2rpp,R_ACt4pp,R_CA2t3pp,R_CAt6pp	4
6	R_ADK1,R_ADK3,R_NDPK1	3
7	R_ADK1,R_ADK3,R_NDPK1,R_PPM,R_PRPPS,R_R15BPK,R_R1PK	7
8	R_ACKr,R_ACS,R_ADK3,R_NDPK1,R_PPKr,R_PTAr	6
9	R_ALATA_L,R_VALTA,R_VPAMT	3
10	R_CA2t3pp,R_CAt6pp,R_SERt2rpp,R_SERt4pp	4
11	R_CA2t3pp,R_CAt6pp,R_PROt2rpp,R_PROt4pp	4
12	R_CA2t3pp,R_CAt6pp,R_GLYCLTt2rpp,R_GLYCLTt4pp	4
13	R_CRNDt2rpp,R_CRNt2rpp,R_CRNt8pp	3
14	R_GLBRAN2,R_GLCP2,R_GLCS1,R_GLDBRAN2,R_GLGC,R_PPKr	6
15	R_GLCP,R_GLCS1,R_GLGC,R_PPKr	4
16	R_GLBRAN2,R_GLCP2,R_GLCS1,R_GLGC,R_PPKr	5
17	R_GLUt2rpp,R_GLUt4pp,R_NAt3pp	3
18	R_GLYCLTt2rpp,R_GLYCLTt4pp,R_NAt3pp	3
19	R_HPYRI,R_HPYRRx,R_TRSARr	3
20	R_NAt3pp,R_SERt2rpp,R_SERt4pp	3
21	R_NAt3pp,R_PROt2rpp,R_PROt4pp	3
22	R_ACCOAL,R_PPAKr,R_PTA2	3
23	R_ACCOAL,R_PPCSCT,R_SUCOAS	3
24	R_ADK1,R_PPM,R_PRPPS,R_R15BPK,R_R1PK	5
25	R_ACKr,R_ACS,R_PPKr,R_PPM,R_PRPPS,R_PTAr,R_R15BPK,R_R1PK	8
26	R_NAt3pp,R_THRt2rpp,R_THRt4pp	3

Table 9 non-trivial loops in iMM904 loops

sn	reactions in loop	loop length
1	R_3MOPtm,R_ASPTA,R_ASPTAm,R_ASPt2m,R_ILETA,R_ILETAm,R_ILEtmi, R_OAAt2m	8
2	R_4HGLSDm,R_PHCDm,R_PHCHGSm	3
3	R_ACALDtm,R_ALCD2ir,R_ALCD2irm,R_ALCD2x,R_ETOHtm,R_MALtm, R_MDH,R_MDHm,R_OAAt2m,R_PIt2m	10
4	R_ACALDtm,R_ALCD2if,R_ALCD2irm,R_ETOHtm,R_FRDcm,R_FUM, R_FUMm,R_H2Otm,R_MDH,R_MDHm,R_OAAt2m,R_PIt2m,R_SUCCtm, R_SUCD1m	14
5	R_ACONT,R_ACONTm,R_CITtcm	3
6	R_ACt2r,R_ACtr,R_PTRCt3i,R_PTRCtex2	4
7	R_ACt2r,R_ACtr,R_GLYCt,R_GLYCt2	4
8	R_ACt2r,R_ACtr,R_PYRt,R_PYRt2	4
9	R_ADK1,R_ADK3,R_NDPK1	3
10	R_ADK1,R_ADK4,R_NDPK9	3
11	R_ADK1,R_ADK3,R_ADK4,R_NDPK1,R_NDPK9	5
12	R_ADK3,R_ADK4,R_NDPK1,R_NDPK9	4
13	R_AKGMAL,R_AKGt2r,R_MALt2r	3
14	R_ACALDtm,R_ALCD2if,R_ALCD2ir,R_ALCD2irm,R_ETOHtm,R_FRDcm ,R_FUM,R_FUMm,R_H2Otm,R_MALtm,R_MDH,R_MDHm,R_OAAt2m,R_PIt2 m, R_SUCCtm,R_SUCD1m	16
15	R_ACALDtm,R_ALCD2ir,R_ALCD2irm,R_ALCD2x,R_ETOHtm,R_FRDcm, R_FUM,R_FUMm, R_H2Otm,R_MALtm,R_MDH,R_MDHm,R_OAAt2m,R_PIt2m,R_SUCCtm, R_SUCD2_u6m, R_SUCD3_u6m	17
16	R_ACALDtm,R_ALCD2if,R_ALCD2irm,R_ETOHtm,R_MALtm,R_MDH, R_MDHm,R_OAAt2m,R_PIt2m	9
17	R_ALCD2if,R_ALCD2ir,R_ALCD2x	3
18	R_ASPt2n,R_ASPt5n,R_CO2tn,R_H2Otn,R_HCO3E,R_HCO3En,R_HCO3tn	7
19	R_CYSTGL,R_SHSL1,R_SHSL4r	3
20	R_CYTK2,R_DCMPDA,R_DCTPD,R_NDPK6,R_NDPK7,R_URIDK2r	6
21	R_D_LACt2m,R_D_LACtm,R_PYRt2m	3
22	R_FRDcm,R_FUM,R_FUMm,R_H2Otm,R_MALtm,R_SUCCtm,R_SUCD2_u6m, R_SUCD3_u6m,R_SUCFUMtm	9
23	R_FRDm,R_SUCD1m,R_SUCD2_u6m,R_SUCD3_u6m	4
24	R_FRDcm,R_FUM,R_FUMm,R_H2Otm,R_MALtm,R_SUCCtm,R_SUCD2_u6m, R_SUCD3_u6m	8
25	R_FRDcm,R_FUM,R_FUMm,R_H2Otm,R_MALtm,R_SUCCtm,R_SUCD1m	7
26	R_G6PI,R_G6PI3,R_PGI	3
27	R_GALT,R_GALU,R_UGLT	3

28	R_GK1,R_GK2,R_NDPK8	3
29	R_ACALDtm,R_ALCD2irm,R_ALCD2x,R_ETOHtm,R_MALtm,R_MDH, R_MDHm,R_OAAt2m,R_PIt2m	9
30	R_GLYCt,R_GLYCt2,R_PTRCt3i,R_PTRCtex2	4
31	R_PTRCt3i,R_PTRCtex2,R_PYRt,R_PYRt2	4
32	R_ACt2r,R_ACtr,R_GLYCt,R_GLYCt2,R_PTRCt3i,R_PTRCtex2	6
33	R_ACt2r,R_ACtr,R_PYRt,R_PYRt2,R_SPMDt3i,R_SPMDtex2	6
34	R_SUCD1m,R_SUCD2_u6m,R_SUCD3_u6m	3
35	R_ACALDtm,R_ALCD2irm,R_ALCD2x,R_ETOHtm,R_FRDcm,R_FUM, R_FUMm,R_H2Otm,R_MALtm,R_MDH,R_MDHm,R_OAAt2m,R_PIt2m, R_SUCCtm, R_SUCD1m	15
36	R_FRDcm,R_SUCD1m,R_SUCFUMtm	3
37	R_FRDcm,R_FUM,R_FUMm,R_H2Otm,R_MALtm,R_SUCCtm,R_SUCD1m, R_SUCD2_u6m, R_SUCD3_u6m	9
38	R_FRDm,R_SUCD2_u6m,R_SUCD3_u6m	3
39	R_FRDcm,R_SUCD2_u6m,R_SUCD3_u6m,R_SUCFUMtm	4
40	R_FRDcm,R_SUCD1m,R_SUCD2_u6m,R_SUCD3_u6m,R_SUCFUMtm	5

References

1. Stephanopoulos G. AAA, Nielsen J.: *Metabolic engineering—Principles and methodologies.* San Diego, CA: Academic Press; 1998.
2. Palsson BO: *Systems Biology: Properties of Reconstructed Networks.* New York: Cambridge University Press; 2006.
3. Aurich MK, Paglia G, Rolfsson Ó, Hrafnsdóttir S, Magnúsdóttir M, Stefaniak MM, Palsson BØ, Fleming RMT, Thiele I: **Prediction of intracellular metabolic states from extracellular metabolomic data.** *Metabolomics* 2014, **11:**603-619.
4. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, and the rest of the SF, Arkin AP, Bornstein BJ, Bray D, et al: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.** *Bioinformatics* 2003, **19:**524-531.
5. Thiele I, Palsson BØ: **A protocol for generating a high-quality genome-scale metabolic reconstruction.** *Nature protocols* 2010, **5:**93-121.
6. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M: **KEGG for integration and interpretation of large-scale molecular data sets.** *Nucleic acids research* 2012, **40:**D109-114.
7. Thorleifsson SG, Thiele I: **rBioNet: A COBRA toolbox extension for reconstructing high-quality biochemical networks.** *Bioinformatics* 2011, **27:**2009-2010.
8. Schilling CH, Thakar R, Travnik E, Dien SV, Wiback S: **SimPheny™: A Computational Infrastructure for Systems Biology.** *US Department of Energy, Genomic Science Program publications* 2005**:**67-86.
9. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BO: **A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information.** *Mol Syst Biol* 2007, **3:**121.
10. Caspi R, Altman T, Dale JM, Dreher K, Fulcher CA, Gilham F, Kaipa P, Karthikeyan AS, Kothari A, Krummenacker M, et al: **The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.** *Nucleic acids research* 2010, **38:**D473-479.
11. Pharkya P, Nikolaev EV, Maranas CD: **Review of the BRENDA Database.** *Metabolic Engineering* 2003, **5:**71-73.
12. Ganter M, Bernard T, Moretti S, Stelling J, Pagni M: **MetaNetX.org: a website and repository for accessing, analysing and manipulating metabolic networks.** *Bioinformatics* 2013, **29:**815-816.
13. Orth JD, Thiele I, Palsson BO: **What is flux balance analysis?** *Nature biotechnology* 2010, **28:**245-248.
14. Yoan N: *Flux Balance Analysis.* Miss Press; 2012.
15. Feist AM, Herrg\rrd MJ, Thiele I, Reed JL, Palsson BO: **Reconstruction of Biochemical Networks in Microbial Organisms.** *Nature Reviews Microbiology* 2009, **7:**129-143.
16. Feist AM, Palsson BO: **The biomass objective function.** *Current Opinion in Microbiology* 2010, **13:**344-349.
17. Schuetz R, Zamboni N, Zampieri M, Heinemann M, Sauer U: **Multidimensional optimality of microbial metabolism.** *Science* 2012, **336:**601-604.
18. Schuetz R, Kuepfer L, Sauer U: **Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli.** *Molecular systems biology* 2007, **3:**119.

19. Jamshidi N, Palsson BO: **Investigating the metabolic capabilities of Mycobacterium tuberculosis H37Rv using the in silico strain iNJ661 and proposing alternative drug targets.** *BMC Syst Biol* 2007, **1:**26.
20. Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng T-Y, Moody DB, Murray M, Galagan JE: **Interpreting expression data with metabolic flux models: predicting Mycobacterium tuberculosis mycolic acid production.** *Plos Computational Biology* 2009, **5:**e1000489-e1000489.
21. Plata G, Hsiao T-L, Olszewski KL, Llinás M, Vitkup D: **Reconstruction and flux-balance analysis of the Plasmodium falciparum metabolic network.** *Molecular systems biology* 2010, **6:**408-408.
22. Folger O, Jerby L, Frezza C, Gottlieb E, Ruppin E, Shlomi T: **Predicting selective drug targets in cancer through metabolic networks.** *Mol Syst Biol* 2011, **7:**501.
23. Pál C, Papp B, Lercher MJ: **An integrated view of protein evolution.** *Nature reviews Genetics* 2006, **7:**337-348.
24. Pál C, Papp B, Lercher MJ: **Adaptive evolution of bacterial metabolic networks by horizontal gene transfer.** *Nature genetics* 2005, **37:**1372-1375.
25. Barve A, Wagner A: **A latent capacity for evolutionary innovation through exaptation in metabolic systems.** *Nature* 2013, **500:**203-206.
26. Holzhütter H-G: **The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks.** *The Federation of European Biochemical Societies Journal* 2004, **271:**2905-2922.
27. Mahadevan R, Schilling CH: **The effects of alternate optimal solutions in constraint-based genome-scale metabolic models.** *Metabolic Engineering* 2003, **5:**264-276.
28. Segre D, Vitkup D, Church GM: **Analysis of optimality in natural and perturbed metabolic networks.** *Proc Natl Acad Sci U S A* 2002, **99:**15112-15117.
29. Ebrahim A, Lerman JA, Palsson BO, Hyduke DR: **COBRApy : COnstraints-Based Reconstruction and Analysis for Python.** 2013.
30. Becker SA, Feist AM, Mo ML, Hannum G, Palsson BO, Herrgard MJ: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox.** *Nat Protoc* 2007, **2:**727-738.
31. Gelius-Dietrich G, Desouki AA, Fritzemeier CJ, Lercher MJ: **sybil - Efficient constraint-based modelling in R.** *BMC Systems Biology* 2013, **7:**125.
32. Zanghellini J, Ruckerbauer DE, Hanscho M, Jungreuthmayer C: **Elementary flux modes in a nutshell: properties, calculation and applications.** *Biotechnology journal* 2013, **8:**1009-1016.
33. Orth JD, Thiele I, Palsson BØ: **What is flux balance analysis?** *Nature biotechnology* 2010, **28:**245-248.
34. Price ND, Famili I, Beard DA, Palsson B: **Extreme Pathways and Kirchhoff's Second Law.** *Biophysical journal* 2002, **83:**2879-2882.
35. Schilling CH, Letscher D, Palsson BO: **Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective.** *Journal of theoretical biology* 2000, **203:**229-248.
36. Wright J, Wagner A: **Exhaustive identification of steady state cycles in large stoichiometric networks.** *BMC Systems Biology* 2008, **2:**61-61.
37. Schellenberger J, Lewis NE, Palsson BØ: **Elimination of thermodynamically infeasible loops in steady-state metabolic models.** *Biophysical journal* 2011, **100:**544-553.

38. De Martino D, Capuani F, Mori M, De Martino A, Marinari E: **Counting and Correcting Thermodynamically Infeasible Flux Cycles in Genome-Scale Metabolic Networks.** *Metabolites* 2013, **3:**946-966.
39. Flamholz A, Noor E, Bar-Even A, Liebermeister W, Milo R: **Glycolytic strategy as a tradeoff between energy yield and protein cost.** *Proceedings of the National Academy of Sciences of the United States of America* 2013, **110:**10039-10044.
40. Scott M, Mateescu EM, Zhang Z, Hwa T: **Interdependence of Cell Growth Origins and Consequences.** *Science* 2010, **330:**1099-1102.
41. Dekel E, Alon U: **Optimality and evolutionary tuning of the expression level of a protein.** *Nature* 2005, **436:**588-592.
42. Beg QK, Vazquez a, Ernst J, de Menezes Ma, Bar-Joseph Z, Barabási a-L, Oltvai ZN: **Intracellular crowding defines the mode and sequence of substrate uptake by Escherichia coli and constrains its metabolic activity.** *Proceedings of the National Academy of Sciences of the United States of America* 2007, **104:**12663-12668.
43. Goelzer A, Fromion V, Scorletti G: **Cell design in bacteria as a convex optimization problem ☆.** *Automatica* 2011, **47:**1210-1218.
44. Schuster S, Pfeiffer T, Fell DA: **Is maximization of molar yield in metabolic networks favoured by evolution?** *Journal of theoretical biology* 2008, **252:**497-504.
45. Crabtree H: **The carbohydrate metabolism of certain pathological overgrowths. .** *Biochem J* 1928, **22:**1289–1298.
46. Hsu PP, Sabatini DM: **Cancer cell metabolism: Warburg and beyond.** *Cell* 2008, **134:**703-707.
47. Vemuri GN, Altman E, Sangurdekar DP, Khodursky AB, Eiteman MA: **Overflow metabolism in Escherichia coli during steady-state growth: transcriptional regulation and effect of the redox ratio.** *Applied and environmental microbiology* 2006, **72:**3653-3661.
48. Sebastian Bonhoeffer Thomas Pa: **Evolution of Cross-Feeding in Microbial Populations.** *The American Naturalist* 2004, **163:**E126-E135.
49. R Core Team: *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing; 2013.
50. **Comprehensive R Archive Network (CRAN)** [http://cran.r-project.org]
51. **glpkAPI** [http://CRAN.R-project.org/package=glpkAPI]
52. **cplexAPI** [http://CRAN.R-project.org/package=cplexAPI]
53. **clpAPI** [http://CRAN.R-project.org/package=clpAPI]
54. **lpSolveAPI** [http://CRAN.R-project.org/package=lpSolveAPI]
55. **GNU Linear Programming Kit (GLPK)** [http://www.gnu.org/software/glpk/]
56. **IBM ILOG CPLEX** [https://www.ibm.com/developerworks/university/academicinitiative/]
57. **COIN OR Clp** [https://projects.coin-or.org/Clp]
58. **lp_solve** [http://lpsolve.sourceforge.net/5.5/index.htm]
59. **Gurobi** [http://www.gurobi.com]
60. Bornstein BJ, Keating SM, Jouraku A, Hucka M: **LibSBML: an API library for SBML.** *Bioinformatics* 2008, **24:**880-881.
61. Desouki AA, Jarre F, Gelius-dietrich G, Lercher MJ: **CycleFreeFlux : efficient removal of thermodynamically infeasible loops from flux distributions.** *Bioinformatics* 2015, **31:**2159-2165.
62. Papin Ja, Price ND, Palsson BØ: **Extreme pathway lengths and reaction participation in genome-scale metabolic networks.** *Genome research* 2002,

12:1889-1900.
63. van Berlo RJP, de Ridder D, Daran J-M, Daran-Lapujade PaS, Teusink B, Reinders MJT: **Predicting metabolic fluxes using gene expression differences as constraints.** *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM* 2011, **8:**206-216.
64. Beard DA, Liang S-d, Qian H: **Energy balance for analysis of complex metabolic networks.** *Biophysical journal* 2002, **83:**79-86.
65. Jonathan M. Borwein ASL: *Convex Analysis and Nonlinear Optimization.* Second edition edn: Springer; 2006.
66. Schellenberger J, Park JO, Conrad TM, Palsson BØ: **BiGG : a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions Database.** *BMC bioinformatics* 2010, **11:**213-213.
67. Mo ML, Palsson BO, Herrgård MJ: **Connecting extracellular metabolomic measurements to intracellular flux states in yeast.** *BMC Systems Biology* 2009, **3:**37-37.
68. Bordel S, Agren R, Nielsen J: **Sampling the solution space in genome-scale metabolic networks reveals transcriptional regulation in key enzymes.** *Plos Computational Biology* 2010, **6:**e1000859-e1000859.
69. Müller AC, Bockmayr A: **Fast thermodynamically constrained flux variability analysis.** *Bioinformatics (Oxford, England)* 2013, **29:**903-909.
70. Duarte NC, Becker SA, Jamshidi N, Thiele I, Mo ML, Vo TD, Srivas R, Palsson BO: **Global reconstruction of the human metabolic network based on genomic and bibliomic data.** *Proc Natl Acad Sci U S A* 2007, **104:**1777-1782.
71. Adadi R, Volkmer B, Milo R, Heinemann M, Shlomi T: **Prediction of Microbial Growth Rate versus Biomass Yield by a Metabolic Network with Kinetic Parameters.** *Plos Computational Biology* 2012, **8**.
72. Muller S, Regensburger G, Steuer R: **Enzyme allocation problems in kinetic metabolic networks: optimal solutions are elementary flux modes.** *Journal of theoretical biology* 2014, **347:**182-190.
73. Schellenberger J, Park JO, Conrad TM, Palsson BO: **BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions.** *BMC bioinformatics* 2010, **11:**213.
74. Wittig U, Kania R, Golebiewski M, Rey M, Shi L, Jong L, Algaa E, Weidemann A, Sauer-Danzwith H, Mir S, et al: **SABIO-RK--database for biochemical reaction kinetics.** *Nucleic acids research* 2012, **40:**D790-796.
75. Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, Palsson BO: **A comprehensive genome-scale reconstruction of Escherichia coli metabolism--2011.** *Mol Syst Biol* 2011, **7:**535.
76. Palsson BO: **Genome-Scale Models Can Compute Proteome Allocation.** In *Book Genome-Scale Models Can Compute Proteome Allocation* (Editor ed.^eds.). City; 2015.
77. Mo ML, Palsson BO, Herrgard MJ: **Connecting extracellular metabolomic measurements to intracellular flux states in yeast.** *BMC Syst Biol* 2009, **3:**37.
78. Richter C: **Kosten und Effizienz von Enzymen als zusaetzliche Bedingung fuer die Flussverteilung in metabolischen Netzwerken.** *Diploma Thesis.* Humboldt University, Institute for Biology; 2011.
79. Hoppe A, Richter C, Holzhütter H-G: **Enzyme maintenance effort as criterion for the characterization of alternative pathways and length distribution of isofunctional enzymes.** *Bio Systems* 2011, **105:**122-129.
80. Machado D, Herrgård M: **Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism.** *Plos*

Computational Biology 2014, **10:**e1003580-e1003580.
81. Kim MK, Lun DS: **Methods for integration of transcriptomic data in genome-scale metabolic models.** *Computational and Structural Biotechnology Journal* 2014, **11:**59-65.
82. Becker Sa, Palsson BO: **Context-specific metabolic networks are consistent with experiments.** *Plos Computational Biology* 2008, **4:**e1000082-e1000082.
83. Jensen Pa, Papin Ja: **Functional integration of a metabolic network model and expression data without arbitrary thresholding.** *Bioinformatics (Oxford, England)* 2011, **27:**541-547.
84. Lee D, Smallbone K, Dunn WB, Murabito E, Winder CL, Kell DB, Mendes P, Swainston N: **Improving metabolic flux predictions using absolute gene expression data.** *BMC Systems Biology* 2012, **6:**73-73.
85. Heavner BD, Smallbone K, Barker B, Mendes P, Walker LP: **Yeast 5 - an expanded reconstruction of the Saccharomyces cerevisiae metabolic network.** *BMC Syst Biol* 2012, **6:**55.
86. Akesson M, Forster J, Nielsen J: **Integration of gene expression data into genome-scale metabolic models.** *Metab Eng* 2004, **6:**285-293.
87. Knijnenburg TA, Daran JM, van den Broek MA, Daran-Lapujade PA, de Winde JH, Pronk JT, Reinders MJ, Wessels LF: **Combinatorial effects of environmental parameters on transcriptional regulation in Saccharomyces cerevisiae: a quantitative analysis of a compendium of chemostat-based transcriptome data.** *BMC genomics* 2009, **10:**53.
88. Holm AK, Blank LM, Oldiges M, Schmid A, Solem C, Jensen PR, Vemuri GN: **Metabolic and transcriptional response to cofactor perturbations in Escherichia coli.** *The Journal of biological chemistry* 2010, **285:**17498-17506.
89. Ishii N, Nakahigashi K, Baba T, Robert M, Soga T, Kanai A, Hirasawa T, Naba M, Hirai K, Hoque A, et al: **Multiple high-throughput analyses monitor the response of E. coli to perturbations.** *Science (New York, NY)* 2007, **316:**593-597.
90. Rintala E, Toivari M, Pitkänen J-p, Wiebe MG, Ruohonen L, Penttilä M: **metabolism in Saccharomyces cerevisiae.** 2009, **19:**1-19.
91. Kim J, Reed JL: **RELATCH: relative optimality in metabolic networks explains robust metabolic and regulatory responses to perturbations.** *Genome Biology* 2012, **13:**R78-R78.
92. Navid A, Almaas E: **Genome-level transcription data of Yersinia pestis analyzed with a new metabolic constraint-based approach.** *BMC Systems Biology* 2012, **6:**150-150.
93. Yizhak K, Benyamini T, Liebermeister W, Ruppin E, Shlomi T: **Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model.** *Bioinformatics (Oxford, England)* 2010, **26:**i255-260.
94. Shlomi T, Eisenberg Y, Sharan R, Ruppin E: **A genome-scale computational study of the interplay between transcriptional regulation and metabolism.** *Mol Syst Biol* 2007, **3:**101.
95. Shlomi T, Cabili MN, Herrgard MJ, Palsson BO, Ruppin E: **Network-based prediction of human tissue-specific metabolism.** *Nature biotechnology* 2008, **26:**1003-1010.
96. O'Brien EJ, Monk JM, Palsson BO: **Using Genome-scale Models to Predict Biological Capabilities.** *Cell* 2015, **161:**971-987.
97. Burgard AP, Nikolaev EV, Schilling CH, Maranas CD: **Flux Coupling Analysis of Genome-Scale Metabolic Network Reconstructions.** *Genome research* 2004, **14:**301-312.

98. Shlomi T, Benyamini T, Gottlieb E, Sharan R, Ruppin E: **Genome-scale metabolic modeling elucidates the role of proliferative adaptation in causing the Warburg effect.** *PLoS Comput Biol* 2011, **7:**e1002018.
99. Atkinson DE: **Limitation of metabolite concentrations and the conservation of solvent capacity in the living cell.** *Current topics in cellular regulation* 1969, **1:**29-43.
100. Schuster S, Heinrich R: **Minimization of intermediate concentrations as a suggested optimality principle for biochemical networks.** *Journal of mathematical biology* 1991, **29:**425-442.
101. Basan M, Hui S, Okano H, Zhang Z, Shen Y, Williamson JR, Hwa T: **Overflow metabolism in Escherichia coli results from efficient proteome allocation.** *Nature* 2015, **528:**99-104.
102. O'Brien EJ, Lerman Ja, Chang RL, Hyduke DR, Palsson BØ: **Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction.** *Molecular systems biology* 2013, **9:**693.
103. Haverkorn van Rijsewijk BR, Nanchen A, Nallet S, Kleijn RJ, Sauer U: **Large-scale 13C-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in Escherichia coli.** *Mol Syst Biol* 2011, **7:**477.
104. Zhuang K, Vemuri GN, Mahadevan R: **Economics of membrane occupancy and respiro-fermentation.** *Molecular systems biology* 2011, **7:**500.
105. Yan S: **Prediction of Michaelis-Menten Constant in Beta-Cellobiosidase's Reaction with Lactoside as Substrate.** *Enzyme Engineering* 2012, **01**.
106. Borger S, Liebermeister W, Klipp E: **Prediction of enzyme kinetic parameters based on statistical learning.** *Genome Informatics* 2006, **17:**80-87.
107. Crockford D: **The application/json media type for javascript object notation (json).** 2006.
108. Varma A, Palsson BO, Varma A, Palsson BO: **Stoichiometric Flux Balance Models Quantitatively Predict Growth and Metabolic By-Product Secretion in Wild-Type Escherichia coli W3110.** *Appl Environ Microbiol* 1994, **60:**3724-3731.
109. Barker B, Sadagopan N, Wang Y, Smallbone K, Myers CR, Xi H, Locasale JW, Gu Z: **A robust and efficient method for estimating enzyme complex abundance and metabolic flux from expression data.** *arXiv preprint arXiv:14044755* 2014.