Automation, miniaturization, and parallelization of isotopic labeling experiments for the advanced analysis of microbial systems

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

M. Sc. Jochen Nießer
aus
Stadtlohn

Berichter: Prof. Dr. rer. nat. Wolfgang Wiechert
Prof. Dr.-Ing. Lars Blank

Tag der mündlichen Prüfung: 30.01.2025

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

"To understand something is not to be able to define it or describe it. Instead, taking something that we think we already know and making it unknown thrills us afresh with its reality and deepens our understanding of it."
Kenya Hara, Designing Design

"Man wird hier lediglich die Beschreibung eines geistigen Gebrechens im Reinzustand vorfinden."
Albert Camus, Der Mythos des Sisyphos

Eidesstattliche Erklärung

Ich, Jochen Nießer, erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt:

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
3. Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
4. Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
5. Alle wesentlichen Quellen von Unterstützung wurden benannt;
6. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
7. Teile dieser Arbeit wurden zuvor veröffentlicht. Die Auflistung ist auf den nachfolgenden Seiten zu finden.

Jochen Nießer
Aachen, 2025

Abstract

The generation and optimization of bioprocesses and strains for industrial application as well as the investigation of fundamental biological research hypotheses require adequate phenotyping experiments. Generally, there is a trade-off between informativeness and experimental throughput which became ever more relevant as both the creation of genetic diversity and the cultivation of mutant strain variants were increasingly accelerated. Isotopic labeling experiments are located at the extreme of high informativeness and low throughput with the additional limitation of significant associated costs per experiment. Commonly, they are conducted in lab-scale bioreactors, shaking flasks, and as the result of recent advances in mini-bioreactors at a scale ranging from liters to milliliters.

In the present dissertation, an automated, miniaturized, and parallelized experimental setup taking advantage of modern liquid handling robots and microbioreactors is established and validated. The development of an automated quenching method for this workflow enables the analysis of labeling patterns from free amino acids and intermediates of the central carbon metabolism, even at a microliter scale. It is then embedded into an overarching integrated pipeline for isotopic labeling experiments and applied to biological case studies. In order to realize such a pipeline, multiple Python programs are constructed and most notably the open source package `PeakPerformance` using an innovative peak fitting approach by Bayesian inference is developed and utilized for the evaluation of chromatographic peak data.

For the first application study, a novel bioprocess modelling approach for estimating intracellular metabolite pool sizes based on ^{13}C-labeling data is developed and demonstrated in *Corynebacterium glutamicum*. Thereby, the pool sizes of multiple amino acids the synthesis pathways of which are branching from the glycolysis were identified with a relatively high certainty.

For the second study, the first ever automated isotopically non-stationary ^{13}C-metabolic flux analysis is conducted at an unprecedented microliter scale to elucidate the fluxome of the evolved strain *C. glutamicum* WT_EtOH-Evo grown on ethanol as the sole carbon source. Since no fluxome of *C. glutamicum* grown exclusively on ethanol had been published prior, new insight regarding the pertaining pathway usage was generated, in particular an increased glyoxylate shunt activity compared to other substrates entering the central carbon metabolism via acetyl-CoA.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Wolfgang and Stephan for the opportunity to conduct this PhD thesis during which I learned so much and grew as a human being – or so I like to think. The exceptionally great working atmosphere at the IBG-1 and the guidance by Stephan in particular have contributed significantly to the successful finalization of my doctoral project. It has to be attributed to the fantastic structure and personnel of the institute that despite starting during the early phase of the COVID-19 pandemic in an unfamiliar working environment, I felt well supported throughout my time. Next, I would like to thank some specific colleagues who have contributed to my work in various ways.

Michael who has driven and motivated me to pick up new skills and push through many initially quite intimidating projects which all came to a successful close. Thanks for all the advice, help, and inspiration over the years!

My main collaborator in flux-related business, Anton, for the fruitful cooperation and tireless work on pushing ^{13}C-MFA forward. Naturally, my thanks extends to every member of our monthly ^{13}C coffee break, especially Katharina and Martin. Here, many insightful conversations were had during which I learned more than I can possibly name about isotopic labeling experiments and also that coffee is to be drunk black with 3 g 80 % 1-^{13}C and 20 % U^{13}C D-glucose.

Further, I am eternally grateful to Alexander for his tutelage concerning all things LC-MS/MS, especially in times when the device was acting out what I can only refer to as a refusal to work.

Yet another person who deserves recognition for helping me out with her tremendous experience regarding bioanalytical devices is Bianca.

I am grateful for the jolly cooperation with my one and only Master's student Tobias who has excelled despite a difficult topic and achieved much more than one could reasonably expect for a Master's thesis. I am confident you can accomplish anything you set your mind to.

Furthermore, I thank Mo for his great work on automated quenching methods setting the stage for my doctoral project.

A big thanks goes out to all members past and present of the Microphen and Biopro groups who are some of the finest people I have ever met and I can truly say it has been a pleasure and an honor to work with and amongst you.

Last but not least I am grateful for the support of my family – especially my parents – and my friends who have had my back all the way.

I would particularly distinguish a certain group of friends who have stuck together since the beginning of the Bachelors and have enriched my life ever since. You know who you are and I would ask you to feel hugged.

Published Work

Publications

- Nießer, J.; Müller, M. F.; Kappelmann J.; Wiechert, W.; Noack, S.: Hot isopropopanol quenching procedure for automated microtiter plate scale ^{13}C-labeling experiments. *Microbial Cell Factories* 2022;21.
- Nießer, J.; Osthege, M.; von Lieres, E.; Wiechert, W.; Noack, S.: PeakPerformance - A tool for Bayesian inference-based fitting of LC-MS/MS peaks. *Journal of Open Source Software* 2024;9.

In preparation

- Nießer, J.*; Stratmann, A.*; Beyß, M.; Wiechert, W.; Nöh, K.; Noack, S.: Automated and parallel isotopically non-stationary MFA of evolved *C. glutamicum* strain on ethanol. *equal contributions by these authors
- Beyß, M.; Nießer, J.; Noack, S.; Nöh, K.: Meta-tool for Streamlining the Natural Abundance Correction of Mass Isotope Data.

Code repositories

- Nießer, J.; Osthege, M.: `PeakPerformance`: A Python toolbox for Bayesian inference of peak areas. https://doi.org/10.5281/zenodo.10255543.
- Osthege, M.; Helleckes, L. M.; Nießer, J.; Halle, L.; Noack, S.; Kosonocky, C.; Müller, C.; Steier, V.: `robotools`: Pythonic in-silico liquid handling and creation of Tecan FreedomEVO worklists. https://doi.org/10.5281/zenodo.4697605.
- Osthege, M.; Tenhaef, N.; Noack, S.; Helleckes, L. M.; Nießer, J.; Reiter, A.; Geinitz, B.; Hamel, R.; Halle, L.; Müller, C.; Schito, S.: `bletl`: Parsing and data analysis with BioLector microbioreactor data. https://doi.org/10.5281/zenodo.5101434.

Poster presentations

- Nießer, J.; Müller, M. F.; Wiechert, W.; Noack, S. Miniaturization and automation of isotopically non-stationary (INST) labeling experiments for advanced microbial phenotyping. PhD workshop and Fall Meeting Topic 7: Towards a Sustainable Bioeconomy – Resources, Utilization, Engineering and AgroEcosystems, November 2022, Leipzig, Germany.
- Nießer, J.; Müller, M. F.; Latour, T.; Wiechert, W.; Noack, S. Automated and miniaturized ^{13}C-isotopic labeling experiments at microtiter plate-scale. Himmelfahrtstagung on Bioprocess Engineering 2023 - Novel production routes and processes for bio-pharmaceuticals and industrial bioeconomy, May 2023, Weimar, Germany.

- Nießer, J.; Müller, M. F.; Latour, T.; Wiechert, W.; Noack, S. Automated and miniaturized ^{13}C-isotopic labeling experiments at microtiter plate-scale. Society for Laboratory Automation and Screening (SLAS) Europe 2023, May 2023, Brussels, Belgium.

Contents

List of Figures

List of Tables

List of Equations

List of Listings

Nomenclature

AC	alternating current
ANSI	American National Standards Institute
BPMN	Business Process Model and Notation
CCM	central carbon metabolism
CDW	cell dry weight
CE	collision energy
CER	carbon dioxide evolution rate
CID	collision-induced dissociation
CoI	confidence interval
cps	counts per second
CrI	credible interval
CRM	charged residue model
csv	comma-separated value
cumomer	cumulative isotopomer
DAE	differential-algebraic system of equations
DAG	directed acyclical graph
DC	direct current
DCS	DigInBio process control system
DDW	double-distilled water
DO	dissolved oxygen
DoE	design of experiments
DOF	degrees of freedom
DP	declustering potential
DWP	deep well plate
ECS	experimental control script

ED	Entner-Doudoroff
EI	electron impact
EIC	extracted ion chromatogram
elpd	expected log pointwise predictive density
EMP	Embden-Meyerhof-Parnas
EMU	elementary metabolite unit
ESI	electrospray ionization
ess	effective sample size
exp	export
FAB	fast atom bombardment
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
GC	gas chromatography
GRAS	generally recognized as safe
GUI	graphical user interface
HDI	highest density interval
HEX	hexadecimal
HILIC	hydrophilic interaction liquid chromatography
HPLC	high pressure liquid chromatography
IEM	ion evaporation model
IEX	ion exchange
ILE	isotopic labeling experiment
ISS	isotopic steady-state
IUPAC	International Union of Pure and Applied Chemistry
INST	isotopically non-stationary
isotopomer	isotope isomer
lb	lower bound

LC-MS/MS	liquid chromatography tandem mass spectrometry
LiHa	liquid handling
LOO-PIT	Pareto-smoothed importance sampling leave-one-out cross-validation
MCMC	Markov Chain Monte Carlo
MCP	microchannel plate
MFA	metabolic flux analysis
ML	machine learning
MLE	maximum likelihood estimator
MOPS	3-(N-morpholino)propanesulfonic acid
MQ	MultiQuant
MRM	multiple reaction monitoring
MS	mass spectrometry
MSS	metabolic steady-state
MTP	microtiter plate
m/z ratio	mass to charge ratio
negLL	negative log-likelihood
NMR	nuclear magnetic resonance
ODE	ordinary differential equation
OD_{600}	optical density at 600 nm
PBS	phosphate buffer saline
png	portable network graphics
posterior	posterior probability distribution
PP	PeakPerformance
ppc	posterior predictive check
PPP	pentose phosphate pathway
prior	prior probability distribution
PTS	phosphotransferase system

Q	quadrupole
QqQ	triple quadrupole
QqTOF-MS	quadrupole-time-of-flight tandem MS
rDOF	remaining degrees of freedom
RF	radiofrequency
RoMa	robotic manipulator
RQ	respiratory quotient
SBML	Systems Biology Markup Language
sd	standard deviation
SSR	weighted sum of squared residuals
svg	scalable vector graphics
TCA	tricarboxylic acid
TMID	tandem mass isotopomer distribution
TOF	time-of-flight
ub	upper bound
upt	uptake
UV	ultraviolet
WAIC	widely applicable information criterion
WT	wild type

Metabolites and enzymes

AcCoA	acetyl-Coenzyme A
Ace	acetate
AceA	isocitrate lyase
AceB	malate synthetase
Adh	alcohol dehydrogenase
AckA	acetate kinase

AKG	α-ketoglutarate
Ala	L-alanine
Aldh	acetaldehyde dehydrogenase
ATP	adenosine triphosphate
Cit	citrate
Citr	L-citrulline
FBP	fructose-1,6-bisphosphate
Fum	fumarate
GAP	glyceraldehyde 3-phosphate
Glc	D-glucose
Gln	L-glutamine
Glu	L-glutamate
Glx	glyoxylate
Gly	L-glycine
GTP	guanosine-5'-triphosphate
G6P	glucose-6-phosphate
Hser	L-homoserine
His	L-histidine
Icit	isocitrate
Ile	L-isoleucine
KIV	α-ketoisovalerate
Leu	L-leucine
Lys	L-lysine
Mal	malate
MalE	malic enzyme
MQH_2	menaquinol
NADH	nicotinamide adenine dinucleotide (reduced form)

NADPH	nicotinamide adenine dinucleotide phosphate (reduced form)
OAA	oxaloacetate
Odx	oxaloacetate decarboxylase
Orn	L-ornithine
Pyc	pyruvate carboxylase
PEP	phosphoenolpyruvate
Pck	phosphoenolpyruvate carboxykinase
Ppc	phosphoenolpyruvate carboxylase
Pgd	6-phosphogluconate dehydrogenase
Phe	L-phenylalanine
Pta	phosphotransacetylase
Pyr	pyruvate
Ser	L-serine
Suc	succinate
S7P	sedoheptulose 7-phosphate
THF	tetrahydrofolate
Trp	L-tryptophane
Val	L-valine
XR5P	collective 5-carbon pool in pentose phosphate pathway

1 Introduction

1.1 The principle of isotopic labeling experiments

Isotopes as defined by International Union of Pure and Applied Chemistry (IUPAC) recommendations are "nuclides having the same atomic number but different mass numbers" [1], i.e. elements with an identical number of protons but a varying number of neutrons in their core. A selection of isotopes relevant to the field of isotopic labeling experiments (ILEs) may include but not be limited to ^{13}C, ^{14}C, ^{18}O, ^{2}D, and ^{15}N. These heavy isotopes can be observed in nature as a minuscule fraction of their respective elemental species – e.g. ^{12}C makes up [98.84, 99.04] % and ^{13}C [0.96, 1.16] % of the total carbon on Earth [2] – and analytically differentiated by their mass difference of about 1 Da. Since there are sufficiently sensitive analytical methods available to detect this difference, they can be thought of and utilized as molecular labels.
ILEs, then, revolve around the incorporation of substrate species enriched with heavy isotopes at specific positions into a target organism and the subsequent analysis of arising metabolite labeling states (figure 1.1).

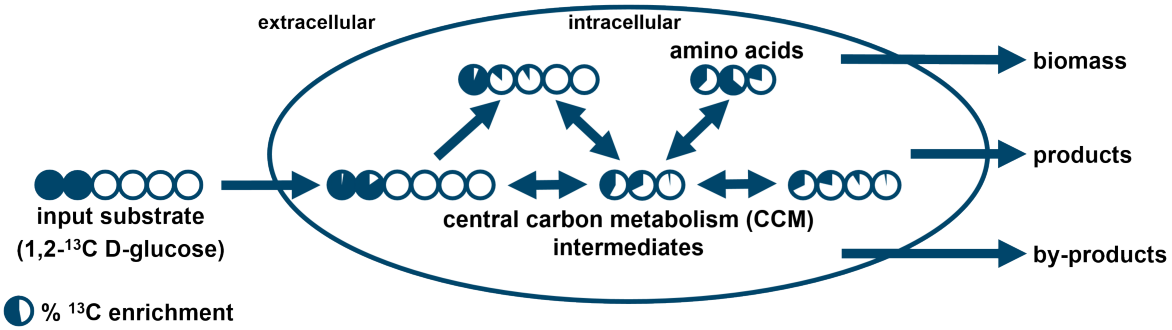


Figure 1.1: Portrayal of the ^{13}C-ILE principle. Cells are grown on a substrate enriched with the heavy carbon isotope ^{13}C at pre-defined positions and the labeling states of either free CCM intermediates, free amino acids, or proteinogenic amino acids are analyzed.

In the portrayed example, the substrate D-glucose (Glc) is labeled at the first and second carbon atom and therefore denominated as 1,2-^{13}C Glc. While there are applications for all listed heavy isotopes, most ILEs rely on carbon isotopes since the central carbon metabolism (CCM) contains many reactions facilitating positional changes of C atoms, thereby increasing the variety and thus informativeness of the resulting metabolite labeling patterns. During the distribution of heavy isotopes across the metabolic network, the positional enrichment of intracellular metabolites, products, by-products, proteins, and biomass is a result of the mixture of labeled substrate species, the network structure, pathway usage, and intracellular reaction rates [3]. The observation of these labeling states accordingly allows the inference of aspects of the organism's phenotype. Commonly, an experiment is focused on one subset of metabolites referring to either proteinogenic amino acids, free amino acids or free CCM intermediates such as sugar phosphates and organic acids.

After an ILE has been conducted, the labeling states of the selected metabolites can be detected with several analytical methods, i.e. nuclear magnetic resonance (NMR), gas chromatography coupled to mass spectrometry (GC-MS(/MS)), and liquid chromatography coupled to mass spectrometry (LC-MS(/MS)). These will be addressed in more detail in one of the following sub-chapters (see 1.3) but suffice to say that all these methods are adequate for the purpose of ILEs and have specific up- and downsides. In actuality, though, due to the large investment cost of such devices the experimenter will have to work with what is present at their laboratory.
Carbon labeling experiments can be further subdivided depending on whether the stable ^{13}C or the radioactive ^{14}C were employed. Historically, ^{14}C tracer experiments were conducted e.g. to elucidate metabolic pathways of microorganisms [4, 5] but were almost entirely superseded by ^{13}C. While the detection of a labeled molecule is not problematic with ^{14}C labels, the positional resolution of a specific label within a molecule necessitates the extraction and complete chemical degradation of that molecule whereas the aforementioned analytical devices are able to measure the labeling states of a molecule within a mixture [6]. There remain some applications for ^{14}C isotopes in ILEs, though, e.g. in plant research a dual approach using both ^{13}C and ^{14}C has been applied in recent publications [7, 8]. Regarding other isotope species, ^{18}O has been used in proteomics [9] and ^{15}N in field [10] and marine [11] experiments as well as to increase the positional information obtained in LC-MS/MS measurements [12], to name just a few examples. The present thesis is focused entirely on ^{13}C-ILEs which, too, have been employed for pathway elucidation [13] and to determine intracellular reaction rates of parts [14] or since the publication of a pioneering study the whole of the CCM [15].
Having addressed some of the possible applications, a state of the art ILE workflow is presented in figure 1.2. The first step comprises pre-experimental considerations not all of which may apply for every ILE. For example, while the conscious choice of a labeled substrate, also denominated as a tracer, or a mixture thereof is a pre-requisite for any ILE, design of experiments (DoE) referring to tracer selection based on simulation studies maximizing an information criterion [3] is not necessarily mandatory. Qualitative ILEs where the usage of a certain pathway can be proven when a specific labeling pattern arises in a target metabolite, require a knowledge of carbon transitions and potentially available pathways but not a traditional DoE.
The ILE itself is oftentimes conducted in parallel lab-scale bioreactors where the measurement of pH and dissolved oxygen (DO) as well as exhaust gas analysis are performed online and all other samples are taken and processed manually. However, a sample for the analysis of metabolite labeling states cannot simply be drawn regularly, but the metabolism of the cells within the sample has to be quenched as fast as possible so that there is no residual enzyme activity during sample processing. In the state of the art workflow, this is done by cold methanol quenching according to a published protocol [16]. This particular quenching method relies on an extremely low, i.e. unphysiological, temperature to stop enzymatically catalyzed reactions immediately after sampling. While keeping the temperature low, cells are separated from supernatant via centrifugation and subsequently cell lysis and metabolite extraction are performed in methanol-chloroform. The supernatant can be used to gauge the effect of metabolite leakage known to occur during this type of quenching [17, 18] and the final extract to measure intracellular concentrations or pool sizes. Additionally, the extracellular metabolite concentrations can be determined with a separate cell-free

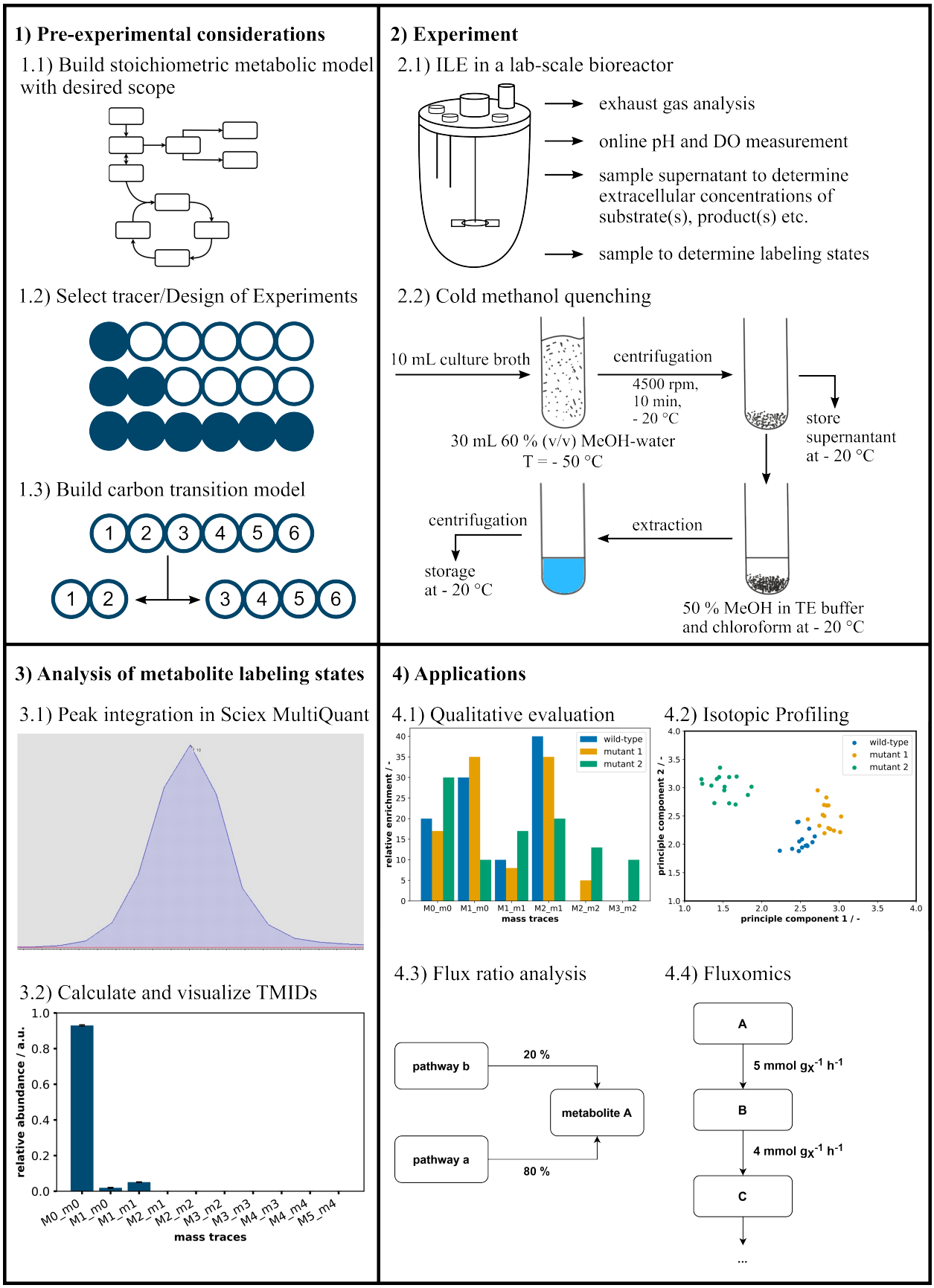


Figure 1.2: Depiction of the manual state of the art ILE workflow subdivided into four phases. The cold methanol quenching protocol was taken from [16].

sample [19]. For the sake of completeness, it should be noted that both divergent cold methanol quenching protocols [20] and different quenching methods altogether [21] have been in regular use.

The third segment deals with the analysis of metabolite labeling states. Labeling experiments can give rise to many differently labeled species of a molecule which are referred to as isotope isomers or isotopomers. Concentrating on carbon labeling, a metabolite with n carbon atoms can form 2^n different isotopomers the distribution of which amounts to the hitherto mentioned labeling state. As alluded to previously, there are multiple viable options but in the state of the art workflow, amino acids and CCM intermediates are analyzed via LC-MS/MS. Since the goal of this section is the calculation of labeling distributions for each metabolite based on peak areas, the LC-MS/MS peaks need to be integrated which is performed using vendor software, e.g. Sciex MultiQuant [22]. Finally, due to the aforementioned natural occurrence of heavy isotopes, the labeling distributions need to be corrected so that they reflect only those labels which have been introduced via the labeled substrate(s) during the ILE. There are multiple readily available open source software solutions for this purpose, e.g. IsoCor [23, 24] and the isotope correction toolbox [25].

Subsequent to raw data processing, ILEs can be interpreted in different ways depending on their design and the experimenter's intent. Sometimes, purely contrasting the labeling distributions arising under different conditions or amongst a group of strains may suffice to prove or falsify an hypothesis. In contrast, the more complex interpretations require additional modelling and computation work like the flux ratio analysis yielding ratios pertaining to the relative utilization of metabolic pathways or ^{13}C-metabolic flux analysis (MFA) absolutely quantifying intracellular reaction rates (see 1.4.3). Generally, not least due to the underlying measurements these methods investigate the inner workings of an assumed average cell, representative for the population. This selection of potential applications is by no means exhaustive but should communicate the broad range of use cases for ILEs in biological phenotyping experimentation.

The choice to structure this workflow as a sequence of ordered steps instead of a cycle, where the results of one ILE inform the design of the next, is meant to illustrate the fact that while such an iterative approach would be beneficial, ILEs are generally too expensive and time-consuming to accommodate multiple consecutive experiments. Since a setup of four parallel bioreactors allows for a maximum of two conditions – i.e. label mixtures, strains, media etc. – in biological duplicates and one run including a pre-culture and time for preparing and cleaning the bioreactors can easily take a week's time, the throughput of ILEs is severely limited. Disregarding temporal issues, the sheer cost of labeled substrate in a batch experiment with a lab-scale bioreactor of at least 1 L filling volume poses an additional constraint, when e.g. U^{13}C Glc is priced at about 539 € g^{-1} and 1,2-^{13}C Glc [26] at about 1290 € g^{-1} [27]. A less important issue but an issue, nonetheless, is the lack of experimental standardization among practitioners of ILEs, especially with regards to the quenching protocol.

The solution to all these drawbacks, then, is the miniaturization, parallelization, and automation of ILEs to achieve a new and improved overarching ILE workflow. When the reaction volume is decreased, the required mass of labeled substrate per replicate is reduced proportionally and so are the costs. A significantly higher degree of parallelization diminishes the time investment per replicate and increases the throughput. Naturally, miniaturization and parallelization behave

synergistically as smaller reaction vessels enable a larger number of simultaneous cultivations. Finally, automation requires the implementation of standardized workflows, thereby offering the chance to reduce the variety of protocols for ILEs.
Previously, automation of ILEs had progressed to the point where a parallelized setup of minibioreactors with a filling volume between 7 mL and 15 mL had been employed for conducting automated experiments investigating proteinogenic amino acids, albeit without quenching [28]. Regarding miniaturization, there have been experimental protocols for a small scale [29, 30], e.g. in deep well plates (DWPs), yet those were still performed manually. To realize an automated ILE workflow while improving on these previous approaches, then, is the supreme objective of the present dissertation.

1.2 *Corynebacterium glutamicum* as a model organism

Biological research has long been conducted by studying phenomena and hypotheses of interest based on those model organisms which most conveniently enable such investigations and of which it can be assumed that the underlying principle can be transferred to other organisms [31, 32]. Accordingly, this dissertation makes extensive use of *C. glutamicum* as a model organism for validation and proof of concept experiments for innovative automated experimental procedures to conduct ILEs.
The facultatively anaerobic, Gram-positive soil bacterium *C. glutamicum* [33] was first isolated in 1957 by Kinoshita et al. [34] in order to identify a natural producer of L-glutamate (Glu). What they discovered was the later to be renamed *Micrococcus* strain No. 534 or *Micrococcus glutamicus* with a Glu yield of 0.25 mol_{Glu} mol_{Glc}^{-1}. In the following years and decades, numerous production processes were established with *C. glutamicum* as the platform organism for amino acids like Glu, L-lysine (Lys), L-ornithine (Orn), and L-valine (Val) making extensive use of auxotrophic [35] and analogue-resistant [33] mutants. This ascent to an industrial workhorse was certainly aided by its generally recognized as safe (GRAS) status and its comparatively high growth rate among other factors [36]. Fast-forwarding to more recent times, *C. glutamicum* is still utilized as an industrial Glu producer with an annual production of 3 million tons in 2014 [37] which is continuously growing. In 2022, the whole glutamate market size was valued at 12.63 billion US dollars with an expected compound annual growth rate of 5.9 % from 2023 to 2032 [38].
Indeed, whereas the Glu production merely requires biotin-limiting conditions [33], many amino acid pathways are subject to strict regulation, wherefore considerable effort has been dedicated to the investigation and circumvention of such cellular regulation systems [39]. Yet, the product palette was not restricted to amino acids as researchers applied *C. glutamicum* to the production of putrescine [40], diaminopentane [41], 5-aminovaleric acid [42], isobutanol [43], hydrobenzoic acids [44] etc., as well.
In the wake of these developments, *C. glutamicum* became the object of many a fundamental research project leading to the identification of its complete genome [45, 46] and the application of omics techniques and modelling to strains of *C. glutamicum* [47]. This extensive degree of study certainly predestines it as a model organism but focusing on systems biology in particular,

it has the advantage of availability of previous research data for comparison and contextualization of results. Additionally, cellular characteristics like the biomass composition, the stoichiometric biomass equation [48], and the specific cellular volume [49] have been published for the wild type (WT). More importantly, though, when validating a method for metabolic quenching, cell lysis, and metabolite extraction as intended in this dissertation, positive results with a Gram-positive organism such as *C. glutamicum* can reasonably be assumed to be transferable to Gram-negative organisms with their less sturdy cellular barrier.

Regarding the central carbon metabolism (CCM) of *C. glutamicum*, it consists of the Embden-Meyerhof-Parnas (EMP) pathway, gluconeogenesis, the pentose phosphate pathway (PPP), the tricarboxylic acid (TCA) cycle including a glyoxylate shunt and anaplerotic reactions, amino acid biosynthesis pathways, and biomass formation [50]. Notably absent is the Entner-Doudoroff (ED) pathway as an alternative avenue for glycolysis. These reactions combine to maintain a supply of energy molecules such as adenosine triphosphate (ATP) and guanosine-5'-triphosphate (GTP), reduction equivalents such as the reduced forms of nicotinamide adenine dinucleotide (NADH) and nicotinamide adenine dinucleotide phosphate (NADPH), and biomass precursors to enable cellular function and growth via the catabolism and oxidation of substrates. More precisely, one could state that the role of the PPP is to regenerate NADPH, provide C_4 and C_5 molecules as building blocks for amino acid and biomass formation, and re-supply the EMP pathway at the level of fructose 6-phosphate (F6P) and glyceraldehyde 3-phosphate (GAP) [51]. The role of the TCA cycle, then, is the complete oxidation of acetyl-Coenzyme A (acetyl-CoA or henceforth AcCoA) originating from any carbon source in order to generate biomass and amino acid precursors [52] while regenerating NADH and menaquinol (MQH_2).

A key regulator of carbon flow to and from the TCA cycle is constituted by the anaplerotic reactions linking it with the EMP pathway. In *C. glutamicum*, these traditionally comprise

$$\begin{aligned}
\mathrm{Ppc\ (EC4.1.1.31)}: &\quad \mathrm{PEP} + \mathrm{HCO_3^-} = \mathrm{OAA} + \mathrm{P_i} \\
\mathrm{Pck\ (EC4.1.1.32)}: &\quad \mathrm{OAA} + \mathrm{GTP} = \mathrm{PEP} + \mathrm{GDP} + \mathrm{CO_2} \\
\mathrm{Pyc\ (EC6.4.1.1)}: &\quad \mathrm{Pyr} + \mathrm{ATP} + \mathrm{HCO_3^-} = \mathrm{OAA} + \mathrm{ADP} + \mathrm{P_i} \\
\mathrm{Odx\ (EC4.1.1.3)}: &\quad \mathrm{OAA} = \mathrm{Pyr} + \mathrm{CO_2} \\
\mathrm{MalE\ (EC1.1.1.40)}: &\quad \mathrm{Mal} + \mathrm{NADP^+} = \mathrm{Pyr} + \mathrm{NADPH} + \mathrm{H^+} + \mathrm{CO_2}
\end{aligned}$$

catalyzed by the enzymes phosphoenolpyruvate carboxylase (Ppc), phosphoenolpyruvate carboxykinase (Pck), pyruvate carboxylase (Pyc), oxaloacetate decarboxylase (Odx), and malic enzyme (MalE). As stated in the dissertation of Kappelmann [53], though, any intracellular reaction can be considered anaplerotic if its net activity contributes carbon atoms to the TCA cycle.

Defining these 5 reactions separately and reversibly grants the maximum degree of freedom but there have been investigations restricting the directionality of some involved enzymes. These studies inquiring into the activity of anaplerotic enzymes made extensive use of various deletion mutants. Thereby, it was found that a Δ*pck* mutant could not grow on substrates entering on the level of AcCoA and thus requiring gluconeogenesis showing that the reverse reaction of Ppc is not active [54]. Single deletion mutants in glycolytic conditions, however, behaved normally with

the exception of the Δ*pck* mutant which exhibited a slightly lowered growth rate despite the lack of required gluconeogensis [55]. With respect to the carboxylases, this result implies that they can replace each other without affecting the growth rate. There were conflicting reports about the effects of a Δ(*ppc pyc*) double deletion for which either no growth [56] or a significantly reduced growth rate after evolving the strain by prolonged cultivation were observed [55]. In the latter study it was surmised that the missing carboxylation reactions were not replaced by their counterparts Pck and Odx but instead via the glyoxylate shunt [55].
With regards to ^{13}C-MFA, the anaplerotic reactions are unified into 3 reactions as there is no point in differentiating between Ppc and Pck as well as PCx and Odx activities from the perspective of net fluxes. Even so, it was demonstrated by identifiability analyses that the anaplerotic node in *C. glutamicum* is structurally unidentifiable [57].
A notable characteristic of *C. glutamicum* is its natural ability to metabolize a large spectrum of carbon sources. The uptake of various different sugars such as D-glucose, D-fructose, and D-sucrose is conducted via distinct phosphoenolpyruvate (PEP)-dependent phosphotransferase systems (PTS) [58, 59]. Accordingly, D-glucose is converted to glucose 6-phosphate (G6P) during uptake. As these PTS are expressed constitutively, *C. glutamicum* exhibits monophasic growth on a mixture of these substrates and others like lactate, propionate, and pyruvate (Pyr) [60, 61].
The catabolism of ethanol and acetate, on the other hand, is subject to catabolite repression which manifests in a biphasic growth behavior [62]. The degradation pathway of these substrates is partially identical, merely ethanol's entry point is two additional reaction farther away from the CCM. In particular, ethanol is oxidized twice in sequential reactions via acetaldehyde to acetate, catalyzed by the NAD-dependent enzymes alcohol dehydrogenase (Adh) and acetaldehyde dehydrogenase (Aldh). Acetate is then converted to its closest CCM intermediate AcCoA via phosphorylation by acetate kinase (AckA) and CoA-activation by phosphotransacetylase (Pta), thus entering the CCM downstream of the EMP pathway [62]. Needless to state that the respective catabolic enzymes towards AcCoA are essential for the growth on acetate and ethanol [62, 63].
Depending on the entry point of the given substrate(s) into the metabolic network, the fractional usage and direction of certain reactions and whole pathways may change. Under glycolytic conditions, both EMP and PPP are utilized but the larger share of G6P is directed towards the latter [64]. The TCA cycle is active yet expression of isocitrate lyase (AceA) and malate synthetase (AceB) – the enzymes of the glyoxylate shunt encoded by the genes *aceA* and *aceB* – is subject to catabolite repression [65]. The net carbon flux into the TCA cycle via the anaplerotic reactions stated above is positive. In detail, the net Ppc/Pck reaction removes carbon from the TCA cycle acting in a cataplerotic manner and the net PCx/Odx reaction is anaplerotic while MalE appeared to be inconsequential, yet non-identifiable [66].
In contradistinction, growth on substrates entering the CCM on the level of AcCoA like ethanol and acetate induces *aceA* and *aceB* which has been substantiated by transcriptome and proteome analyses [52, 62]. A deletion mutant without these two genes could grow neither on acetate nor ethanol as the sole carbon source meaning glyoxylate shunt activity is indeed essential for such substrates [67] and can be viewed as another anaplerotic reaction in this context [52, 62]. Beyond the experimental proof, the necessity of the glyoxylate shunt usage is intuitively clear as a supply of the TCA cycle with fresh oxaloacetate (OAA) is not possible when a C_2 molecule like AcCoA is the

only influx into a cycle with two decarboxylation steps while still providing biomass precursors. This biological role for the glyoxylate shunt has been hypothesized and re-iterated since the pathway's discovery in 1957 [68–70]. More precisely, one turn of the TCA cycle without glyoxylate shunt activity requires 1 OAA and 1 AcCoA yielding 1 OAA, 2 CO_2, 1 GTP, 3 NADH, 1 MQH_2, and 1 CoA while one turn with glyoxylate shunt activity consumes 1 OAA and 2 AcCoA generating 2 OAA, 2 AcCoA, 1 NADH, and 1 MQH_2. The increased regeneration of the reduction equivalent NADH at the loss of carbon is particularly noteworthy. When related back to the substrate, one could simplify this balance to the fact that 2 molecules of acetate or ethanol are converted to one C_4 molecule [68].
The hitherto conducted ^{13}C-MFAs on acetate lumped the PEP and Pyr as well as the malate (Mal) and OAA pools so that only the direction of the overall net carbon flux was determined. Said net flux flowed towards PEP/Pyr so that carbon is actively removed from the TCA cycle via the conventional five anaplerotic reactions [64, 71] in favor of the gluconeogenetic flux while carbon is supplied to the TCA cycle via acetate degradation and glyoxylate shunt activity.
One topic of this dissertation, then, is to analyze the intracellular reaction rates of *C. glutamicum* when grown on ethanol as the sole carbon source to gain a direct insight into the quantitative pathway usage which the available transcriptome and proteome data merely elude to.

1.3 LC-MS

Since ILE aim to draw information from the labeling states of free metabolites, proteins or biomass, the labeling distributions of the target compounds of choice need to be observed. Historically, GC-MS and NMR techniques were favored, yet in recent years LC-MS(/MS) methods have gained traction. In this thesis, an LC-MS/MS device was chosen due to its low limit of detection, the existence of established, previously validated methods at the IBG-1 [72], and the independence from sample derivatization required by GC-MS enabling high-throughput workflows. The following section, therefore, describes the working principles of high pressure liquid chromatography (HPLC) and electrospray ionization quadrupole-time-of-flight tandem MS (ESI-QqTOF-MS) applied in this dissertation.

1.3.1 HPLC

The general objective of HPLC is to separate compounds dissolved in a mobile liquid phase along a chromatography column by interaction with a solid phase. The strength of said interaction determines the retention time of a given compound. Accordingly, there are many different types of liquid chromatography depending on the column material, e.g. size exclusion chromatography, ion exchange chromatography (IEX), and reverse-phase chromatography to name just a select few. The HPLC principle as well as an exemplary column material for IEX in benzene sulfonic acid coupled to silica particles are portrayed in figure 1.3.
Both for a successful separation of target compounds and for peak data evaluation, it is important to address the physical and chemical factors resulting in a dispersion, i.e. the dilution of the injected sample during elution, and in turn influencing peak shape and width. Within the HPLC

column, analytes travel through a packed bed of solid phase particles with a diameter in the low micrometer range. Since the particle arrangement of the packed bed is not perfect, though, the retention times of particles from the same molecular species may differ due to deviating paths through the column (compare the green and black lines between points a and b in figure 1.3) - a phenomenon denominated as eddy diffusion.

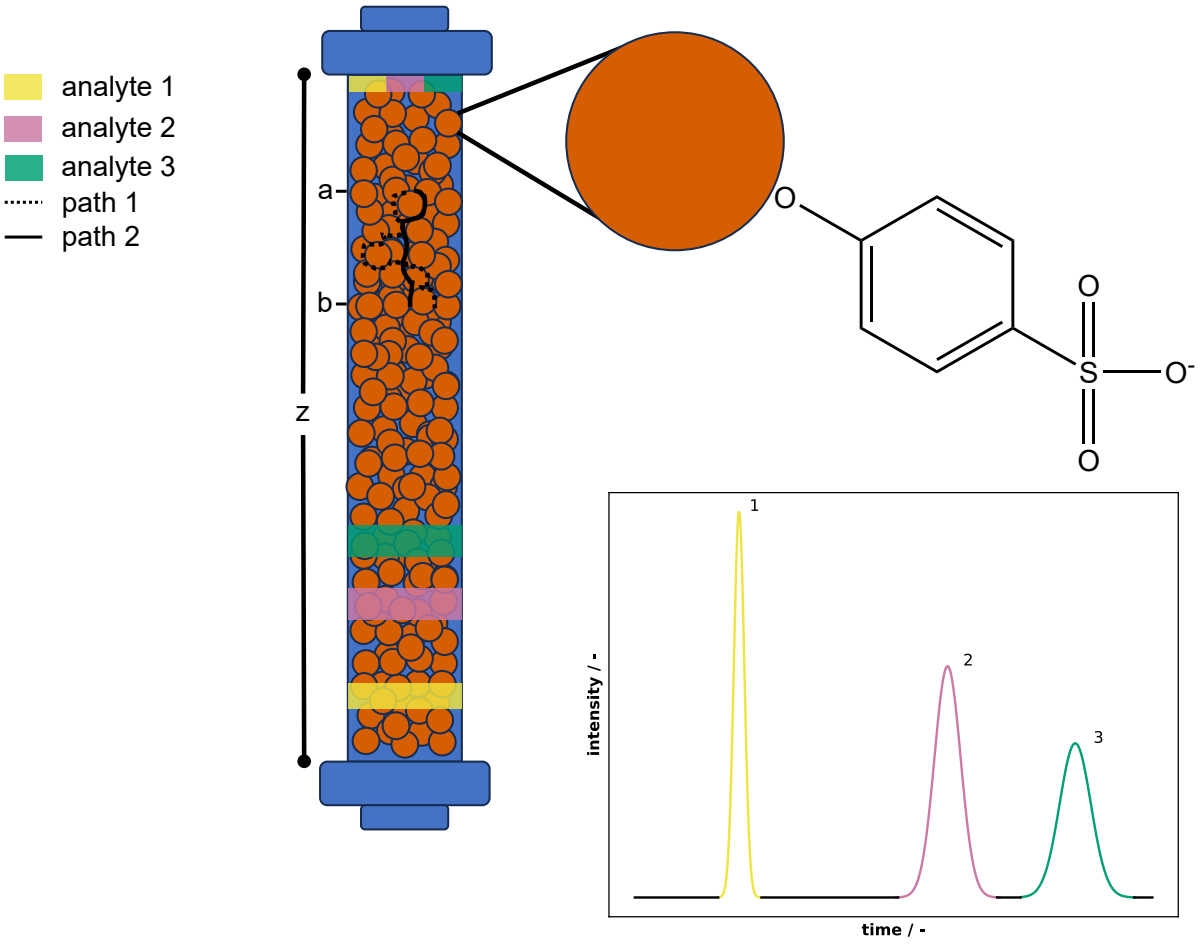


Figure 1.3: Depiction of a HPLC column separating three analytes resulting in a qualitative chromatogram on the bottom right. The paths between points a and b marked with the full and dashed lines are meant to represent Eddy diffusion as their lengths deviate despite bridging the same fraction of z. Furthermore, an exemplary coating material of silica particles for IEX in benzene sulfonic acid is portrayed.

Additionally, since the sample is injected in a concentrated manner at a low volume in the microliter range into a column with a diameter in the millimeter range, there is a considerable effect of axial diffusion along the length z of the column. This effect grows more pronounced the larger z and the lower the interstitial velocity are. Lastly, the mass transfer, i.e. the adsorption and desorption kinetics, impact the retention time of a compound and cause peak broadening, especially at higher interstitial velocities.
These effects were combined in the van Deemter equation

$$H_i = A_i + \frac{B_i}{u} + C_i u \tag{1.1}$$

in which H_i represents the height of an equivalent theoretical plate, A_i the Eddy diffusion term, B_i the axial diffusion term, C_i the mass transfer resistance, and u the interstitial velocity [73].
Elaborating on the adsorption behavior exhibited by analytes, it has been quantified experimentally and numerous adsorption isotherms describing the relation of the dissolved analyte concentration q_i and the adsorbed analyte concentration c_i at equilibrium have been postulated. Many of these are variations of the Langmuir isotherm which follows the semi-empirical kinetic

$$q_i = q_{i,\max} \cdot \frac{bc_i}{1 + bc_i} \tag{1.2}$$

with $q_{i,\max}$ as the maximum adsorbed analyte concentration (or maximum adsorption capacity) and b as the Langmuir coefficient [74]. The Langmuir model is based on the assumption of a homogeneous surface where molecules are bound in a monolayer and each adsorption site may only bind one molecule. Further, all sites are considered energetically equal and interactions between molecules or effects such as cooperativity are not taken into account [75]
The interstitial velocity depends on the isotherm's slope which in turn is dependent on the local dissolved analyte concentration. Since in case of the convex Langmuir isotherm there is an inverse relationship between the slope $\delta q_i \ \delta c_i^{-1}$ and u_i (see 1.4), the fractions with a higher local dissolved analyte concentration move faster and thus elute earlier, causing a peak distortion referred to as tailing.

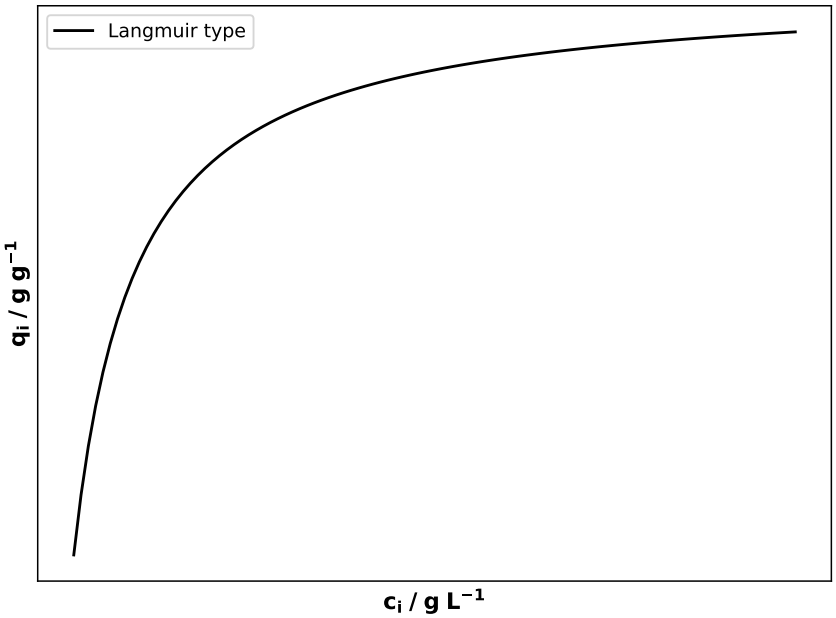


Figure 1.4: Qualitative depiction of the Langmuir adsorption isotherm.

This is but one of many effects potentially influencing the peak shape and thereby complicating peak recognition and integration.
Upon leaving the HPLC column, analytes would usually be detected with a refractive index or an ultraviolet (UV) light detector but in LC-MS/MS the MS essentially serves as the detector, instead. HPLC's role as a preceding step in the LC-MS/MS workflow, then, is to a) reduce the complexity of a sample's biological matrix to enhance the subsequent MS analysis by diminishing

ion suppression effects and b) to separate isobaric compounds which could not be differentiated using a stand-alone MS device.

1.3.2 ESI-QqTOF-MS

The subsequently discussed MS device is comprised of three main types of components, namely an ion source, mass analyzer(s), and a detector [76]. After exiting the HPLC column, the analytes of interest remain as dissolved molecules in a liquid mobile phase regularly consisting of solvents such as acetonitril, methanol, and water or mixtures thereof. Before MS analysis, they must hence be converted to an ion stream in the gas phase which is realized in the ion source. The actual analysis of ions is subsequently performed by the mass analyzers which separate ions based on their mass to charge (m/z) ratio. Finally, the detector records the number of arriving ions of a given m/z ratio and the signal is converted with an analogue-digital converter.
Historically, there have been numerous ionization methods such as fast atom bombardment (FAB), plasma desorption, and laser desorption where the ionization and the greatly endoergic transfer of ions from liquid to gas phase were accomplished by high energy collision and heating [77]. These methods led to in-source fragmentation of analytes and had additional limitations. FAB, which was suited for biological samples, yielded mostly singly charged ions, thus effectively imposing an upper limit on the mass of analyzable molecules [76]. A similar method, electron impact ionization (EI), was published in 1918 [78] and is still routinely used in GC-TOF applications today [79, 80]. Here, an electron stream originating from a glowing thread induces ionization by collision or near-collision with the gaseous analytes leaving the GC column. Due to the electrical forces of a passing electron, the analytes loose an electron themselves resulting in a positively charged ion but also in in-source fragmentation [81].
For LC-MS/MS, the ubiquitously applied technique of choice is electrospray ionization (ESI), a "soft" ionization technique imparting little additional energy onto the analytes thus avoiding in-source fragmentation [77] and preserving even non-covalent bounds [76]. It has the further advantages of applicability to a broad spectrum of molecules and the ability to produce multiply charged $[M + zH]^{z+}$ ions enabling the analysis of molecules in the MDa range even on MS devices with limited m/z ranges [82, 83]. Thereby, the analysis of macromolecules such as nucleic acids and proteins was enabled which single-handedly led to the foundation of the field of proteomics [76, 77, 82]. Concomitantly, ESI facilitates the detection of low molecular weight compounds such as metabolites [82]. Another reason for its dominant usage in LC-MS devices, in particular, is that ESI accommodates utilizing the same polar solvents which are already ubiquitous in analytical chemistry [77, 82, 83]. The technique has been first applied to MS by Yamashita and Fenn in 1984 [84, 85] for which Fenn received a Nobel Prize in 2007. As an interesting historical note, electrospraying existed for much longer and had been used in the car industry for coating since the 1950s [86].
The basic principle of ESI, then, lies in producing an electrospray of small highly charged droplets within a strong electric field which are continuously reduced in diameter until the dissolved ions are transferred to the gas phase by one of multiple effects [77]. Going into more detail, the liquid phase from the HPLC enters the ion source through a probe with a metal capillary or electrode

with an inner diameter of about 0.1 mm located 1 cm - 3 cm opposite of a counterelectrode which corresponds to the entrance of the MS [77]. Inside the ion source, a strong electric field of several kilovolts and a temperature of several hundred degrees Celsius are applied at atmospheric pressure [82]. In an electrochemical redox reaction with the electrode, an oxidation or reduction of redox active analytes or solvent occurs at the interface of electrode and solution [87]. The following description is focused on the positive ionization mode, i.e. with a positive potential of the capillary [77], as it was predominantly used in this dissertation (figure 1.5). Exemplary oxidation reactions for the common solvents water and methanol are [87]

$$\begin{aligned} 2H_2O &= O_2 + 4H^+ + 2e^- \\ CH_3OH &= HCHO + 2H^+ + 2e^- \end{aligned}$$

The produced protons and all positively charged ions then gather at the liquid surface for they are repelled by the identically charged electrode [76] and conversely attracted by the MS entrance. Despite this, the ions cannot leave the liquid so in a sufficiently strong electric field, the accumulation of positive charges destabilizes the liquid surface deforming it in downfield direction into a Taylor cone [77, 88].

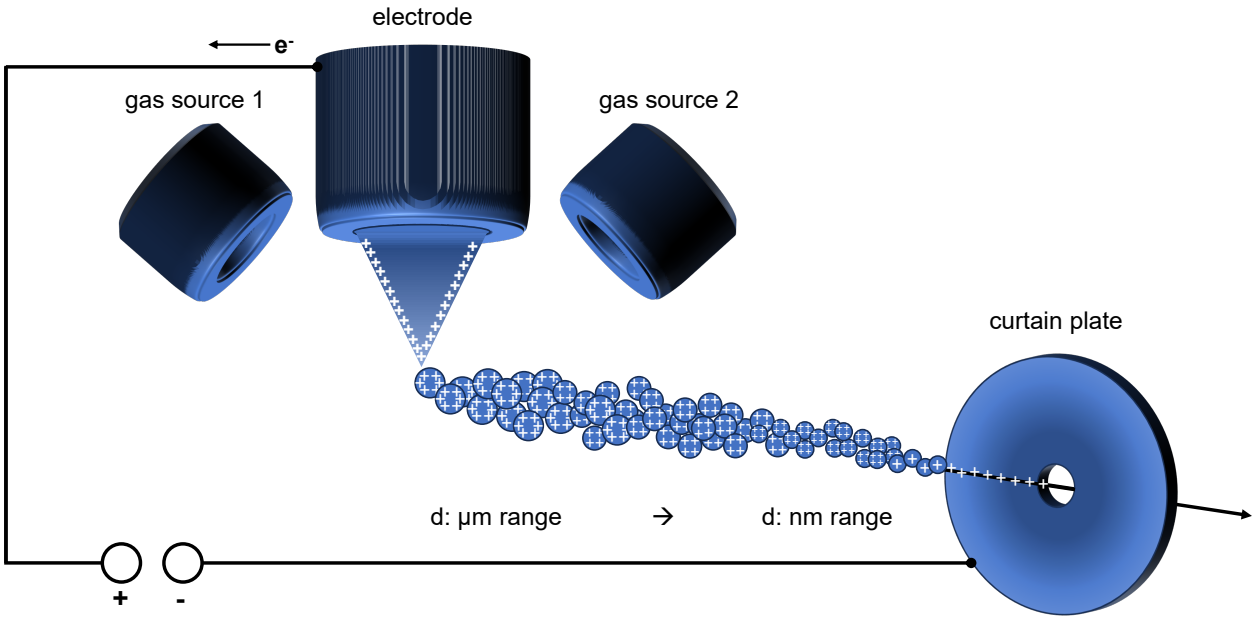


Figure 1.5: Portrayal of the ion source of a MS device performing ESI in positive ionization mode.

The stability of the Taylor cone is dependent on an equilibrium of electric field strength and surface tension. Upon instability, a jet of charged droplets with diameters of few micrometers is released from the cone's tip where the charges behave according to Gauss' law and once again reside in an equidistant formation at the surface of the droplets. Aided by the high temperature and by applying a nitrogen gas stream [76], the solvent evaporates until the Coulomb repulsion between the positive charges becomes larger than the surface tension, a relationship which is described in the Rayleigh equation [89, 90]. Upon exceeding this so-called Rayleigh limit, a Coulomb fission occurs where multiple smaller droplets are emitted from their larger progenitor. By repetition of

this process, the droplet diameter is reduced to several nanometers but still the ions remain within the liquid phase. Regarding their transfer to the gas phase, multiple models exist and the predominant mode is dependent on the molecular weight of the analyte in question. When droplets with a diameter below 10 nm are formed, an emission of gas phase ions may occur instead of the aforementioned Coulomb fission [77]. This ion evaporation model (IEM) applies mostly to such ions with low molecular weight [91] like e.g. metabolites. For larger analytes like proteins, the charged residue model (CRM) states that small droplets containing merely a single analyte may be dried completely and impart their charge to the analyte, thereby creating a gas phase ion [82]. Further models are described elsewhere in literature [76, 82].
When gas phase ions have been formed, they are guided into the MS and then focused by ion optics, i.e. lenses and multipoles, to which an axial direct current (DC) voltage differential is applied. The resulting DC gradient is comprised of increasingly negative voltages only to rise immediately preceding the mass analyzer(s) to decelerate the positively charged ions [92]. Since the optics are run in radiofrequency (RF) mode, the ions are radially constricted and no separation is performed yet. This also serves to separate the ion source and its atmospheric pressure environment from the MS device's high vacuum in the range of 10^{-3} torr to 10^{-6} torr [76].
The two mass analyzers that will be addressed here are the quadrupole (Q) and the time-of-flight (TOF) analyzer. Starting with the quadrupole, it consists of two sets of two oppositely placed metal rods in a diagonal arrangement. A DC voltage with the same sign is applied to each set and based on a RF – i.e. an alternating current (AC) – the voltages are exchanged between the two sets causing a radial oscillation of the ions. If the DC voltages of the two sets had the same amplitude, this would amount to the so-called RF mode addressed above but due to the application of a DC offset, only ions with a certain m/z ratio move on a stable trajectory and leave the quadrupole [93]. The separation based on m/z values balances two principles: the attraction towards one set of rods at any point in time (DC) versus the radial force caused by the RF. Ions with a higher molecular weight experience a higher degree of inertia so the DC parameter acts as a filter for m/z ratios above a certain threshold. On the other hand, the RF parameter is used to filter out lower molecular weight ions for they are accelerated faster with the same force than heavier ions would be. If the axial motion induced by the RF grows too large, lighter ions collide with the quadrupole rods. However, when changing the RF and DC amplitudes in a fixed ratio, only ions with a certain m/z ratio are selected [94]. Another way to conceptualize the separation by a quadrupole mass filter is via the a and q parameters of the Mathieu equation as visualized in stability diagrams [93].
To be more precise, a quadrupole is usually run in unit resolution meaning one m/z ratio window is restricted to a width of 1 Da. As only one range can be focused on at a time, a quadrupole is referred to as a scanning mass analyzer. Therefore, when several m/z ratios are supposed to be selected, this is performed successively in iterative cycles thereby imposing the concept of cycle time, i.e. the time it takes to complete one such measurement cycle. When connected to a HPLC, the number of data points per peak is intended to be as high as possible so the cycle time effectively limits the number of m/z ratios which can be measured per sample or time period within the sample.
The second mass analyzer introduced here is the TOF, the first of which was built in 1948 [95].

Quickly superseded by quadrupoles with superior mass resolution and sensitivity, early TOF devices were held back by technical limitations such as e.g. the speed of recording mass spectra [96, 97]. Since then, numerous developments have alleviated these shortcomings so that currently TOF devices are regularly used to both identify and quantify unknown compounds separated by HPLC [97].
In contradistinction to the quadrupole, a TOF is not a scanning mass analyzer but instead all ions within a discrete ion packet are analyzed simultaneously, thus removing the concept of cycle time introduced above. Such ion packets are generated by modulation optics with a pulse, i.e. by briefly switching from a negative to a positive potential, thereby controlling the transmission of ions into the acceleration region of the TOF [96]. Subsequently, ions in said region are accelerated orthogonally with a strong positive pulse by a repeller electrode and hit the detector after traversing the drift region. In principle, when being subjected to a pulse of equal energy, ions with a higher m/z ratio will be slower and thus arrive at the detector later than those with a lower m/z ratio. However, in practice the resolution of a TOF is impacted negatively by the initial distribution of ions with regards to their velocity and position in the acceleration region [97, 98]. This large energy and positional spread is counteracted by reflectrons [99], also referred to as ion mirrors. Such an ion mirror is constituted by an electrostatic field after a first drift region which serves to decelerate and eventually reverse the ions. As ions with a higher velocity will penetrate more deeply into this fields, they will spend more time there which compensates for their higher speed and according lower residence time in the drift regions [99]. This way, ions of equal m/z ratio and diverging energies and positions are essentially focused before reaching the detector [97]. Thusly, this technique comprises a space-efficient way to increase the length of the drift region [97] and for TOF devices with reflectrons, such an increase does indeed positively influence the resolution proportionally [98].
Modern MS devices use electron multiplier detectors like microchannel plates (MCPs) where incoming ions collide with metal plates which emit multiple secondary electrons striking further plates, thereby amplifying the signal with a gain of roughly 10^6. Finally, the voltage signal inferred from this current is converted to an intensity value in counts per second (cps) of a mass spectrum with an analog-to-digital converter [76].
Combining the techniques and devices introduced in the preceding paragraphs, an ESI-QqTOF device is comprised of an ion source performing ESI, a first quadrupole (Q1), a collision cell (Q2), a TOF mass analyzer and a detector (figure 1.6). The Q1 constitutes the first MS stage, selecting intact precursor ions which are subsequently fragmented in Q2 by collision with inert gas particles in a process called collision-induced dissociation (CID). Q2 is run in radiofrequency-mode and does not act as a mass analyzer. The result of CID depends on the kinetic energy of the ions, i.e. on the difference in potential between the entrance of the MS and Q2 which is defined as collision energy (CE). The fragments or product ions are then analyzed in the second stage of MS comprised by the TOF device to resolve some structural information about the ion which could not be obtained in single stage MS.
By combining quadrupole and TOF in this way instead of using a triple quadrupole device (QqQ), only the first MS stage is subject to cycle time allowing the observation of more mass transitions, i.e. pairs of precursor and product ion exact m/z ratios [53].

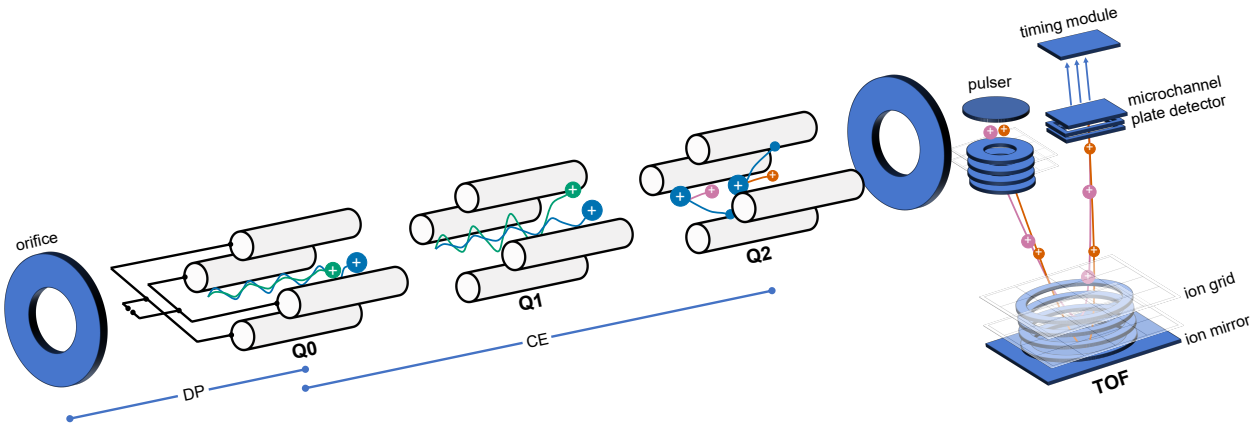


Figure 1.6: Exploded view of a QqTOF device featuring quadrupoles employed for focusing (Q0), mass selection (Q1), as a collision cell (Q2), and finally a TOF mass analyzer with a microchannel plate detector. The difference in potential between orifice and Q0 is denominated as declustering potential (DP) and the one between Q0 and Q2 as CE.

When applying the QqTOF concept to ILEs in order to detect molecular labeling states, the concept of mass traces has to be introduced. Given the case of labeling exclusively with ^{13}C, single stage MS would not allow the measurement of all 2^n isotopomers [100] but only of the mass isotopomers, i.e. groups of isotopomers with an equal number of labeled atoms without regard for their position. Mass isotopomers are commonly expressed as the mass of the molecule m plus the number of incorporated labels, e.g. m+1 for one label, m+2 for two labels and so forth. Tandem MS, then, generates additional information about the position of labels by fragmentation. If for example the first carbon atom of a precursor ion is dissociated during fragmentation and the compound was labeled at that position, the number of labels of the product ion will be reduced by one. In case this was the only label, this is denominated as the mass trace M1_m0, where the M1 pertains to the mass of the precursor ion plus its single label and the m0 to the mass of the product ion without any labels. While this still does not fully determine the positions of all labels and thus uncover all isotopomers, it nonetheless yields additional information, especially when observing multiple product ions with different carbon backbones per precursor ion. Obtaining peak data for all theoretically possible mass traces of a mass transition, then, enables the calculation of a tandem mass isotopomer distribution (TMID) by normalization. Such a distribution is often depicted as a TMID vector where each element pertains to the relative abundance of a mass trace.
In summary, using ESI-QqTOF analytics tandem mass isotopomers are detected and the obtained peak data processed to TMIDs which in turn serve as raw data for further evaluations via model-based approaches or are directly interpreted to experimentally test a hypothesis.

1.4 Applications for experimental data

1.4.1 Bioprocess modelling

In general, models serve as abstractions and simplifications of reality and are designed to describe specific phenomena while being subject to certain assumptions. They can be used to unite and explain data and even predict a system's behaviour under altered conditions. In accordance with

Occam's razor, an often propagated principle of modelling states that they should be as simple as possible, yet as complex as necessary.
In biotechnology, a common use case for modelling, perhaps the most prominent one, is constituted by bioprocess modelling. This term refers to the mathematical depiction of fermentations in general but with respect to the contents of the present thesis, it will be related to microbial cultivations specifically. Over the course of such cultivations in a bioreactor, online measurements (e.g. pH, DO) are complemented with sampling to monitor the extracellular concentrations of substrates, products, and possibly by-products as well as the cell dry weight (CDW). During the bioprocess, this data can be used to e.g. trigger events such as the administration of additional substrate or an increase of the stirrer speed upon low DO values. Subsequently, performance indicators such as yield and space-time yield can be calculated and the concentration data can be utilized to determine extracellular rates such as the substrate uptake rate, product formation rate, and growth rate. Traditionally, specific rates, i.e. the formation or consumption of component i with respect to time t and biomass X, were often computed in a linearized manner according to

$$v_i = \frac{c_{i,t_{i+1}} - c_{i,t_i}}{(X_{i,t_{i+1}} - X_{i,t_i})(t_{i+1} - t_i)} \tag{1.3}$$

Since growth – baring a limitation or inhibition – generally follows an exponential behavior, this approximation is not very faithful to begin with but grows especially erroneous close to the transition between different growth phases. Sometimes this has been applied to calculate an average rate over the whole course of the exponential growth phase, meaning that only measurements from two time points would be considered, granting an undue weight to these data points. Bioprocess models, on the other hand, routinely unite all measurements from multiple quantities, thereby enabling a much more faithful representation of the experimental reality.
An additional advantage of model-based data evaluation is the estimation of quantities that could not be measured. This is particularly relevant when reducing the scale from a bioreactor to a microbioreactor. The latter has the advantages of miniaturization and parallelization of cultivations but it lacks access to exhaust gas analysis and biomass is usually indirectly measured via backscatter, i.e. a density measurement. Even when mimicking the sampling to directly measure cell dry weight, the available biomass is much lower and a cultivation well needs to be sacrificed which may not be compatible with a given experimental setup. Thus, the use of bioprocess models can fill the gap caused by the reduction of informative data in some areas.
At their core, bioprocess models are deduced from mass balances around system components such as metabolite pools. The formulation of ordinary differential equations (ODEs) expressing these balances depends on the chosen system boundaries. These are usually placed either around a cultivation vessel with defined conditions or around an assumed average cell representing the cell population inside the vessel. The former will henceforth be referred to as the macroscopic view and the latter as the microscopic view of a cultivation. Furthermore, the mode of operation requires consideration since an in- and/or efflux of material may need to be taken into account.

Starting with the macroscopic view, the general mass balance holds that [101]

$$\frac{dm_i}{dt} = \begin{cases} \dot{m}_{i,\mathrm{in}} + \dot{m}_{i,\mathrm{out}} \pm r_i V_R & \text{continuous} \\ \dot{m}_{i,\mathrm{in}} \pm r_i V_R & \text{fed-batch} \\ \pm r_i V_R & \text{batch} \end{cases} \tag{1.4}$$

where m_i refers to the mass of component i, r_i to its production or consumption rate, and the reactor volume V_R is defined as

$$\frac{dV_R}{dt} = \begin{cases} \dot{V}_{\mathrm{in}} - \dot{V}_{\mathrm{out}} & \text{continuous} \\ \dot{V}_{\mathrm{in}} & \text{fed-batch} \\ 0 & \text{batch} \end{cases} \tag{1.5}$$

In case of a continuous cultivation, the system influx $\dot{m}_{i,\mathrm{in}}$ and efflux $\dot{m}_{i,\mathrm{out}}$ of component i across the system boundaries are included whereas a fed-batch cultivation merely necessitates the influx term and a batch cultivation features no transport into or from the vessel.
For a simple process featuring only one limiting substrate S and biomass formation, equation 1.4 is adapted yielding

$$\frac{dm_S}{dt} = \frac{(c_S V_R)}{dt} = \begin{cases} c_{S,\mathrm{in}}\dot{V}_{\mathrm{in}} - c_S\dot{V}_{\mathrm{out}} - v_{\mathrm{upt},S}c_X V_{\mathrm{cell}} V_R - MX & \text{continuous} \\ c_{S,\mathrm{in}}\dot{V}_{\mathrm{in}} - v_{\mathrm{upt},S}c_X V_{\mathrm{cell}} V_R - MX & \text{fed-batch} \\ -v_{\mathrm{upt},S}c_X V_{\mathrm{cell}} V_R - MX & \text{batch} \end{cases} \tag{1.6}$$

and

$$\frac{dm_X}{dt} = \frac{(c_X V_R)}{dt} = \begin{cases} -c_X\dot{V}_{\mathrm{out}} + \mu c_X V_R & \text{continuous} \\ \mu c_X V_R & \text{fed-batch} \\ \mu c_X V_R & \text{batch} \end{cases} \tag{1.7}$$

To present the most general form of the balance around S, the metabolic maintenance term MX was included although it is omitted in many applications. The cell-specific volume V_{cell} in $L_{cell}\ g_X^{-1}$ is introduced to connect the macroscopic and microscopic views of the cultivation as the substrate uptake rate $v_{\mathrm{upt},S}$ is defined in $mmol_{substrate}\ L_{cell}^{-1}\ h^{-1}$ [101].
Microbial growth in the absence of limitations or inhibitions is most commonly expressed by the Monod kinetics [102]

$$\mu = \mu_{\max}\ \frac{c_S}{c_S + K_S} \tag{1.8a}$$

$$v_{\mathrm{upt},S} = v_{\mathrm{upt},S_{\max}}\ \frac{c_S}{c_S + K_S} \tag{1.8b}$$

These equations cover the simple case of one limiting substrate but they may express effects like multiple substrates or inhibitions by expanding the formulae with suitable terms.
The microscopic view, then, enables the introduction of mass balances around intracellular metabo-

lite pools. For intracellular metabolite A it holds that [101]

$$\frac{d}{dt}c_A = \dot{c}_A = \sum_{i=1}^{n} v_{\mathrm{in},i} - \sum_{j=1}^{m} v_{\mathrm{eff},j} - \mu\frac{Y_{A/X}}{V_{\mathrm{cell}}} - \mu c_A \tag{1.9}$$

Here, A is assumed to be a biomass precursor, hence the introduction of the term $\mu\frac{Y_{A/X}}{V_{\mathrm{cell}}}$ representing biomass drain in dependence of the growth rate μ, the biomass-specific yield $Y_{A/X}$ in $\mathrm{mmol_A\ g_X^{-1}}$, and the cell-specific volume V_{cell} in $\mathrm{L_{cell}\ g_X^{-1}}$. Due to the shift from macroscopic to microscopic view, growth by mitosis has to be conceived of as an effective dilution of intracellular pools in biomass which is mathematically expressed as μc_A.
The realization of bioprocess models can be conducted freely in a programming language like Python or Matlab or alternatively by using a specialized software suit such as pyFOOMB [103]. Event handling, particularly with regards to modelling ILE, is facilitated by complementing the system of ODEs with a differential-algebraic system of equations (DAE). For example, starting a feed with one labeled substrate species requires splitting the substrate concentration into two distinct quantities, i.e. the initial unlabeled substrate and the labeled substrate which can be summed up to obtain the total substrate concentration. A disadvantage of established software, then, can be a restrictiveness in their design as for example pyFOOMB deals exclusively with ODEs and does not allow the formulation of DAEs.
Moreover, to facilitate communication and cooperation within the scientific community, standardization is highly advantageous so the use of a widely adopted modelling language such as Modelica [104, 105] is encouraged. Here, models can be defined in a straightforward way including both DOEs and DAEs. This factor has also been considered in the development of the software estim8 [106] which is in essence a followup project to pyFOOMB allowing the definition of models in Modelica and performing forward simulations, parameter estimations, and uncertainty quantification. Ultimately, many avenues are available on the software side and should be selected based on the given model structure and the personal abilities and experiences of the experimenter.

1.4.2 Statistical considerations

In general, data generating processes such as biological experiments can and have to be viewed as stochastic since they are subject to numerous sources of errors wherefore the resulting data carries uncertainty [107]. For example, measurements are afflicted with noise, samples are taken and media prepared with a certain precision and so forth. Hence, the uncertainty of quantities computed from experimental data needs to be taken into account. When performing statistical analyses, the experimenter is, however, confronted with the decision between two divergent schools of thought in the field of statistics: Frequentist and Bayesian inference.
In Frequentist thought, the existence of true parameters θ_{true} with exact values is assumed and measured data is characterized as exact values y_i^* with a measurement error ϵ. Here, the goal is to obtain point estimators such as the maximum likelihood estimator (MLE) for the parameters based on measurements the uncertainty of which is expressed in terms of the deviation to their true counterparts. Measurement data is accordingly conceptualized as a random set of draws

which eventually with an increasing number of samples would converge towards the ground truth, hence the value of obtaining a point estimate. The pertaining uncertainty of a parameter y_i, then, manifests as a confidence interval (CoI) stated as a range with lower and upper bounds lb_i and ub_i, respectively, for which it holds that

$$p(\mathrm{lb}_i \leq \theta_{\mathrm{true},i} \leq \mathrm{ub}_i) = 1 - \alpha \tag{1.10}$$

Thus, at a confidence level of α, a CoI is an interval containing the true parameter value $100(1-\alpha)$ % of the time if the experiment were repeated with different data sets [108]. Fundamentally, probability in the Frequentist sense is intricately linked to frequency, hence the name. In practice, confidence intervals can be approximated with several methods one of which is referred to as profile likelihoods [109]. Here, the uncertainty of measurements is mapped onto the parameter values in a nonlinear fashion. The bounds lb_i and ub_i are then obtained by minimizing and maximizing y_i iteratively as long as a given value passes a likelihood ratio test [110]. This is oftentimes the method of choice in systems biology applications like ^{13}C-MFA. However, the interpretation of the resulting uncertainty quantification is complicated by the fact that different methods may generate different CoIs, especially when a parameter mapping is either highly nonlinear or close to an inequality constraint [110]. Additionally, errors arise both due to the approximative nature of CoI determination and for numerical reasons due to the optimization procedure [110].
In contrast, Bayesian thinking conceptualizes parameters as random variables with probability distributions and does away with the notion of true parameter values. The general concept of probability as a quantitative measure for the uncertainty of any observed or unobserved statement [108] allows hypothesizing even about non-repeatable or counterfactual events. Experimental data is then used to update parameters in a "reallocation of probabilities" [111] according to Bayes' theorem. The latter can be derived by first formulating the conditional probabilities of the parameters θ given the data y

$$p(\theta|y) = \frac{p(\theta \cap y)}{p(\theta)} \tag{1.11}$$

and of the data given the parameters

$$p(y|\theta) = \frac{p(y \cap \theta)}{p(y)} \tag{1.12}$$

Since $p(\theta \cap y) = p(y \cap \theta)$, equations 1.11 and 1.12 are combined to form Bayes' theorem [112, 113]

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{1.13}$$

Discussing equation 1.13 in more detail, the term $p(\theta)$ is referred to as the prior probability distribution (prior) of the parameters reflecting previous knowledge about them independent of the present experiment's data. By choosing a suitable distribution for a given parameter based on e.g. expert knowledge or previous experiments, the informativeness or strength of a prior can vary according to the degree of uncertainty assigned to it by the experimenter. For example, a horizontal line across a range of parameter values would constitute an uninformative, flat, so-called "uniform"

prior as it assigns the same probability density to all values within the range. Even at this conceptual stage, i.e. when formulating the priors and not having observed the actual experimental data, it is possible to check the model by performing a prior predictive check [114]. Here, the range of data, which can be accommodated by the model, is assessed by drawing replicated data y^{rep} from the priors according to [115]

$$p(y^{\mathrm{rep}}) = \int_{\theta} p(y^{\mathrm{rep}}|\theta)p(\theta)d\theta \tag{1.14}$$

The likelihood $p(y|\theta) = \mathcal{L}(y|\theta)$, then, represents a model which assigns probability to the data given the parameters thus linking the two.
By updating previous knowledge (priors) with new data via the likelihood, a posterior probability distribution (posterior) is obtained – a multivariate distribution describing the current knowledge about the parameters. Once new data is generated, this posterior may serve as a prior and be continuously updated and refined. Since the resulting distributions for single parameters are of relevance for an experimenter, a so-called marginal posterior distribution can be determined for each parameter θ_i. In case of a model with two parameters θ_1 and θ_2 and the posterior $p(\theta_1, \theta_2|y)$, the marginal posterior distribution of θ_1 is calculated by integrating out θ_2 according to [116]

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y)d\theta_2 \tag{1.15}$$

Finally, the denominator of Bayes' theorem $p(y)$ is called marginal likelihood or evidence and is defined as the probability of observing the data across all parameter ranges. Above all, $p(y)$ acts as a normalization term assuring that the resulting posterior is a valid probability distribution, i.e. that the sum in case of a discrete distribution or the integral in case of a continuous distribution amount to 1 [117]. Mathematically, this is expressed with the integral

$$p(y) = \int_{\theta} p(y|\theta)p(\theta)d\theta \tag{1.16}$$

the calculation of which is practically unfeasible for continuous parameters due to its high dimensionality [118]. Hence, the posterior cannot be calculated exactly in a direct manner but instead has to be obtained indirectly by approximation, i.e. by drawing samples from it. More specifically, only the numerator term of equation 1.13 is sampled which corresponds to the simplification of Bayes' theorem to

$$p(\theta|y) \sim p(y|\theta)p(\theta) \tag{1.17}$$

Since the denominator term is not dependent on the parameters, the numerator suffices to determine the shape of the posterior distribution represented by the relative number of draws at one point compared to others [118]. Accordingly, absolute values of the probability density are not necessary to estimate the posterior and its summary statistics. Importantly, for the same reasons of the computational intractability of the denominator and the inefficiency of the pertaining algorithms, independent sampling cannot be performed. Instead, dependent samples are drawn using Markov Chain Monte Carlo (MCMC) methods.
Monte Carlo sampling is an umbrella term for algorithms simulating random numbers from a target distribution, thereby being able to approximate integrals [112]. Such a method is utilized to receive

an estimated distribution of the posterior from which sets of parameters are drawn sequentially in multiple independent Markov chains. Each chain is initiated at a starting point θ^0 whereupon the following values $\theta^1, \theta^2, \theta^3, ..., \theta^n$ are drawn from a transition distribution $T_t(\theta^t, \theta^{t-1},)$ [119]. First-order Markov chains are sequences of such parameter draws where the next value θ^{t+1} depends only on the current value θ^t. More significant than this so-called Markov property, however, is the possibility to create various Markov chains for any posterior - including unnormalized ones - which by iteratively drawing from a distribution converge towards said posterior. This property was proven by showing that the Markov chain has a unique stationary distribution, i.e. a unique solution, and that this solution is equal to the posterior [119, 120].

In order to perform MCMC, a simulation algorithm is needed like e.g. the commonly used ones belonging to the Metropolis-Hastings family of algorithms. Generally, a Metropolis algorithm [121] works by first drawing the starting value θ_0 with a probability given the data greater than 0 from a starting distribution $p_0(\theta)$. Then, in an iterative process proposals θ^* are sampled from a proposal distribution $J_t(\theta^*|\theta^{t-1})$ dependent on the previous point $t-1$ and are accepted or rejected based on the ratio of probability densities

$$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)} \tag{1.18}$$

In case of increasing probability density, the proposal is always accepted, and upon decreasing it is only accepted with a probability equal to r [119]. Since the chains' starting points are probably not close to the posterior and it may require numerous samples to reach stationarity, the early sampling phase is sometimes denominated as warm-up and the pertaining samples as tuning samples. Especially for models with an increasingly large number of parameters, the performance of the Metropolis algorithm with regards to convergence suffers due to inefficient random walk behavior [120]. Hamiltonian Monte Carlo algorithms improved on this by introducing Hamiltonian dynamics [122, 123] and subsequently No U-Turn Samplers (NUTS) were developed removing the need for manually tuning the step size parameter [124]. This combination of innovations thus tackled issues of efficiency and usability.

As convergence to the stationary distribution and thus the posterior is critical, a metric to monitor this is the potential scale reduction factor $\hat{R}$ determined in dependence of the ratio of an estimated marginal posterior variance $\hat{\mathrm{var}}^+(\theta|y)$ and the within-sequence variation W according to

$$\hat{R} = \sqrt{\frac{\hat{\mathrm{var}}^+(\theta|y)}{W}} \tag{1.19}$$

Due to the difference in starting points and since $\hat{\mathrm{var}}^+(\theta|y)$ contains the between-sequence variance, it will generally overestimate the marginal posterior variance at the start of the process while the within-sequence variation tends to underestimate it. When having reached stationarity and approaching convergence, both parameters close in on the same value so that $\hat{R}$ converges towards 1 for unlimited samples. While $\hat{R}$ is above a certain threshold, the between- and within-sequence variations imply an improved inference result upon further sampling [119]. When originally proposed, an $\hat{R}$ of 1.1 or lower was recommended to indicate proper convergence [125] but after improvements on the statistics and continual insights during usage this was lowered to 1.01 [126]. Returning to the Markov property or dependence of each draw on its direct predecessor, draws

within a Markov chain are subject to autocorrelation and the number of samples cannot simply be interpreted as the sample size of independent draws. To obtain a measure for this important metric, nonetheless, the effective sample size (ess) was defined as

$$\mathrm{ess} = \frac{mn}{-1 + 2\sum_{t'=0}^{k} \hat{\rho}_{t'}} \tag{1.20}$$

with m Markov chains, n samples, and $\hat{\rho}_{t'}$ as the estimated autocorrelations [126]. The ess should only be calculated after the warm-up once the Markov chains have reached their stationary distribution.

Having obtained a posterior distribution and since the assumption of the existence of true parameters is omitted, the summary statistics communicating the results of an inference differ from the MLE and CoI of Frequentist affiliation. The Bayesian equivalents to the former are the maximum a posteriori, which is the global maximum of the posterior's probability density, and the expected value which corresponds to the mean of the posterior's density [110]. To represent the uncertainty of such metrics, credible intervals (CrIs) are defined as [127]

$$p(\mathrm{lb}_i \leq \theta_i \leq \mathrm{ub}_i) = 1 - \alpha \tag{1.21}$$

and differ from CoIs in that they contain $100(1-\alpha)$ % of the probability density of a given parameter. In other words, the CrI contains its parameter with a probability of $1 - \alpha$.

Another evaluation enabled by the knowledge of the posterior, is posterior predictive checking (ppc). Here, the posterior predictive distribution

$$p(\hat{y}|y) = \int p(\hat{y}|\theta)p(\theta|y)d\theta \tag{1.22}$$

is approximated iteratively by sampling parameter values θ from the posterior and subsequently using these to obtain predicted data points $\tilde{y}$ from the likelihood [107]. There are several use cases for $\tilde{y}$, most crucially the comparison with the original measurement data set to gauge whether it can be explained by the model as well as the prediction of future data sets [107, 119].

Bayesian methods have been on the rise recently, not least due to the ease of accessibility to a level of computational power even with just a desktop computer not attainable in the past enabling the execution of demanding sampling algorithms. Simultaneously, Python packages such as PyMC [128] and hopsy [129] among others and samplers have become ever more efficient thus lowering the cost of entry. Due to the variability of results for CoIs depending on the utilized method and the ability to infer CrIs to an in theory arbitrary precision [110], Bayesian methods have a palpable advantage but since many software frameworks currently rely on the Frequentist paradigm, further developments on the part of Bayesian tools are necessary.

Accordingly, in this thesis both approaches will be used depending on the scenario and contributions to enable Bayesian inference for evaluating biological experiments will be made.

1.4.3 ^{13}C-MFA

The aforementioned method of ^{13}C-MFA (1.1) integrates experimentally determined extracellular rates with ^{13}C-labeling data to quantify an organism's intracellular reaction rates. As the rates of enzymatically catalyzed reactions or transportation steps are denominated as "fluxes", the totality of a system's fluxes yields a "fluxome" and accordingly another term for ^{13}C-MFA is "quantitative fluxomics" [3]. Within the larger spectrum of omics techniques, it stands apart in that a) the intracellular fluxes are not directly measurable but instead inferred with a computational procedure and b) the obtained fluxome represents the effective phenotype resulting after all layers of regulation in contradistinction to the cellular potential uncovered by other omics approaches. For example, it has been shown that in *C. glutamicum* ATCC13032 the CCM enzyme concentrations measured by proteomics were not indicative of their associated flux values [101]. Due to this unique property, ^{13}C-MFA can be used in phenotyping experiments to e.g. identify bottlenecks, thereby facilitating rational strain development for bioprocesses.
Regarding the requirements for conducting ^{13}C-MFA, as initially stated the extracellular concentrations of all carbon sources, products, and by-products have to be measured and used for the estimation of extracellular rates via a bioprocess model. A further prerequisite is constituted by a valid metabolic network model of the organism's CCM including all reactions potentially influencing the metabolite labeling patterns. Since the combination of a network model and extracellular rates forms an underdetermined system, the innovation of ^{13}C-MFA is the addition of metabolite labeling data to reduce the remaining degrees of freedom (rDOF) to 0 and resolve fluxes inaccessible with a purely stoichiometrical MFA.
When conducting an ILE, two distinct types of steady-states have to be considered. The first is the metabolic steady-state (MSS) which is presumed to occur in a chemostat, i.e. a continuous cultivation, and during the exponential growth phase of a batch experiment [3]. It is characterized by funneling imported carbon towards maximum growth and thus all pool sizes and fluxes are assumed to be constant. Realistically, the first time derivative of a given metabolite's pool size and connected fluxes might not amount to exactly 0 but at least approach it or be significantly lower than the flux values themselves, thus amounting to a quasi-MSS.
The second is the isotopic steady-state (ISS) referring to the equilibrium of a given metabolite's labeling pattern which is established some time after the onset of label incorporation. During the preceding isotopically transient or instationary state, some isotopomers may be observed which are washed out before reaching the ISS and thus cannot be observed in isotopically stationary ILEs. The duration until ISS is highly dependent on the metabolite in question, its proximity within the metabolic network to the labeled substrate(s) and the fluxome.
Having defined basic terminology, the following general assumptions must hold in order to conduct ^{13}C-MFA:

1. Molecules are homogeneously distributed both intra- and extracellularly.
2. During the ILE, the MSS assumption must hold true.
3. All relevant reactions and their carbon transitions are known.

4. There are no kinetic mass effects, i.e. all isotopomers of an enzymatic educt are indiscriminately catalyzed at the same reaction velocity.
5. The experimental measurements do not alter the measured quantities.
6. The experimental measurements do not alter the conditions of the cultivation.

Points 1 and 4 ensure the equal treatment of differently labeled molecular species by enzymes which clearly needs to be the case when intending to link labeling states with fluxes. The analytical demands stated in points 5 and 6 refer to all measurements associated with the ILE including biomass, exhaust gas analysis, supernatant sampling, and sampling to determine labeling states. Non-invasive online measurements such as exhaust gas analysis in a bioreactor and backscatter in a microbioreactor are generally uncritical. In terms of sampling from a cultivation, the sample volume should generally either be low enough relative to the total cultivation volume to be insignificant or the cultivation vessel may be sacrificed, i.e. discontinued after taking the sample. When increasing the degree of miniaturization and parallelization, sacrifice sampling presents a valid option. With respect to quenching, the decomposition of targeted metabolites could prevent their detection, alter their pool size or in the worst case be mass-dependent thus influencing the labeling state.
These original requirements and assumptions have been partly amended due to more recent developments. Firstly, the rise of isotopically non-stationary (INST) ^{13}C-MFA introduced the time-resolved measurement of INST labeling patterns following the administration of a labeled substrate mixture [130]. As the resulting time-course of label incorporation is caused both by the associated fluxes as well as the metabolite pool size, the latter quantity has to be measured. Secondly, there have been forays into a fully dynamic ^{13}C-MFA where neither the metabolic nor the isotopic steady-state are enforced [131]. Nonetheless and to re-iterate, the established techniques of isotopically stationary and INST ^{13}C-MFA still require the MSS.
The advantages of INST justifying a decision against the fully stationary approach comprise a shorter experimental duration as there is no need to wait for the ISS and a significantly reduced amount of expended labeled substrate per experiment which also diminishes the associated costs. Additionally while the measurement of at least some metabolite pool sizes is required, even more are estimated thus increasing the informational content of the final result.
The fundamental structure of a ^{13}C-model will be illustrated with a toy example (figure 1.7) inspired by [3, 132]. As portrayed in the example, some reactions are assumed as unidirectional or irreversible and some as bidirectional or reversible. When the Gibbs free energy difference of a reaction is low, i.e. the reaction operates close to a thermodynamic equilibrium, it can be considered reversible and when said difference is large, the model can be simplified by assuming irreversibility *in vivo* [133]. As the reversibility of reactions influences the resulting metabolite labeling patterns, the inclusion of labeling data allows inference of both directions [133] whereas imposing unidirectionality can introduce structural errors.
Instead of defining forward ($v^{\rightarrow}$) and backward ($v^{\leftarrow}$) reactions, the notion of net fluxes [134]

$$v_i^{\mathrm{net}} = v_i^{\rightarrow} - v_i^{\leftarrow} \tag{1.23}$$

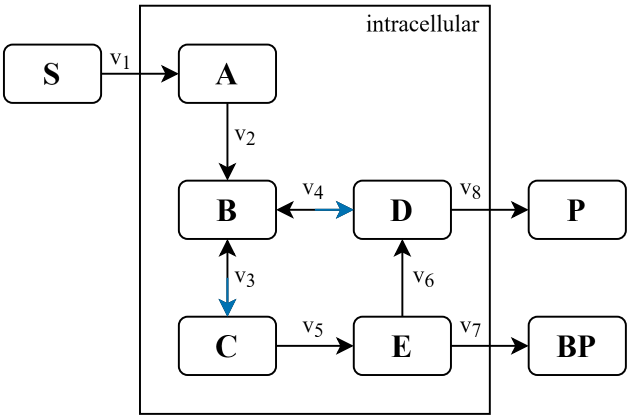


Figure 1.7: Small toy example of a metabolic network with a substrate S, a by-product BP, and a product P as well as the intracellular metabolites A, B, C, D, and E. Net directions of reversible reactions are indicated with blue arrows.

and exchange fluxes

$$v_i^{\text{xch}} = \min(v_i^{\rightarrow}, v_i^{\leftarrow}) \tag{1.24}$$

for a given reaction i has emerged [133]. In case of the toy example, the net directions of reversible reactions v_3 and v_4 have been highlighted with superimposed blue arrows.
A ^{13}C-model features two kinds of mass balances with distinct system boundaries. The first type is a mass balance around each total metabolite pool yielding a system of ODEs

$$\frac{d}{dt}c_A = v_1 - v_2 \tag{1.25a}$$
$$\frac{d}{dt}c_B = v_2 - v_3 - v_4 \tag{1.25b}$$
$$\frac{d}{dt}c_C = v_3 - v_5 \tag{1.25c}$$
$$\frac{d}{dt}c_D = v_4 + v_6 - v_8 \tag{1.25d}$$
$$\frac{d}{dt}c_E = v_5 - v_6 - v_7 \tag{1.25e}$$

Due to the MSS assumption, the ODE system is effectively simplified to a system of linear equations as the pool sizes and fluxes are constant over time, i.e. $\frac{d}{dt}c_i = \frac{d}{dt}v_i = 0$. In contradistinction, the extracellular pools do not exhibit a constant pool size and are instead defined as

$$\frac{d}{dt}c_S = -v_1 \tag{1.26a}$$
$$\frac{d}{dt}c_P = v_8 \tag{1.26b}$$
$$\frac{d}{dt}c_{\text{BP}} = v_7 \tag{1.26c}$$

These systems can be jointly expressed in matrix form using the stoichiometric matrix $\mathbf{N}$ containing only stoichiometric coefficients and the measurement matrix $\mathbf{M}_r$ 1.27. Here, each row of $\mathbf{N}$ pertains to a metabolite pool and each column to a particular flux. By multiplication with the flux

vector, the metabolite mass balances stated in equations 1.25a - 1.25e are recovered.

$$\begin{pmatrix} 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & -1 & -1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & -1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & -1 \\ \cdot & \cdot & \cdot & \cdot & 1 & -1 & -1 & \cdot \\ -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{pmatrix} = \begin{pmatrix} \mathbf{N} \\ \mathbf{M}_r \end{pmatrix} \vec{\mathbf{v}} = \begin{pmatrix} 0 \\ \mathbf{r} \end{pmatrix} \tag{1.27}$$

The degree of freedom (DOF) corresponding to the number of free fluxes of this system can be computed by

$$\mathrm{DOF} = \dim(\mathbf{v}) - \mathrm{rank}(\mathbf{N}) \tag{1.28}$$

in dependence of the rank, i.e. the number of independent linear equations, of $\mathbf{N}$.
For the second type of mass balance the system boundaries are constricted to the fraction of the pool size pertaining to a given isotopomer and thus incorporate labeling data. As these constitute the labeling mass balances, their formulation is dependent on which assumptions regarding the ISS and MSS are enforced. Consequently, they exist in three layers of complexity according to the previously established fully dynamic, INST, and fully stationary variants.
Generally, the pool size of just one labeled species of metabolite E is defined as the product of its fractional enrichment e_1 and the pertaining total pool size c_E [132]

$$\frac{d}{dt}[e_1(t)\ c_E(t)] = c_E(t)\frac{d}{dt}e_1(t) + e_1(t)\frac{d}{dt}c_E(t) \tag{1.29}$$

The right-hand side of the equation, then, contains products of all incoming and outgoing flux rates each with the pool size of its educt labeling species. Considering only the uniformly labeled mass trace of metabolite E, the three variants are expressed by

$$c_E(t)\frac{d}{dt}e_1(t) + e_1(t)\frac{d}{dt}c_E(t) = v_5(t)c_1(t) - v_6(t)e_1(t) - v_7(t)e_1(t) \tag{1.30a}$$

$$c_E\ \frac{d}{dt}e_1(t) = v_5c_1(t) - v_6e_1(t) - v_7e_1(t) \qquad \frac{d}{dt}c_E = 0 \tag{1.30b}$$

$$0 = v_5c_1 - v_6e_1 - v_7e_1 \qquad \frac{d}{dt}c_E = \frac{d}{dt}e_1 = 0 \tag{1.30c}$$

in descending order of complexity. Here, the lower case letters denominate fractional labeling enrichment and their subscript 1 a fully labeled isotopomer. As only the fully dynamic variant (equation 1.30a) does not assume a metabolic steady-state, the metabolite pool sizes and fluxes are time-dependent. In the INST case (equation 1.30b), only the labeling states remain time-dependent and in the fully stationary case (equation 1.30c), pool sizes, fluxes, and labeling states are constant.
Alternative formulations of labeling balances like cumomers meaning cumulative isotopomers [100] and elementary metabolite units (EMUs) [135] have been developed to increase computa-

tional efficiency. When using cumomers, isotopomers are united in fractions based on the labeling of certain positions. For example, the 1-cumomer fraction of the amino acid Leu with 6 carbon atoms would be denominated as Leu_{1XXXXX} and contain all isotopomers labeled at the first position without regards to whether the residual positions each portrayed by an X contain labels or not [100].
Aside from the mass balances, a mechanism mapping the carbon transitions of reactions needs to be in place to propagate their distribution throughout the metabolic network. This can be realized with a carbon transition model which may be formulated in FluxML [136].
When a model comprising these three components (metabolite mass balances, labeling balances, and carbon transitions) has been built and given the fully stationary case (equation 1.30c), flux solutions for $\vec{\mathbf{v}}$ in equation 1.31 can be calculated by determining the kernel matrix $\mathbf{K}$ and solving [137]

$$\vec{\mathbf{v}} = \begin{pmatrix} \vec{\mathbf{v}}_{\mathrm{dep}} \\ \vec{\mathbf{v}}_{\mathrm{free}} \end{pmatrix} = K \begin{pmatrix} 1 \\ \vec{\mathbf{v}}_{\mathrm{free}} \end{pmatrix} \tag{1.31}$$

such that

$$\mathbf{NK} = 0 \tag{1.32}$$

In this process, the flux vector $\vec{\mathbf{v}}$ is subdivided into free and dependent fluxes $\vec{\mathbf{v}}_{\mathrm{free}}$ and $\vec{\mathbf{v}}_{\mathrm{dep}}$, respectively, and the kernel matrix encodes linear combinations of the dependent fluxes [137].
Having obtained a flux solution, a simple forward simulation can be performed to determine its resulting metabolite labeling states. However, the inference of fluxes constitutes the inverse problem so an optimization procedure is conducted where the simulated labeling states and their experimentally measured counterparts are compared with a metric like the weighted sum of squared residuals (SSR) and their differences are minimized through iterative parameter fitting. This least squares regression [138] comprises the initially mentioned computational method of ^{13}C-MFA and is defined as

$$\text{minimize SSR} = \sum \frac{(r_i - \overline{r}_i)^2}{\sigma_{r_i}^2} + \sum \frac{(x_i - \overline{x}_i)^2}{\sigma_{x_i}^2} \tag{1.33}$$

in dependence of the flux rate r_i and the fractional labeling enrichment x_i. In the INST case this equation is expanded to contain the SSR of pool sizes, as well. It is important to note that utilizing the SSR metric implicitly carries another assumption, namely that of a normally distributed error as the SSR calculation requires a mean and a standard deviation.

Summarizing the ^{13}C-MFA approach, in the most general way one could portray the INST case as [130, 139]

$$\mathrm{diag}(\mathbf{C}_M)\dot{\mathbf{x}} = f(\vec{\mathbf{v}}, \mathbf{x}^{\mathrm{input}}, \mathbf{x}) \tag{1.34a}$$

$$\mathbf{y}_i = \mathbf{g}(t_i, \dot{\mathbf{x}}, \vec{\mathbf{v}}, \mathbf{C}_M, \mathbf{x}^{\mathrm{input}}) \tag{1.34b}$$

$$\mathbf{Y} = \mathbf{h}(\mathbf{C}_M) \tag{1.34c}$$

$$\mathbf{w} = \mathbf{k}(\vec{\mathbf{v}}) \tag{1.34d}$$

The forward simulation is performed by solving equation 1.34a and the following comprise the additional data to be considered during parameter fitting, i.e. INST labeling data $\mathbf{y}_i$ (equation 1.34b), pool sizes $\mathbf{Y}$ (equation 1.34c), and extracellular rates $\mathbf{w}$ (equation 1.34d) [130]. Here, $\mathbf{C}_M$ is a matrix containing pool size measurements and its product with the transient labeling state vector $\dot{x}$ is equivalent to the left part of equation 1.30b. For the metabolically and isotopically stationary case this is simplified to

$$0 = f(\vec{\mathbf{v}}, \mathbf{x}^{\mathrm{input}}, \mathbf{x}) \tag{1.35a}$$

$$\mathbf{y}_i = \mathbf{g}(\mathbf{x}, \vec{\mathbf{v}}, \mathbf{x}^{\mathrm{input}}) \tag{1.35b}$$

$$\mathbf{w} = \mathbf{k}(\vec{\mathbf{v}}) \tag{1.35c}$$

in accordance with equation 1.30c. When performing the parameter fitting, it is common practice to use a multi-start method with varying initial parameters since the starting location may influence the regression and the final result. There is always the possibility of identifying a local instead of the global optimum as the best solution and this risk is decreased by using many starting points. After obtaining an optimized fit, Frequentist statistics are the most commonly implemented choice for uncertainty quantification via linearized statistics, profile likelihoods or Monte Carlo sampling [110] in currently available software suites. One reason for this circumstance is that these methods interconnect well to the input data for ^{13}C-MFA. Extracellular fluxes – despite being usually obtained through parameter fitting with a separate bioprocess model where they are defined as parameters – are treated as measurements in the ^{13}C-model. Accordingly, they already come with a mean value, i.e. MLE, and a standard deviation used for mapping uncertainty to the fluxes and pools (see 1.4.2). Labeling data is provided as points and the standard deviation originates from replicate measurements or is chosen in a way that reflects the measurement errors expected from the utilized experimental and analytical workflows. Since metrics like the SSR require normally distributed data, software for ^{13}C-MFA is not equipped to readily accept data with different distributions which may occur in Bayesian statistics. On the other hand, this constitutes – as stated previously – another assumption which could be removed or at least validated by switching to the Bayesian paradigm. A truly Bayesian approach would, however, necessarily have to start with the raw data, i.e. at the level of bioprocess modelling and peak data evaluation, thus requiring another data pipeline entirely. Therefore, in the status quo it is most practical to work with Frequentist methodology.

1.5 Motivation and outline of the thesis

Isotopic labeling experiments have granted deep insight into the metabolism of many organisms. They come in different shapes and forms ranging from qualitative measurements with specific tracer molecules to quantitative methods such as fluxomics. Yet, despite the value of the procured data in e.g. aiding rational strain design, their use as a regularly applied screening technique in biotechnology is inhibited by virtue of their low throughput, high associated cost, and complexity.
To enable conducting automated ILE at a microliter scale, a suitable quenching method constitutes a prerequisite when analyzing free metabolites and is certainly advantageous even when targeting proteinogenic amino acids. Consequently, the development of such a method is the first aim of the present dissertation and represents the basis for the subsequent automation of the ILE experimental workflow to build an integrated pipeline boasting a much increased throughput. Beyond productivity, automation grants standardization almost as a positive side effect by the need of establishing scripts and protocols and can be utilized to reduce complexity, thereby lowering the barrier of entry – qualities which would benefit the field of ILE like few others. Due to the differences in measurement capabilities between lab-scale bioreactors and microbioreactors the latter of which are used throughout this thesis, it is demonstrated that valid isotopically stationary and instationary labeling data of free metabolites can be obtained, interpreted, and even used to conduct ^{13}C-MFAs. To this end, *C. glutamicum* is utilized as a model organism as per its success as an industrial workhorse as well as its prominence in biotechnological research in general and in systems biology in particular.
Since the experimental section is not the only bottleneck in ILE, the raw data evaluation after analysis of labeling states is addressed by developing successive, interlocked Python programs performing the necessary modelling, computation, and visualization tasks. Innovative modelling solutions for chromatographic peak data characterization by Bayesian inference and a data pipeline for estimating intracellular metabolite pool sizes are created and established on a Airflow computation cluster for increased performance and parallelization.
Finally, the assembled ILE pipeline culminating from the described efforts will be presented and critically discussed alongside future perspectives in the field of ILEs.

2 Material and Methods

2.1 Microbial strains

All experiments in the present dissertation except for the INST ILEs on ethanol were conducted with *C. glutamicum* ATCC13032 (WT). For the ethanol labeling experiments, cryo stocks of the strain *C. glutamicum* WT_EtOH-Evo were kindly provided by Lars Halle. The strain itself was obtained by adaptive laboratory evolution from the WT to enable more efficient growth on ethanol as the sole carbon source [140].

2.2 Strain maintenance

Cryo stocks for the WT were produced by performing a cultivation in a 500 mL shaking flask with 50 mL filling volume at 300 rpm and 30 °C. During the exponential growth phase, cells were harvested, centrifuged at 13000 rpm for 5 min, washed with 0.9 % NaCl solution, centrifuged again, and finally resuspended in a pre-cooled NaCl-glycerol mixture containing 20 % (v/v) glycerol. Subsequently, the cells were frozen at - 20 °C until being thawed immediately preceding experimental usage.

2.3 Growth media

When not explicitly stated otherwise, all presented experiments were conducted with CGXII medium [141] composed of 20 g L^{-1} D-glucose, 42 g L^{-1} 3-(N-morpholino)propanesulfonic acid (MOPS) buffer, 5 g L^{-1} urea, 20 g L^{-1} ammonium sulfate, 1 g L^{-1} KH_2PO_4, 1 g L^{-1} K_2HPO_4, 13.25 mg L^{-1} $CaCl_2$ • $2H_2O$, 0.25 g L^{-1} $MgSO_4$ • $7H_2O$, 10 mg L^{-1} $FeSO_4$ • $7H_2O$, 10 mg L^{-1} $MnSO_4$ • H_2O, 0.02 mg L^{-1} $NiCl_2$ • $6H_2O$, 0.313 mg L^{-1} $CuSO_4$ • $5H_2O$, 1 mg L^{-1} $ZnSO_4$ • $7H_2O$, 0.2 mg L^{-1} biotin, and 30 mg L^{-1} protocatechuaic acid.
For the ILE pertaining to case study I (3.5), i.e. the estimation of pool sizes, a minimized variant of this CGXII medium was utilized in order to decrease the complexity of the biological matrix of samples and thus obtain improved results from LC-MS/MS analysis. This variant deviated from the aforementioned CGXII medium only by omitting MOPS buffer completely and reducing the concentration of ammonium sulfate by 90 % to 2 g L^{-1}.
Cultivations for the ILEs on ethanol serving as case study II (3.6) were conducted with CGXII medium with 1 %(v/v) ethanol instead of D-glucose.

2.4 Components of the robotic platforms or Mini Pilot Plants

The robotic platforms or Mini Pilot Plants used in this work all consisted of a Tecan Freedom Evo 200 liquid handler (Tecan Deutschland GmbH, Crailsheim, Germany) equipped with a liquid han-

dling (LiHa) arm with 8 fixed steel tips with a Teflon coating and a robotic manipulator (RoMa) arm and interfaced with numerous third party devices forming a general framework granting the freedom to perform various kinds of biological experiments. For cultivations, a microbioreactor – either a BioLector I, II or Pro (Beckman Coulter GmbH, Baesweiler, Germany) – was included so that samples could be taken mid-cultivation with the LiHa in an autonomous manner. Further devices were an automated centrifuge, i.e. a Sigma 4-5KRL centrifuge (Sigma Laborzentrifugen GmbH, Osterode am Harz, Germany) or a Hettich Rotanta 460 Robotic centrifuge (Andreas Hettich GmbH & Co. KG, Tuttlingen, Germany), for sample processing, a BioShake 3000T-elm (QInstruments GmbH, Jena, Germany) for shaking and heating microtiter plates (MTPs), a spectrophotometer Tecan Infinite M Nano$^+$ or Tecan Infinite 200 PRO (Tecan Deutschland GmbH, Crailsheim, Germany), and a cryostate Lauda MC 600 (Lauda Dr. R. Wobser GmbH & Co. KG, Lauda-Königshofen, Germany) for on-deck cooling of MTPs.

2.5 Automated ILEs

For all automated ILEs, pre-cultures were conducted in 4-baffled 500 mL shaking flasks with filling volumes of 50 mL at a temperature of 30 °C and shaking frequencies of either 250 rpm or 300 rpm. Usually, they were inoculated in the early evening and run over-night before being used in the following morning for inoculation of the main culture performed in a BioLector. After measuring the optical density at a wavelength of 600 nm (OD_{600}) and calculating the necessary inoculation volume via

$$V_{\text{inoculation}} = \frac{\text{OD}_{600,\text{ pre-culture}} V_{\text{pre-culture}}}{\text{OD}_{600,\text{ main culture}}} \tag{2.1}$$

the main culture was inoculated and transferred to a FlowerPlate manually. Before inoculation, samples from the pre-culture were centrifuged at > 4000 g for 5 min at 4 °C, washed with phosphate buffer saline (PBS), centrifuged again, and finally the cell pellet was resuspended in main culture medium.
The BioLector protocol for the cultivations of all *C. glutamicum* strains used a temperature of 30 °C and a shaking frequency of 1400 rpm. A backscatter gain of 20 was applied for the BioLector I and 3 for the BioLector II and Pro.

2.5.1 Hot isopropanol quenching: Validation

The original experiment used 12 biological replicates published in [142] but later-on a repeat experiment was conducted with 6 biological replicates. The results presented in this dissertation originate from the repeat experiment so this section accordingly details the setup of this experiment.
The main culture was inoculated targeting a starting OD_{600} of 1. Each well had a filling volume of 800 µL. During the mid-exponential growth phase, samples were taken concomitantly from all replicates and quenched using automated hot isopropanol quenching as detailed in section 3.1 and published in literature [142]. To investigate whether residual enzyme activity was present

during quenching or the subsequent sample processing via centrifugation, $U^{13}C$ D-glucose was added to the isopropanol solution to a final concentration of 0.2 g L^{-1}. The principle behind this approach to validating a quenching method has been previously described [21, 143, 144]. The samples were stored at - 20 °C until LC-MS/MS analysis of the metabolite labeling states.

2.5.2 Hot isopropanol quenching: INST proof of concept

This experiment was conducted separately for amino acids and CCM intermediates. Both experiments were mostly identical in conditions and execution, though. The only exceptions were that the amino acid experiment was conducted on the robotic platform Frank with an intended starting OD_{600} of 1.5 while the experiment for the CCM intermediates was performed on the robotic platform Pahpshmir with an intended starting OD_{600} of 1.
All wells pertaining to the INST-ILE had an initial filling volume of 750 µL and during the mid-exponential growth phase, a pulse of 50 µL of a 80 g L^{-1} 100 % $U^{13}C$ D-glucose solution was administered to the wells as described in detail in section 3.1.3. The delays between pulsing and quenching were 25 s, 30 s, 40 s, 50 s, 60 s, 70 s, 90 s, 120 s and 300 s for the amino acid experiment and 25 s, 28 s, 30 s, 35 s, 40 s, 45 s, 55 s, 85 s, 115 s and 205 s for the CCM intermediates experiment. All given time points were sampled in biological triplicates. After pulsing and quenching groups of wells successively in a column-wise manner, the samples were stored at - 20 °C until LC-MS/MS analysis of the metabolite labeling states.

2.5.3 ILE for pool size estimation

This experiment used the robotic platform Frank. Main cultures had an initial volume of 750 µL per well until the mid-exponential growth phase, when a pulse of 50 µL of a 80 g L^{-1} 100 % $U^{13}C$ D-glucose solution was administered. The delays between pulsing and quenching amounted to 20 s, 25 s, 40 s, 50 s, 60 s, 75 s, 90 s, 105 s, 140 s, 215 s, 315 s, 440 s, and 690 s. After pulsing and quenching groups of wells successively in a column-wise manner, the samples were stored at - 20 °C until LC-MS/MS analysis of the metabolite labeling states.

2.5.4 Ethanol ILEs

Except for the substrate input labeling mixture, both separately conducted INST ILEs with *C. glutamicum* WT_EtOH-Evo were identical with regards to experimental conditions. The first experiment used the WT and WT_EtOH-Evo strains with 100 % 1-^{13}C-ethanol (Cambridge Isotope Laboratories, Andover, MA 01810 USA; 99 % purity) as the tracer molecule. The second experiment used only WT_EtOH-Evo but the wells were subdivided into two groups receiving either 100 % 2-^{13}C-ethanol or 100 % $U^{13}C$-ethanol (Santa Cruz Biotechnology, Inc., Heidelberg, Germany; 99 % purity). However, the evaluation of the second ethanol ILE is beyond the scope of the present thesis but since its design was a product of the established workflow, its experimental details have been included here for the sake of completeness. The unlabeled ethanol supplied initially was ROTIPURAN Ethanol $\geq$ 99.8 % p.a. (Carl Roth GmbH + Co. KG, Karlsruhe, Germany) and the cultivations were inoculated with an intended starting OD_{600} of 0.5. The delays between pulsing

and quenching amounted to 24 s, 35 s, 60 s, 120 s, 180 s, 1200 s, and 1800 s for first ILE. Based on the results of the first ILE, the time points were optimized for the second ILE and set to 20.6 s, 47.9 s, 59.4 s, 111.3 s, 540.6 s, 875.6 s, 1995.8 s, and 2703.2 s. The sequence of pulsing, sampling, and quenching was scheduled in such a manner as to minimize the temporal spread of pulsing events, thus keeping the remaining unlabeled ethanol concentration as even as possible. After pulsing and quenching groups of wells, the samples were stored at - 20 °C until LC-MS/MS analysis of the metabolite labeling states.

2.6 LC-MS/MS analyses

All LC-MS/MS analyses were conducted with an Agilent 1260 Infinity II HPLC system (Agilent Technologies, Waldbronn, Germany) connected to a Sciex TripleTOF6600 QqTOF device (AB Sciex Germany GmbH, Darmstadt, Germany) equipped with a Turbo V ion source.

2.6.1 LC-MS/MS analysis of free amino acids

For the analysis of labeling states of free amino acids, extracts obtained from hot isopropanol quenching were diluted 1:4 in double-distilled water (DDW) before injection at a volume of 5 µL. The chromatographic separation was performed with a 150 mm x 2 mm Phenomenex Luna SCX column (Phenomenex Ltd., Aschaffenburg, Germany) with a pore size of 100 Å and a particle size of 5 µm preceded by a 4 x 2 mm SCX Security Guard cartridge (Phenomenex Ltd., Aschaffenburg, Germany). For the gradient elution, a 5 % (v/v) acetic acid solution (A) and a 15 mM ammonium sulfate solution adjusted to pH 6 with 100 % acetic acid (B) were pumped at a flow rate of 0.4 mL min^{-1} and a temperature of 60 °C according to the following gradient: 15 % B at 0 min, 15 % B at 10 min, 100 % B at 16 min, 100 % B at 28 min, 15 % B at 30 min, 15 % B at 35 min. This method has been described and validated previously [72]. Newly optimized ion source parameters are listed in table 2.1 and the amino acid-specific parameters collision energy and declustering potential in table 2.2.

Table 2.1: Ion source parameters for the LC-MS/MS analysis of free amino acids.

parameter	**value**
gas source 1	45 psi
gas source 2	65 psi
curtain gas	25 psi
temperature	630 °C
IonSpray Float Voltage	4600 V

Table 2.2: Analyte-specific collision energy and declustering potential values for the LC-MS/MS analysis of free amino acids. The most intense product ion is displayed in bold print.

analyte	precursor ion m/z ratio / Da	declustering potential (DP) / V	collision energy (CE) / V	product ion m/z ratio(s) / Da
aspartate	134	36	21	**74**, 43, 46, 70, 88, 116
glutamate	148	46	21	**84**, 130, 56
threonine/homoserine	120	41	17	**74**, 56
serine	106	31	15	**60**, 70, 88
glutamine	147	46	23	**84**, 130, 56
tyrosine	182	46	19	**136**, 165, 123
glycine	76	36	11	**30**, 48, 59
proline	116	46	23	**70**
methionine	150	46	15	**104**, 133, 56
alanine	90	36	21	**44**
valine	118	41	15	**72**, 55
phenylalanine	166	41	19	**120**
isoleucine/leucine	132	36	17	**86**, 69 (only Ile)
tryptophane	205	41	15	**146**, 188
lysine	147	41	15	**84**, 130
histidine	156	46	21	**110**
arginine	175	51	35	**70**, 60
ornithine	133	46	31	**70**
citrulline	176	31	41	**70**, 113
G6P	259	- 20	- 15	**97**, 259, 199, 169, 139, 79
FBP	339	- 40	- 20	**339**
X5P/R5P	229	- 20	- 18	**97**, 79, 139, 161, 229
S7P	289	- 80	- 15	**289**, 199, 97
GAP	169	- 20	- 6	**169**, 97

2.6.2 LC-MS/MS analysis of free CCM intermediates

For the analysis of labeling states of free CCM intermediates, extracts obtained from hot isopropanol quenching were diluted 1:4 in a 60 % (v/v) acetonitril-DDW solution before injection at a volume of 5 µL. Chromatographic separation was performed with a hydrophilic interaction liquid chromatography (HILIC) technique using a 150 mm x 2.1 mm SeQuant ZIC-pHILIC peek coated HPLC column (Merck KGaA, Darmstadt, Germany) with a particle size of 5 µm preceded by a 20 x 2.1 mm SeQuant ZIC-pHILIC guard column (Merck KGaA, Darmstadt, Germany). For the gradient elution, acetonitril (A) and a 10 mM ammonium sulfate solution adjusted to pH 9.2 with 25 % (v/v) ammonium hydroxide (B) were pumped at a flow rate of 0.2 mL min^{-1} and a temperature of 40 °C according to the following gradient: 10 % B at 0 min, 10 % B at 1 min, 70 % B at 31 min, 90 % B at 35 min, 90 % B at 45 min, 10 % B at 55 min, 10 % B at 80 min. This method has been described and validated previously [145]. Here, the isocratic hold at 90 % B and the final equilibration step at 10 % B have been elongated to ensure identical conditions for each sample. Newly optimized ion source parameters are listed in table 2.3 and the metabolite-specific parameters collision energy and declustering potential in table 2.2.

Table 2.3: Ion source parameters for the LC-MS/MS analysis of free amino acids.

parameter	value
gas source 1	50 psi
gas source 2	65 psi
curtain gas	25 psi
temperature	500 °C
IonSpray Float Voltage	- 3600 V

2.6.3 Chromatographic peak recognition and integration

Peak data generated with the TripleTOF6600 QqTOF device was loaded into the vendor software Sciex MultiQuant (version 3.0.3) [22] with a quantitation method specifying mass traces and their pertaining m/z ranges. Peak recognition and integration was performed with the MQ4 algorithm with default settings. Subsequently, results were reviewed visually and if necessary the baseline was manually adjusted, false negative or mislabeled peaks were corrected manually, and false positive were de-selected. Upon finishing these checks, the data was copied into Microsoft Excel [146] and saved as a comma-separated value (csv) file. The calculation of TMIDs was conducted with a self-authored Python script the structure and workings of which are more closely related in section 3.4.

2.6.4 Validation of PeakPerformance

Several stages of validation were employed to prove the suitability of `PeakPerformance` for chromatographic peak data analysis. The goals were to showcase the efficacy of `PeakPerformance` utilizing noisy synthetic data, investigate cases where a peak could reasonably be fit with either of the single peak models, and finally use experimental data to compare results obtained with `PeakPerformance` to those from the commercial vendor software Sciex MultiQuant [22].
For the first test, 500 random data sets were generated with the NumPy random module by drawing from the normal-shaped models detailed in Table 2.4 except for the mean parameter which was held constant at a value of 6. Subsequently, normally distributed random noise ($\mathcal{N}(0, 0.6)$ or $\mathcal{N}(0, 1.2)$ for data sets with the tag "higher noise") was added to each data point. The amount of data points per time was chosen based on an LC-MS/MS method routinely utilized by the authors and accordingly set to one data point per 1.8 s.

Table 2.4: Normal-shaped models from which parameters were drawn randomly to create synthetic data sets for the validation of `PeakPerformance`.

	model	
parameter	**1st test**	**2nd test**
area	$\mathcal{N}(8, 0.5)$	-
standard deviation	$\mathcal{N}(0.5, 0.1)$	$\mathcal{N}(0.5, 0.1)$
skewness	$\mathcal{N}(0, 2)$	-
baseline intercept	$\mathcal{N}(25, 1)$	$\mathcal{N}(25, 1)$
baseline slope	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$

In marginal cases when the shape of a single peak had a slight skew, the automated model selection would at times settle on a normal or a skew normal model. Therefore, it was relevant to investigate whether this choice would lead to a significant discrepancy in estimated peak parameters. Accordingly, for the second test synthetic data sets were generated with the NumPy random module according to Table 2.4 and noise was added as described before. The residual parameters were held constant, i.e. the mean was fixed to 6, the area to 8, and the skewness parameter α to 1.

For the third and final test, experimental peak data was analyzed with both `PeakPerformance` (version 0.7.0) and Sciex MultiQuant (version 3.0.3) [22] with human supervision, i.e. the results were visually inspected and corrected if necessary. The data set consisted of 192 signals comprised of 123 single peaks, 50 peaks as part of double peaks, and 19 noise signals.

2.7 Modelling

2.7.1 Apache Airflow computation cluster

The design and performance of remote workflows in this dissertation has been conducted with the open source platform Apache Airflow [147] which allows the implementation, scheduling, and monitoring of directed acyclical graphs (DAGs). A DAG is a batch run consisting of different parallel or sequential tasks which may exchange data between one another and the execution of which depends on the success of previous tasks. The DAG's code is written entirely in Python and is subject to version control via GitLab. An Airflow webserver is used to start batch runs manually or dependent on a schedule. Furthermore, it facilitates process monitoring and error analysis via a logging framework. In the present case, the celery executor [148] is used to scale the number of workers which are distributed on a computation cluster consisting of multiple PCs.

2.7.2 Parameter estimation with estim8

Parameter estimations were performed with the in-house developed Python tool estim8 [106]. Metabolic sub-network models v0 and v1 (3.5.1) were implemented with the modeling language Modelica in OpenModelica (v1.19.2) [104, 105]. Harnessing the advantages of the Functional Mockup Interface (FMI) standard, they were exported as Functional Mockup Units (FMUs) in `ModelExchange` mode. The parameter estimations in estim8 were performed with its `FmuModel` class and parallelized via the Python Parallel Global Multiobjective Optimizer package [149], particularly with its implemented optimizers self-adaptive differential evolutionary algorithm (DE1220), particle swarm optimization, and the sequential evolutionary algorithm. As a likelihood function, the negative log-likelihood (negLL) was utilized and to enable its calculation, absolute errors of 0.02, i.e. a relative abundance of 2 %, were assumed for the labeling fractions of all experimentally determined mass traces. Upon reaching stagnant parameter estimations with regards to the first decimal of the objective function for 100 steps, the process was stopped. The boundaries or inequality constraints for the parameter estimation are listed in table 2.5.

Table 2.5: Applied boundaries for parameter estimation with estim8.

parameter	lower boundary	upper boundary
c_X0_1st	0.0014	0.81
c_X0_2nd	0.0014	0.81
c_X0_3rd	0.0014	0.81
mu_max	0.16	0.74
v_upt_S_max	1500	2500
k_scale_Gly_exp	0.0001	0.01
c_EMPUP_1st	0.001	10
c_EMPUP_2nd	0.001	10
c_EMPUP_3rd	0.001	10
c_Ser_1st	0.1	9.7
c_Ser_2nd	0.1	9.7
c_Ser_3rd	0.1	9.7
c_Cys_1st	0.1	19.7
c_Cys_2nd	0.1	19.7
c_Cys_3rd	0.1	19.7
c_Gly_intra_1st	2.1	25.4
c_Gly_intra_2nd	2.1	25.4
c_Gly_intra_3rd	2.1	25.4

2.7.3 Bioprocess modelling: Sampling with the MCMC pipeline

The statistical inference with the Python package hopsy [129] was performed with a Gaussian hit and run MCMC sampling technique preceded by step size tuning and using polytope rounding. The models and settings of the different approaches are explained in table 3.5. Approaches 2 and 3 utilized the same parameter boundaries as the estim8 parameter estimation (table 2.5). Approach 4 expanded these boundaries with additional ones for the parameters exclusive to model v1 (table 2.6).

Table 2.6: Additional boundaries for inference of approach 4.

parameter	lower boundary	upper boundary
c_PEP_1st	0.1	50
c_PEP_2nd	0.1	50
c_PEP_3rd	0.1	50
c_Pyr_1st	0.1	50
c_Pyr_2nd	0.1	50
c_Pyr_3rd	0.1	50
c_KIV_1st	0.01	20
c_KIV_2nd	0.01	20
c_KIV_3rd	0.01	20
c_Ala_1st	0.1	30
c_Ala_2nd	0.1	30
c_Ala_3rd	0.1	30
c_Val_1st	0.01	20
c_Val_2nd	0.01	20
c_Val_3rd	0.01	20
c_Leu_1st	0.01	11.5
c_Leu_2nd	0.01	11.5
c_Leu_3rd	0.01	11.5

Approach 1 which omitted the batch phase of the bioprocess and started instead at the time of the pulse required some additional or altered parameters replacing their previous definitions, as well (table 2.7).

Table 2.7: Additional or replaced boundaries for inference of approach 1.

parameter	lower boundary	upper boundary
c_X0_1st	7.549	8.493
c_X0_2nd	7.597	8.543
c_X0_3rd	7.415	8.351
c_Ser_1st	0.1	9.6
c_Ser_2nd	0.1	9.6
c_Ser_3rd	0.1	9.6
c_Gly0_extra_0_1st	0.0001	1
c_Gly0_extra_0_2nd	0.0001	1
c_Gly0_extra_0_3rd	0.0001	1
c_S0_0_1st	20	60
c_S0_0_2nd	20	60
c_S0_0_3rd	20	60

2.7.4 INST ^{13}C-MFA

For the first INST ^{13}C-MFA, the ethanol uptake rate and maximum growth rates were estimated with a black box model based on cell dry weight and ethanol concentration data from a published bioreactor experiment [140]. Regarding the model-based determination of extracellular rates with backscatter data from the ILE with 1-^{13}C ethanol (2.5.4), the estim8 tool [106] was selected. Here,

online backscatter data from the BioLector I system from 21 parallel batch cultivations until the time of the pulse with labeled substrate and 3 parallel batch cultivations grown into stationarity were utilized. A simple bioprocess model (equations 2.2a and 2.2b) was implemented in OpenModelica [104, 105] following Monod kinetics to estimate growth rate and ethanol uptake rate.

$$\frac{dX}{dt} = \mu\, X \quad \text{with} \quad \mu = \mu_{max}\, \frac{c_{\mathrm{EtOH}}}{K_{\mathrm{EtOH}} + c_{\mathrm{EtOH}}} \quad \text{and} \quad X(t_0) = X_0 \tag{2.2a}$$

$$\frac{dc_{\mathrm{EtOH}}}{dt} = q_{\mathrm{EtOH}}\, X \quad \text{with} \quad q_{\mathrm{EtOH}} = -\frac{\mu}{Y_{X/\mathrm{EtOH}}} \quad \text{and} \quad c_{\mathrm{EtOH}}(t_0) = c_{\mathrm{EtOH},0} \tag{2.2b}$$

A pooled approach for parameter fitting was chosen where parameters were divided into global or strain-specific and local or replicate-specific parameters. The former comprised the affinity constant K_{EtOH} and the yield coefficient $Y_{X/\mathrm{EtOH}}$ while the latter included the maximum specific growth rate $\mu_{\max}$ and the initial biomass concentration X_0. The backscatter raw data was mapped to cell dry weight via a linear model originating from an earlier calibration experiment performed by Lars Halle where the following linear dynamic range

$$\mathrm{BS} = 13.4254\ \mathrm{CDW} + 15.3524 \tag{2.3}$$

had been observed.

The metabolic network model for the INST ^{13}C-MFA was adapted from [57] and contains the CCM of *C. glutamicum*. Supplement 1 of this publication contains a detailed description of all model reactions and carbon transitions. Since the model was originally intended for growth on Glc, necessary changes included the omission of Glc uptake and the addition of EtOH uptake and the glyoxylate shunt. Furthermore, gluconeogenesis was enabled by granting bidirectionality to the pertaining reactions in the EMP pathway. The amino acid metabolism and the ethanol uptake pathway are simplified in the model by lumping some linear reaction pathways and the biomass equation is based on [150]. In total, the metabolic model comprises 70 balanced intracellular metabolites and 78 intracellular reactions 76 of which are bidirectional and 2 unidirectional. Accordingly, the model has 29 free flux parameters made up of 7 net and 22 exchange fluxes and 69 metabolite pool size parameters, implying that in total 98 parameters are to be estimated from 1078 TMIDs plus the growth and ethanol uptake rates. Model expansion, validation, and visualization were conducted in Omix (ver. 2.1.2) [151]. The model was formulated in FluxML [136].

Estimation of the model parameters was performed using the 13CFLUX2 software [152]. A given set of measurements was incorporated in the ^{13}C-model and TMIDs $y(\boldsymbol{\theta})$ were predicted based on the parameters $\boldsymbol{\theta}$, consisting of fluxes ($\mathbf{v}$) and pool sizes ($\mathbf{X}$). Parameter estimates were computed by minimizing the weighted sum of squared residuals via [153]

$$\mathrm{SSR}(\boldsymbol{\theta}) = \sum_i \left(\frac{y_i(\boldsymbol{\theta}) - y_i^{\mathrm{meas}}}{\sigma_{y_i}} \right)^2 + \sum_j \left(\frac{v_j - v_j^{\mathrm{meas}}}{\sigma_{v_j}} \right)^2 \tag{2.4}$$

Local minima were computed by the interior-point optimizer IPOPT (v3.14.14) [154]. A multi-start with 1,000 random starting points determined by the sampling library hopsy [129] was applied to maximize the chance of finding the global optimum. Frequentist statistical analysis to determine

CoIs was performed using the profile likelihood approach [155] where the CoI for the i-th parameter is defined as

$$\mathrm{CoI}_i(\alpha) := \left\{ \bar{\theta}_i \;\middle|\; 2\left(\max_{\boldsymbol{\theta}} \mathrm{LL}(\boldsymbol{\theta}) - \max_{\boldsymbol{\theta}, \theta_i = \bar{\theta}_i} \mathrm{LL}(\boldsymbol{\theta}) \right) \leq q_{\chi^2_1,\, 1-\alpha} \right\} \tag{2.5}$$

Here, $\mathrm{LL}(\boldsymbol{\theta}) = -\frac{1}{2}\mathrm{SSR}(\boldsymbol{\theta})$ corresponds to the log-likelihood function and $q_{\chi^2_1,\, 1-\alpha}$ to the $(1-\alpha)$-quantile of the chi-squared distribution with one degree of freedom. To estimate 95 % CoI bounds, a binary search was employed between the best fit parameter value and its absolute lower and upper bound, respectively, with $\alpha = 0.05$.

3 Results and Discussion

3.1 Developing an automated quenching method

This chapter is based partially on the publication "Hot isopropanol quenching procedure for automated microtiter plate scale ^{13}C-labeling experiments" [142] first-authored by JN. The figures 3.3 and 3.6 were originally created by JN and have been adapted from said publication.

As detailed in the introduction (1.1), many orthogonal quenching methods to choose from have been published, yet none of them had been transferred to an automated liquid handling system. Accordingly, the first step on the way to an automated ILE workflow would have to be the adaptation of a quenching method onto the automated platform. Generally and above all, a quenching method has to stop enzymatic activity sufficiently fast, i.e. on a sub-second scale, but beyond that most basic of requirements it needs to connect to the subsequent analytical steps as seamlessly as possible when targeting a higher throughput of ILEs. This means it should not necessitate complex processing steps before the LC-MS/MS analysis.
The latter criteria already ruled out methods relying on a pH shift due to the need to remove excess salts in order to avoid ion suppression effects during ESI. For a manual process, the decision would have been an easy one in favor of the most commonly applied method of cold methanol quenching. However, upon creating an automated ILE workflow, the choice is complicated by the fact that the infrastructure in place dictates what method can sensibly and successfully be adapted and thus adds a criteria of device availability to the decision making process.
The Mini Pilot Plant on which the ILE workflow was to be established – for a more thorough description see section 2.4 – featured several devices with cooling capabilities but the lowest attainable temperature amounted to - 20 °C with the Hettich centrifuge. This theoretically lowest point, however, would a) not be reached in practice and b) transform the centrifuge into a glorified freezer since at temperatures approaching 0 or below, it cannot be operated regularly within specifications any more without risking damage to the device. As the sample processing would involve removing cells and cell debris via centrifugation to obtain only the supernatant for LC-MS/MS analysis, this was not a tenable design choice.
The next lowest temperature could be achieved by the cryostate which is able to cool up to a temperature of - 10 °C. Consequently, this precluded cold methanol quenching as all considered protocols required lower temperatures, for example the - 50 °C from the protocol referenced in the state of the art workflow [16].
Even disregarding this practical limitation and assuming unlimited investment regarding devices, cold methanol quenching presents several challenges in an automated setting. Firstly, methanol as a toxic solvent exacts abiding by safety regularities and having a working environment with sufficient ventilation provided by a fume hood which may pose a sterical challenge if not a financial one. The same holds true for chloroform which is commonly used in the later extraction stage of the cold methanol quenching procedure. Secondly, cold methanol quenching involves several

pipetting steps with a solvent-water mixture. While there are specified liquid classes for such a purpose with optimized parameters like aspiration and dispense speed as well as the volumes of the liquid handler's air gaps, it can be difficult to avoid dripping tips when dealing with variously concentrated solvents. Thirdly, a device placed on or in the vicinity of a robotic deck cooling down to - 40 °C would cause a difficult to contain volume of condensed water. This has already emerged as a hassle with a cooling rack at a temperature of 4 °C and would surely evolve into a serious problem at increasingly low temperatures. Fourthly and finally, cold methanol quenching protocols have always suffered from metabolite leakage during quenching particularly influencing the results of quantitative metabolomics experiments.

When pH shifts and the temperature shift to the low extreme are conceptually not feasible, the most obvious remaining option is constituted by a temperature shift to the high extreme. While cold methanol quenching aims to stop metabolic activity by lowering the temperature so far that the reaction speed decreases to virtually zero, the working principle of hot quenching methods is the denaturation of catalysts. The latter has the advantage that extreme temperature conditions need not be sustained once quenching has occurred whereas cold quenching methods demand a prolonging of low temperature conditions throughout the whole residual sample processing workflow. There is, however, yet another fundamental difference with respect to the extraction of target molecules whose labeling states are to be determined. As portrayed in figure 1.2, cold methanol quenching features separate stages for quenching and extraction which enables the quantification of intracellular metabolite pool sizes. These constitute valuable data points in isolation as well as being traditionally required for ^{13}C-INST-MFA. Contradistinctly, hot quenching utilizes agents "that would destroy permeability barriers of the cell" [156] like ethanol. Accordingly, this results in a single-step process encompassing quenching and extraction in a concomitant manner denominated as whole broth sampling. Combined with the irreproducibility of solvent evaporation at high temperatures causing an unknown reference volume, this entails that a) endo- and exometabolome cannot be differentiated but are instead observed in unity and b) a quantification of metabolite pool sizes with an acceptable measure of certainty is not contingent. It does, however, alleviate the aforementioned problems of numerous pipetting steps with different solvents and the toxicity drawback of methanol and chloroform predestining it for an automated process.

A further upside is that the Mini Pilot Plant houses a BioShake device able to heat its surface up to 99 °C which would be more than sufficiently high for hot quenching. Since miniaturized, automated bioprocesses rely on single-use plastic labware like MTPs and DWPs featuring notoriously low heat transmission, the resulting temperatures inside the liquid phase of a given well amounted to roughly 60 °C - 65 °C. This was deemed a lower temperature than likely necessary to reliably perform hot quenching wherefore a custom aluminum labware with a better heat transfer was designed and built by Daniel Klein and co-designed by Moritz-Fabian Müller.

This so-called "6x8 septum vial holder" labware is shown in figure 3.1 and is intended to hold 1.5 mL vitreous vials or HPLC vials as they are often referred to due to their ubiquitous use in that field. Overall, utilizing these vials improves both heat transfer and retention and is enabled by the ability of the liquid handler's fixed steel tips which can pierce their septa. The lid was added to the design after it was found that the vials would stick to and be hoisted along with the fixed tips and therefore needed to be retained by a barrier. In fact, the association between tips and septa

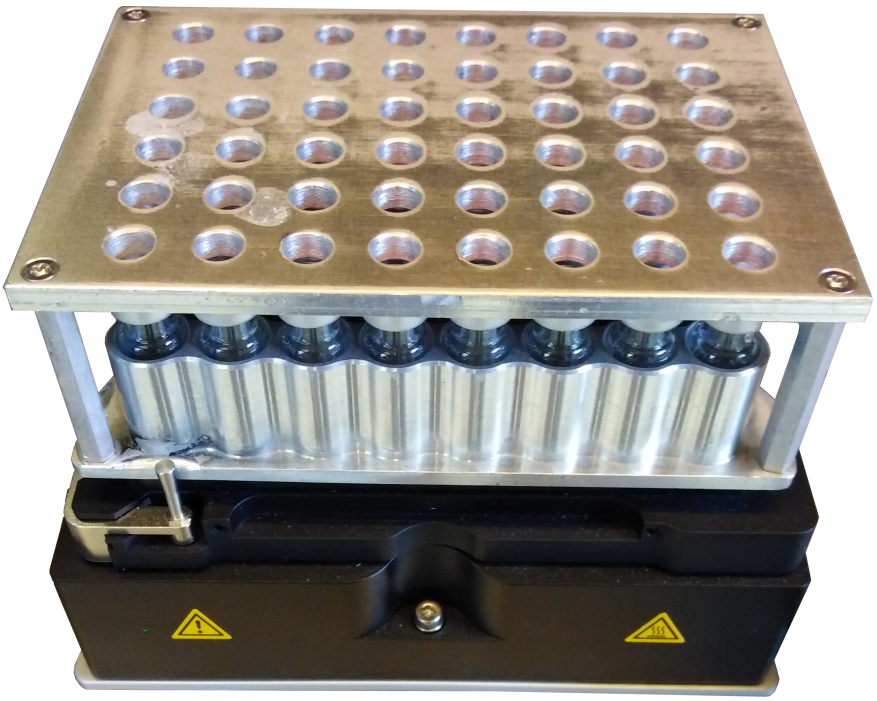


Figure 3.1: The aluminum plate equipped with closed 1.5 mL vitreous vials on top of a BioShake 3000-T elm with the capacity to heat up to 99 °C.

was strong enough to lift the whole aluminum plate when it was not securely tied down which was addressed on the software side by allowing the digital placement of the aluminum plate labware only on the deck position of the BioShake which has a lock functionality. On any other site, the software would accordingly produce an error message and would not be able to start the script to avoid damages to the system.

After these practical issues had been addressed, an open question remained regarding the precise choice of the liquid for the combined quenching and extraction procedure. Hence, the technique was hitherto referred to in a general manner as "hot solvent quenching" or simply "hot quenching" since up to this point in development, the final solvent had not been settled on and there was certainly no lack of candidates. While substances like trichloroacetic acid [156] had previously been precluded for similar reasons as the methodology of quenching via pH shift, many solvents would theoretically suit the task. However, it was suspected that especially for Gram-positive organisms with durable cell walls, the boiling of the solvent may enhance the permeabilization of the cell.

Based on this assumption, the prime candidates in consideration were ethanol and isopropanol. Since isopropanol had the benefit of non-interference with the substrate metabolic pool and labeling state in cases when ethanol served as a carbon source, it was the preferred solvent although ultimately both options would have worked.

With the solvent issue attended to, the design of an automated ILE workflow featuring hot isopropanol quenching as well as its validation and proof of concept were performed as described in the next sections.

3.1.1 Designing a workflow for automated ILEs with hot isopropanol quenching

Automated ILEs were realized as a continuous workflow featuring a main loop tied to the cultivation via online measurements, thus accounting for biological variability. On a basic level, the automated workflow (figure 3.2) could be divided into two distinct sections: cultivation and sample processing including hot isopropanol quenching.
Upon manually placing a FlowerPlate with a main culture in the BioLector microbioreactor system and having supplied the robotic deck with the necessary plates for a given experiment, the workflow is started by the user via a web server. This web server is part of the in-house developed DigInBio process control system (DCS) which also enables cross-communication between different third-party devices assembled within the robotic platform and features an extensive logging framework [157].
The first steps include the execution of possible pre-loop methods with preparatory steps for the experiment, i.e. washing the system, preparing a balance plate for subsequent centrifugation steps and so forth. Thereafter, the aforementioned main loop is entered and synchronized with the measurement cycles of the BioLector. The duration of these cycles depends on the number of filters selected in a BioLector protocol where one filter pertains to a specific type of online measurement, e.g. measurement of backscatter with a certain gain, pH within a certain range, DO etc. Each measurement is estimated to take 3 min with an additional 1 min allotted (in total) for data transfer and downtime, hence the usual minimum cycle time for such an experiment would amount to 10 min since backscatter, pH and DO data are of interest. Accordingly, every 10 min the current online data is downloaded from the BioLector, plotted, and send to the experimenter via Slack in the form of a line diagram and a heat map portraying the backscatter data across the FlowerPlate. Based on this online data and threshold values defined in separate configuration files, several decisions are evaluated before waiting for the completion of the next BioLector cycle.
The first decision is whether to prepare the quenching by heating up the BioShake. As the main cultivation for an ILE has to last until at least the mid-exponential growth phase, it did not seem sensible to include this step in the pre-loop methods. Instead, it is coupled to a time threshold which should ideally be triggered at least 30 min before hot isopropanol quenching, thus requiring previous knowledge about the cultivation. Alternatively, it can be tied to a backscatter threshold which is reached at least an hour before any quenching events are expected to be scheduled.
The second decision concerns which if any of the main sampling operations should be undertaken. First and foremost, this refers to the quenching procedure or rather its two variations for isotopically stationary and instationary labeling experiments but after the establishment of the quenching workflow, an option for supernatant sampling was added to enable measurements allowing the calculation of extracellular rates for ^{13}C-MFA. All of these operations are commonly tied to backscatter thresholds but naturally the choice of threshold is up to the user and could just as well fall on pH, DO or time. At least for the ILEs, though, a backscatter threshold is sensible to ensure having reached the mid-exponential growth phase.
The final decision simply checks whether any active wells remain which have not been sampled yet and if this is not the case, the experiment is ended after device-specific shutdown methods have been executed.

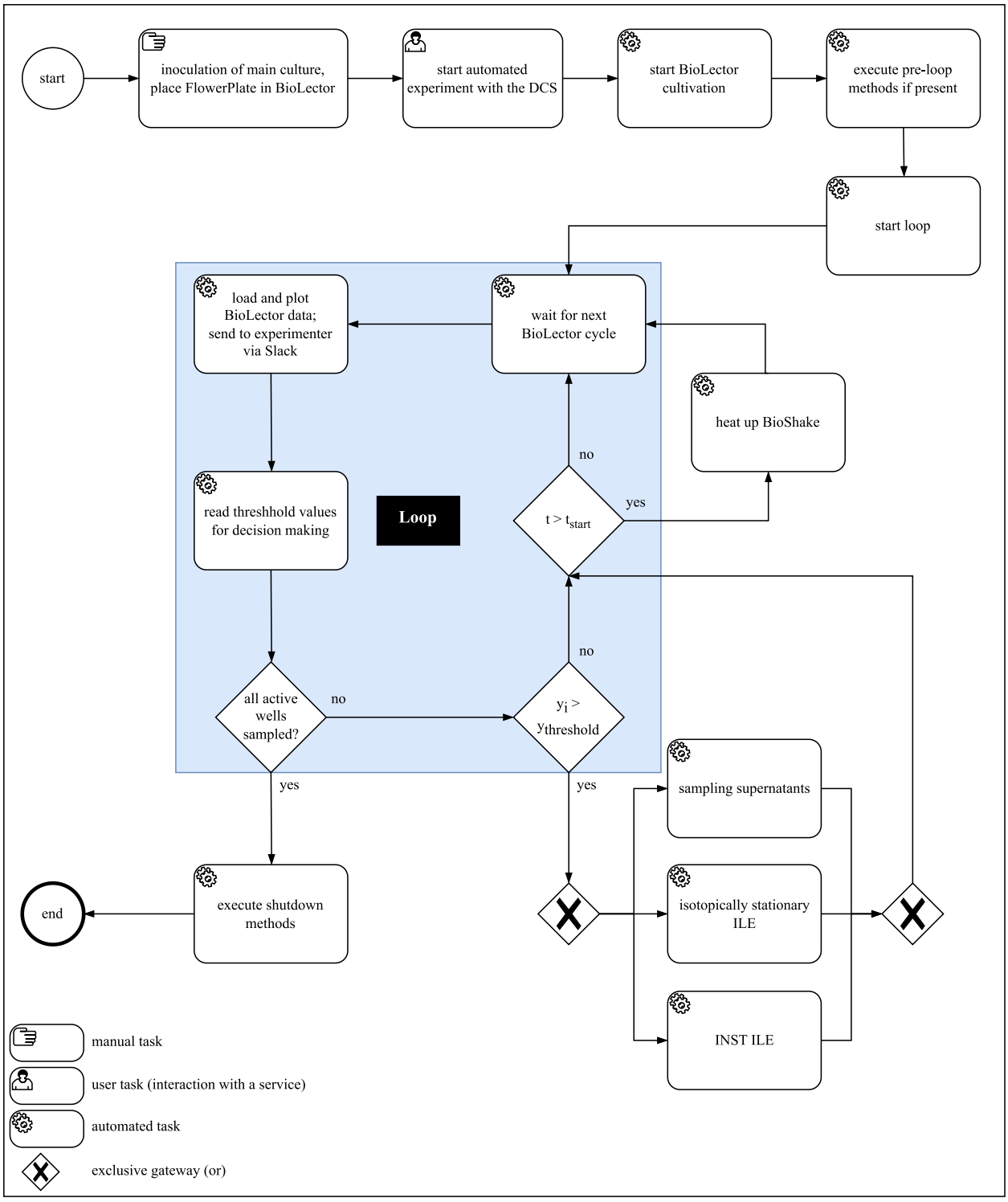


Figure 3.2: Business Process Model and Notation (BPMN) 2.0-inspired flow scheme portraying the automated ILE workflow.

Discussing the quenching workflow (figure 3.3) itself in more detail, it has to be emphasized that quenching is performed strictly column-wise - at least with the present robotic setup with a liquid handling arm with 8 pipettes. Accordingly, up to 6 wells, i.e. one full FlowerPlate column, can be harvested and quenched simultaneously. Since the aluminum plate's and a FlowerPlate's layout

are identical with 6 rows and 8 columns, no additional mapping of wells is required.
Going step by step through the process, two preparatory operations are constituted by piercing the vials' septa to decrease the built-up pressure from heating, thus avoiding a so-called "plunger overload" error of the robotic system, and by providing the 94 % (v/v) isopropanol solution for quenching. Next, hot isopropanol quenching is initiated by taking samples from the BioLector which are concomitantly quenched, lysed, and extracted upon injection into the vials containing hot isopropanol. After transferring the quenched samples to a DWP, the iterative quenching procedure re-starts with the next column until all wells relevant to the quenching operation have been attended to as described. In the subsequent sample processing stage, the DWP is centrifuged to separate cell debris from the supernatant, the latter of which is then transferred into a fresh DWP placed upon a cooling rack at 4 °C for storage for the remaining duration of the automated ILE workflow. At the very end of the ILE workflow, the samples are transferred into 1.5 mL Eppendorf tubes which have to be manually placed in a box inside a - 20 °C freezer.

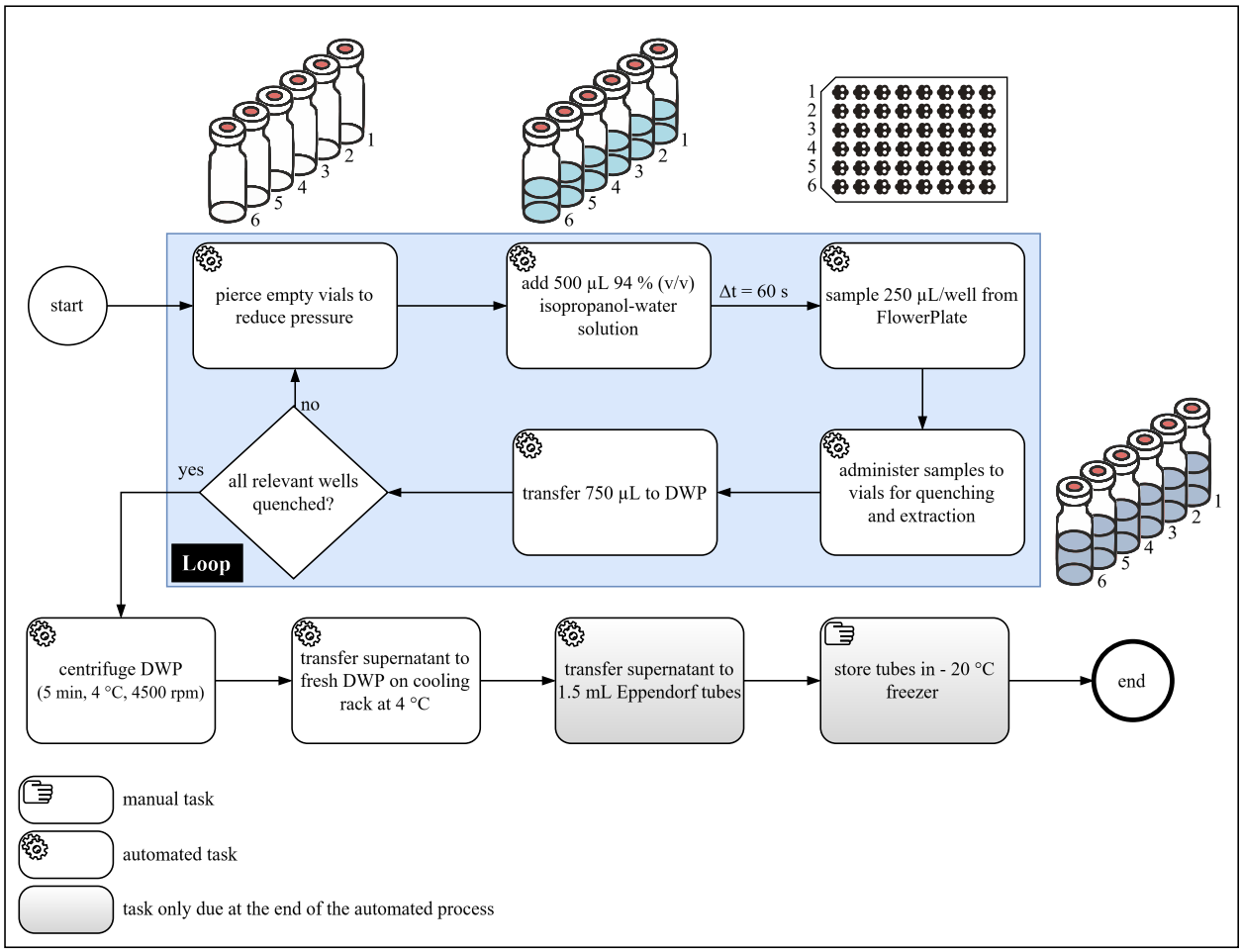


Figure 3.3: BPMN 2.0-inspired flow scheme portraying the automated hot isopropanol quenching workflow. The column-wise approach of the workflow is emphasized with the central loop. The single manual step is due only at the end of the overarching automated experiment run.

By comparison, the supernatant sampling unit operation is much simpler. After transferring 500 µL per sacrificed well from the FlowerPlate to a DWP and preparing an according balance plate with water, the plates are centrifuged for 5 min at at 4 °C and 4500 rpm. The process is concluded

by securing the supernatants in a fresh DWP and removing the water from the balance plate to provide it for the next execution of this operation.

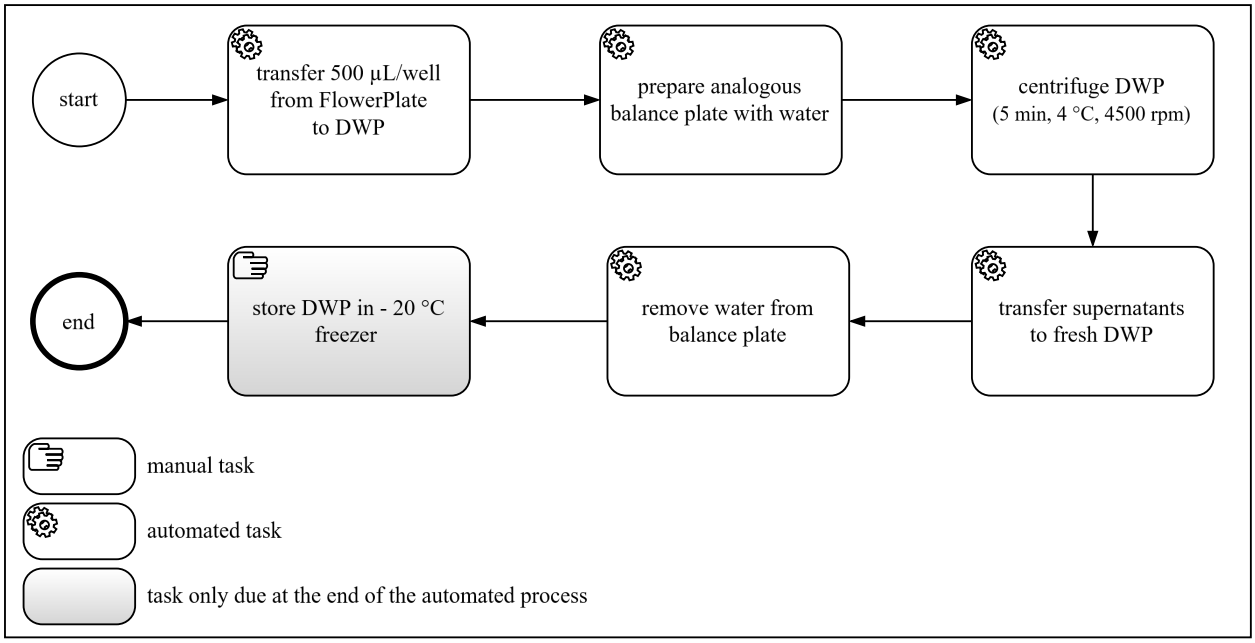


Figure 3.4: BPMN 2.0-inspired flow scheme portraying the optional supernatant sampling sub-procedure of the automated ILE workflow. The single manual step is due only at the end of the overarching automated experiment run.

It would be amiss not to address the remaining manual steps despite their positions at the very start and end of the otherwise automated process. The growth of the pre-culture and the inoculation of the main culture could definitely be automated, e.g. by using a few wells of the FlowerPlate strictly as pre-culture wells and placing media components on the robotic deck, in as sterile a fashion as possible given the circumstances. The labware containing said media components would be sterilized with UV light before being loaded under a clean bench and would be sealed before transfer to the robotic deck. Also, the available robotic platforms are covered with high efficient particulate air filters in order to approach sterility with a laminar flow, yet due to the open sides of the systems this cannot be guaranteed. The lack of automating the pre-culture was therefore a conscious decision specifically for the ILE workflow based on a risk-reward assessment. If an experiment fails due to contamination, it would lead in the best case to a loss of time and some material, most notably the FlowerPlate which is the most expensive piece of single-use equipment involved. In the worst case, however, when a contamination is only detected at the end of the experiment, the waste of labeled material and additional time would increase the overall loss significantly. As for the upside, the gain is only the avoidance of a fairly minor amount of manual work and time for an experimenter.
Regarding the storage of quenched samples at the end of the automated ILE workflow, there is the possibility to e.g. immediately distribute the samples across several MTPs and use an automated freezer as intermediary storage. The MTP format has the added benefit of compatibility with HPLC autosamplers allowing for a rather seamless transition to the bioanalytical stage. The workflow can be easily changed to accommodate this and the decision is left up to the user. However, since the

samples are diluted in water or solvents before LC-MS/MS analysis and the dilution factor is not always known at the time of conducting an ILE, it can be beneficial to store samples in an undiluted and on top of that centralized manner. Furthermore, not every robotic system was equipped with an automated freezer so access to one was not guaranteed and would therefore pose an unwelcome limitation. This was identified as an area of improvement for future developments and alterations to the workflow.
Software-wise, each run of the automated ILE workflow is governed by the DCS which executes a Python-based experimental control script (ECS). The download and parsing of online data from the BioLector is performed using the Python package bletl [157–159] and liquid handling is programmed either in the vendor software EVOware [160] or in worklists created with the Python package robotools [157, 161]. Generally, the EVOware is still utilized to execute robotic scripts triggered by the ECS but the actual content of those scripts and even their generation can now be relegated entirely to the existing Python framework (see section 3.2).

3.1.2 Validation of automated hot isopropanol quenching

In order to validate automated hot isopropanol quenching, a regular cultivation was performed on unlabeled glucose and the isopropanol solution for quenching was spiked with uniformly labeled D-glucose. If no incorporation of labeled carbon atoms beyond natural labeling could be detected by LC-MS/MS analysis, then it would be surmised that no residual enzyme activity occurred during quenching and thus that the quenching method is sufficiently fast for valid use in experiments. This validation technique has been published previously, albeit in an opposite fashion, i.e. with fully labeled cells and unlabeled substrate [162, 163]. As elaborated in the introduction, the model organism used for this (and subsequent) experiments - *C. glutamicum* ATCC13032 - was chosen in part due to its relatively sturdy cell wall as a Gram-positive organism which should render the cell lysis more difficult than with Gram-negative organisms. If, then, the process works with this model organism, it should be transferable to other microorganisms.
Results-wise, neither the closest glycolytic intermediate glucose-6-phosphate (G6P) nor the closest amino acid L-serine (Ser) to the substrate glucose showed any fully labeled mass trace (figure 3.5). Since glucose is imported via a PTS and phosphorylated in the process, G6P constitutes literally the first metabolite originating from the substrate so that the observation of any other free intermediate or amino acid would be unnecessary if the label from glucose did not reach it. Nevertheless, the experiment was fully evaluated and no signs of label incorporation were found (see figures A1, A2, and A3). It was concluded that automated hot isopropanol quenching is indeed a valid quenching method fit for use in ILE. The fully labeled mass traces observed for some amino acids in the published variant of this experiment [142] were later found to have originated from the cell-free fully ^{13}C- and ^{15}N-labeled amino acid standard mixture used in the publication. Since the fully ^{13}C-labeled state and the fully ^{13}C- and ^{15}N-labeled state of amino acids are often merely separated by 1 Da, a bleed-over occurred due to the lacking purity of the standard mixture which was then misinterpreted as being caused by residual enzyme activity.

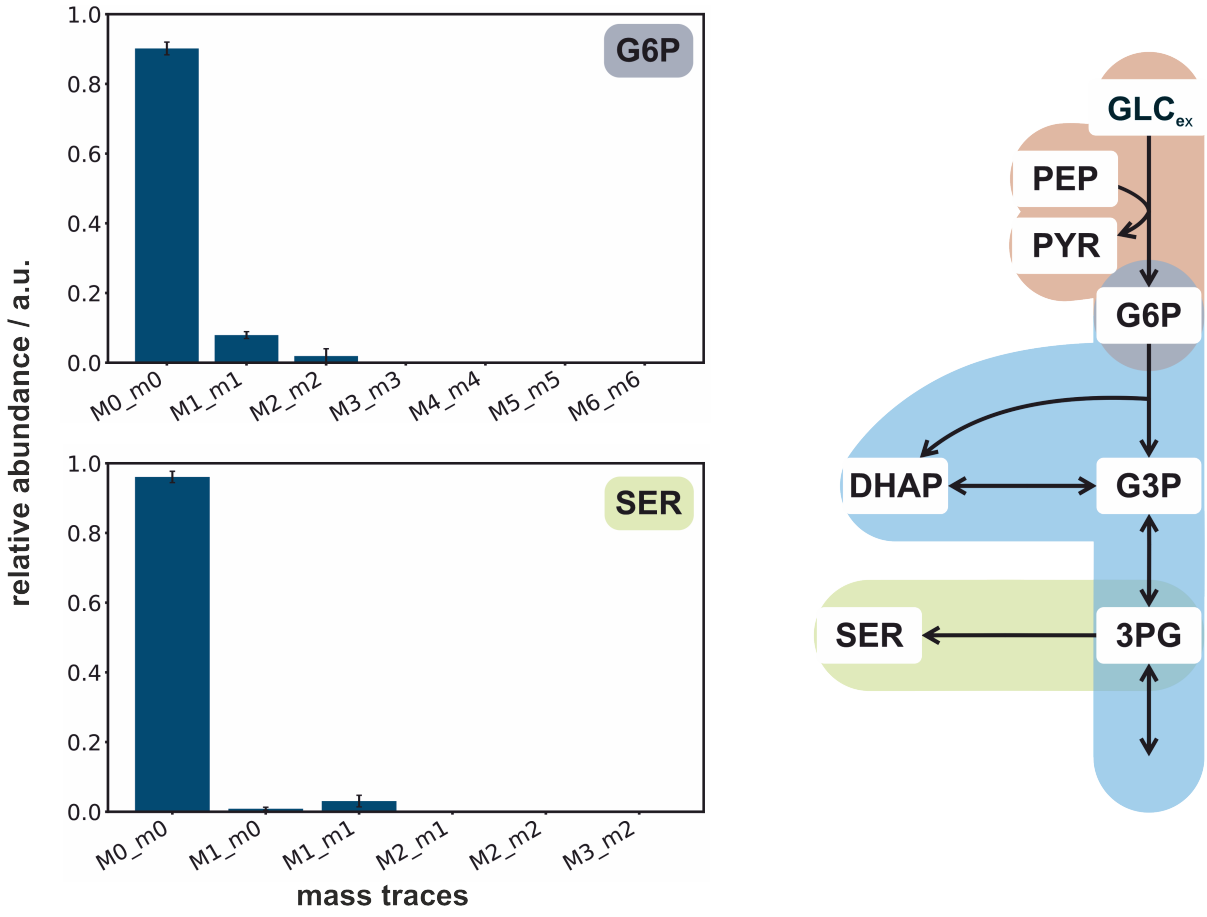


Figure 3.5: The most relevant results of the validation experiment for the automated hot isopropanol quenching method are portrayed as bar plots exhibiting the uncorrected TMIDs of G6P and Ser. The simplified depiction of the upper glycolysis and Ser synthesis of the model organism *C. glutamicum* WT serves to contextualize the metabolite's position in the metabolic network.

3.1.3 Proof of concept: Performing an automated INST labeling experiment

As a proof of concept showcasing the unique capabilities of the automated platform, an INST labeling experiment was conducted. For this purpose, the automated hot isopropanol quenching workflow was expanded with a pulsing step which would be executed column-wise right before sampling and quenching as portrayed in figure 3.3. Here, a labeling mixture was introduced to otherwise unlabeled cultures immediately prior to sampling and quenching. The time between pulsing and quenching is henceforth denominated as "delay" and corresponds to the time portrayed in INST plots in this work, meaning the time of the pulse acts as the zero point. In order to attain insightful kinetic labeling data, this delay had to be as short as possible for the first data points to gain access to the isotopically instationary period. Usually, after a liquid handling action like delivering the pulse, the tips of the liquid handling arm would be washed to regenerate system air gaps and clean the tips since residuals may adhere to the tips' inner walls. Depending on the wash parameters, this might take around 20 s but can be sidestepped by delivering the pulse with labeled material with the first three tips and and immediately sampling with the next three tips. Even in this scenario, though, the time to perforate the FlowerPlate's foil and the vials' septa in addition to the travel time, amounts to roughly 20 s which is hence the first time point accessible

for the automated INST. By sampling biological triplicates successively with increasing delays, up to 16 different time points can be observed on a single 6x8 FlowerPlate. The proof of concept experiment was designed with 10 time points in mind ranging from 24.5 s to 600 s (figure 3.6). The portrayed time courses of mass traces are the result of the metabolic network structure, the input labeling mixture and crucially both flux rates and metabolic pool sizes. Since pool sizes could not be quantified with automated hot isopropanol quenching, the latter parameters cannot be discerned limiting the use of a straightforward interpretation of the data but some useful conclusions regarding the cellular phenotype can be drawn, nonetheless.

One might be tempted to hypothesize a connection between the stoichiometric biomass requirement of an amino acid and the observed labeling time course but this assumption does not hold. Exemplary for this is the discrepancy between the faster time course of label incorporation into L-serine compared to L-alanine (Ala) despite Ala's almost fivefold higher biomass requirement of 1.15 mmol g^{-1}. An interesting observation with regard to Ser is the observed lack of transfer of labeled material to its direct neighbor L-glycine (Gly) within the first minute after the pulse. This particular finding has been replicated across multiple experiments with several biological replicates each so a faulty measurement can be excluded. An obvious cause would be constituted by a large pool size of Gly but even then small fractions on the labeled mass traces should have been detected. Due to the blending of endo- and exometabolome, a significant extracellular Gly pool as described previously [19] could explain this phenomenon, as well. Another hypothesis is that the reaction has a low exchange flux and the net reaction is oriented towards Ser, possibly due to its connection to the one-carbon metabolism through its cofactor tetrahydrofolate (THF).

In a previous publication it was hypothesized that labeling of TCA intermediates was delayed on account of large buffer pools of L-glutamate and L-glutamine (Gln) but unfortunately the data for Glu was not included due to excessive measurement noise [164]. The present data set supports this notion as the label incorporation into Glu commenced only slowly after a late onset at 24.5 s, implying a large pool size. Crucially, a much larger delay of more than 2 min was observed for Gln causing the amino acids like L-arginine and L-ornithine further downstream in the metabolic network to remain unlabeled during the observed time frame.

Aside from these biological findings, it could first and foremost be demonstrated that informative data from the isotopically instationary phase can be observed with the ILE workflow, even in a fast growing organism such as *C. glutamicum*. Since this experiment was intended as a proof of concept, this most important result was clearly achieved.

After the initial success when focusing on amino acids, the experiment was repeated to investigate whether the isotopically instationary phases of some intermediates of glycolysis, PPP, and TCA cycle could be observed, after all, despite the expected higher turnover of these molecules [165] compared to free amino acids. As can be seen in figure 3.7, the glycolytic intermediates G6P, fructose-1,6-bisphosphate (FBP), and glyceraldehyde 3-phosphate (GAP) were already in the stationary phase when the first data point was recorded which is in agreement with data from literature [164]. For the collective pentose phosphate pool denominated as "XR5P" and sedoheptulose 7-phosphate (S7P), sections of the isotopically instationary phase could be observed demonstrating that for such molecules which are more distant from the substrate's entry into the metabolic network, more informative data can be generated. As indicated by the scarcity of shown

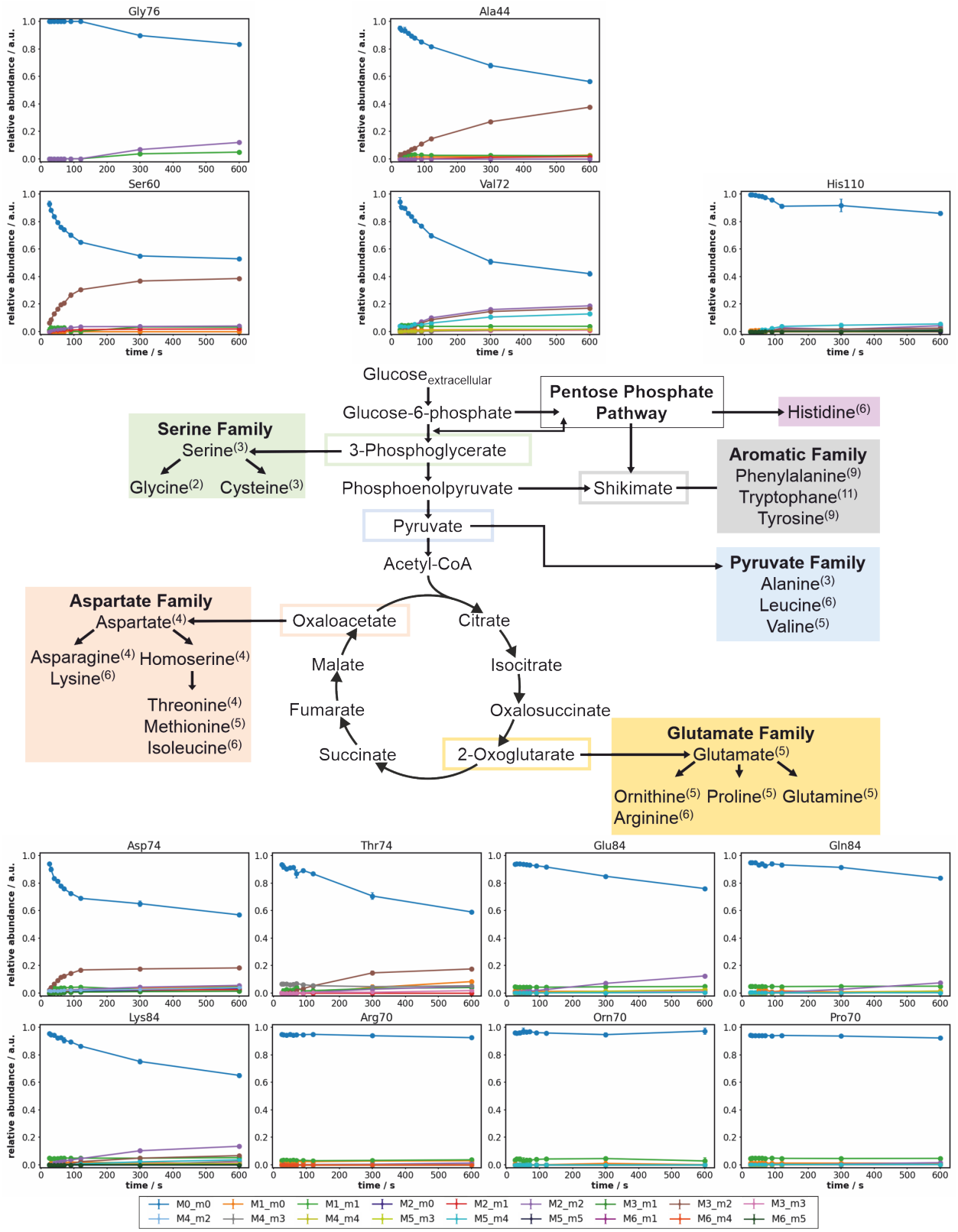


Figure 3.6: Results of the proof of concept experiment featuring an automated INST ILE focused on free amino acids alongside a simplified network of the model organism *C. glutamicum* WT. The line diagrams exhibit the TMIDs of the titled amino acid fragments denominated by the three letter code of the amino acid and the m/z ratio of the fragment.

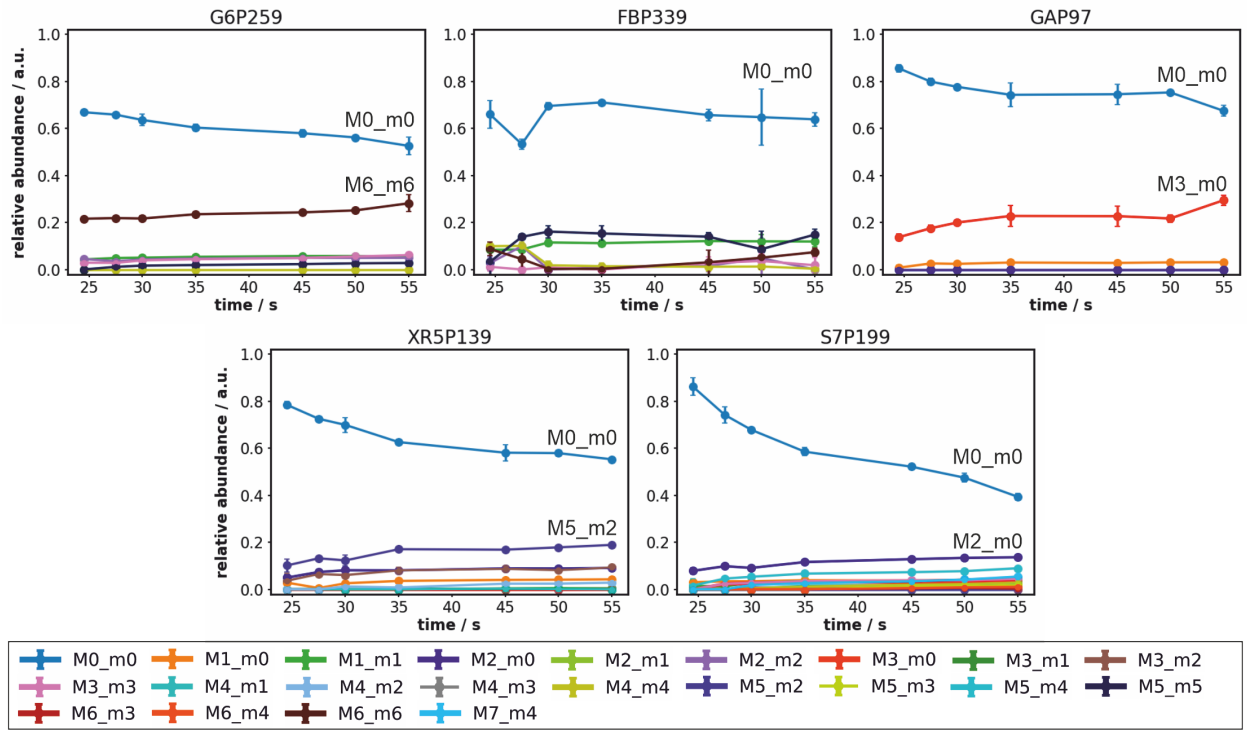


Figure 3.7: Results of the automated INST ILE focused on free intermediates. The line diagrams exhibit the TMIDs of the titled fragments denominated by the abbreviation of the intermediate and the m/z ratio of the fragment.

intermediates, their detection turned out rather challenging, even with a HILIC column and optimized MS parameters. This is most likely due to the dilution effect of hot isopropanol quenching where - upon cell lysis - the metabolites are extracted from the relatively low cellular volume into a solution of up to 750 µL inhibiting the detection of especially compounds with low intracellular concentrations. Thus, when using the described workflow INST labeling data from intermediates of glycolysis, PPP, and TCA cycle cannot be generated in a comprehensive manner and serves more of an auxiliary function by using the stationary data points and thereby concomitantly introducing an upper limit for the onset of the ISS for a given compound. At least, this is the case for the closest intermediates to the substrate. The instationary phase of more distant intermediates from the carbon source may be observed, after all.

3.1.4 Comparative evaluation and limitations of the automated INST labeling experiment

Since the column-wise nature of the workflow and the sacrifice sampling approach stay in place, the fundamental difference to a lab-scale bioreactor experiment has to be emphasized. In a bioreactor cultivation in batch or continuous mode, an INST experiment would involve a similar spike with or switch to labeled material, respectively, but each reactor would be sampled multiple times thereafter to record the time course of label incorporation. When performing sacrifice sampling, this is obviously not possible so instead the time course is observed by choosing different delays for each biological triplicate. It is presumed, then, that while the cultures in different wells grow

comparably and remain in the exponential phase, the assumption of an average cell still holds true and that the derived time points can be assembled into one unified labeling time course. In the opposite extreme, one may view this setup as a massively parallel ILE where every single well is regarded as a unique ILE.

Yet another assumption is that of one input labeling mixture shared by all wells. When pulsing - like in the present experiment - with a constant volume of a solution with 100 % labeled substrate while cultures were grown on unlabeled substrate, the input labeling mixture depends ultimately on the residual concentration of unlabeled substrate c_{S0} in a given well. Clearly, if the growth between wells deviates significantly, c_{S0} is not constant and thus, the input labeling is not. Even when this requirement is fulfilled, however, since groups of wells with different delays and/or located in different columns are handled successively and all the while the cultivation continues in yet untreated wells, this, too, leads to a small, systematic offset in c_{S0} and the input labeling mixture. This could theoretically be addressed after the fact by a) recording the temporal offsets between the pulsing events and allowing for local variations in input labeling mixture or b) using hierarchical models to simultaneously treat all wells as separate ILEs while imposing a framework with global parameters. As the latter, however, is a computationally burdening approach, it is recommended to minimize this systematic effect by simply structuring the experiment more efficiently. Accordingly, when later time points are to be observed, those can be pulsed first and their long delays between pulsing and quenching can be used to handle earlier time points with shorter delays. Another time save is constituted by front-loading the preparatory steps like piercing the vials and administering isopropanol and completing them for all vials before the first pulse. This way, all pulses are in closer temporal proximity, thereby minimizing the described effect of a difference in input labeling mixture.

Another key difference pertains to the question of speed. In INST experiments it is paramount to observe the kinetic of label incorporation before isotopic stationarity. During growth on Glc, this is especially critical for intermediates from glycolysis and PPP which can already reach the ISS after a short time frame of around 10 s [164]. For bioreactor cultivations, a rapid sampling device [166, 167] may be used to sample and quench at a sub-second scale in order to observe even energy storage molecules such as GTP among others. As detailed above, the robotic system is comparatively limited in terms of speed allowing observations only after about 20 s because the LiHa is slowed down significantly by having to pierce the FlowerPlate's foil and the vitreous vials' septa.

A final important differentiation to make is with regards to the obtained experimental data. Online backscatter is used as a substitute to direct CDW measurements and hence requires conversion, either via a separately generated calibration model [157, 168, 169] or by subsequently estimating the parameters for the conversion during bioprocess modeling. In theory, a direct CDW measurement as is customary in bioreactor experiments can be conducted with BioLector samples, as well, but that would require sacrificing wells for this purpose which could then not be used for hot isopropanol quenching. One compromise would be to use the cell pellets from supernatant sampling to determine CDW at some points during the cultivation, at least. A straight-up downside of the BioLector cultivation lies in the absent capability for off-gas analysis. Therefore, no carbon dioxide evolution rate (CER) nor the respiratory quotient (RQ) can be determined leaving a gap in the

carbon balance around the reaction vessel. Since the secretion of carbon dioxide is traditionally an extracellular rate provided for ^{13}C-MFA, its omission presents an additional degree of freedom. With regards to the original intentions, the combination of miniaturization and the INST methodology led to a synergistic effect lowering experimental costs significantly as both reduce the amount of required labeled substrate. Moreover, the established automated ILE workflow's capacity to conduct both isotopically stationary and INST ILEs at a much increased throughput has been demonstrated and additional functionality such as supernatant sampling was introduced to enable at the very least the fully stationary ^{13}C-MFA variant. In contrast to prior publications, the development of the automated hot isopropanol quenching method enabled the generation of labeling data of free metabolites at a microliter scale for the first time. While the primary target of the method were free amino acids, it was shown that a limited investigation of free intermediates from glycolysis, PPP, and TCA cycle was possible, as well. Finally, to impress upon the reader the magnitude of time saved by the automated ILE workflow, the proof of concept experiment was conducted (excluding the pre-culture) within a single work day which was comprised mostly of walk-away time for the experimenter.

3.2 Automated generation of experimental scripts

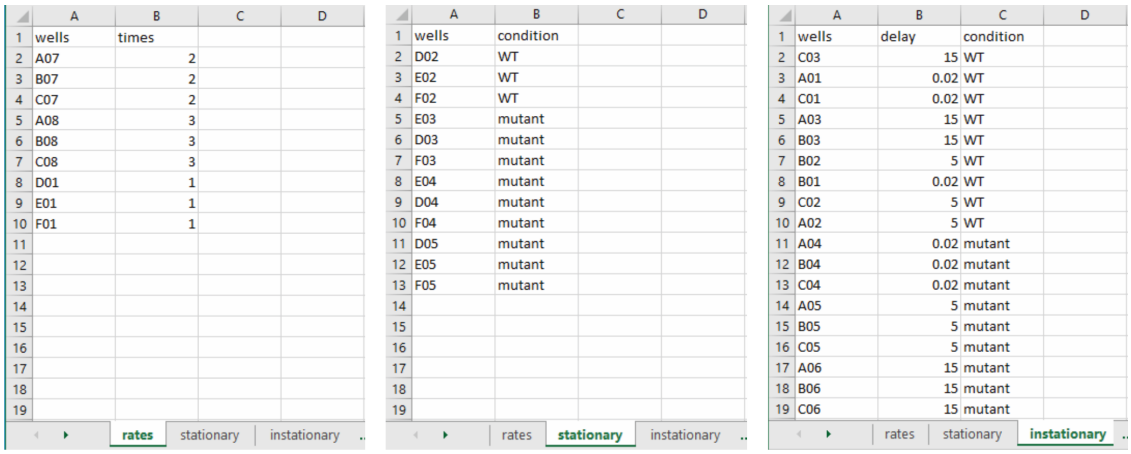
As much as the throughput of biological experiments may be enhanced by automation, additional tasks which do not apply to manual experiments to the same degree if at all do arise. As detailed in the previous section (see 3.1), an ECS meaning a Python script governing the automated workflow [157] has to be created, tested, and adapted when either the experimental setup or a utilized Python package is altered or updated. Aside from slowing down the potential gain in walk-away time and throughput, these scripts can be fairly expansive and complex, thus requiring knowledge of Python and the DCS [157] when alterations are in order. Since a stated goal of the overarching ILE workflow is to lower the barrier of entry in every facet, it was decided to also automate the generation of such a script. While this does not eliminate the need for testing and should serve rather as a basis which may be varied and expanded upon, it should yet grant less experienced users access to automated ILEs.

3.2.1 Implementation

To enable swift experimentation with as low a barrier of entry as possible, an ECS generating program should fulfill the following criteria:

1. User input should be sourced from as few files as possible and only rudimentary programming skills should be necessary to supply said input.
2. The generated ECS needs to have the option to sample supernatants and perform both isotopically stationary and instationary labeling experiments.
3. The generated ECS should be independent of any one robotic platform so that it can be readily executed on all Mini Pilot Plants.

The first two design principles could be upheld by writing a template containing the most general form of the ECS encompassing all variants of an ILE. Using the Jinja2 Python package [170], such a template can be parameterized and an experimental logic can be introduced via if-statements and for-loops. Upon rendering the template from a Jupyter notebook, these parameters must be delivered and the ECS is created in dependence of them. E.g. when a Boolean variable decides whether or not an INST experiment is to be included and it reads "False", then the generated ECS will not contain any section exclusively relevant to the INST workflow. The entry of this information is kept simple by only defining a few variables within a provided example notebook and outsourcing all other settings and parameters to an Excel file with three sheets and one `*.json` file. For example, if any of the three main sampling unit operations depicted in the ILE workflow (figure 3.2) are not part of the planned experiment, the user can just leave the respective Excel sheet empty and the pertaining Boolean will automatically be set to "False". Such simplifications streamline the automated ECS generation for the user.
The third design principle of creating an ECS independent of any single robotic system proved much more difficult. It should be noted that while the ECS governs an automated experiment, the control of the liquid handling system is still enacted via its vendor software EVOware [160].



	A	B	C	D
1	wells	times		
2	A07	2		
3	B07	2		
4	C07	2		
5	A08	3		
6	B08	3		
7	C08	3		
8	D01	1		
9	E01	1		
10	F01	1		
11				
12				
13				
14				
15				
16				
17				
18				
19				

rates | stationary | instationary

	A	B	C	D
1	wells	condition		
2	D02	WT		
3	E02	WT		
4	F02	WT		
5	E03	mutant		
6	D03	mutant		
7	F03	mutant		
8	E04	mutant		
9	D04	mutant		
10	F04	mutant		
11	D05	mutant		
12	E05	mutant		
13	F05	mutant		
14				
15				
16				
17				
18				
19				

rates | stationary | instationary

	A	B	C	D
1	wells	delay	condition	
2	C03	15	WT	
3	A01	0.02	WT	
4	C01	0.02	WT	
5	A03	15	WT	
6	B03	15	WT	
7	B02	5	WT	
8	B01	0.02	WT	
9	C02	5	WT	
10	A02	5	WT	
11	A04	0.02	mutant	
12	B04	0.02	mutant	
13	C04	0.02	mutant	
14	A05	5	mutant	
15	B05	5	mutant	
16	C05	5	mutant	
17	A06	15	mutant	
18	B06	15	mutant	
19	C06	15	mutant	

rates | stationary | instationary

Figure 3.8: Configuration Excel file "configuration.xlsx" with settings for the automated ECS generation. It features three sheets for the different sampling workflows: supernatant sampling as well as whole broth sampling for both isotopically stationary and INST labeling experiments.

Whereas pipetting schemes can be separately defined and read into the software as so-called worklists, an EVOware script or `*.esc` file can and unfortunately does in the present case differ between robotic platforms due to deviations in deck layouts etc. Hence, each `*.esc` file would need to be prepared in multiple variations to cover all Mini Pilot Plants. A second problem was constituted by the difference in liquid handling commands between EVOware scripts and worklists. As mentioned in 3.1.3, it is critical to access the earliest time point possible wherefore lengthy washing steps have to be avoided and different tips are used for pulsing and sampling, instead. However, the standard pipetting commands within worklists - depending on how they are phrased - either insert an unavoidable washing step or automatically group successive pipetting actions in a (for the present purpose) nonsensical manner. Within the EVOware, pipetting commands are stated more freely but this would require hard-coding them into these scripts and work against a unified approach of programming entire automated workflows in Python. Additionally, any changes would require blocking a robotic system in order to access a PC with an EVOware license and to individually update each relevant pipetting command in each `*.esc` file. Thus, the shift to Python has tangible advantages and its pursuit is worthwhile.
Upon testing it was noticed that the EVOware pipetting commands could be copied into a worklist preceded by a break command and would function as intended. For a more detailed explanation of their constituents, see figure 3.9. If one could reproduce these commands *in silico*, it would therefore provide users with a much higher degree of freedom and control when designing pipetting schemes via the robotools package in Python [157, 161].
By far the most complex part of their syntax was the well selection string which used a bitmap where 7 wells are codified by one byte or one character in the code string. Some examples of this are shown in table 3.1. The first 4 characters describe the labware dimensions in hexadecimal (HEX), e.g. for a plate with 12 columns and 8 rows it reads "0C08" or for a FlowerPlate "0806". Subsequent characters specify the selected bit-coded wells by summation of their decimal values which are finally represented in the code string using American National Standards Institute (ANSI) characters. To avoid unprintable characters, 48 is added on top of the sum of selected wells so

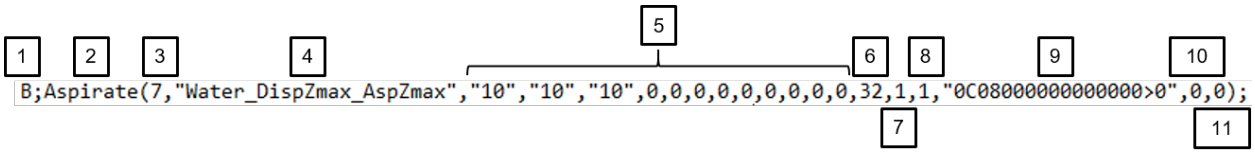


Figure 3.9: Exemplary EVOware pipetting command as appearing in worklists. The annotated positions represent a break command within worklists (**1**), the pipetting action (aspirate or dispense, **2**), bit-coded tip selection (**3**), liquid class (**4**), individual volumes for up to 12 tips (**5**), grid (**6**) and site (**7**) of the targeted labware object, tip spacing (**8**), bit-coded well selection (**9**), number of loops (**10**) and liquid handling arm selection (**11**). In case a loop is selected, additional loop parameters are included (not shown).

that the selection of only the first well in a group amounts to a "1" in the code string (ANSI 48 corresponds to 0 in decimal). As portrayed in the final line of the table, this sum may exceed the 127 ANSI characters thus requiring use of an extended character sheet based on the Windows-1252 (CP-1252) standard. Up to this point, the information was available in the EVOware extended help file. Furthermore, a code snippet to recreate the well selection string written in C was included which was then translated to Python by Martin Beyß. It required additional testing, though, to figure out the correct encoding when saving worklists. With the default `UTF-8` encoding, the extended character set would not be printed correctly so the `latin_1` encoding was selected, instead.

Table 3.1: Well selection of EVOware pipetting commands for a 8x12 plate.

well(s)	well number(s)	binary	decimal	decimal + 48	ANSI	code string
A01	1	0000001	1	49	1	0C08**1**0000000000000
B01	2	0000010	2	50	2	0C08**2**0000000000000
C01	3	0000100	4	52	4	0C08**4**0000000000000
D01	4	0001000	8	56	8	0C08**8**0000000000000
E01	5	0010000	16	64	@	0C08@0000000000000
F01	6	0100000	32	80	P	0C08**P**0000000000000
G01	7	1000000	64	112	p	0C08**p**0000000000000
H01	8	0000001	1	49	1	0C08**1**0000000000000
A02	9	0000010	2	50	2	0C08**2**0000000000000
A01 - G01	1 - 7	1111111	127	175	¯	0C08¯0000000000000

Once the code strings could successfully be reproduced, a number of functions generating aspirate, dispense, and wash commands in the EVOware style were added to robotools in a massive update with roughly 1300 added lines of code [171]. Aside from the core functions producing the code strings, additional ones for checking user input as well as extensive unit tests were supplied. The current implementation of these new commands in robotools is shown in figure A4 for aspirate and dispense commands and in figure A5 for wash commands. In each case, the user interacts with one outward-facing function returning the created EVOware pipetting command after passing through a number of internal functions which are not depicted in the aforementioned figures. As can be seen, there are many, partly optional, parameters for the user to specify enabling tight control of pipetting actions down to minuscule details such as the volume of the system trailing airgap. For a code example of how to apply these newly established EVOware commands, see listing 3.1. The example shows a pipetting scheme for an INST experiment with just three biological replicates sampled at the earliest time point after the pulse. Henceforth, it was possible to code the entirety

of all necessary liquid handling steps in worklists within the ECS, i.e. in Python, and have these worklists be created automatically during the execution of the automated experiment. In addition to the *in silico* unit tests written for robotools, the EVOware pipetting commands were naturally tested on the Freedom Evo robotic systems, as well, and have been utilized successfully in e.g. the ethanol labeling experiments presented in later sections of this thesis.

```
from robotools import liquidhandling
from robotools import evotools

# define labware objects
flowerplate = liquidhandling.Labware('flowerplate', 6, 8, min_volume=0, max_volume=2000, initial_volumes=800)
inst_dwp = liquidhandling.Labware('labeled substrate', 8, 12, min_volume=0, max_volume=2000, initial_volumes=1000)
aluplate = liquidhandling.Labware("aluplate", 6, 8, min_volume=0, max_volume=2000, initial_volumes=0)

# create worklist
with evotools.Worklist("test.gwl") as wl:
    wl.evo_aspirate(
        wells=["D01", "E01", "F01"],
        labware=inst_dwp,
        labware_position=(44,2),
        tips=[1, 2, 3],
        volumes=50,
        liquid_class="Water_DispZmax-1_AspZmax-1",
    )
    wl.evo_dispense(
        wells=["D01", "E01", "F01"],
        labware=flowerplate,
        labware_position=(62,1),
        tips=[1, 2, 3],
        volumes=50,
        liquid_class="Water_DispZmax-1_AspZmax-1",
    )
    wl.evo_aspirate(
        wells=["D01", "E01", "F01"],
        labware=flowerplate,
        labware_position=(62,1),
        tips=[4, 5, 6],
        volumes=250,
        liquid_class="Water_DispZmax-1_AspZmax-1",
    )
    wl.evo_dispense(
        wells=["D01", "E01", "F01"],
        labware=aluplate,
        labware_position=(26,3),
        tips=[4, 5, 6],
        volumes=250,
        liquid_class="Water_DispZmax-1_AspZmax-1",
    )
    wl.evo_wash(
        tips=[1,2,3,4,5,6,7,8],
        waste_location=(52,2),
        cleaner_location=(52,1)
    )
```

Listing 3.1: Code example for usage of EVOware pipetting commands in worklists to code INST pulsing, sampling, and quenching.

The remaining grievance of needing multiple versions of the EVOware scripts for different automation platforms was tackled by an effort synergizing with the investigation of EVOware pipetting commands. The next logical step was constituted by constructing a Python package featuring Pythonic reading and writing of EVOware scripts. This necessitated an understanding of the complete structure of EVOware scripts - a topic which was not covered in the documentation and required reverse-engineering to solve. The resulting Python package was a joint effort of several PhD students including the author with smaller contribution by other institute members and will not be covered in detail here. Suffice it to say, the combination of the sizeable robotools update and the establishment of the Python package for EVOware script writing enabled the realization of the

automated ECS generation in its presented form fulfilling the third requirement of the initially stated design principles. Additionally, these efforts furthered digitization of laboratory work since entire automated workflows could now be composed and partly tested remotely. Especially when time slots of robotic platforms are a limiting factor, enabling Pythonic `*.esc` file generation independent of limited EVOware licenses becomes particularly valuable.

3.2.2 Limitations of the automatically created ECS files

It is imperative to note the limitations of the generated ECS and thus when user action for customization is required.
The first such limitation pertains to how sampling actions are triggered. The function responsible for selecting wells for sampling upon exceeding a threshold cycles through the conditions stated in the input Excel file. As soon as the criteria is fulfilled for any one condition, all wells pertaining to it are flagged for sampling and this list of wells is returned immediately without checking further conditions. Accordingly, sampling and quenching events can only be triggered for one single condition per cycle so in case of a concomitant passing of thresholds by two groups of wells attached to different conditions they are dealt with successively. This was deemed acceptable since ILEs merely require sampling within the exponential growth phase to maintain the assumption of a metabolic steady-state but as long as this is guaranteed the precise timing of sampling is of lesser importance.
As stated before, crossing a threshold leads to sampling of all wells affiliated with a specific condition and not only to those above the threshold assuming a rather uniform or at the very least comparable growth behavior between wells. For isotopically stationary ILEs, this could be changed to a mode with individual wells but for INST ILEs, it is imperative that the pulsing is performed in a timely fashion across all wells and simultaneously for biological replicates (see subsection 3.1.4). In both cases, piercing the vials and preparing isopropanol is finished before the actual sampling starts. With regards to INST, this measure saves time between administration of the successive pulses, as discussed before.
Finally, to greatly simplify the ECS, the last liquid transfer of all samples from a DWP into 1.5 mL Eppendorf tubes is performed for all 48 wells on the DWP which could potentially be filled, regardless of whether they actually are. This is due to the difficulty in generalizing the mapping of wells from a 8x12 well plate such as a DWP to the 16x1 carrier holding the 1.5 mL Eppendorf tubes. Since this is only executed at the very end of an automated workflow, the delay caused by this simplified approach is inconsequential.
None of these limitations are particularly restrictive and all of them can be amended by the user after the ECS has been generated but they nonetheless have to be acknowledged. As stated at the onset of this chapter, the motivation was to create a generalized chassis to save time on the preparation of future experiments and this was fulfilled.

3.3 Realizing a novel approach for peak integration and uncertainty quantification of LC-MS/MS raw data

This chapter is based partially on the publication "PeakPerformance - A tool for Bayesian inference-based fitting of LC-MS/MS peaks" [172] first-authored by JN with text revision by Michael Osthege and Stephan Noack. All figures were originally created by JN and all but figure 3.16 have appeared in said publication. The pertaining section of Materials and Methods (2.6.4) was taken from said publication. The sections 3.3.2, 3.3.3, and 3.3.4 are heavily based on their counterparts in the publication. As stated in the publication, `PeakPerformance` *was conceptualized by JN and MO. Software implementation was conducted by JN with code review by MO.*

Since the previous chapter focused on all aspects of the automated experimental workflow, i.e. the data generation, this marks the beginning of the data evaluation section. Having conducted an automated ILE, the experimenter will have obtained cell-free extracts composed of free metabolites dissolved in an isopropanol-water solution at a volume of a few hundred microliters per well. Next, the isotope labeling patterns of target compounds have to be determined, in this case those of free amino acids, by way of LC-MS/MS. The bioanalytical techniques themselves are not the focus of the present thesis and accordingly are omitted here but the generated data which may be presented in the form of extracted ion chromatograms (EIC) is highly relevant.
Essentially, these EIC portray time series of the intensities of mass traces over the course of the entire LC-MS/MS run or measurement period, thereby combining the detected quantity of the MS with the time course of the HPLC separation. In order to calculate TMIDs, the peak areas of all mass traces corresponding to the mass isotopomers of one fragment need to be normalized. Therefore, one must first obtain these peak areas by integrating the measured LC-MS/MS peaks pertaining to said mass traces.
There are different approaches to peak recognition and integration, but the first workflow to be presented here is based on commercially available vendor software and was deemed representative for many other laboratories besides the ones at the IBG-1. Here, feature recognition and peak integration were performed using the Sciex MultiQuant software version 3.0.3 [22] specifically with its MQ4 algorithm and default integration settings. While this process is – in theory – automated, the mandatory visual inspection and occasional manual re-integration by the user due to the high frequency of false positives, false negatives or incorrectly determined baselines, ultimately downgrades it to a semi-automated process.
Beyond the requirement of human labor, its associated costs, and the tediousness of the task when faced with a large number of peaks, it not only constitutes a bottleneck in a high-throughput pipeline but necessarily introduces additional sources of errors. One major error is caused by applying manual and algorithm-based integration within the same evaluation procedure. Another originates from user-specific differences since especially over the course of a longer project the manual corrections may not be restricted to one singular user but multiple ones. In a previous dissertation by Max von Haugwitz [173], a small-scale study was conducted to investigate these differences by integrating a data set with an algorithm and a group of 10 users. For most tested

metabolites it was found that the relative standard deviations of all user results varied between 11 % and 27 % with threonine as an outlier at 57 %. Among the findings it was particularly striking that a clear correlation between single challenging features of a chromatogram and high inter-user deviations could not be determined underlining the complexity of the problem.
Since the MultiQuant approach was still in use for peak integration of all data from experiments detailed in this dissertation, there is a wealth of data to support claims about the frequency of manual intervention mentioned initially. Across 3 labeling experiments with a combined total of 28593 potential peaks, one third was revised manually. Focusing on a smaller example data set of 192 signals originating from one experiment and consisting to a large fraction of double peaks, the manual share was increased to 52 % of all signals, of which 22 % were false positives and 78 % were manually re-integrated. While manual interventions require the most time, the prevalence of mistakes necessitates at least the visual inspection of all peaks. These examples should illustrate the scale of human effort involved amounting to hours upon hours of monotonous screen work (the exact duration is difficult to track) which can be re-contextualized as an additional operating cost of a LC-MS/MS system.
Aside from the problems with semi-automated integration, a further downside of proprietary software in general is the limited number and high price of licenses so that commonly very few PCs are equipped with the software. This creates another bottleneck when multiple users need to analyze LC-MS/MS data concurrently.
Hence, it was decided to attempt building an in-house solution for LC-MS/MS peak integration.

3.3.1 Introducing PeakPerformance: A Python package for peak fitting and uncertainty quantification by Bayesian inference

The solution presented here takes the form of the open source Python package `PeakPerformance` applying Bayesian inference to chromatographic peak fitting. This became viable only recently due to the development of high performance, open source software packages enabling complex Bayesian modeling on regular PCs. All relevant peak parameters – i.e. baseline, peak area and height, mean, signal-to-noise ratio etc. – are encompassed in a singular model and are subsequently estimated simultaneously via MCMC. The key difference to previous attempts at peak fitting is that these peak parameters are defined as random variables and the result of the parameter estimations are distributions for each such parameter. Hence, instead of a point estimate for e.g. the peak area, a probability distribution is obtained meaning the provision of uncertainty quantification is built-in.
This development is synergistic to the rise of Bayesian methods such as Bayesian model averaging [174] and others in the field of ^{13}C-MFA which is directly connected to and based on the labeling data analyzed here in the form of LC-MS/MS peaks. Also, it constitutes a more realistic and honest representation of the experimental and analytical reality that a LC-MS/MS measurement is noise-afflicted and the resulting peak areas carry an uncertainty which has not been taken into account previously. Finally, the uncertainty can be used to exclude false positive peaks by defining a relative cut-off of the standard deviation with respect to the mean of a parameter's marginal posterior of e.g. 30 %. When combining this criterion with MCMC convergence checks such as

the potential scale reduction factor [125], this makes for a more effective filtering system than what has been available in e.g. MultiQuant. Since `PeakPerformance` was to be realized as a Python package, it could be installed on as many PCs as necessary without any additional associated license fees and would thus resolve the bottleneck of concomitant work of multiple users. Using the aforementioned new quality metrics, the accuracy of peak detection and accordingly the degree of automation are increased reducing the time investment into human supervision. Moreover, the Bayesian models and the distributions of their constituents are clearly defined, reproducible, and any changes to them can be documented comprehensively by version control in contrast to the deviations arising from manual integration by different personnel.
In summation, the stated approach was selected to first and foremost address the initially discussed issues with vendor software and connect with state of the art modelling methodology downstream in the overarching ILE workflow. A further mission statement was to enable less experienced users - both with regards to Python programming and to Bayesian statistics - to work with the software.
Regarding software implementation, `PeakPerformance` was created as an open source Python package freely available as a code repository on GitHub and intended for use on Windows and Linux systems. It is subdivided into the three modules `pipeline`, `models`, and `plotting`. The `pipeline` module is concerned with functions pertaining to raw data handling, sampling, filtering out false positive signals, and reporting results. It also features a ready-to-use example data pipeline showcasing the capacities of the program and serving as a convenience function for less experienced users to enable the usage of `PeakPerformance` out of the box. The `models` module contains all functions related to model definition and the `plotting` module encompasses all implemented visualizations.
Due to its modular design, `PeakPerformance` can easily be expanded by adding new models for deviating peak shapes or additional, e.g. diagnostic, plots. Aside from notes on the data format of raw data, the GitHub repository contains detailed instructions for installation and expansion of `PeakPerformance`. Bayesian inference is conducted utilizing the PyMC package [128, 175] with the external sampler `nutpie` [176] for improved performance. Both model selection and analysis of inference data objects were realized with the ArviZ package [177]. Since the inference data is stored alongside graphs and report sheets, users may employ the ArviZ package or others for further analysis of results if necessary.

3.3.2 Composition and assumptions of peak models in PeakPerformance

To realize a peak fitting approach of any kind, it is necessary to identify one model or a collection thereof which are able to describe the LC-MS/MS data of interest and express the peak distortions mentioned in the introduction (see section 1.3.1) while avoiding over- or underfitting. Although it is necessary to continue adding models when expanding the software to data from other chromatographic methods, a starting set of models was to be supplied which could adequately deal with the hitherto obtained LC-MS/MS data in this thesis.
The most basic and ideal peak shape took the form of a Gaussian or normal distribution which is assumed to be the combined result of mass transfer, longitudinal diffusion, and eddy diffusion

inside the HPLC column. A skew normal distribution [178] - i.e. a family of distributions with an additional skewness parameter α allowing for a one-sided distortion of the peak - is used to describe frequent phenomena such as tailing ($\alpha > 0$) and fronting ($\alpha < 0$) which are dependent on the adsorption isotherm of a given analyte. In case of $\alpha = 0$, the distribution simply results in identity to a normal distribution. Another regular phenomenon is the occurrence of double peaks meaning partly overlapping peaks without baseline separation. In the context of peak fitting, peaks with low resolution eluting in close succession need to be considered in a similar vein. For this purpose, double normal and double skew normal peak models were included applying the same models to the double peak case.

Discussing the model composition in detail, first some general assumptions and commonalities among the models will be established before addressing diverging parameters. Peak models in `PeakPerformance` require the definition of prior probability distributions for their parameters as well as the choice of an intensity function and a likelihood function. Generally, priors are derived from a given time series and assigned a weakly informative parametrization, such that the resulting inferences of parameters like the peak height are primarily based on the data. While defining priors in a data-dependent manner is generally to be avoided, it is clearly not tenable to define legitimate priors for all kinds of different peaks with heights and areas varying by multiple orders of magnitude and retention times, i.e. mean values, scattered across the whole run time of the LC-MS/MS method. In order to flexibly build models for all these peaks in an automated manner and embedded in a standardized data pipeline, some parameter priors had to be based on the raw data. If specific distributions or their parameters had to be restricted to certain value ranges, error handling was incorporated. For example, when only positive values were acceptable or when 0 was not a permissive value, a lower bound was defined using NumPy's `clip` function [179].

Regarding shared model elements across all intensity functions, one such component of all models presented hereafter is the likelihood function

$$L \sim \mathrm{Normal}(y, \mathrm{noise}) \tag{3.1}$$

with y as the predicted intensity and noise as the random variable expressing the standard deviation of measurement noise. This definition contains the assumption that observed intensities are the result of normally distributed noise around the true intensity values of a peak which is justified by the device-specific measurement noise of the QqTOF and the complexity of the biological matrix. In turn, the noise parameter is defined as

$$\mathrm{noise} \sim \mathrm{LogNormal}(\log_{10} \max(10, \mathrm{noise}_{\mathrm{guess}}), 1) \tag{3.2}$$

The log-normal distribution where the logarithm of the random variable follows a normal distribution was chosen partly to exclude negative values from the solution space and also due to its shape attributing a higher fraction of the probability mass to lower values provided the standard deviation is defined sufficiently high. This prior is defined in a raw data-dependent manner as the $\mathrm{noise}_{\mathrm{guess}}$ amounts to the standard deviation of the difference of the first and final 15 % of intensity values included in a given time frame and their respective mean values.

The intensity function itself is defined as the sum of a linear baseline function and a peak intensity function, the latter of which is composed of a given distribution's probability density function (PDF) scaled up to the peak size by the area or height parameter. The linear baseline

$$y_{\mathrm{baseline}}(t) = at + b \tag{3.3}$$

features the slope and intersect parameters a and b, respectively, both of which were assigned a normally distributed prior. The data-dependent guesses for these priors are obtained by constructing a line through the mean of the first and last three data points of a given intensity data set which oftentimes already resulted in a good fit. Hence, the determined values for slope (a_{guess}) and intercept (b_{guess}) are used as the mean values for their pertaining priors and the standard definitions are defined as fractions of them with minima set to 0.5 and 0.05, respectively. Here, the exact definition of the standard deviations was less important than simply obtaining an uninformative prior which, while based on the rough fit for the baseline, possesses a sufficient degree of independence from it, thus allowing deviations by the Bayesian parameter estimation.

$$a \sim \begin{cases} \mathrm{Normal}(a_{\mathrm{guess}}, \frac{|a_{\mathrm{guess}}|}{5}) & \frac{|a_{\mathrm{guess}}|}{5} \geq 0.5 \\ \mathrm{Normal}(a_{\mathrm{guess}}, 0.5) & \frac{|a_{\mathrm{guess}}|}{5} < 0.5 \end{cases} \tag{3.4}$$

$$b \sim \begin{cases} \mathrm{Normal}(b_{\mathrm{guess}}, \frac{|b_{\mathrm{guess}}|}{6}) & \frac{|b_{\mathrm{guess}}|}{6} \geq 0.05 \\ \mathrm{Normal}(b_{\mathrm{guess}}, 0.05) & \frac{|b_{\mathrm{guess}}|}{6} < 0.05 \end{cases} \tag{3.5}$$

Software-wise, the deterministic variables $\mathrm{noise}_{\mathrm{guess}}$, a_{guess}, and b_{guess} are calculated by the function `initial_guesses()` from the `models` submodule.

Beyond this point, it is sensible to subdivide models into single and double peak models since these subgroups share a common basis. Starting with the single peak models (figure 3.10), the normal-shaped model requires only three additional parameters for defining its intensity function. The mean value μ has a normally distributed prior with the center of the selected time frame $\min(t) + \frac{\Delta t}{2}$ as its mean and $\frac{\Delta t}{2}$ as the standard deviation where Δt corresponds to the length of the time frame. Accordingly, the resulting prior is rather compressed and weakly informative. The prior for the standard deviation of the normal-shaped peak model was defined with a half-normal distribution, once again to avoid values equaling or below 0. As a half normal distribution only features a standard deviation, this was set to $\frac{\Delta t}{3}$. The final parameter is the peak height used for scaling up the distribution to match the size of the peak. Here, a rather uninformative half-normal distribution with a standard deviation amounting to 95 % of the highest intensity value in the time frame was selected.

The second featured single peak model is based on the skew normal distribution which has an additional skewness parameter α enabling a one-sided distortion of the peak or resulting in a normal distribution when $\alpha = 0$. Hence, the prior of α is constituted by a normal distribution centered on 0 with a standard deviation of 3.5 to allow for a sufficiently large range of possible values for α and thus a realistic skew. Instead of the peak height, the peak area was utilized to scale the distribution, albeit with an identical prior.

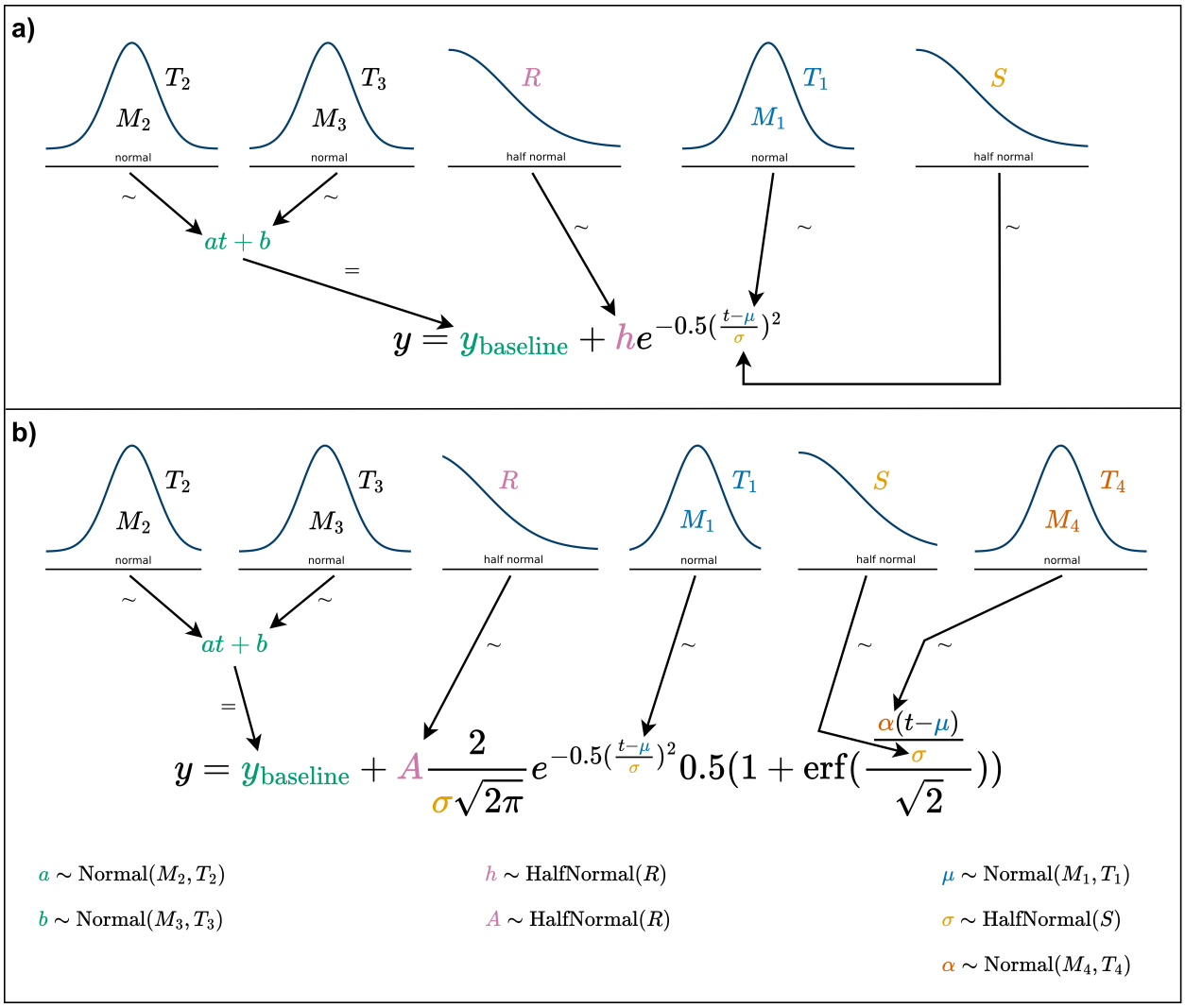


Figure 3.10: The intensity functions of normal (**a**) and skew normal peak models (**b**) as well as the prior probability distributions of their parameters are shown in the style of a Kruschke diagram [180]. Connections with $\sim$ imply stochastic and with $=$ deterministic relationships. In case of variables with multiple occurrences in one formula, the prior was only connected to one such instance to preserve visual clarity. The variables M_i and O_i describe mean values and T_i, R, and S standard deviations.

The double peak models (figure 3.11) featured many of the same variables as their single peak counterparts so only the differences will be highlighted here. All variables pertaining to the actual peak were turned into vectors with two entries and labeled with 0 and 1 by adding a named dimension to that effect. Aside from that, their priors remained unaltered except for the peak mean μ.

An early attempt to define double peak mean priors used an ordered transformation to enforce that $\mu_0 < \mu_1$, thereby avoiding confusion between the two peaks. However, this solution relied on two normally distributed priors for μ_0 and μ_1 whose location was anchored at one quarter and three quarters of the time frame, respectively. This setup had trouble fitting double peaks which were not centered quite exactly in the time frame but slightly shifted to the right or left. Since the HPLC retention time of a metabolite may change slightly over the course of a batch run of samples, it is not tenable to force a user to provide time frames exactly centered on the given double peak

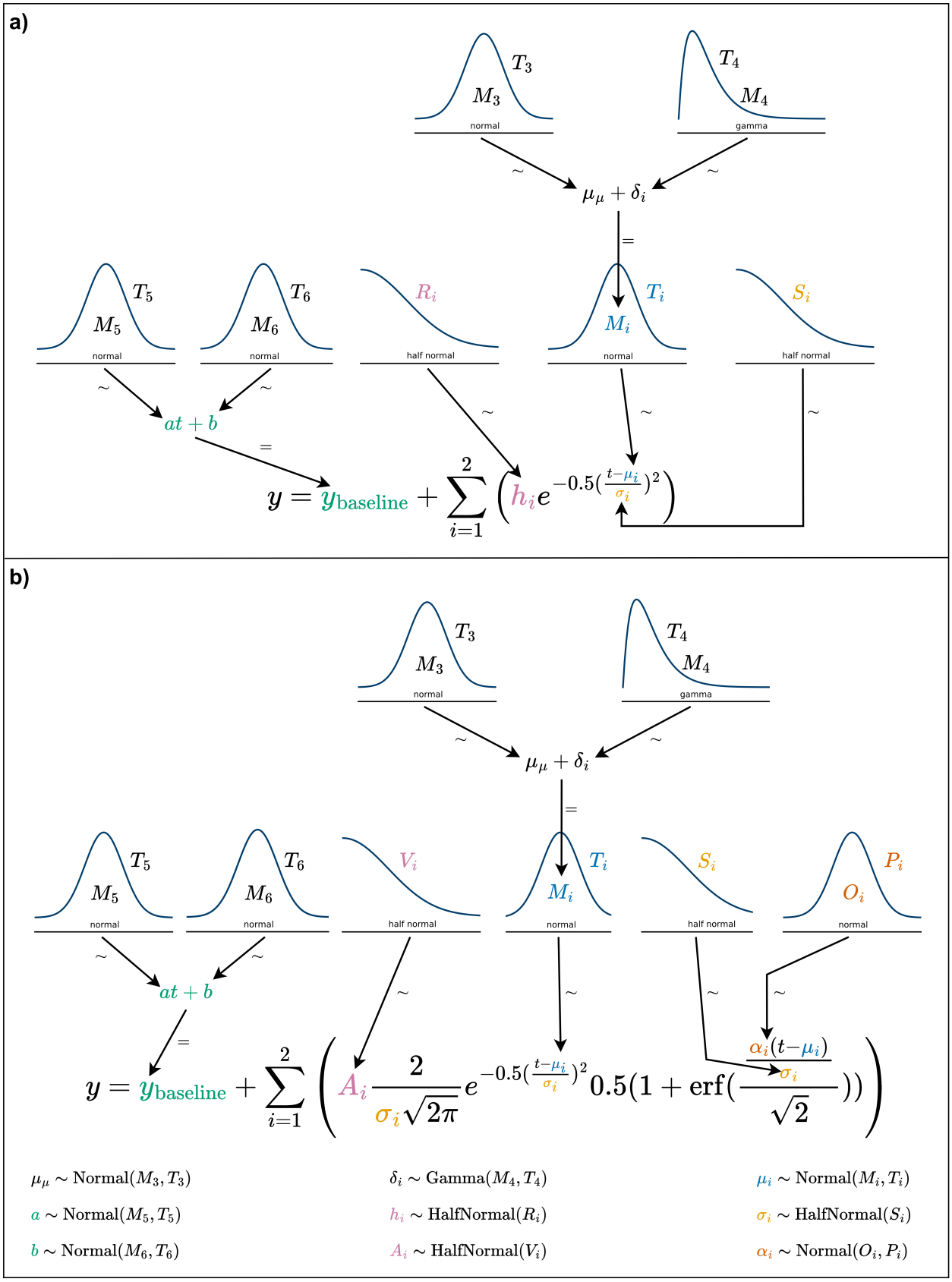


Figure 3.11: The intensity functions of double normal (**a**) and double skew normal peak models (**b**) as well as the prior probability distributions of their parameters are shown in the style of a Kruschke diagram [180]. Connections with $\sim$ imply stochastic and with $=$ deterministic relationships. In case of variables with multiple occurrences in one formula, the prior was only connected to one such instance to preserve visual clarity. The variables M_i and O_i describe mean values and T_i, S_i, P_i, and V_i standard deviations.

for each and every sample. Instead, a more flexible solution was found and implemented which is able to find double peak means across the whole time frame.
This was accomplished by adding several additional parameters to aid the creation of sub peak mean priors. More precisely, the mean of both peaks or group mean was introduced as hyperprior 3.6 with a broad normal prior which enabled it to vary across the time frame as needed.

$$\mu_\mu \sim \mathrm{Normal}\left(\min(t) + \frac{\Delta t}{2}, \frac{\Delta t}{6}\right) \tag{3.6}$$

By defining a separation parameter representing the distance between the sub-peaks of a double peak

$$\mathrm{separation} \sim \mathrm{Gamma}\left(\frac{\Delta t}{6}, \frac{\Delta t}{12}\right) \tag{3.7}$$

the offset of each peak's mean parameter from the group mean is calculated as

$$\delta = \begin{bmatrix} -\frac{\mathrm{separation}}{2} \\ \frac{\mathrm{separation}}{2} \end{bmatrix} \tag{3.8}$$

The priors for the mean parameters of each subpeak were then defined in dependence of μ_μ and δ as

$$\mu = \mu_\mu + \delta \tag{3.9}$$

Furthermore, `PeakPerformance` has the capacity of providing mean priors for multi peaks beyond double peaks using a similar approach in spirit but distinct in its implementation. Once again, the group mean and the sub peak means were defined according to equations 3.6 and 3.9, respectively. However, the major difference is that δ was defined as the zero-sum-normal (ZSN) distribution

$$\delta \sim \mathrm{ZeroSumNormal}(1) = \mathrm{Normal}(0, 1) \tag{3.10}$$

meaning that the sum of all values of the vector δ, containing draws from the ZSN, must amount to 0. Accordingly, this framework can accommodate two peaks or more since the length of variables μ_μ, δ, and μ are never explicitly stated. In theory, this approach could also be used for double peaks but in practice the ZSN attributes a high probability to a distance between the two sub peak means of 0 or close to it which could potentially lead to confusion between or a merge of the two peaks. However, when more than two sub peaks are present, the likelihood of such a mistake decreases significantly.
To illustrate the mode of operation of the ZSN approach, a short example will be presented (equation 3.11). In case of a triple peak located within a time frame from 10 to 14 min, 3 values x_1, x_2, and x_3 are drawn from the ZSN distribution. Their mean value μ_x is then subtracted from each resulting in a proposal for δ_1, δ_2, and δ_3. Note that the sum of the proposed values add up to 0. Assuming a proposal for the group mean μ_μ of 12 min, the subpeak means μ_1, μ_2, and μ_3 are obtained by summation of μ_μ and the pertaining value of δ_i amounting in this toy example to 10.7 min, 11.8 min and 13.5 min.

$$
\begin{aligned}
x_1 &= -0.3 \\
x_2 &= 0.8 \\
x_3 &= 2.5 \\
\mu_x &= 1 \\
\delta_1 = x_1 - \mu_x &= -1.3 \\
\delta_2 = x_2 - \mu_x &= -0.2 \\
\delta_3 = x_3 - \mu_x &= 1.5 \\
\delta_1 + \delta_2 + \delta_3 &= -1.3 - 0.2 + 1.5 = 0 \\
\mu_1 = \mu_\mu + \delta_1 &= 12 - 1.3 = 10.7 \\
\mu_2 = \mu_\mu + \delta_2 &= 12 - 0.2 = 11.8 \\
\mu_3 = \mu_\mu + \delta_3 &= 12 + 1.5 = 13.5
\end{aligned}
\tag{3.11}
$$

While all aforementioned parameters are necessary for the models, not all are of equal relevance for the user. A user's primary interest for consecutive data analysis generally lies in obtaining mean values, peak areas and perhaps - usually to a much lesser degree - peak heights. Since only one of the latter two parameters is strictly required for scaling purposes, different models as shown in figures 3.10 and 3.11 will feature only one or the other. Nonetheless, both peak area and peak height should be supplied to the user, hence the missing one was included as a deterministic model variable and thus equally accessible by the user. In case of normal-shaped peaks, the peak height h was used for scaling and the area A was calculated by

$$A = \frac{h}{\frac{1}{\sigma\sqrt{2\pi}}} \tag{3.12}$$

For skew normal-shaped peaks, the scaling parameter was the peak area. Since the mode and mean of a skewed distribution are – in contrast to normal distributions – distinct, the calculation of the height was nontrivial and ultimately a numerical approximation was added to the skewed models (listing A1).

Beyond these key peak parameters, all PyMC models created by `PeakPerformance` contain additional constant data variables and deterministic model variables. For example, the time series, i.e. the analyzed raw data, as well as the initial guesses for noise, baseline slope, and baseline intercept are kept as constant data variables to facilitate debugging and reproducibility. Examples for deterministic model variables in addition to peak area or height are the predicted intensity values and the signal-to-noise ratio defined here as

$$\mathrm{sn} = \frac{h}{\mathrm{noise}} \tag{3.13}$$

3.3.3 Structure and results of the PeakPerformance workflow

`PeakPerformance` accommodates the use of a pre-manufactured data pipeline for standard applications as well as the creation of custom data pipelines using only its core functions. The provided data analysis pipeline was designed in a user-friendly way and requires minimal programming knowledge (figure 3.12). As portrayed in an example notebook in the code repository, only a few simple Python commands need to be executed. Instead of relying on these convenience functions, experienced users can also directly access the core functions of `PeakPerformance` for a more flexible application which is demonstrated in yet another example notebook.
Before using `PeakPerformance`, the user has to supply raw data files containing a NumPy array with time in the first and intensity in the second dimension. For each peak, such a file has to be provided according to the naming convention specified in `PeakPerformance`'s documentation and listing 3.2. All these files then have to be gathered in one raw data directory.

```
<acquisition name>_<precursor ion m/z or experiment number>_<product ion m/z start>_<product ion m/z end>.npy
```

Listing 3.2: Generalized naming scheme for `PeakPerformance` raw data files.

Naturally, following the naming convention is only relevant when using `PeakPerformance`'s convenience functions. It is entirely possible and encouraged for more experienced users to create their own data pipeline based on `PeakPerformance`'s core functions. These core functions still require access to the raw data divided into sequences of time and intensity but the manner in which those are supplied is entirely up to the user.
If a complete time series of a 30 min - 90 min LC-MS/MS run were to be submitted to the program, however, the target peak would make up an extremely small portion of this data. Additionally, other peaks with the same m/z ratio and fragmentation pattern may have been observed at different retention times. Therefore, it was decided from the outset that in order to enable proper peak fitting, only a fraction of such a time series with a range of 3 - 5 times the peak width and roughly centered on the target peak would be accepted as an input. This guarantees that there is a sufficient number of data points at the beginning and end of the time frame for estimating the baseline and noise level, as well.
The provided data pipeline starts by defining a path to this raw data directory and one to a local clone of the `PeakPerformance` code repository. Using the `prepare_model_selection()` method, an Excel template file ("Template.xlsx") for inputting user information is prepared and stored in the raw data directory. It is the user's task, then, to select the settings for the pipeline within the file itself and they will be mentioned here only when becoming relevant to the workflow. Accordingly, the file contains detailed explanations of all settings and the parsing functions of the software feature clear error messages in case mandatory entries are missing or filled out incorrectly.
Since targeted LC-MS/MS analyses essentially cycle through a list of mass traces for every sample, a model type has to be assigned to each mass trace. Preferably, this is done by the user which is of course only possible when the model choice is self-evident. If this is not the case, an optional automated model selection step can be performed, where one exemplary peak per mass trace is analyzed with all models to identify the most appropriate one. It is then assumed that within one

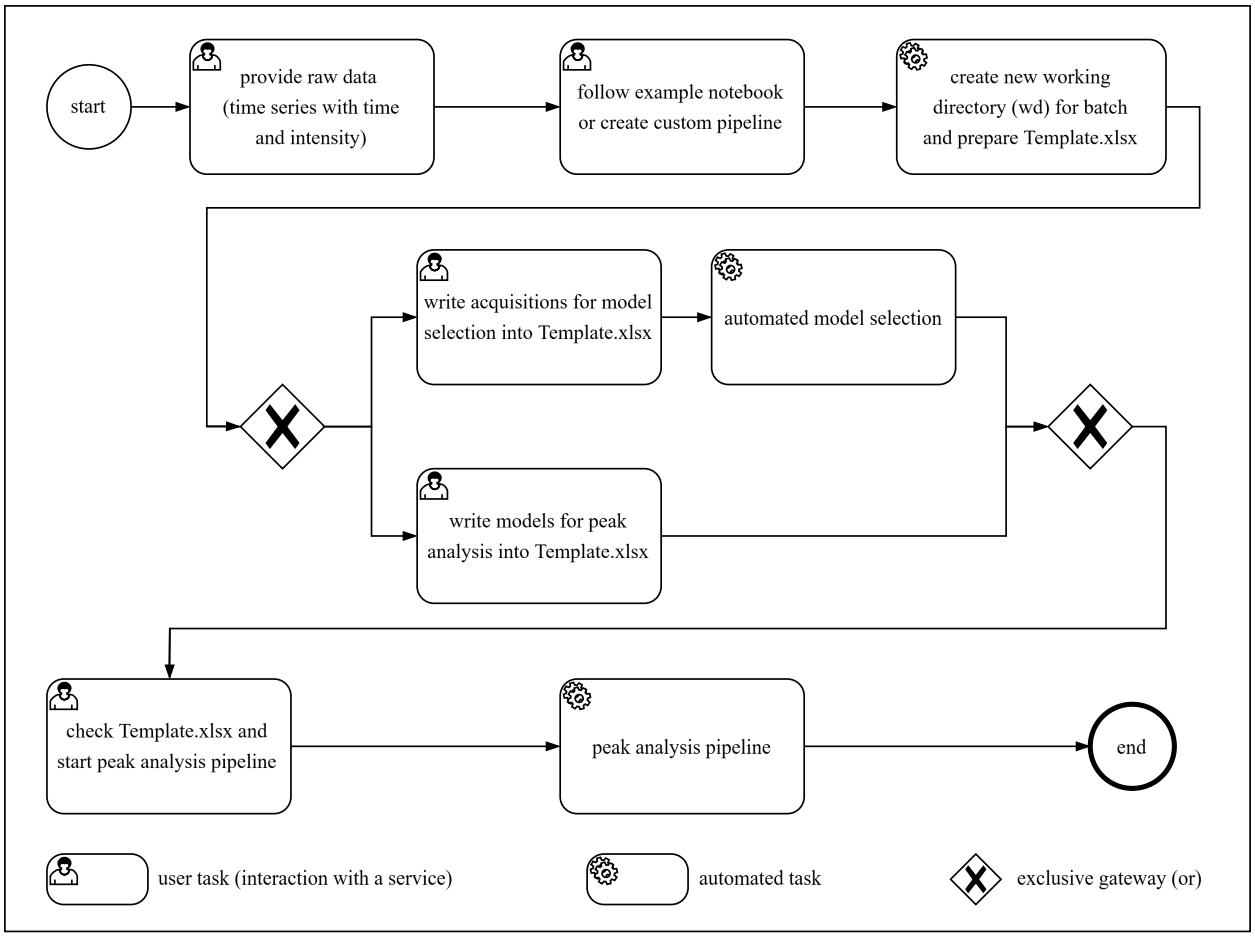


Figure 3.12: BPMN 2.0-inspired flow scheme depicting an overview of the pre-manufactured data analysis pipeline featured in `PeakPerformance`.

batch run, all instances of a mass trace across all acquisitions can be fitted with the same type of model. For this purpose, the user must provide the name of an acquisition, i.e. sample, where a representative peak for the given mass trace was observed. If e.g. a standard mixture containing all targets was measured, this would be considered a prime candidate. An additional feature lets the user exclude specific model types to save computation time and improve the accuracy of model selection by for example excluding double peak models when a single peak was observed. Upon provision of the required information, the automated model selection can be started using the `model_selection()` function from the `pipeline` module and will be performed successively for each mass trace. Essentially, every type of model which has not been excluded by the user needs to be instantiated, sampled, and the log-likelihood needs to be calculated. Subsequently, the results for each model are ranked with the `compare()` function from the ArviZ package based on an information criterion - either Pareto-smoothed importance sampling leave-one-out cross-validation (LOO-PIT) or the widely applicable information criterion (WAIC) [181, 182]. By default, the former is selected in `PeakPerformance`. This function returns a pandas DataFrame [183, 184] showing the results of the models in order of their placement on the ranking which is decided by the expected log pointwise predictive density (elpd). The best model for each mass trace is then written to "Template.xlsx".

Unfortunately, when testing both single and double peak models for a given mass trace, the double peak models can gain an advantage due to their relatively increased complexity by overfitting, meaning that even for obvious single peaks, double peak models would at times outperform single peak models in the ranking. Therefore, an additional function was implemented comparing the elpd scores of single and double peak models and offsetting the latter by an empirical constant. While this ameliorated the problem, it could not be fixed entirely so it is always recommended to exclude models which clearly should not be applied from the selection process which is in the user's interest anyway since it drastically decreases the computation time for the automated model selection.

After a model was chosen either manually or automatically for each mass trace, the peak analysis pipeline can be started with the function `pipeline()` from the `pipeline` module. The first step consists of parsing the information from "Template.xlsx". Since the data pipeline, just like model selection, acts successively, a time series is read from its raw data file next and the information contained in the name of the file according to the naming convention is parsed. All this information is combined in an instance of `PeakPerformance`'s `UserInput` class acting as a centralized source of data for the program.

Depending on whether the "pre-filtering" setting was selected, an optional filtering step will be executed to reject signals where clearly no peak is present before sampling, thus saving computation time. This filtering step uses the `find_peaks()` function from the SciPy package [185] which simply checks for data points directly neighbored by points with lower intensity values. If no data points within a certain range around the expected retention time of an analyte fulfill this most basic requirement of a peak, the signal is rejected. Furthermore, if none of the candidate data points exceed a signal-to-noise ratio threshold defined by the user in "Template.xlsx", the signal will also be discarded. Depending on the origin of the samples, this crude filtering step may reject a great many signals before sampling saving potentially hours of computation time across a batch run of the `PeakPerformance` pipeline. For instance, in bioreactor cultivations, a product might be quantified but if it is only produced during the stationary growth phase, it will not show up in early samples. Another pertinent example of such a use case are isotopic labeling experiments for which every theoretically achievable mass isotopomer needs to be investigated, yet depending on the input labeling mixture, the majority of them might not be present in actuality. In such a case, the number of time series with and without an actual peak might approach parity underlining the need for such a filtering step.

Upon passing the first filter, a MCMC simulation is conducted using a No-U-Turn Sampler [124], preferably - if installed in the Python environment - the nutpie sampler [176] due to its highly increased performance compared to the default sampler of PyMC. Before sampling from the posterior distribution, a prior predictive check is performed the results of which can be accessed and evaluated after the fact.

When a posterior distribution has been obtained, the main filtering step is next in line. The first criterion is constituted by checking the convergence of the Markov chains towards a common solution for the posterior represented by the potential scale reduction factor [125], also referred to as the $\hat{R}$ statistic or Gelman-Rubin diagnostic. If this factor is above 1.05 for any parameter, convergence was not reached and the sampling will be repeated once with a much higher number

of tuning samples. If the filter is not passed a second time, the pertaining signal is rejected.
Harnessing the advantages of uncertainty quantification, a second criterion calculates the ratio of the resulting standard deviation of a peak parameter to its mean and discards signals exceeding a threshold. Usually, false positives passing the first criterion are rather noisy signals where a fit was achieved but the uncertainty on the peak parameters is extremely high. These signals will then be rejected by the second criterion, ultimately reducing the number of false positive peaks significantly if not eliminating them.
If a signal was accepted as a peak, the final simulation step is a posterior predictive check which is added to the inference data object resulting from the model simulation.
After completing a cycle of the data pipeline or prematurely exiting it through one of the filters, the results need to be communicated and made available to the user. This is done in multiple ways. The most complete report is found in an Excel file called "peak_data_summary.xlsx". Here, each analyzed time series has multiple rows (one per peak parameter) with the columns containing estimation results in the form of mean and standard deviation (sd) of the marginal posterior distribution, highest density interval (HDI), and the $\hat{R}$ statistic among other metrics. Additional columns provide information on the acquisition and mass trace in question. Finally, there are columns stating whether the signal was recognized as a peak, if applicable the reason for the rejection of the signal, the utilized model for the simulation, and in case of a double peak a column specifying the peak number ("1st" or "2nd"). Accordingly, when a signal is rejected, it will nonetheless be added to the Excel report file and the exact reason for its rejection is detailed.
The second Excel file created is denominated as "area_summary.xlsx" and is a more handy version of "peak_data_summary.xlsx" with a reduced degree of detail. As implied by the name, from the peak parameters only the peak area remains and the columns are trimmed down to the essentials. Since subsequent data analyses will most likely rely on the peak area, this sheet should facilitate the further usage of the data.
The most valuable result, however, are the inference data objects saved to disk for each signal for which a peak function was successfully fitted. Conveniently, the inference data objects saved as `*.nc` files contain all data and metadata related to the Bayesian parameter estimation, enabling the user to perform diagnostics or create custom visualizations not already provided by `PeakPerformance`.
In case the user selects the "plotting" option, the results of the fit will additionally be visualized using the matplotlib Python package [186, 187]. Data formats can be specified when calling the plotting functions but the default arguments contain portable network graphics (`*.png`) and scalable vector graphics (`*.svg`).
For rejected signals, the time series is simply portrayed as a scatter plot so that - when in doubt - it can be checked visually whether the assessment was correct or whether a peak was present, after all. Regarding data visualization of accepted signals, `PeakPerformance`'s `plots` module offers the generation of two diagram types for each successfully fitted peak. The posterior plot presents the fit of the intensity function alongside the raw data points. The first row of figure 3.13 exhibits two such examples where the single peak diagram shows the histidine (His) fragment with a m/z ratio of 110 Da and the double peak diagram the leucine (Leu) and isoleucine (Ile) fragments with a m/z ratio of 86 Da. The posterior predictive plots in the second row of Figure 4 are provided for

the purpose of visual posterior predictive checking by comparing the observed and predicted data distributions. Since a ppc is based on sampling parameters from the posterior and subsequently obtaining predicted data points from the likelihood function, the result represents the theoretical range of values encompassed by the model. Accordingly, this plot enables users to judge whether the selected model can accurately explain the data.
To complete the example, table 3.2 shows the results of the fit in the form of mean, standard deviation, and HDI of each parameter's marginal posterior. In this case, the fits were successful and convergence was reached for all parameters. Most notably and for the first time, the measurement noise was taken into account when determining the peak area as represented by its standard deviation and as can be observed in the posterior predictive plots where the noisy data points fall within the boundary of the 94 % HDI.

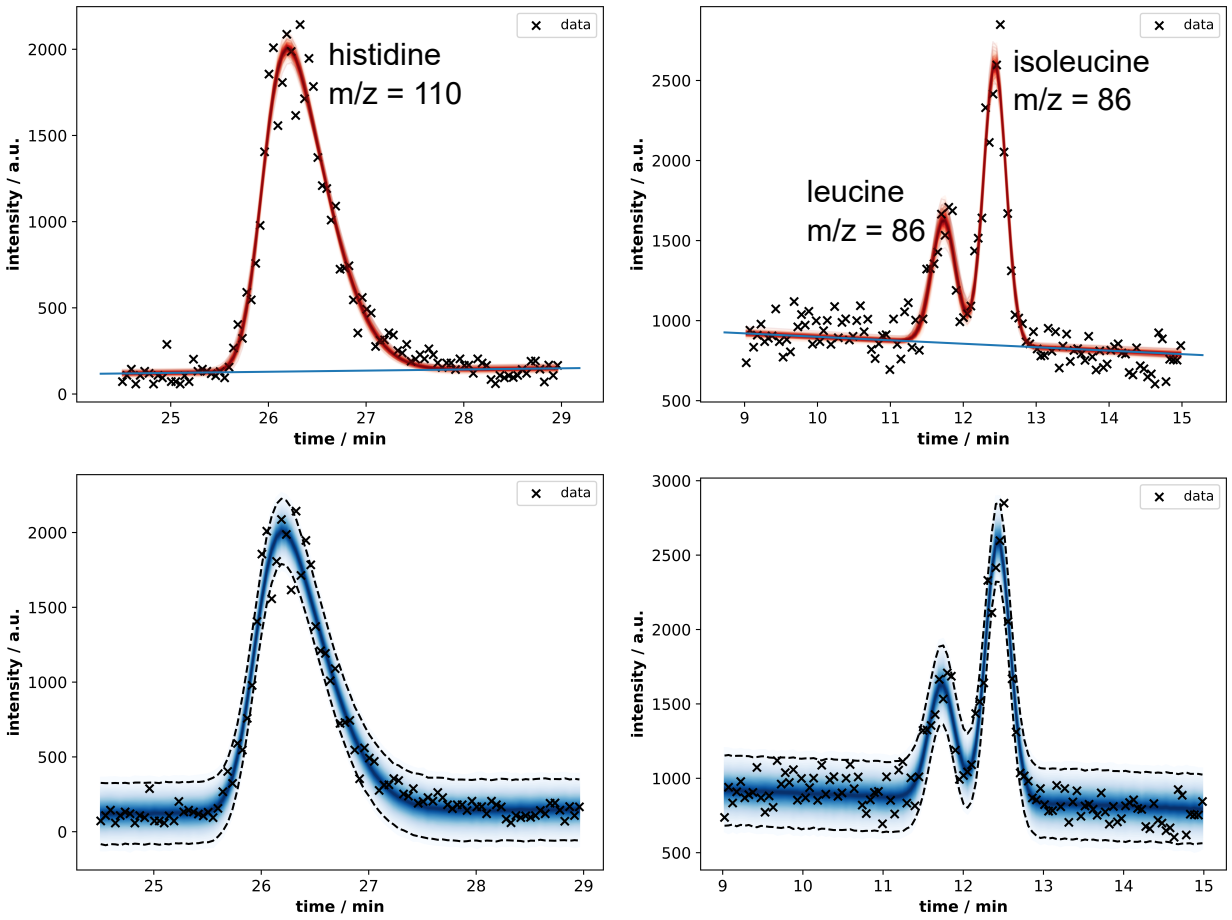


Figure 3.13: Results plots for a single His peak and a double Leu and Ile peak depicting the peak fit (first row) and the posterior predictive checks (second row) alongside the raw data. The dashed lines in the second row plots signify the borders of the 94 % HDI and the blue lines within this range represent individual posterior predictive samples. Thus, a darker blue region corresponds to a higher probability density. The numerical results are listed table 3.2.

Another important feature of `PeakPerformance` is constituted by the easy access to diagnostic metrics for extensive quality control. Using the data stored in an inference data object of a fit, the user can utilize the ArviZ package to generate various diagnostic plots. A particularly useful one is the cumulative posterior predictive plot portrayed in figure 3.14. This plot enables users to

Table 3.2: Depiction of `PeakPerformance` results for a single peak fit with the skew normal model and a double peak fit with the double normal model. Mean, area, and height have been highlighted in bold print as they constitute the most relevant parameters for further data evaluation purposes. The results correspond to the fits exhibited in Figure 3.13.

Parameter	**single peak** (skew normal model)				**double peak** (double normal model)			
	mean	**sd**	**hdi_3%**	**hdi_97%**	**mean**	**sd**	**hdi_3%**	**hdi_97%**
baseline_intercept	-43.94	7.41	-57.88	-30.02	1115.40	38.69	1040.14	1185.07
baseline_slope	6.66	0.51	5.71	7.63	-21.65	3.09	-27.50	-15.94
noise	103.63	7.51	89.50	117.26	118.63	8.01	103.52	133.29
mean	**25.95**	**0.01**	25.93	25.97	**11.73** **12.43**	**0.02** **0.01**	11.70 12.42	11.76 12.45
area	**1512.32**	**37.31**	1441.25	1581.37	**317.16** **674.34**	**28.84** **26.34**	263.23 623.47	370.56 722.88
height	**1879.72**	**37.71**	1809.30	1950.64	**774.99** **1762.66**	**65.28** **64.04**	653.50 1639.62	897.88 1881.66
std	0.53	0.02	0.48	0.56	0.16 0.15	0.02 0.01	0.13 0.14	0.20 0.17
sn	18.24	1.37	15.69	20.76	6.56 14.93	0.72 1.14	5.22 12.82	7.88 17.14
alpha	2.96	0.39	2.27	3.71	-	-	-	-

judge the quality of a fit and identify instances of lack-of-fit. As can be seen in the left plot, some predicted intensity values in the lowest quantile of the single peak example show a minimal lack-of-fit. Importantly, such a deviation can be observed, judged and is quantifiable which intrinsically represents a large improvement over the status quo.

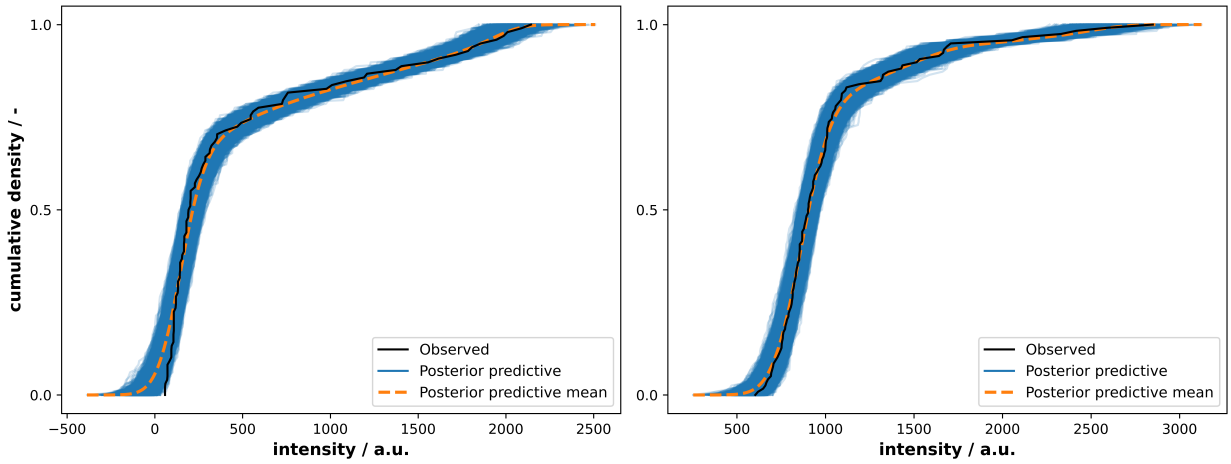


Figure 3.14: Cumulative posterior predictive plots created with the ArviZ package and pertaining to the example data of the single His peak (left) and the double Leu and Ile peak (right). The empirical cumulative density function (black) is in good agreement with the median posterior predictive (orange) and lies within the predicted variance (blue band), visually signifying that the model provides an adequate prediction irrespective of the intensity value.

The pipeline as laid out in detail in the preceding paragraphs is intended as a demonstration and to enable new users but it is entirely possible and encouraged to use `PeakPerformance` by directly calling upon its core functions and building a custom pipeline. Said functions encompass model definitions, parts of the model selection, MCMC sampling, and returning results in the shape of Excel sheets and plots. This way, experienced users are able to build pipelines closely aligned with their individual purpose and type of data.

3.3.4 Validation of PeakPerformance results

To validate the peak fitting approach as implemented in `PeakPerformance`, tests with multiple approaches were performed. The first testing stage employed synthetic, noise-afflicted data of each implemented distribution to check whether the original parameters would be recovered by inference with `PeakPerformance`. In particular, 500 random data sets were drawn and afflicted with normally distributed noise as described in section 2.6.4.
The arithmetic means portrayed in figure 3.15a were calculated based on a measure of similarity

$$F_{y/\hat{y}} = \frac{y}{\hat{y}} \tag{3.14}$$

where y represents the estimated parameter value and $\hat{y}$ its pertaining ground truth. The plot was focused on the most important peak parameters which were restricted to mean, standard deviation, area, and height. As the mean of all 500 tests was then portrayed along with its standard deviation, a value of 1 implies a near identity between estimation and ground truth with a higher precision and certainty the lower the exhibited variance is. As can be observed here, the normal and skew normal single peak models from `PeakPerformance` were able to uncover the ground truth of their respective noisy distributions well regardless of the noise level.
Beyond reproducing appropriate data, it was tested how the models would perform when each was paired with noisy data from the other intensity function. The result was still surprisingly accurate with respect to the estimated mean, area, and height, especially for the skew normal model paired with normally distributed, noisy data. The normal model paired with skew normal data merely slightly underestimated the area which can be easily explained as underfitting the skewed tail of the intensity function which cannot be reproduced with a normal shape. Both models failed to reproduce the true standard deviation, though, but this parameter is not essential whilst the remaining peak parameters were correctly estimated.
The next step concentrated on the aspect of the effects of fitting normally distributed data with a skew normal distribution and vice versa. In contradistinction to the previous step, however, marginal cases were observed where the skewness parameter α was fixed at a value of 1 indicating a slight skew which especially after adding noise cannot be clearly discerned as normal-shaped or skewed, any more. More importantly, though, the automated model selection implemented in `PeakPerformance` might settle on either of the two candidate models depending on the case. Therefore, it was important to verify whether the choice of one model above the other would have a systematic effect on the estimation of peak area and height, especially.
In an analogue manner to the first test, 100 random data sets were generated, only this time in addition to the aforementioned mean of 6 and skewness of 1, the area was defined as a constant of 8. Since the comparison now focused not on the ground truth but on the similarity of the estimations, the results of both models for area and height were evaluated as the ratio

$$F_{n/\mathrm{sn}} = \frac{A_{\mathrm{normal}}}{A_{\mathrm{skew\ normal}}} \tag{3.15}$$

where A_{normal} and $A_{\mathrm{skew\ normal}}$ are the estimated areas with normal and skew normal models,

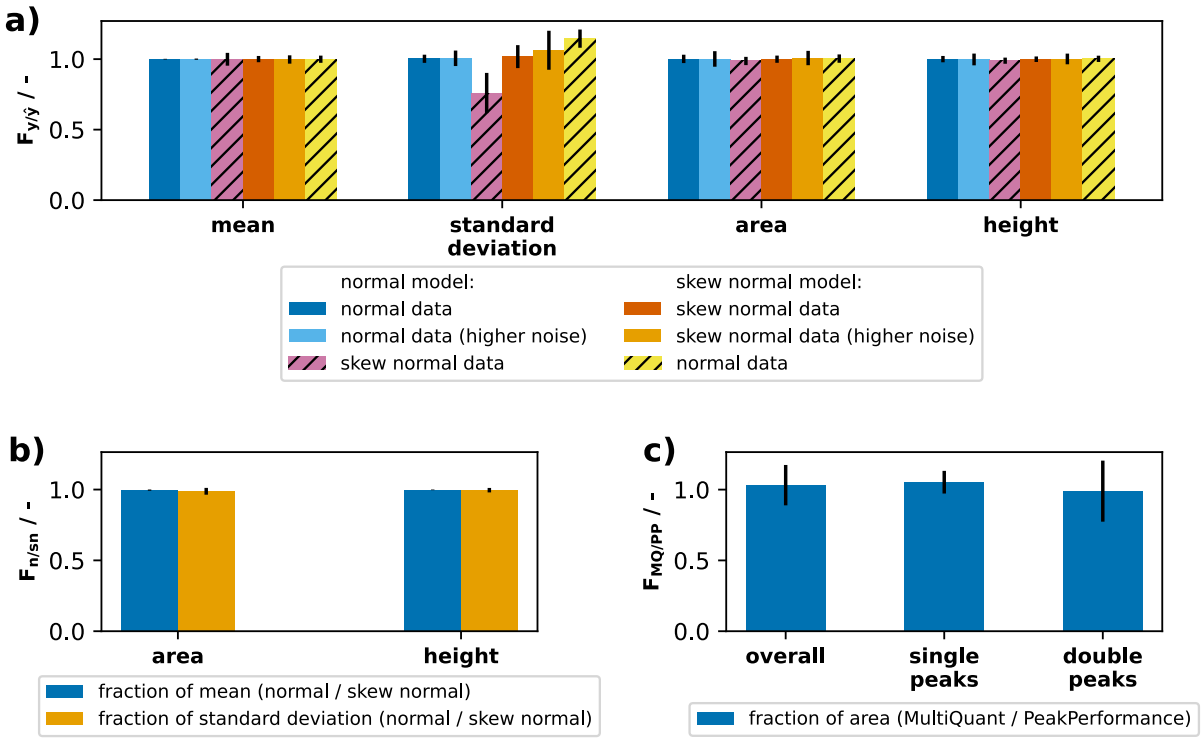


Figure 3.15: Validation of results from `PeakPerformance`. **a)** Noisy synthetic data was randomly generated from one of the implemented distributions and the program's ability to infer the ground truth was observed. Portrayed are the fractions of estimated parameter to ground truth. **b)** The influence of model choice between normal and skew normal model in marginal cases with little to no skew was tested and the ratios between results from both models are plotted. **c)** Lastly, experimental data was analyzed with `PeakPerformance` (version 0.7.0) and compared to results achieved with the commercial software Sciex MultiQuant (version 3.0.3).

respectively. Figure 3.15b, then, exhibits the mean values and standard deviations of the ratios $F_{n/\mathrm{sn}}$ from all 100 tests. As one of `PeakPerformance`'s unique selling points is the quantification of uncertainty for each peak, it was also relevant to investigate whether the estimated uncertainties would be comparable between the models. Therefore, both the ratios of the estimated means and standard deviations are shown in figure 3.15b.

As demonstrated by all portrayed values approaching 1 with very narrow standard deviations, the resulting estimates for area and height as well as their uncertainties were near identical between the two models. Hence, the choice of the model in such marginal cases as tested here is inconsequential and the application of any of the eligible models is justified.

The final test comprised a thorough comparison of `PeakPerformance` (version 0.7.0) results with those from the established vendor software Sciex MultiQuant (version 3.0.3) and was accordingly the first test to utilize actual experimental data in place of the synthetic data sets of the previous validations. Since the results generating mechanisms and the underlying assumptions are quite distinct between the two programs, identical results were not expected but they should nonetheless be located in a similar range. Emphasizing the most important parameter for users, the test was focused on the peak area, in particular (figure 3.15c). The plotted means and standard deviations

of the fraction of the obtained areas was determined as

$$F_{\mathrm{MQ/PP}} = \frac{A_{\mathrm{MQ}}}{A_{\mathrm{PP}}} \tag{3.16}$$

where A_{MQ} denominates the area yielded by MultiQuant and A_{PP} the area from `PeakPerformance`. Beyond the comparability of the resulting peak area ratio means, it is relevant to state that 103 signals from MultiQuant (54 % of total signals) were manually modified. Of these, 31 % were false positives and 69 % were manually re-integrated. These figures are the result of a relatively high share of double peaks in the test sample which generally give a lot more cause for manual interference than single peaks. In contrast, however, the `PeakPerformance` pipeline was only started once and merely two single peaks and one double peak were fit once more with a different model and/or increased sample size after the original pipeline batch run had finished. Among the 192 signals of the test data set, there were 7 noisy, low intensity signals without a clear peak which were recognized as a peak only by either one or the other software and were hence omitted from this comparison. By showing not only the mean area ratio of all peaks but also the ones for the single and double peak subgroups, it is evident that the variance is significantly higher for double peaks. In case of this data set, two low quality double peaks in particular inflated the variance significantly which may not be representative for other data sets. It has to be stated, too, that the prevalence of manual re-integration of double peaks in MQ might have introduced a user-specific bias, thereby increasing the final variance. Nevertheless, it could be shown that `PeakPerformance` yields comparable peak area results to a commercially available vendor software.

3.3.5 Considerations regarding the PeakPerformance Python package

When taking on the challenge of devising an alternative to available vendor software, an approach that has surfaced recently is the machine learning-based peak data processing [188, 189]. This is certainly a promising approach but nonetheless it was decided against. Some drawbacks of a ML-based solution are the need for sufficient and representative training data as well as the opacity of its mechanism which essentially constitutes a black box. In contrast, with `PeakPerformance` the root cause of a bad fit can be investigated and the models and priors are clearly defined. Moreover, the ML-based approach - just like the MultiQuant one - does not perform uncertainty quantification yielding merely a point estimate and thus assuming a noise-free measurement which does not faithfully represent the experimental reality. The most important reason for the Bayesian route, however, is the increasing application of Bayesian statistics to the subsequent modelling steps such as Bayesian model averaging and eventually ^{13}C-MFA, as well. When using `PeakPerformance` in sequence with these techniques, a holistic Bayesian data evaluation pipeline can be achieved.

Despite the success of `PeakPerformance` as a stand-alone Python package with the designated goal of dealing with LC-MS/MS data, some drawbacks remain, namely the complete lack of parallelization leading to inflated computation times, the need to execute the software on the user's local machine, and the focus on QqTOF data in particular.

To truly realize `PeakPerformance`'s potential in an automated high-throughput workflow, all these

points were addressed by creating a directed acyclical graph (DAG) based on the `PeakPerformance` package on an Apache Airflow [147] computation cluster which is described in detail in the following section. This was combined with a MS data cluster being concomitantly established at the IBG-1 by a third party so that the Airflow `PeakPerformance` pipeline was largely independent of the local file system. The mentioned MS data cluster itself is not part of the present thesis, the author merely provided beta testing for it.

3.3.6 Parallelization and scale-up of PeakPerformance on an Airflow cluster

For the Airflow approach to work, some changes were necessary since `PeakPerformance`'s `pipeline` functions were not designed from the ground up with this goal in mind. However, the previously mentioned core functions like model definition and sampling could be used as is, merely the framework embedding them had to be altered.
The start of the Airflow pipeline or DAG depicted in figure 3.16 remained fairly similar to its stand-alone counterpart, opening with the creation of a run directory or working directory based on the current date and time. The pertaining task `prepare_run_inputs()` also imports the necessary Python packages PyTensor [190], PyMC, and `PeakPerformance` so that import errors are noticed at the onset of the process. Thereafter, the user has to move an Excel file denominated as "inputs.xlsx" containing the user information into the working directory, upon which it is read and sanity checks are performed. Ideally, "inputs.xlsx" should be prepared beforehand since a timer to await the arrival of the file for 15 min was incorporated into this first task.
Naturally, the option for an automated model selection was implemented in the DAG, as well. Just like in the stand-alone version, the user merely has to provide an exemplary acquisition per mass trace and retains the option to exclude specific model types from the selection. The first major difference, then, is constituted by the parallelization of model selection across the number of workers currently available with the correct worker image. Making use of Airflow's dynamic task mapping, it is possible to provide a task with a list of jobs and for each entry in the list, a separate instance of the task is created. These instances are then completed by as many workers as currently available in parallel and without having to state the number of jobs beforehand, hence the dynamism. It is further possible to provide a task with two lists whereupon all combinations of entries form a new job list. The Airflow `PeakPerformance` DAG uses both of these variations to its advantage. For the model selection, the preceding task `get_mass_traces()` cycles through all acquisitions of the given LC-MS/MS batch present on the MS data cluster and through the rows of the "download" tab of "inputs.xlsx". Whenever an acquisition name equals one stated in the "acquisition_for_choosing_model_type" column, the rest of the given information is used to download the time series from the MS data cluster.
Here, the device used to generate the data led to two distinct versions of this step. When using a QqTOF for a product ion scan, the acquisition method for the device only contains precursor m/z ratios for the Q1 and a TOF m/z range. Therefore, the user has to provide more narrow product m/z start and end values within this TOF range for each target fragment since this information is not contained in the raw data files stored on the MS data cluster. For example, in the QqTOF measurements of the present thesis, a TOF range of 0 to 350 was selected and extracted ion

chromatograms were created with product m/z widths of 0.2 - 0.4 Da. On the other hand, when performing a multiple reaction monitoring (MRM) experiment on a QqQ, both precursor m/z ratios for Q1 and product m/z ratios for Q3 must be stated before the analysis and are thus present in the raw data files. Hence, much less user input is necessary and all mass traces of a batch can be downloaded automatically from the MS data cluster.

At this point, a limitation of Airflow comes into play, namely the restrictions placed upon the transfer of data between tasks. It it not intended to move large volumes of data between tasks. More importantly for the `PeakPerformance` DAG, though, the data types which can be transferred at all are limited to only those that are JSON serializable. These are comprised by dictionaries, sequences, integers, floats, Booleans, and None. Most notably absent from this list are custom classes and common data types such as pandas DataFrames and NumPy arrays. Accordingly, it was not possible to simply store all relevant metadata and raw data of a mass trace in a DataFrame to be imported in the next task. To sidestep this problem, the NumPy array with the time series is stored locally in the working directory as a `*.npy` file instead of being returned while the rest of the mass trace's data is collected in a dictionary and appended to a list. This list is then returned and serves as the job list for the subsequent `model_selection()` task of which the aforementioned parallelized instances are created via dynamic task mapping. Whenever the user had already specified a model type for a mass trace in "inputs.xlsx", this particular instance of the task is skipped. Upon a successful model selection, each dictionary containing the information regarding a single mass trace is updated with the chosen model type. Simultaneous with the model selection, the task `get_acquisition_list()` downloads a list of all acquisitions of the given batch from the MS data cluster subtracted by an optional list of acquisitions to exclude which the user may provide in "inputs.xlsx".

After all instances of the model selection task have been successfully completed, the list of acquisitions and the list of mass trace dictionaries originating from the two parallel tracks are multiplied to expand the `peak_analysis()` task which is then performed at the highest degree of parallelization the cluster can offer. This represents the aforementioned dynamic task mapping based on two input lists. Since the time series of most combinations of acquisitions and mass traces were not downloaded before, this is performed here. Then, the actual peak fitting is executed in much the same way as in the data pipeline provided in the stand-alone version.

One point of departure is the way the Excel results files are created. The stand-alone version updates one DataFrame successively which is possible due to the lack of parallelization. When numerous instances of the same task work concomitantly, some of them may require access to the Excel file at the same time which causes errors. At first, it was attempted to solve this issue by using the filelock Python package [191] to manage access to the file but upon encountering further problems it was decided to instead create the results files in a separate and final task. Once again the results could not just be transferred as inference data objects or DataFrames since these are not JSON serializable. Hence, the inference data objects are stored locally and the DataFrames with the results are converted to dictionaries before being delivered to the final task.

In `collect_results_and_report()`, these dictionaries are gathered, re-converted into DataFrames whereupon Excel report sheets are created just like in the stand-alone version of `PeakPerformance`.

In summary, the `PeakPerformance` DAG on the Apache Airflow cluster is a highly parallelized,

alternative version of the program which is thus much more efficient and explicitly suitable for larger volumes of data and high-throughput processes. Due to the computation times and the strictly successive, one-track peak analysis, this would be difficult to achieve with just the stand-alone version of the program. The state of the art workflow based on MultiQuant would most certainly be inept for high-throughput data as the degree of human supervision and work time necessary are simply not feasible.

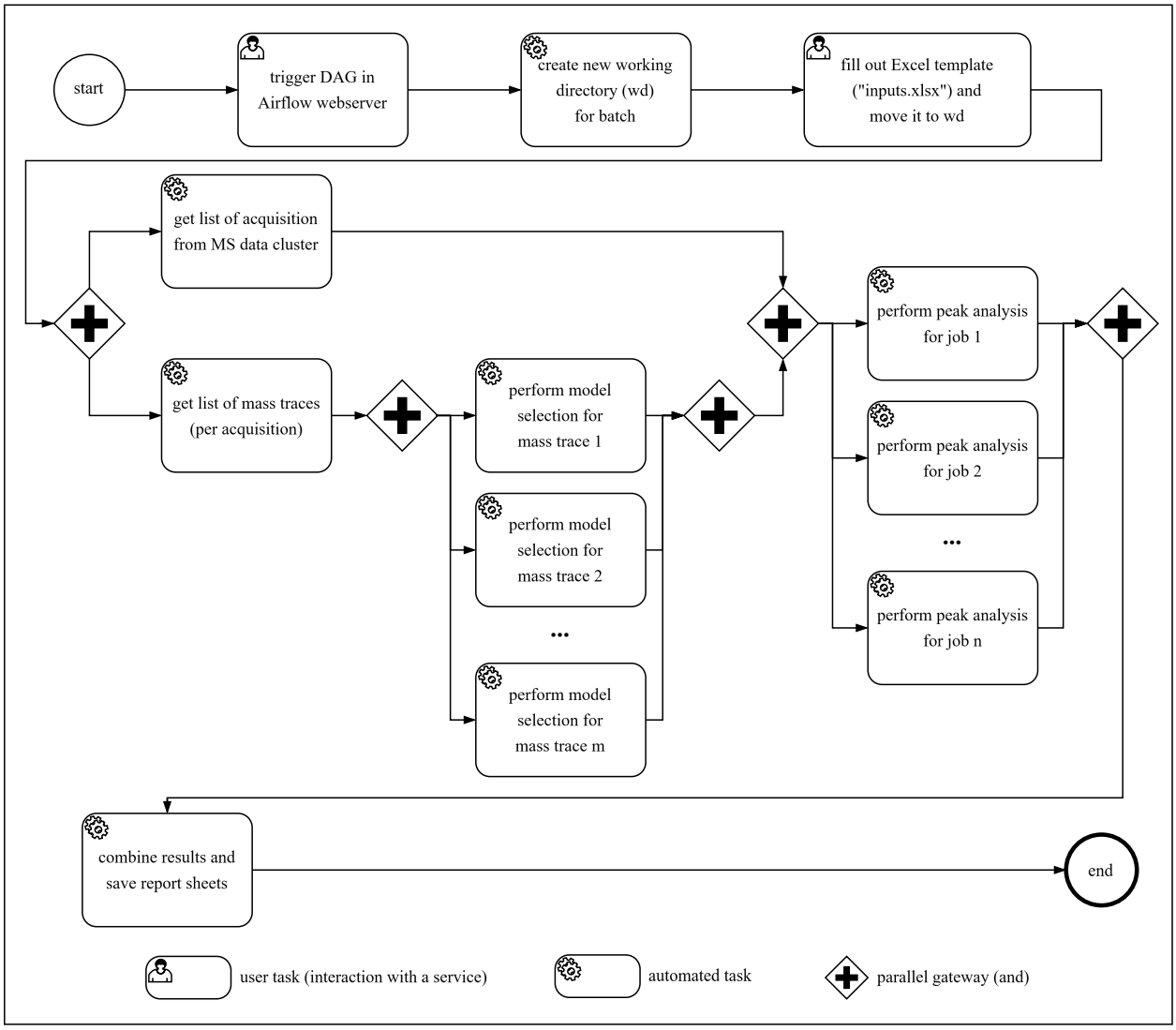


Figure 3.16: BPMN 2.0-inspired flow scheme portraying the general workflow of the `PeakPerformance` Airflow DAG.

3.3.7 Conclusion and outlook for PeakPerformance

Regarding `PeakPerformance` in general, i.e. independent of the specifics of its stand-alone and Airflow versions, the program will expand in versatility and applicability to different types of chromatographic data by increasing the number and variety of peak models. To facilitate this development, a guide was included in the code repository on GitHub detailing the necessary steps to introduce a new model. However, an adequate model type alone does not guarantee a legitimate fit, the priors

of random variables might require adaptation to the data in question, as well. As much as the barrier of entry for `PeakPerformance` was kept low, some expertise and time investment remains necessary to establish a model for data from a particular chromatographic method.
A loosely related feature that is still missing but would benefit the program massively is the recombination of different single peak models into a mixed multi peak model. Currently, both double peak models are essentially sums of the linear baseline and two identical single peak models so a combination of a normal-shaped peak and a skew normal-shaped peak is not implemented. Problematic or outright failed double or multi peak fits could potentially be amended by using such a mixed double peak model. An envisioned future implementation of this feature could work by dividing the model definition for multi peak models across different sub-functions and searching for the best combination of single peak models. A detriment to this idea, however, is the exponentially increasing number of potential double peak model candidates when the combination of every implemented single peak model with each other is tested during model selection. To limit the negative impact on computation time, the model selection would ideally need to be further parallelized, not just by job but on a job and model variant basis which is theoretically possible on Airflow but in practice demands a larger number of worker nodes than can be feasibly spared with the current infrastructure.
Regarding the elimination of false positive peaks, currently the peak parameter uncertainties and the $\hat{R}$ statistic are used but other metrics like e.g. ess could be included in the decision making process, as well. A more robust peak identification was, after all, one of the key targets from the beginning of the `PeakPerformance` development and such a development would only serve to increase said robustness.
One final feature of great value - were it implemented - is constituted by the use of hierarchical models. When e.g. having measured multiple replicates for quantitation, a hierarchical model may be employed to gain one joint resulting posterior from all relevant replicates using a single model.
With respect to the `PeakPerformance` version on the Airflow cluster, there are three main open points. The first is the usage of `PeakPerformance` for LC-MS/MS data from a triple quadrupole (QqQ). The code base was already subdivided into separate tracks for QqTOF and QqQ data but the latter has not been applied to experimental data and can be best described as having reached a prototype status.
The second open topic is the integration of the nutpie package into the Docker [192] image for the Airflow `PeakPerformance` workflow. Due to conflicts with other included packages, it has to date not been possible to add this much more efficient solver which is the default choice in the `PeakPerformance` stand-alone program.
Finally, the third point is one pertaining to the infrastructure of the workflow. While the results are as of yet still stored locally, it is planned to employ a database for the inference data objects. The only files returned locally will be the Excel results sheets which barely demand any memory. Plots, too, will not be saved but instead a graphical user interface (GUI) will be created to allow users to visualize everything stored in the inference data objects, meaning in the first instance raw data, prior predictive checks, posteriors, and posterior predictive checks.
Another option which is enabled by the parallelization in Airflow is to perform model selection for every single signal. This would abolish the assumption that all mass traces across a batch of

acquisitions can be fitted using an identical model type. This might lead to more accurate results and avoid false negatives when peaks are rejected only due to non-convergence of the model when a different model perhaps would have converged. Even with the parallelized Airflow DAG, though, computation times would be increased again and at some point, energy demand and the impact on the cluster workload need to be considered, as well.
If the development of `PeakPerformance` and its Airflow counterpart is continued and the features mentioned above are implemented, this peak data evaluation workflow would increase in efficiency over its present state and vastly so relative to the previous state of the art workflow based on vendor software. Even so, `PeakPerformance` as a tool for automated LC-MS/MS peak data analysis employing Bayesian inference, has improved the degree of automation of said analyses and uniquely introduced uncertainty quantification to LC-MS/MS measurements and accordingly represents a valuable addition to the overarching automated ILE workflow.

3.4 Data evaluation and visualization

Somewhat paradoxically, judging by the efforts described in the previous section, the data processing up to this point with either MultiQuant or `PeakPerformance` yielded peak areas which are still characterized as raw data. To be able to derive qualitative or quantitative information from them, first TMIDs need to be obtained by way of normalization. Since these represent the labeling states of their metabolites, they can serve as the basis for biological interpretations, either directly or indirectly via further processing through modelling. Additionally, by simply plotting TMIDs, a visual inspection before any modelling has occurred can already reveal problems with the LC-MS/MS measurement or peak integration so the ability to visualize is not only practical but may save a significant amount of time, especially when experimenter and modeler are not the same person. Originally starting from MultiQuant data and later-on including a separate track for `PeakPerformance` data (see 3.4.2), the program presented in this chapter is intended not only for TMID calculation and visualization but also to interface as seamlessly as possible with the next step in the overarching ILE workflow, namely the natural isotope correction with uNAC.

3.4.1 Implementation for peak area data from Sciex MultiQuant

The Sciex MultiQuant track of the program is confined in one singular Python file (evaluation_and_visualization_MQ.py) containing mostly the class `MultiQuantDataProcessing`. The file and accompanying exemplary raw data alongside an example notebook are part of a code repository in JuGit. To accommodate the different plotting and data report features, its class attributes encompass the denominations of all mass traces of a given fragment and its elemental formula among others. The general workflow of the script including the data flow is portrayed in figure 3.17.

After supplying the results from MultiQuant, which have to be copied manually from the software, and defining the user information necessary to instantiate the `MultiQuantDataProcessing` class, the workflow starts with the data preparation section. Here, data is prepared for the following steps and all calculations are front-loaded, i.e. TMIDs as well as the arithmetic means and standard deviations of biological replicates are computed. Specifically, the method `open_and_prepare()` parses the raw data file, `calculate_tmids()` performs the normalization to receive TMIDs, and `calculate_mean_stdev()` computes means and standard deviations of biological replicates. These methods also feature basic error handling by e.g. omitting a sample where no mass trace of a given fragment was detected during the calculation of means and standard deviations while informing the user of this action via a displayed statement. The results of these computation steps are stored in DataFrames serving as the input for the subsequent tasks of data visualization and reporting which are thus independent of each other.

Options for visualization include bar diagrams for isotopically stationary labeling data with the `plot_bar_diagram_replicate()` method and line diagrams for INST data with the `plot_tmid_transient()` method. Bar diagrams may feature either a single TMID or the TMID means and standard deviations across a biological replicate. Line diagrams portray the time course of TMIDs so that each value on the time axis pertains to one TMID meaning a much

higher density of information is at display. Additionally, for INST data, both the (ideally isotopically stationary) end point and the kinetics of all mass traces are important, hence the choice of diagram.
To enable the creation of stylistically unified line diagrams for larger figures, an option is included to choose a constant color scheme, whereby each mass trace has a hard-coded color independent of which fragment it pertains to. Due to the sheer number of possible mass traces and combinations thereof, it is nigh impossible to devise a general color scheme which is visually discernible in all cases so users are advised to alter or exchange colors based on which mass traces are needed for a given data set. Moreover, it is recommended to additionally use text labels of the mass trace denominations right next to their respective lines to improve the readability of the resulting diagrams.
Since one of the key advantages of the automated ILE pipeline is the increase in throughput at decreased expenditure, it follows that the number of possible parallel or comparative ILEs should increase, as well. When comparing results from these experiments, then, it is advantageous to plot multiple time courses of TMIDs into the same diagram to enable a direct comparison of the molecular labeling states. Therefore, another type of line diagram was added to enable the joint presentation of data originating from multiple distinct experimental conditions using the `plot_tmid_transient_several_replicates()` method. Conditions might refer to either a different input labeling mixture or e.g. a deviating strain or medium, depending on the experiment. Since colors are utilized to differentiate mass traces, these conditions are distinguished by line style (e.g. dashed, dotted etc.). Whenever possible, the style of all diagrams produced by this program was unified, one of the only exceptions being the line diagrams featuring several mass traces as their expansive legends require more space and thus a larger figure size.
Regarding the data compatibility section, its main focus is the re-formatting of the resulting DataFrames into a shape which can be directly used as the input for the next step in the overarching ILE workflow, namely the natural isotope correction with the in-house developed meta-tool and Python program uNAC. Accordingly, there is one method for isotopically stationary (`prepare_isotopically_stationary_data_template()`) and one for INST data (`prepare_inst_data_template()`) both of which store the DataFrame as an Excel file. Structurally, the resulting Excel files are similar containing the fragment name and replicate ID as well as the elemental formulae, C atom number, and mass shifts of precursor and product ion. The only difference lies in the representation of relative abundances of mass traces which are contained in a single column for isotopically stationary data and in one column per time point for INST data. Here, all relative abundances pertain to single replicates and no mean values or standard deviations are included.
In summary, the output of the present program comprise uncorrected TMIDs, their graphical representations and report files facilitating further data evaluation. The program's code repository in JuGit has been provided with exemplary results from MultiQuant and an example notebook showcasing all discussed methods.

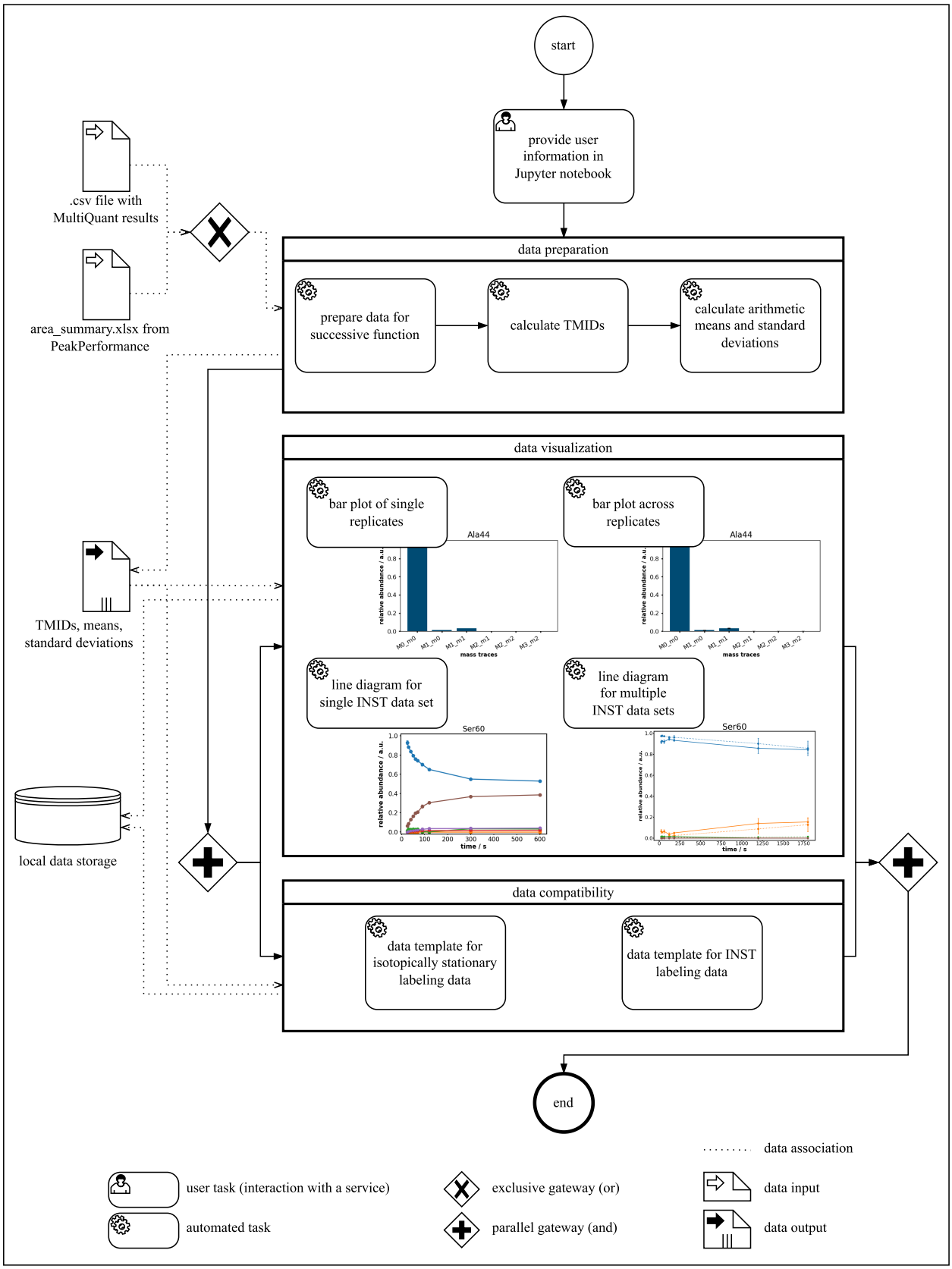


Figure 3.17: BPMN 2.0-inspired flow scheme portraying the functions and data flow of the Python program for data evaluation starting from peak area data. These functions include TMID calculation by normalization, visualization via various bar and line diagrams, and creating data templates for natural isotope correction via the in-house developed Python package uNAC.

3.4.2 Retrofitted compatibility with PeakPerformance results

The inception of the program as described in the previous section occurred long before the development of `PeakPerformance` and so it is wholly incompatible with data originating from it. Not only is the peak data formatted in a different way from MultiQuant which would merely require a new parsing method to be added, but one of the core advantages of `PeakPerformance` is constituted by its use of Bayesian inference to provide uncertainty quantification for each single peak. In this circumstance, the calculation of TMIDs via normalization of peak areas requires Gaussian error propagation. Since this necessitates changes to several methods of the program, it was decided to create a separate module for the version compliant with `PeakPerformance` data. The general workflow remained untouched, though, thus remaining as it is presented in figure 3.17.
When writing this newer module, the opportunity was taken to apply some lessons from the practical experience of applying the MultiQuant side of the program, e.g. the class structure was removed as its instantiation required arguments which were not always strictly necessary for using the desired methods. This complication was avoided by merely defining functions outside of the class framework.
The data input for this program is the "area_summary.xlsx" file from `PeakPerformance` which only contains the peak area instead of all modeled peak parameters. Since the parsing function is trivial and many of the changes were merely intended to recreate the same operations as before for data from a different source, the focus will be placed on the implementation of error propagation and thus on the functions `calculate_tmids_peak_performance()`, `calculate_mean_stdev()`, and `plot_tmid_transient()`. To obtain the respective TMID for a given fragment, the error propagation is conducted in two consecutive steps, just as the TMID calculation itself is. First, a sum of all areas pertaining to a TMID is computed and then the individual areas are divided by that sum. Accordingly, the error of the area sum of n peaks amounts to

$$u_{\text{area sum}} = \sqrt{\sum_{i=1}^{n} u_i^2} \tag{3.17}$$

and the total error of the resulting relative abundance pertaining to each peak i to

$$u_{\text{relative abundance}} = \sqrt{(\frac{1}{\text{area sum}}\sigma_i)^2 + (\frac{\text{area}_i}{\text{area sum}^2}\sigma_{\text{area sum}})^2} \tag{3.18}$$

For visualization purposes, however, the mean and standard deviation of biological replicates were exhibited, meaning the mention of uncertainties resulting from error propagation is restricted to the optionally stored, intermediary Excel file "concatenated_df.xlsx".
The code repository of this program contains a results file from `PeakPerformance` with an example notebook. Furthermore, an Excel file with manually calculated error propagation for some peak areas from the exemplary data set is provided which can be compared to the values determined by the software for testing purposes.

3.5 Case study I: Model-based estimation of metabolic pool sizes

This chapter is based on the Master thesis project of Tobias Latour (TL) supervised by JN. All figures were originally created by TL (if not stated otherwise) and adapted in style and partially edited in content by JN. The text as well as table 3.3 and figure 3.18 were authored by JN for this dissertation.

As discussed in detail earlier (see 3.1), the developed automated hot isopropanol quenching method could not be utilized for accurate metabolic pool size measurements. Since this data is not only relevant for rational strain engineering to identify bottlenecks in the form of rate-limiting steps but also for conducting INST ^{13}C-MFA, this was considered a serious drawback of the method. Therefore, it was investigated whether labeling data of free amino acids generated with the automated, miniaturized and parallelized ILE setup could be used to estimate pool sizes when complemented with an appropriate modelling approach.
Concretely, the idea was to build reduced metabolic network models of *C. glutamicum* WT by constricting the system boundaries and extensively lumping pathways within the central carbon metabolism. When coupling these model structures with an INST ILE where the cultivation is started on unlabeled D-glucose before pulsing with 100 % U^{13}C D-glucose, only a minimal amount of mass balances pertaining to the differently labeled metabolite species need to be taken into account. In fact, when concentrating on the upper part of glycolysis, it can be assumed – and earlier ILEs have confirmed as much (figure 3.6) – that only the fully labeled and the unlabeled species of a given metabolite will be observed. Accordingly, only one ODE per metabolite i is sufficient as the sum of the relative abundances of the unlabeled fraction $x_{i,0}$ and the fully labeled fraction $x_{i,1}$ amount to 1 meaning one of the two fractions and the total pool size constitute the only degrees of freedom.
With such a model design in place, there remains the problem that the experimentally measured dynamic INST labeling data is essentially a hybrid created by the flux rate and the pool size of a given metabolite, hence the requirement of measuring pool sizes to resolve fluxes (see 1.4.3). However, assuming a metabolic steady-state alleviates the complexity of the problem by asserting constant pool sizes and fluxes. When additionally focusing on free amino acids, which act in many cases as biomass precursors, the fluxes are constricted by the biomass drain reactions and the network stoichiometry in such a way as to enable the estimation of pool sizes based on labeling and backscatter data.
The combination of these simplifications leads to a drastic reduction in the number of parameters and a much smaller scope than even in the core metabolic models preferred in ^{13}C-MFA.

3.5.1 Building small metabolic sub-network models for *C. glutamicum* WT

The simplifications discussed in the previous section already narrowed down the available model scopes considerably since this approach is predicated on the assumption of observing only unlabeled and fully labeled mass traces for the amino acids in question which does not hold in many pathways.

Hence, two model variants with different levels of complexity focused on the upper glycolysis were defined (figure 3.18). As it is known that the carbon atoms are shuffled in the PPP due to the enzymatic ping-pong mechanism [193], aromatic amino acids derived from precursors of the PPP could not be included. Similarly, upon inclusion of the TCA cycle, the number and directionality of reactions, especially considering the anaplerotic ones, would increase the model complexity beyond a feasible scope for this approach and undo most of the simplifications.
The first model variant referred to as v0 initially contained just Ser, Gly, and Cys as amino acid pools. Labeling data was recorded only for the former two since Cys could routinely not be detected in biological samples. However, the Gly labeling data showed a delay with respect to the onset of label incorporation of around 1 min which is comparable to the results of the original proof of concept experiment (3.1.3). In accordance with published data on the exometabolome of *C. glutamicum* [19], it was hypothesized that the root cause for this did not lie in the intracellular Gly pool but rather in an extracellular one of considerable size. In the referenced publication, the highest measured extracellular Gly pool over the course of a fermentation amounted to roughly 699 µM compared to the 24 µM of Ser. Early parameter estimation attempts with and without the extracellular Gly pool confirmed that its inclusion led to a significantly improved fit so the model was expanded with this additional pool denominated as $\text{Gly}_{\text{extra}}$.
Aside from the biosynthesis pathways of these amino acids, the biomass formation in dependence of the growth rate was included and the linear upper glycolysis and the PPP were lumped in an artificial pool referred to as EMPUP with reaction v_4 crossing the system boundary toward the lower glycolysis and TCA cycle.
The second model variant v1, then, additionally included the glycolysis up until Pyr and its associated amino acids Ala, Val, and Leu for all of which INST labeling data was measured.
Moving on from the model structure to formulation, the cultivation was characterized as a fed-batch process to describe the pulse with U^{13}C D-glucose appropriately. Accordingly, the macroscopic bioprocess model was defined as

$$\frac{d}{dt}c_X = \mu c_X - \frac{F_S}{V_R}c_X \qquad c_X(t=0) = c_{X_0} \tag{3.19a}$$

$$\frac{d}{dt}V_R = F_S \qquad V_R(t=0) = V_{R,0} \tag{3.19b}$$

$$F_S = \begin{cases} 0 & t < t_{\text{feed}_{\text{start}}} \vee t > t_{\text{feed}_{\text{end}}} \\ \frac{V_{\text{feed}}}{\delta_{t_{\text{feed}}}} & t_{\text{feed}_{\text{start}}} \leq t \leq t_{\text{feed}_{\text{end}}} \end{cases} \tag{3.19c}$$

with the flow rate F_S in L h^{-1}, feed volume V_{feed} in L, and pulse duration $\delta_{t_{\text{feed}}}$ in h. All other parameters and the derivation of the general equations have been established in the pertaining introductory segment (see 1.4.1).
Since two substrate species were planned for the INST experiment, the concentrations of the unlabeled glucose species c_{S0} provided at the start of the experiment and of its uniformly labeled counterpart c_{S1} supplied with the pulse were specified by

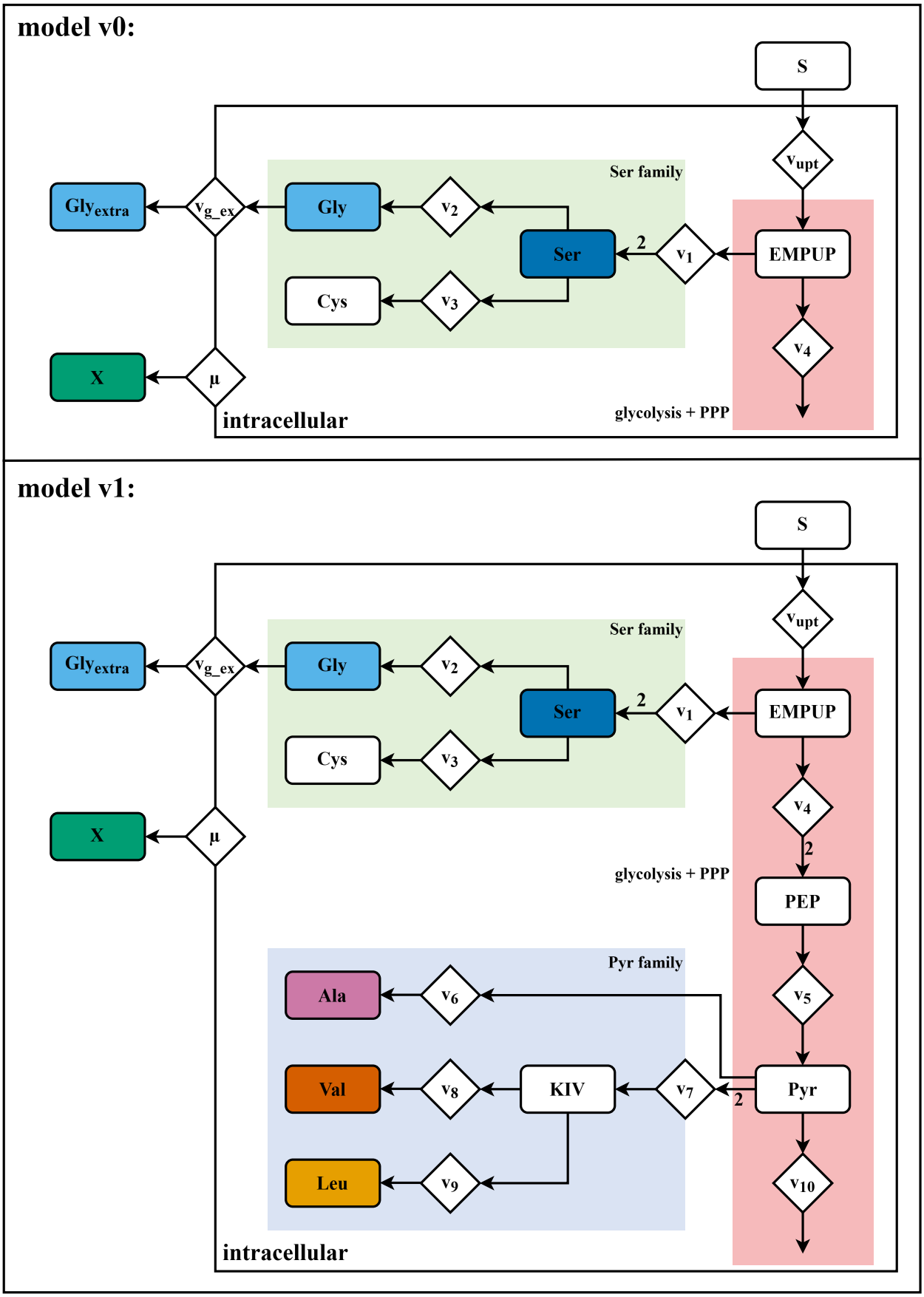


Figure 3.18: Two reduced metabolic network models for *C. glutamicum* WT of different complexity. In both models the upper glycolysis and the PPP are lumped into the artificial pool EMPUP. Model v0 contains only the Ser family of amino acids featuring INST labeling data for Ser and Gly. Model v1 places the system boundary between Pyr and AcCoA, thus including the Pyr family of amino acids and INST labeling data for Ala, Val, and Leu. Biomass drains were omitted here to preserve visual clarity.

$$\frac{d}{dt}c_{S0} = - v_{\mathrm{upt},S}c_X V_{\mathrm{cell}}x_{S0} - \frac{F_S}{V_R}c_{S0} \qquad c_{S0}(t=0) = c_{S0_0} \tag{3.20a}$$

$$\frac{d}{dt}c_{S1} = - v_{\mathrm{upt},S}c_X V_{\mathrm{cell}}x_{S1} - \frac{F_S}{V_R}c_{S1} + \frac{F_S}{V_R}c_{S1_{\mathrm{feed}}} \qquad c_{S1}(t=0) = 0 \tag{3.20b}$$

$$c_S = c_{S0} + c_{S1} \tag{3.20c}$$

The total substrate concentration c_S was accordingly defined as the sum of the concentrations of its two isotopomer species. The extracellular rates excluding the growth rate are notably molar volumetric rates in mmol L^{-1} h^{-1}.
For the relative labeling fractions x_{S0} and x_{S1}, it holds that

$$1 = x_{S1} + x_{S0}, \quad x_{S0}(t=0) = 1 \tag{3.21a}$$

$$x_{S0} = \frac{c_{S0}}{c_S} \tag{3.21b}$$

Crucially, compared to conventional ILEs, the exact substrate mixture is unknown and thus declared a model parameter included in the estimation since it is not feasible to take a sample simultaneously to each of the pulses with the present experimental setup.
To be able to realize DAEs like defining the total substrate concentration c_S as the sum of the concentrations of its present isotopomers, the models were implemented in the estim8 software relying on the Modelica modelling language.
As the growth rate and substrate uptake rate were modelled based on the Monod kinetics as established in the introduction (1.8), the final remaining parts of the bioprocess model were constituted by the glycine export reaction which was coupled to the substrate uptake rate via

$$v_{\mathrm{Gly}_{\mathrm{exp}}}(t) = v_{\mathrm{upt},S}(t)k_{\mathrm{scale}_{\mathrm{Gly,exp}}} \tag{3.22}$$

and the extracellular Gly pool as described by

$$\frac{d}{dt}c_{\mathrm{Gly}_{\mathrm{extra}}} = v_{\mathrm{Gly}_{\mathrm{exp}}}c_X V_{\mathrm{cell}} - \frac{F_S}{V_R}c_{\mathrm{Gly}_{\mathrm{extra}}} \tag{3.23}$$

The microscopic part of the models featured two kinds of mass balances. The first type constraining the flux rates stoichiometrically amounted to a simple mass balance around a total pool size given by for example

$$\frac{d}{dt}c_{\mathrm{Ser}} = 0 = 2v_1 - (v_2 + v_3) - \mu\frac{Y_{\mathrm{Ser},X} + Y_{\mathrm{Trp},X}}{V_{\mathrm{cell}}} - \mu c_{\mathrm{Ser}} \tag{3.24}$$

when based on Ser. Terms representing biomass drain and dilution by growth were included, yet the cellular maintenance term was not. In this particular case, the biomass yield coefficients for Ser and L-tryptophan (Trp) were summed since Ser acts as a co-substrate during Trp biosynthesis. For other amino acids, only their own biomass yield coefficient might be present, instead. Due to

the metabolic steady-state assumption, the pool size c_{Ser} is constant, i.e. $\frac{d}{dt}c_{\mathrm{Ser}} = 0$.
In contrast, the second type refers to only the fully labeled pool fraction of Ser and is defined as

$$\frac{d}{dt}x_{\mathrm{Ser},1} = \frac{2v_1 x_{\mathrm{EMPUP},1} - (v_2 + v_3)x_{\mathrm{Ser},1} - \mu\frac{Y_{\mathrm{Ser},X}+Y_{\mathrm{Trp},X}}{V_{\mathrm{cell}}}x_{\mathrm{Ser},1} - \mu c_{\mathrm{Ser}}x_{\mathrm{Ser},1}}{c_{\mathrm{Ser}}} \tag{3.25}$$

As these exemplary equations were merely meant to outline the principles behind the formulation, the full model definitions are exhibited in the appendix (listings A2 and A3). One slight exception, however, is caused by the existence of the extracellular Gly pool and by the fact that the labeling data pertains to the union of intra- and extracellular pools. To fit the measurements, a total Gly pool had to be defined according to

$$c_{\mathrm{Gly}_{\mathrm{total}}} = c_{\mathrm{Gly}_{\mathrm{intra}}}c_X V_{\mathrm{cell}} + c_{\mathrm{Gly}_{\mathrm{extra}}} \tag{3.26a}$$

$$\frac{d}{dt}c_{\mathrm{Gly}_{\mathrm{extra}}} = \frac{x_{\mathrm{Gly1}_{\mathrm{intra}}}c_{\mathrm{Gly}_{\mathrm{intra}}}c_X V_{\mathrm{cell}} + x_{\mathrm{Gly1}_{\mathrm{extra}}}c_{\mathrm{Gly}_{\mathrm{extra}}}}{c_{\mathrm{Gly}_{\mathrm{total}}}} \tag{3.26b}$$

Upon comparison with a metabolic model encompassing the central carbon metabolism with 92 reactions and 51 pools [57], it is noteworthy that the presented sub-networks merely feature 11 reactions and 7 pools for model v0 and 22 reactions and 13 pools for model v1. Here, the count of reactions includes biomass drain reactions which are products of the fixed yield constants and the growth rate, meaning the number of reaction parameters and the associated degree of freedom is lower than it may seem.
Before presenting the experimental data generated for pool size estimation, it is worth considering some additional and foundational assumptions inherent to this approach which were neither among the chief design choices established in 3.5 nor previously mentioned otherwise.
Firstly, fluxes are assumed to be unidirectional, i.e. irreversible, under glycolytic conditions. Secondly, protein turnover is disregarded. Thirdly, as the biomass drain terms depend on the biomass yield coefficients adopted from [48] which are assumed to be constant and are expressly not included in parameter estimations, there is a strong reliance on the accuracy of and a high sensitivity towards these values. Fourthly, both the wells and the intracellular environments are considered ideally mixed. Fifthly, V_{cell} is assumed to be at a constant value of 1.93 $\mathrm{mL_{cell}\ g_X^{-1}}$ [49]. Finally, except for Gly, the exometabolome of amino acids is deemed negligible. In case of evidence to the contrary for a given amino acid, this can naturally be amended.

3.5.2 INST ILE to generate data for modelling

Having already formulated the model variants, the present INST ILE was conducted as an improved version of the proof of concept experiment (3.1.3) including additional INST time points and a minimal CGXII medium variant to ameliorate LC-MS/MS analyses. The INST ILE was conducted across 39 wells in parallel translating to 15 time points sampled in biological triplicates after administrating the pulse with U^{13}C D-glucose during the exponential growth phase. The backscatter data was converted to biomass via calibr8 [157, 168, 169] and merged into three replicates

based on the rows of the FlowerPlate (figure 3.19). Accordingly, each of the hereinafter referred to three replicates encompass data from 13 wells.
Compared to previous experiments, the growth deviation within the 39 replicates was slightly more pronounced, thus leading to larger differences in the input labeling mixture of each well. This offset propagated into comparatively larger standard deviations of the labeling time courses (figure 3.20) relative to the proof of concept experiment (figure 3.6) but even so they were certainly of an acceptable magnitude.

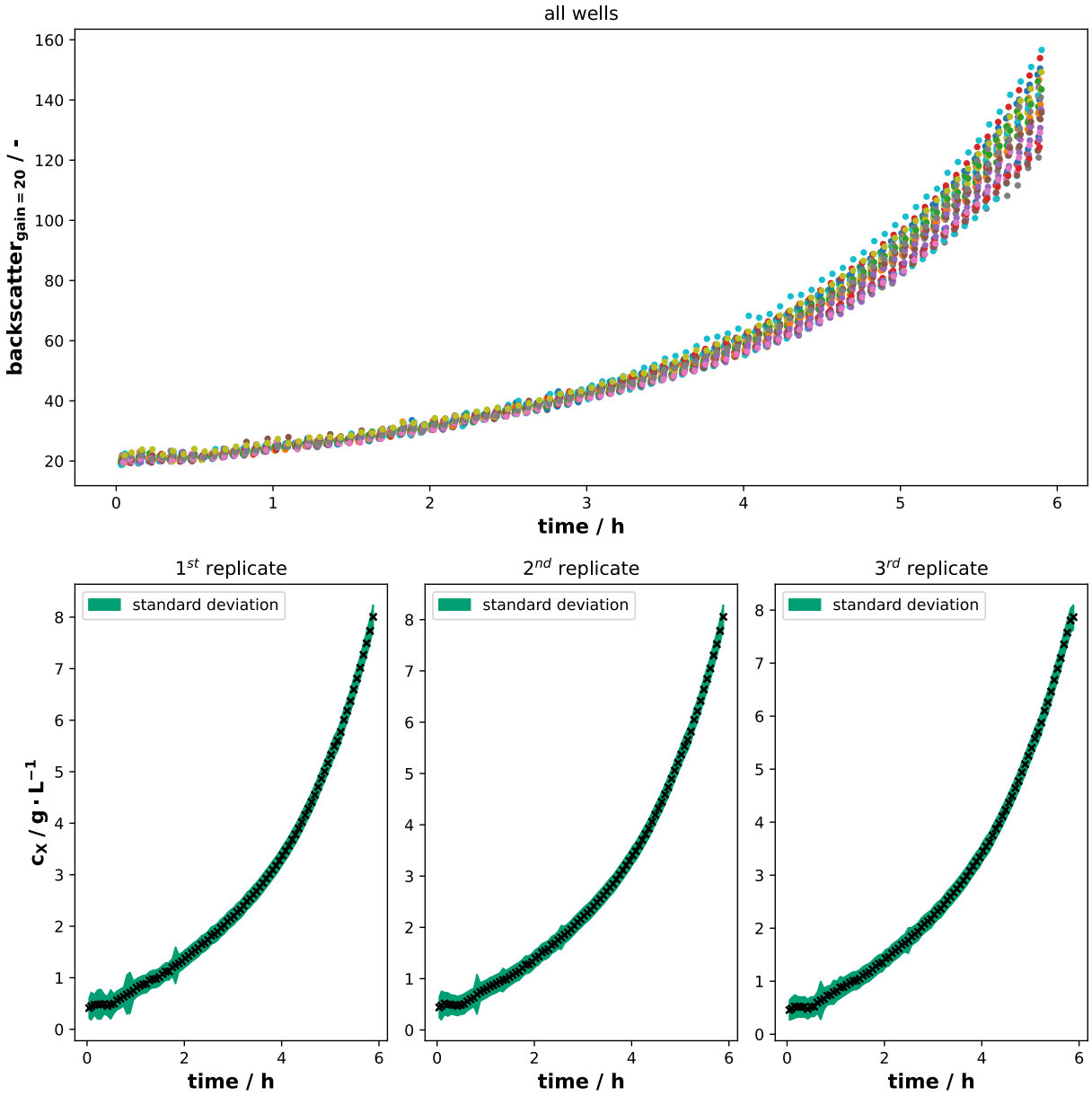


Figure 3.19: Backscatter (top row) and biomass (bottom row) data of the INST ILE conducted to generate experimental data for model-based pool size estimation. Biomass conversion was performed via a nonlinear calibration model.

It can be observed that the assumption of modelling only the unlabeled and fully labeled mass traces is justified by the experimental data of Ser and Ala, in particular. With regards to Gly, merely

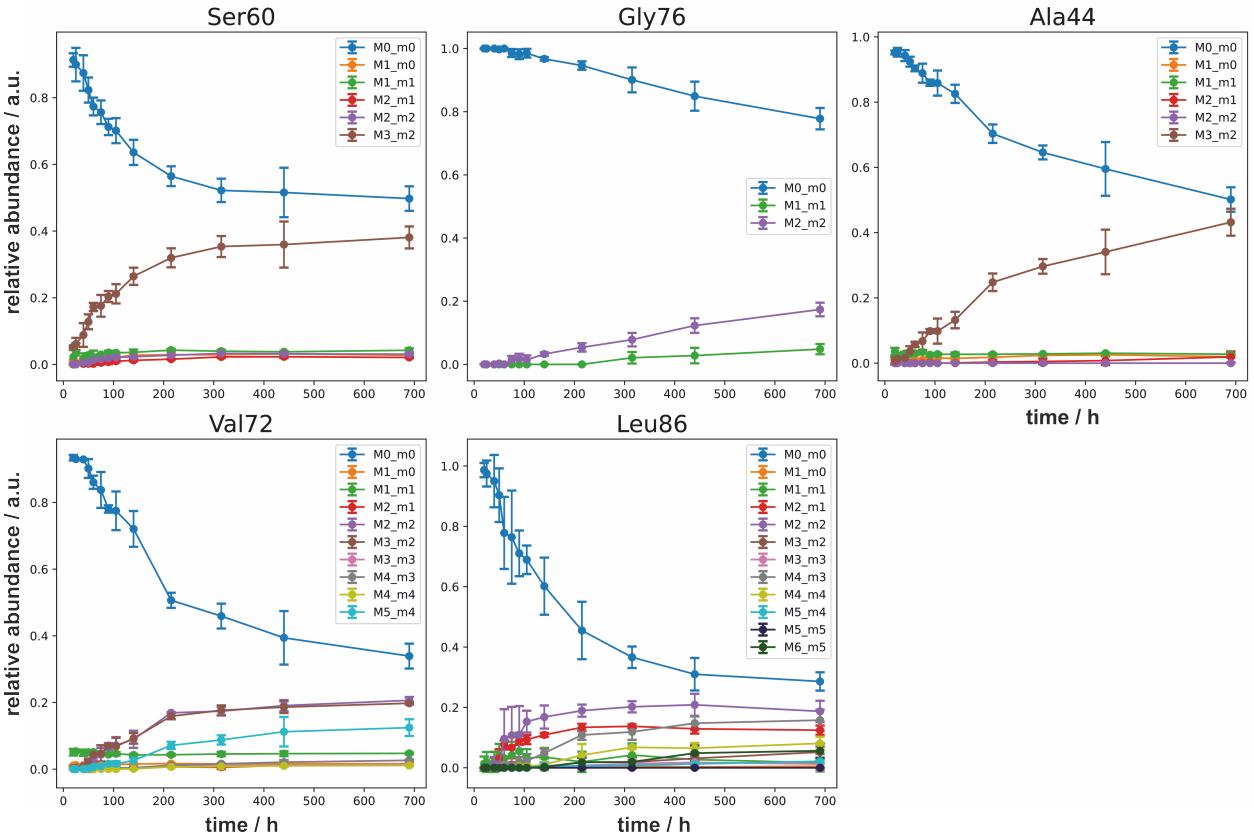


Figure 3.20: Portrayal of the relevant INST ILE data of amino acids located within the system boundaries of the small sub-network models v0 and/or v1.

a slight increase in the M1_m1 mass trace towards the later stages of the ILE was detected so the assumption was insisted on here, as well. However, for Val and Leu this was clearly untenable. Val's data exhibited two additional mass traces M2_m2 and M3_m2 and regarding Leu, the mass traces M2_m1, M2_m2, and M4_m3 were significantly increased. Accordingly, this had to be represented in model v1 in order to faithfully describe the experimental data.

For Val, the case was straightforward since all 5 carbon atoms of its precursor α-ketoisovalerate (KIV) originate from 2 Pyr which in turn can be assumed to be only fully or unlabeled based on the Ala measurements. Since the observed fragment of Val with a m/z ratio of 72 possesses one fewer carbon atom than its precursor ion [194], precisely the two mass traces M2_m2 and M3_m2 would theoretically arise by combination of the two Pyr species. As this is reflected in the Val measurements, the assumption regarding Pyr holds true and the fully labeled Val fraction can be expressed according to the exemplary formula 3.25.

$$\begin{aligned} &\mathrm{Pyr}_{\#\mathrm{abc}} + \mathrm{Pyr}_{\#\mathrm{def}} && \rightarrow \mathrm{KIV}_{\#\mathrm{abefc}} + \mathrm{CO2}_{\#\mathrm{d}} \\ &\mathrm{KIV}_{\#\mathrm{abcde}} && \rightarrow \mathrm{Val}_{\#\mathrm{abcde}} \\ &\mathrm{KIV}_{\#\mathrm{abcde}} + \mathrm{ACCOA}_{\#\mathrm{fg}} && \rightarrow \mathrm{Leu}_{\#\mathrm{fgbcde}} + \mathrm{CO2}_{\#\mathrm{a}} \end{aligned}$$

In contrast, the first two of Leu's carbon atoms originate from AcCoA which is located outside

the system boundary directly downstream from Pyr as it could not be detected in the LC-MS/MS analysis. Here, the fragmentation of Leu during tandem-MS analysis led to the collision-induced dissociation of its first carbon atom [194], hence the fully labeled mass trace is M6_m5. To nevertheless insist on circumventing the introduction of an extensive atom transition model, the cumomer Leu_{XX1111} was included in model v1 instead of M6_m5. As can been seen in the matrix in table 3.3, said cumomer produced by any AcCoA species and fully labeled KIV comprises the mass traces M4_m4, M5_m4, M5_m5, and M6_m5. Of these, only M4_m4 and M6_m5 were detected at any significant level. Interestingly, the matrix also disproves the occurrence of partially labeled AcCoA as none of the detected mass traces for Leu (M2_m1, M2_m2, M4_m3, M4_m4, and M6_m5) would arise from it.

Table 3.3: Matrix mapping isotopomers of KIV and AcCoA to the resulting mass traces of Leu's product ion with a m/z ratio of 86.

KIV \ AcCoA	○○	●●	○●	●○
●●○○	M2_m2	M4_m3	M3_m3	M3_m2
○○●●	M2_m2	M4_m3	M3_m3	M3_m2
●●●●	M4_m4	M6_m5	M5_m5	M5_m4
○○○○	M0_m0	M2_m1	M1_m1	M1_m0

3.5.3 Pool size estimation with the estim8 tool

After aligning the model formulation with the insights gained by interpretation of the experimental labeling data, parameter estimations for both model versions were conducted with the estim8 software [106]. As described earlier, the biomass and labeling data were subdivided into three replicates so for parameter estimation a pooled approach to replicate handling was chosen.
The unknown parameters (table 3.4) maximum growth rate $\mu_{\max}$, maximum substrate uptake rate $v_{\mathrm{upt},S_{\max}}$, and the relative Gly export factor $k_{\mathrm{scale}_{\mathrm{Gly,exp}}}$ were assumed to be strain-specific, global parameters and thus should not be subject to change in each well. In contrast, the starting biomass and pool sizes were defined as local parameters allowing for pipetting errors and imperfect mixture of inoculated medium with regards to the former and batch-specific growth heterogeneity with regards to the latter.
Upon determination of an optimized parameter set, a forward simulation was conducted with the pertaining maximum likelihood estimators yielding the resulting trajectories compared to experimental data points presented in figures 3.21 and 3.22.
Both model fits clearly reproduced the data well. The microbial growth up until the moment of the pulse after ca. 5.9 h of cultivation – clearly demarcated by the sharp decline in the biomass signal – was in agreement with the chosen bioprocess model. Fitting the much more sparsely distributed labeling data was more challenging but ultimately successful following the introduction of the extracellular Gly pool which allowed simulating the delayed label incorporation. Even with the extrapolation due to the increased time frame of the simulation compared to the measurements, Gly did not reach its isotopic steady-state. Moreover, as in particular the final two data points were deviating between replicates, the simulated end point of the first replicate lay at a lower value than

Table 3.4: Unknown global and local parameter mappings of the models v0 and v1. Resulting degrees of freedom upon inclusion of three replicate data sets are listed at the bottom.

	model v0		**model v1**	
parameter	global	local	global	local
c_{X_0}		✓		✓
$\mu_{\max}$	✓		✓	
$v_{\mathrm{upt},S_{\max}}$	✓		✓	
$k_{\mathrm{scale}_{\mathrm{Gly,exp}}}$	✓		✓	
c_{EMPUP}		✓		✓
c_{PEP}				✓
c_{Pyr}				✓
c_{KIV}				✓
c_{Ser}		✓		✓
c_{Cys}		✓		✓
$c_{\mathrm{Gly}_{\mathrm{intra}}}$		✓		✓
c_{Ala}				✓
c_{Leu}				✓
c_{Val}				✓
DOFs	18		36	

the others although they could still converge toward the same equilibrium.

The trajectory of Ser's fully labeled mass trace was fit well and notably, all replicates reached approximately the same steady-state at about 38 %. The labeling data pertaining to the final three time points was more replicate-dependent with points scattered around the 40 % mark so such a uniform outcome is not necessarily expected.

The results of model v1 with regards to the shared parameters was quite similar to those of model v0. The sole difference was that the steady-state of Ser was estimated to be close to 40 % indicating the influence of the additional data. Aside from two outliers as the final points of replicates 2 and 3, Ala's time course was similar to Ser's and their steady-state virtually identical. When using a binary input labeling mixture, amino acids such as Ser and Ala with an almost linear connection to the substrate give an indication of the mixture's ratio. For Val and Leu, the data was slightly more noisy but especially the fit of the trajectory remained convincing.

Since both models gave rise to faithful optimizations, it stands to reason to compare the resulting parameter sets of maximum likelihood estimators (figure 3.23). The biomass parameters, specifically the starting biomass c_{X_0}, the maximum growth rate $\mu_{\max}$, and the Gly export factor $k_{\mathrm{scale}_{\mathrm{Gly,exp}}}$, were estimated virtually identical across the two model variants with the notable exception of the maximum substrate uptake rate $v_{\mathrm{upt},S_{\max}}$. Here, the models exhibit a discrepancy of about 100 mmol L_{cell}^{-1} h^{-1} amounting to values of 1884.7 mmol L_{cell}^{-1} h^{-1} and 1996.3 mmol L_{cell}^{-1} h^{-1} or 3.64 mmol g_X^{-1} h^{-1} and 3.85 mmol g_X^{-1} h^{-1} for models v0 and v1, respectively.

In literature, values of 4.42 mmol g_X^{-1} h^{-1} $\pm$ 0.54 mmol g_X^{-1} h^{-1} or 2290.2 mmol L_{cell}^{-1} h^{-1} $\pm$ 279.8 mmol L_{cell}^{-1} h^{-1} have been reported [195] so the estimated values must be considered lower than expected. However, the result of model v1 is located barely outside of the standard deviation of the literature value and it seems that the expansion of the model scope and the inclusion of additional data have steered the model closer to the expected value.

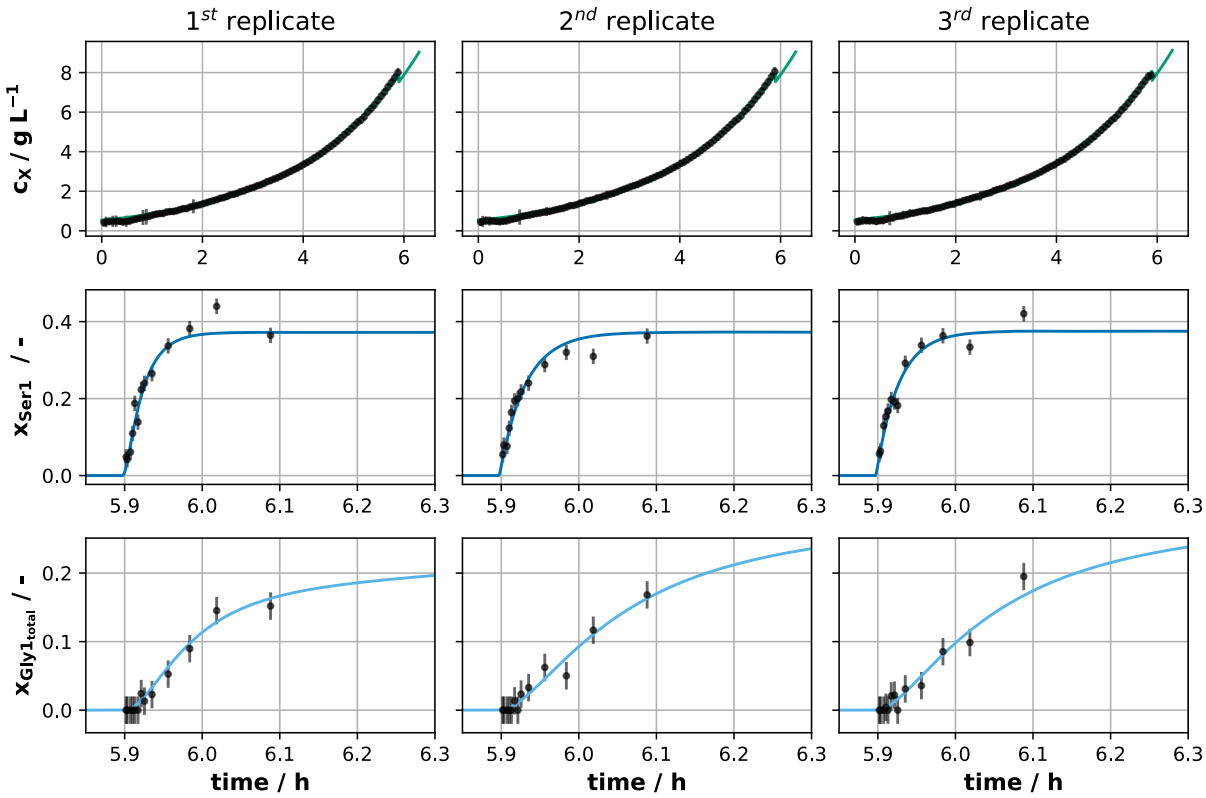


Figure 3.21: Forward simulation of model v0 with the maximum likelihood estimators resulting from the parameter estimation with estim8.

As these parameters were all global, they can – as point estimates, at least – not reflect the replicate variance which turned out particularly large for pools without associated labeling data like PEP, Pyr, and Cys but Ala, too, exhibited a large variance among replicates. The first replicate especially seemed to deviate from the others, featuring much higher EMPUP and PEP pool sizes at the cost of much smaller Pyr, Gly, and Ala pools. For Ser, however, the estimates for all replicates and models were remarkably close and model v1 achieved a fairly low replicate variance with regards to Val and Leu pool sizes.

Using the same models, it would be interesting to see whether successfully obtaining Cys labeling data would influence the result, yet again, since the extension of the model to encompass PPP- or TCA cycle-derived amino acids is not permissible due to the aforementioned model assumptions.

Subsequent to obtaining an optimized set of parameters, traditionally the uncertainty quantification would be the next step as statements about the certainty of a parameter fit and parameter identifiability are paramount while a point estimator alone is of limited value. Despite the many simplifications, though, the number of parameters and complexity of the model led to inflated simulation times in estim8. Due to its open-ended design and reliance on Modelica, it would accommodate even such a presumably unintended combination of bioprocess and ^{13}C-model as was constructed here. Yet, its implemented uncertainty quantification via profile likelihoods or Monte Carlo simulations proved too computationally demanding and inefficient to yield confidence intervals for the presented point estimates.

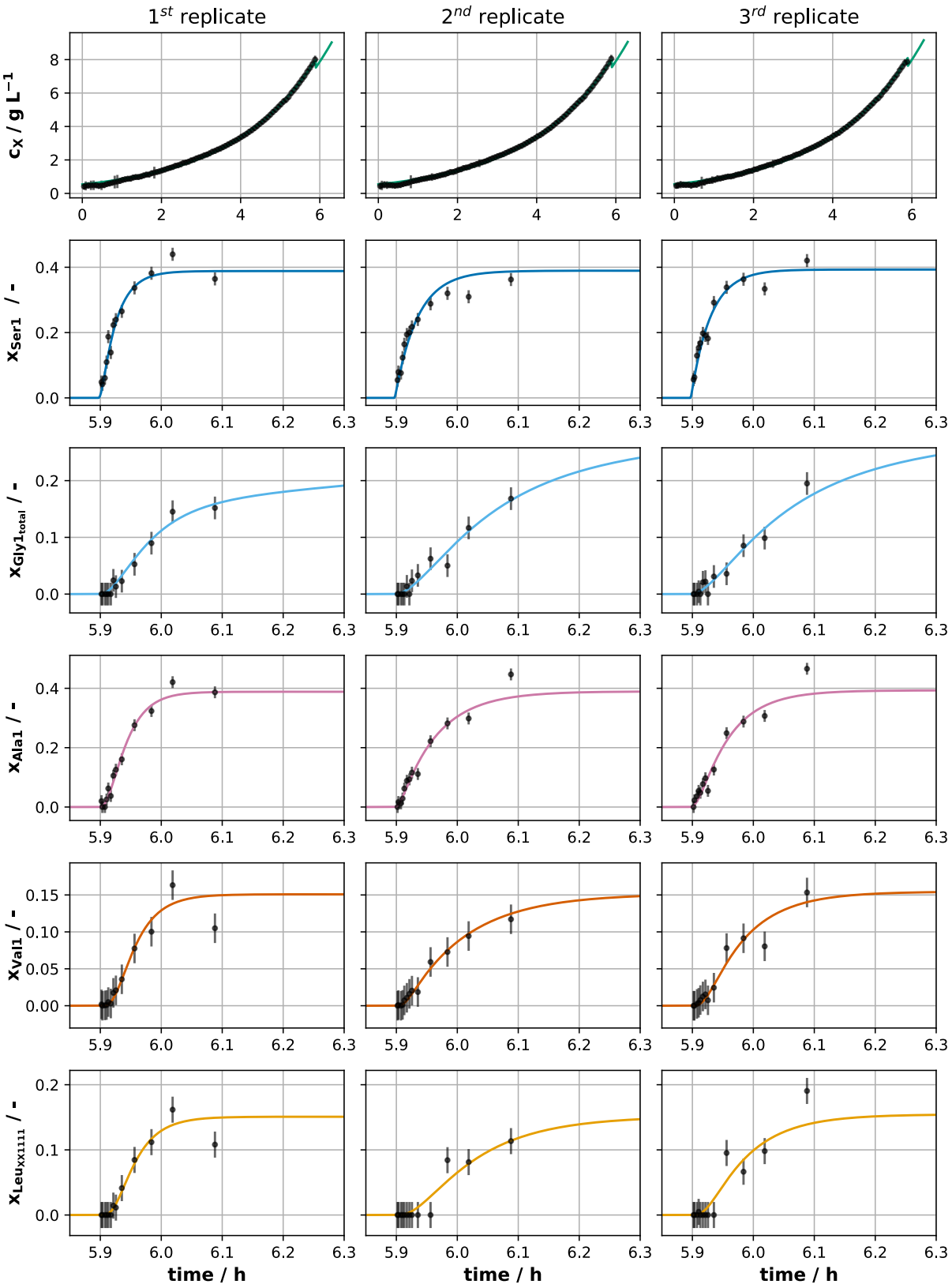


Figure 3.22: Forward simulation of model v1 with the maximum likelihood estimators resulting from the parameter estimation with estim8.

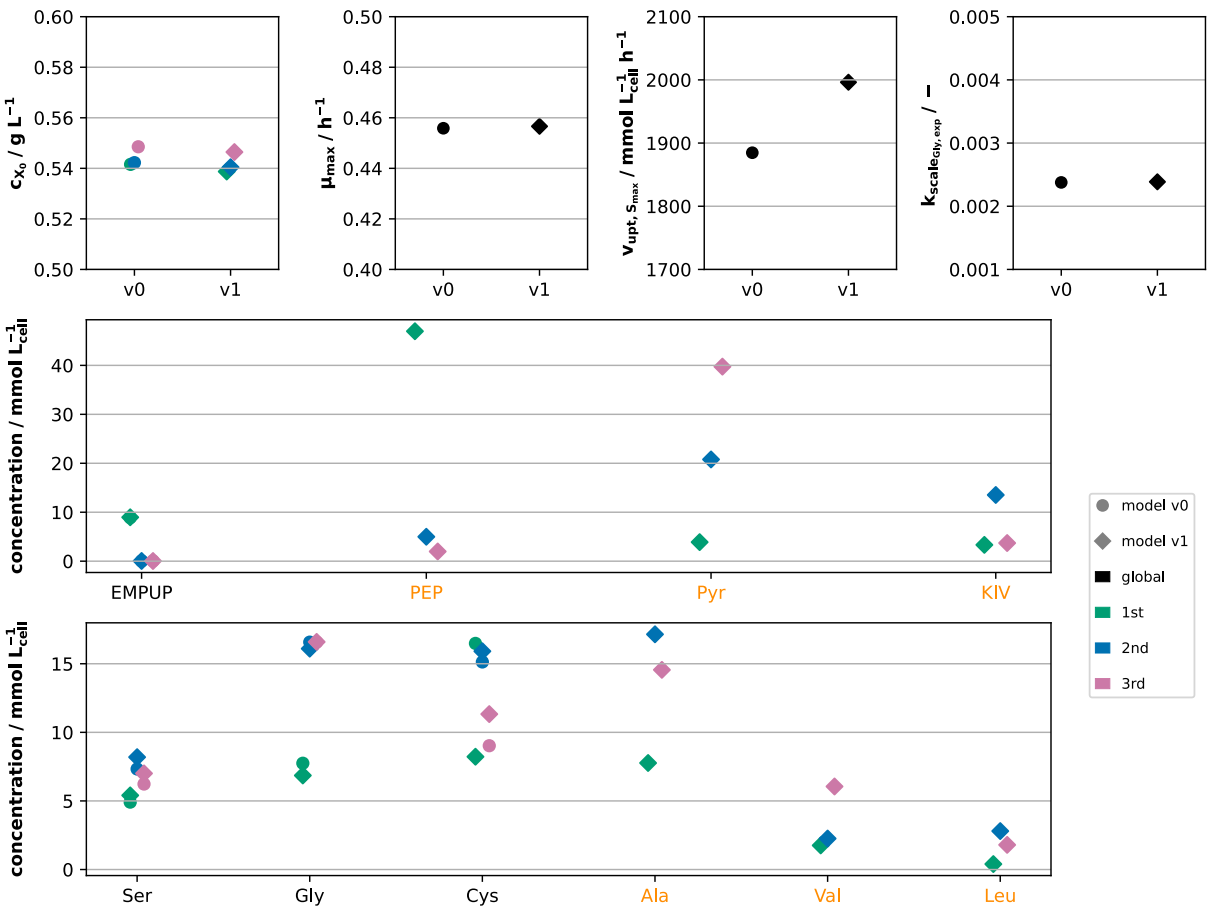


Figure 3.23: Comparison of parameter estimation results of models v0 and v1 as computed with estim8. Parameters which are exclusively contained in v1 are labeled in orange font.

3.5.4 Construction of a highly parallelized data pipeline for uncertainty quantification in bioprocess modelling

The shortcomings of estim8 with respect to the uncertainty quantification of models with a larger number of parameters aside, it is worth returning to one of the core running themes of this dissertation: the implementation of high-throughput workflows. Since the generation of high-throughput labeling data had been realized and each experiment would feature a multiplicity of replicates, a modelling approach with a higher degree of parallelization and independence of one's own local PC was required, anyway. When constructing such a modelling pipeline, some previously utilized frameworks and design approaches were applied once again, namely the usage of the Airflow computation cluster with an emphasis on user-friendliness by lowering the barrier of entry as much as possible.

In a more concrete sense, the goal was to apply Bayesian methods, which have recently been introduced in biology for e.g. nonlinear calibration models [196] and Bayesian model averaging [174], to bioprocess modelling realized by MCMC simulations run in a parallelized manner on the Airflow cluster. To promote a joint usage in conjunction with the estim8 tool and enable the definitions of ODEs and DAEs, the model formulation based on the Modelica language was retained. As

Modelica adheres to the Functional Mockup Interface (FMI) standard [197], this effectively means that the inclusions of models from hundreds of software tools is possible, even more so since the Systems Biology Markup Language (SBML) [198] can be converted into Modelica [199]. As the extensive interaction with the Airflow webserver may be off-putting to new users, it was intended to build a GUI allowing the easy upload of input data guided by clear instructions. The webserver, then, is merely used to start the MCMC simulations with one click. After convergence, the Airflow DAG includes tasks for creating data reports, an inference data object, and plots ripe for further interpretation by the user. In case the number of samples is yet insufficient, the pipeline can simply be restarted to continue the simulation. With regards to replicate handling, the pooled approach used in estim8 was to be established alongside another approach where all replicates are treated as isolated problems without global parameters or a hierarchical structure.
The software components used to realize this pipeline (figure 3.24) are organized either within Airflow or in GitLab repositories for version control. In both cases, they are mounted on the same network drive, thus forming a central location for the necessary data and software. The software written for the pipeline is subdivided into 5 modules supplying functions for the DAG and calling upon third-party Python packages, most importantly the hopsy package for MCMC simulations [129].

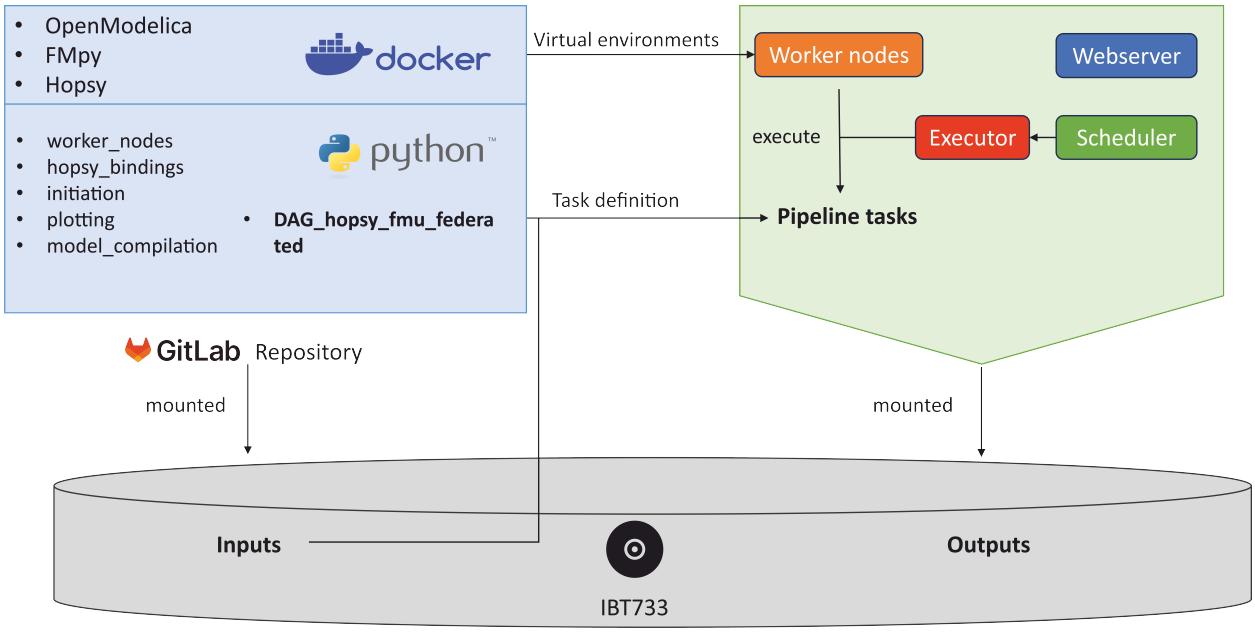


Figure 3.24: Software components of the parallelized MCMC pipeline for bioprocess modelling.

In hopsy, the Python interface for the C++ library HOPS [200], there is a dedicated class for wrapping third-party models which is used to import the Modelica models. The sampling for MCMC simulations, then, is not performed across the entire parameter space but instead the polytope representing the joint solution space of all parameters restricted by parameter bounds. This is realized in hopsy by instantiating the `Problem` class. To deal with issues introduced by the varied magnitude of the parameters (e.g. the growth rate is somewhere between 0 and 1 but the substrate uptake rate may be in the thousands depending on the unit), this polytope is rounded [201] to increase sampling efficiency. Additionally, the initial values for a Markov chain can be based on

point estimators obtained from previous optimizations to save time otherwise spent for the warm-up sampling to reach the higher probability density regions of the polytope.
Every pipeline run is executed in virtual machines for each worker created with Docker [192] and specifically within Python environments created dependent on a Docker image – essentially a snap shot of an environment serving as the basis for its re-creation. This way, a clean and identical environment can be guaranteed for every worker in every process ensuring the correct functionality of the pipeline.
Due to the computational burden of MCMC simulations and the large number of replicate data sets generated with high-throughput workflows, scalability by parallelization across many CPUs is necessary to keep computation times manageable. In the present pipeline, such parallelization is manifested on several levels although the exact structure depends on the type of replicate handling (figure 3.25).

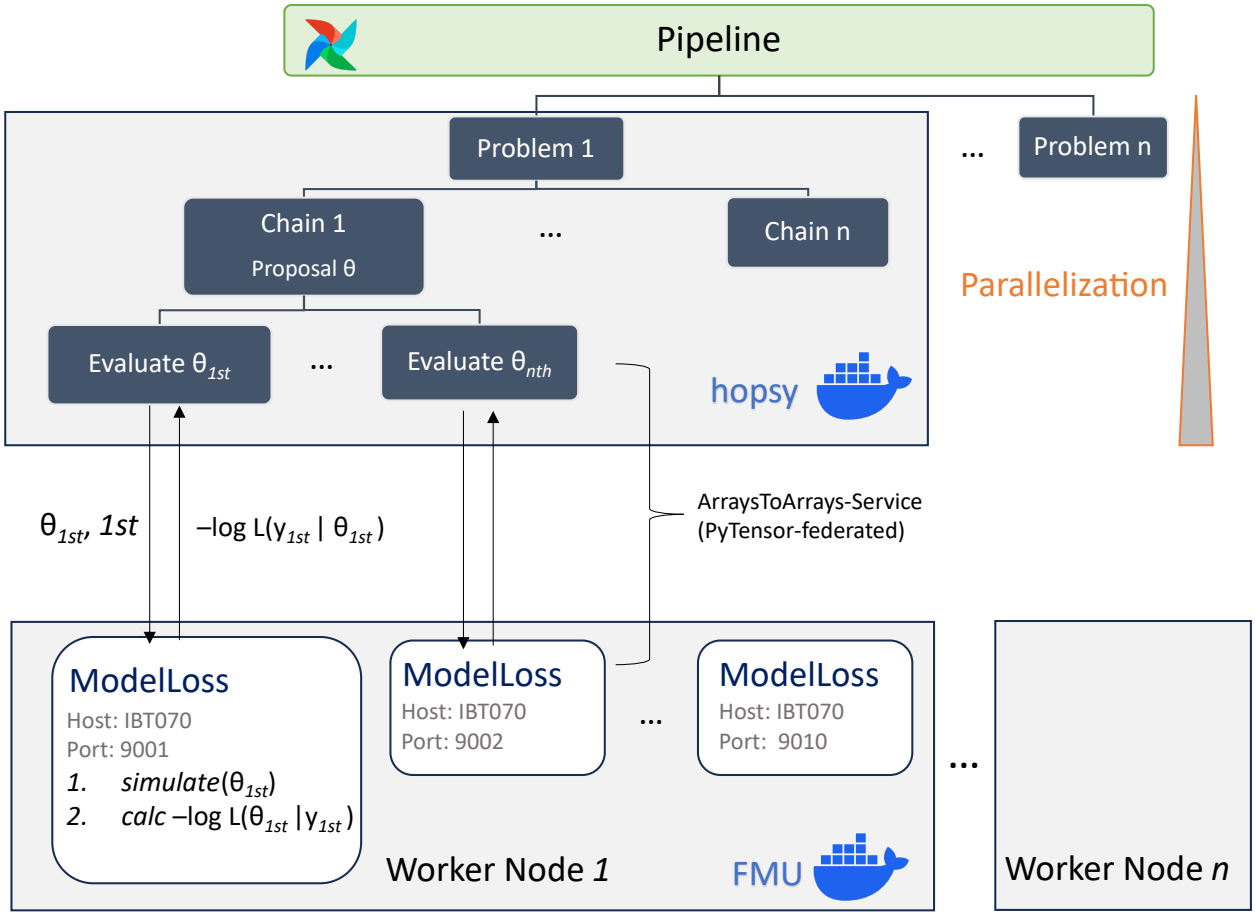


Figure 3.25: Parallelization of the MCMC pipeline for bioprocess modelling by distributing work dependent on the replicate handling approach. When separating replicates, they are each defined as independent hopsy problems and when pooling replicates, they are parallelized on the simulation level.

First off, the aforementioned hopsy problems are initiated in separate Docker containers, i.e. environments, on the higher level. On the lower level, Markov chains are parallelized for each problem. Depending on the type of replicate handling, proposals are either distributed across multiple replicates for the pooled approach or dealt with separately for independent replicates. To convey the scale of the pipeline, a data set with 10 replicates and 4 Markov chains will trigger 40 simultane-

ous evaluations per problem. This level of parallelization is based on using the pytensor-federated [202] package managing the transmission of requests taking the form of a proposal to be evaluated via an `ArraysToArrays` service. After forward simulation and likelihood calculation on part of the `ModelLoss` class, a response is sent containing the results. The usage of different ports for communication and a dynamically created folder structure allows for the creation of an arbitrary number of ModelLoss instances.
Regarding the actual workflow, i.e. the Airflow DAG, the tasks are shown in sequence in figure 3.26. First, a Modelica model is compiled as a Functional Mockup Unit (FMU) and the Docker image for the worker nodes is pulled from GitLab. Then, said worker nodes are launched, tested and an optional step size tuning step may be executed. Finally, the actual MCMC workflow is performed for a given number of samples. Upon successful completion of that workflow, the pipeline is stopped and the results are gathered and stored as described.

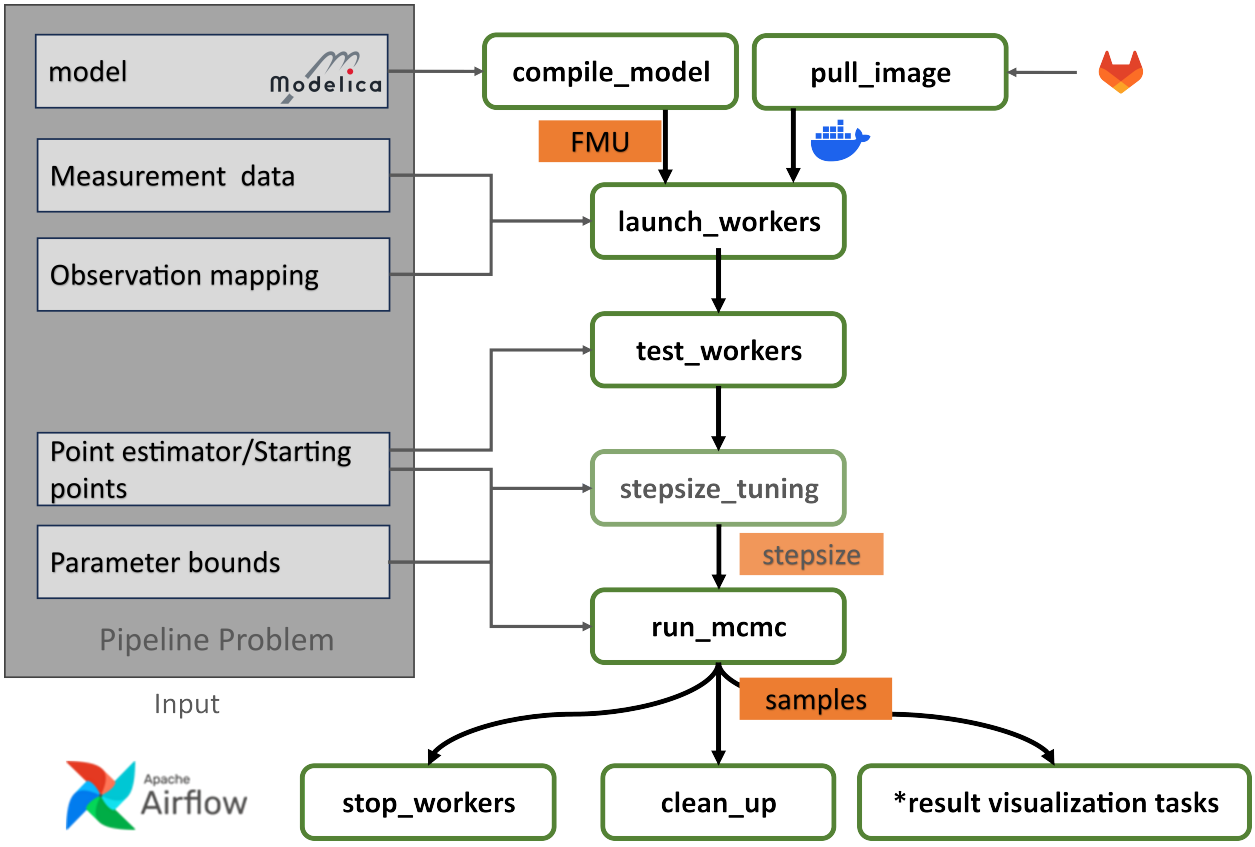


Figure 3.26: Workflow of the MCMC pipeline for bioprocess modelling as established as a DAG on the Apache Airflow computation cluster.

One of the clearest strengths of this pipeline is its dynamic scalability and its connectivity to models from many different sources. Furthermore, when benchmarked against estim8, the simulation time was reduced significantly and especially the use of virtual memory was much more efficient (figure A6). In line with the stated design philosophy, the barrier of entry to accessing MCMC simulations for uncertainty quantification of bioprocess models has been lowered significantly.
With the pipeline in place, the next step was to perform the pool size estimation contrasting the results with estim8 and those found in literature. Aside from this main goal, some related studies

were conducted based on the results of several model approaches listed in table 3.5. By using different settings with regards to replicate handling and the starting time of modelling, the effects of replicate variance and the bioprocess data on the results were investigated and will be presented in the next sections.

Table 3.5: Overview over the model approaches investigated with the data pipeline for uncertainty quantification.

approach	start time t_0	replicate handling	model version
1	t_{pulse}	pooling	v0
2	0 h	pooling	v0
3	0 h	separate	v0
4	0 h	separate	v1

3.5.5 Influence of bioprocess data on parameter identifiability

By comparing the results of the first two model approaches (3.5), the effect of including bioprocess data and indeed the bioprocess model versus only using labeling data and setting the start time of the simulation to the time of the pulse can be investigated.
The latter reduces the simulation time significantly but discards the majority of the data points while introducing new unknown parameters since the biomass and unlabeled concentrations of Glc and Gly at the time of the pulse would be obtained via the bioprocess model.
The biomass at t_{pulse} and $\mu_{\max}$ used priors based on the experimental data which were, however, not updated significantly, i.e. there was no additional information about these parameters in the data.
The extracellular Gly pool size and the remaining concentration of unlabeled Glc at the time of the pulse, on the other hand, were identified, i.e. their posteriors showed clear and relatively sharp peaks. However, the marginal posterior mean of $\mathrm{Gly}_{\mathrm{extra}}$ was below 0.2 mM which is 1 to 2 orders of magnitude below the point estimator for the total Gly pool as determined with estim8 and the remaining Glc concentration varied significantly between the second replicate and the remaining two. The latter results may be explained by the lower fraction of fully labeled Ser in the isotopic steady-state of the second replicate.
As proven by the posterior predictive check of approach 1 (figure 3.27), the labeling data was fit well but the posterior distribution of the Gly pool size was quite broad ranging from less than 5 mM to the upper boundary of 25 mM with all replicates showing the tendency to cross the upper boundary if it were possible. The Ser pool size, though, was identifiable resulting in a marginal posterior distribution with a 95 % highest density interval from 5.2 mM to 8.4 mM (figure 3.28). Hence, it has to follow that the network stoichiometry, the restriction of flux rates by biomass drains, and the labeling data in conjunction sufficed to determine this particular pool size. For Cys, the posterior was approaching uniformity, thus implying unidentifiability which was expected due to the lack of associated labeling data.

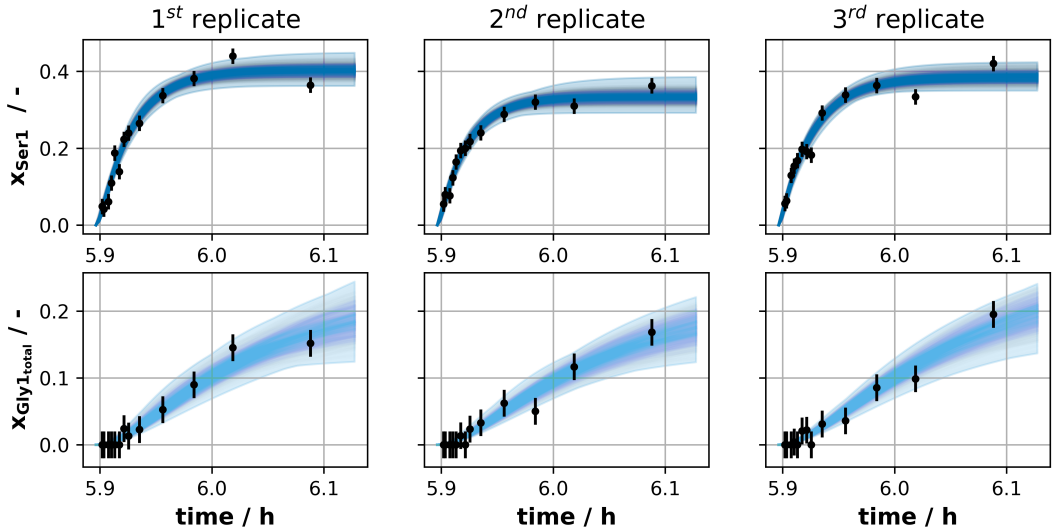


Figure 3.27: Posterior predictive checks computed by performing forward simulations of 1000 random MCMC samples acquired for approach 1.

The posterior predictive check of approach 2 (figure 3.29) was quite comparable to that of approach 1, merely the highest density interval of Ser's isotopic steady-state was narrower and its value was virtually identical among replicates.

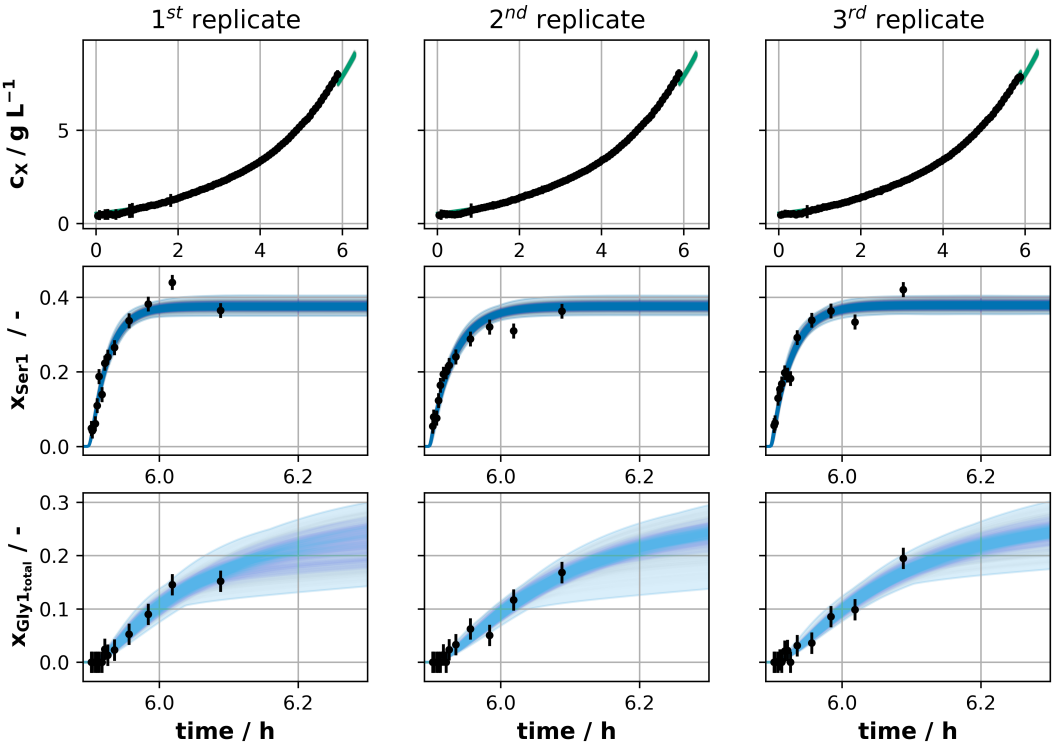


Figure 3.29: Posterior predictive checks computed by performing forward simulations of 1000 random MCMC samples acquired for approach 2.

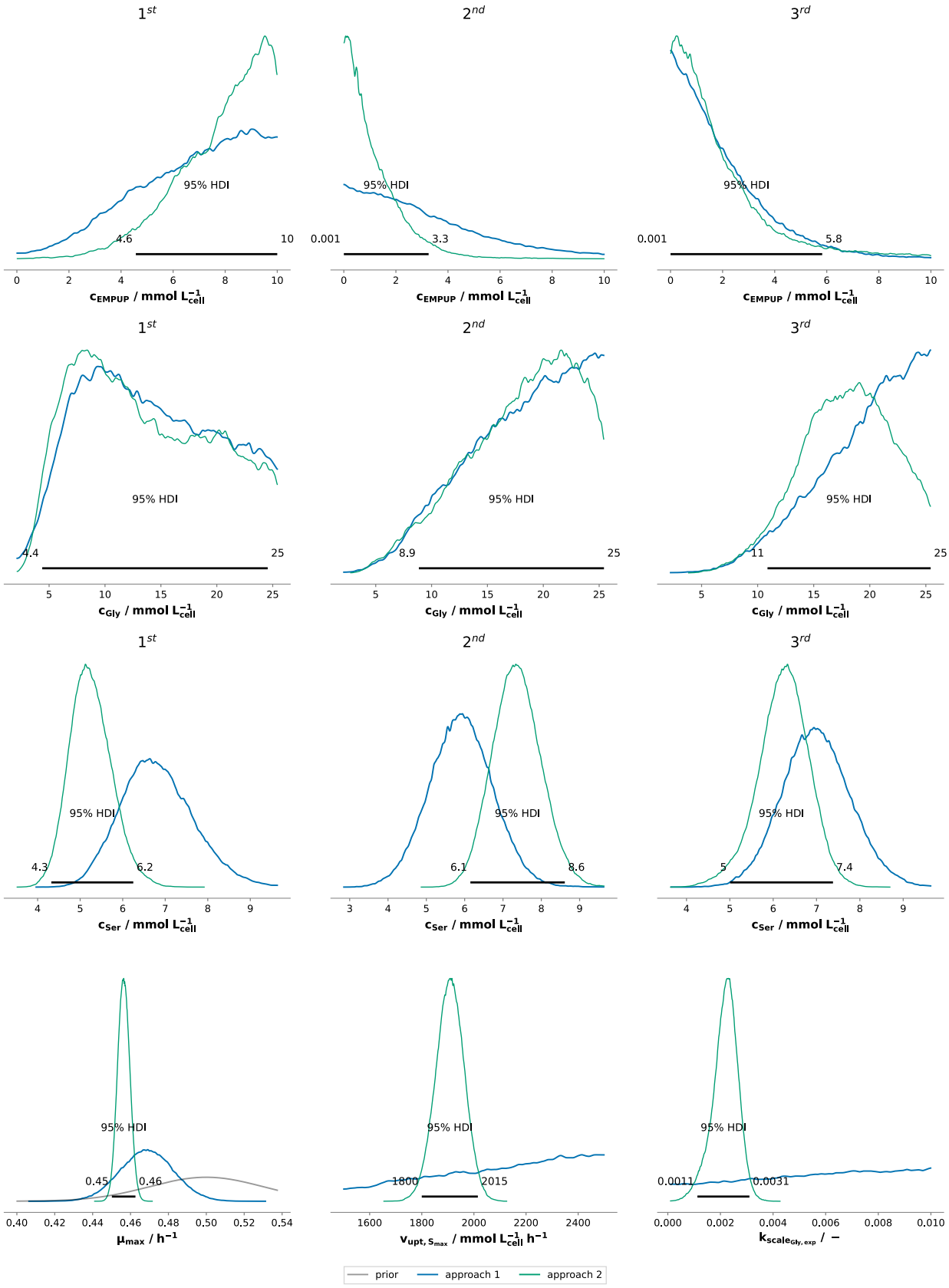


Figure 3.28: Marginal posterior distributions of pool sizes and growth-related parameters from approaches 1 and 2 to investigate the influence of including bioprocess data.

The latter observation can be directly linked to differences in substrate input labeling (figure 3.30) which is estimated as much more congruent among replicates upon inclusion of the bioprocess model. Quantitatively speaking, the means of the input labeling distributions had standard deviations of 0.029 without and 0.0017 with the bioprocess data.
The narrower HDI of Ser, on the other hand, must follow from the bioprocess data as the estimation of the bioprocess led to a much sharper determination of particularly $\mu_{\max}$ (and therefore by extension the biomass drain reactions), $v_{\mathrm{upt},S_{\max}}$, and $k_{\mathrm{scale}_{\mathrm{Gly,exp}}}$ (figure 3.28). As an aside, the estimation of $v_{\mathrm{upt},S_{\max}}$ also corroborates the earlier estim8 results with values approaching 2000 mmol L_{cell}^{-1} h^{-1}. The effect on pool size estimation, then, is generally the determination of sharper posteriors with narrower 95 % HDIs. For Ser, the regions of highest density were additionally shifted towards the upper or lower limits – depending on the replicate – of the marginal posteriors from the first approach. While the artificial lumped pool of EMPUP became more precisely characterized, there was an interesting tendency in both approaches that the first replicate would estimate a large pool size for EMPUP and a relatively small one for Gly and Ser while the second and third replicates exhibited the opposite behaviour. When checking the underlying flux values, $v_{\mathrm{upt},S}$ is basically identical among replicates but reactions v_1 and v_2 leading from EMPUP through Ser to Gly are significantly lower in value for the first replicate.
In summation, the inclusion of the bioprocess model was proven to be crucial as prior knowledge about related parameters was not updated without it and estimations of pool sizes were consequently less certain.

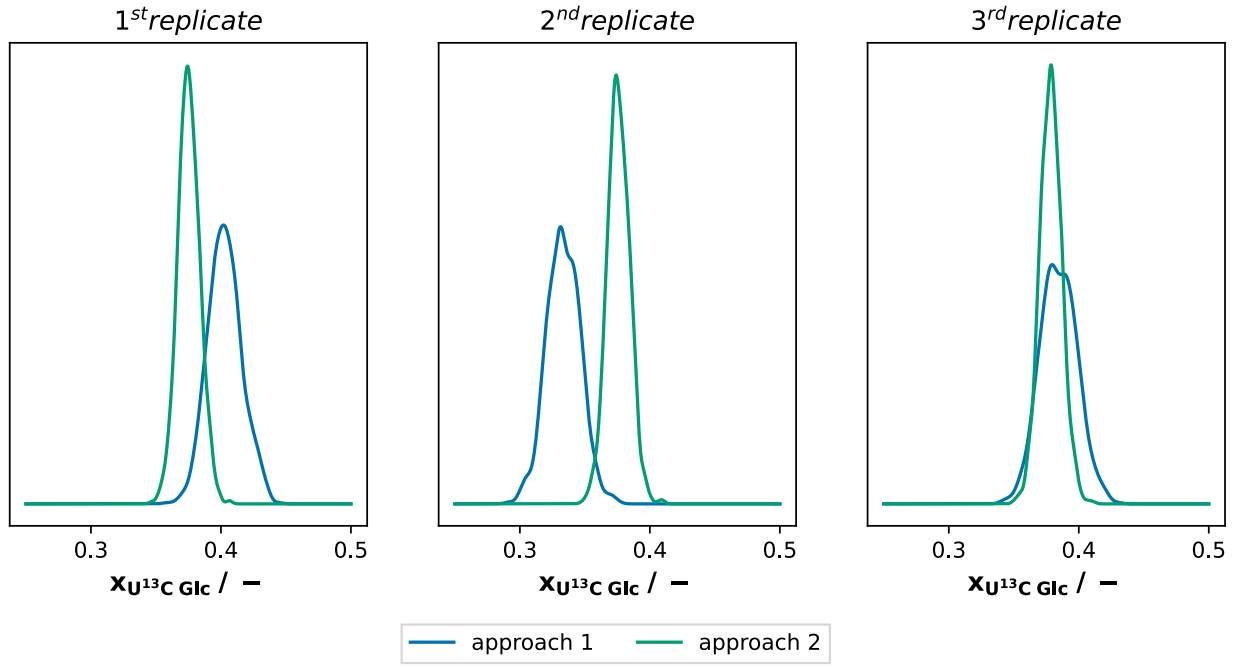


Figure 3.30: Based on the optimized parameters from 1000 MCMC samples for approaches 1 and 2, forward simulations were conducted to determine the relative abundance of uniformly labeled Glc after the pulse and by extension the substrate input labeling mixture.

3.5.6 Influence of replicate handling on variance

To gauge the effect of replicate handling on the results, comparisons are made between pooling replicates (approach 2) versus treating them independently (approach 3). The posterior predictive checks for approach 3 are exhibited in figure 3.31 and once again the data is reproduced well. The only remarkable feature is that similar to approach 1 and in opposition with approach 2, the second replicate of Ser shows the aforementioned lowered steady-state abundance compared to the other replicates.

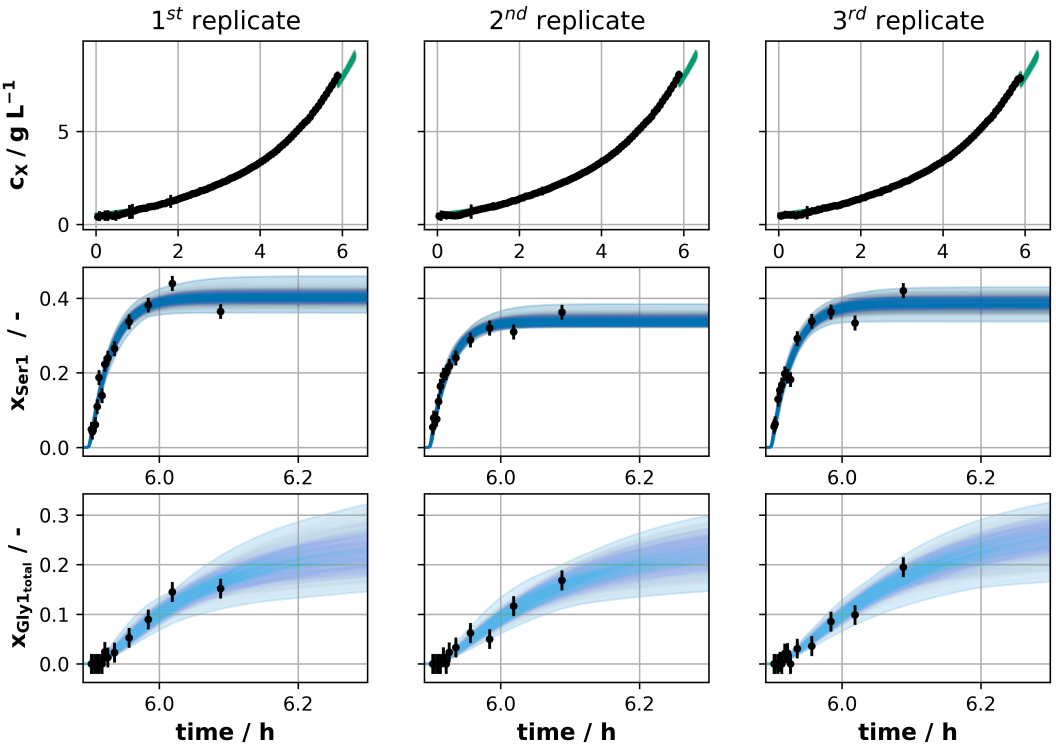


Figure 3.31: Posterior predictive checks computed by performing forward simulations of 1000 random MCMC samples acquired for approach 3.

With regards to the substrate input labeling which is directly connected to the steady-state abundance of fully labeled Ser, it can be observed that the replicate handling is the relevant setting and the influence of bioprocess data seems to be negligible (figure 3.32). It also becomes clear that the variance which is inherent in the data set is not wholly depicted when pooling replicates. Especially the first and second replicates have broad distributions located in large parts outside of their pooled counterparts. Ideally, replicate handling would be conducted in such a way as to reconcile the individuality of biological replicates while still asserting an overarching global structure which is clearly not sufficiently performed when using the pooled approach.

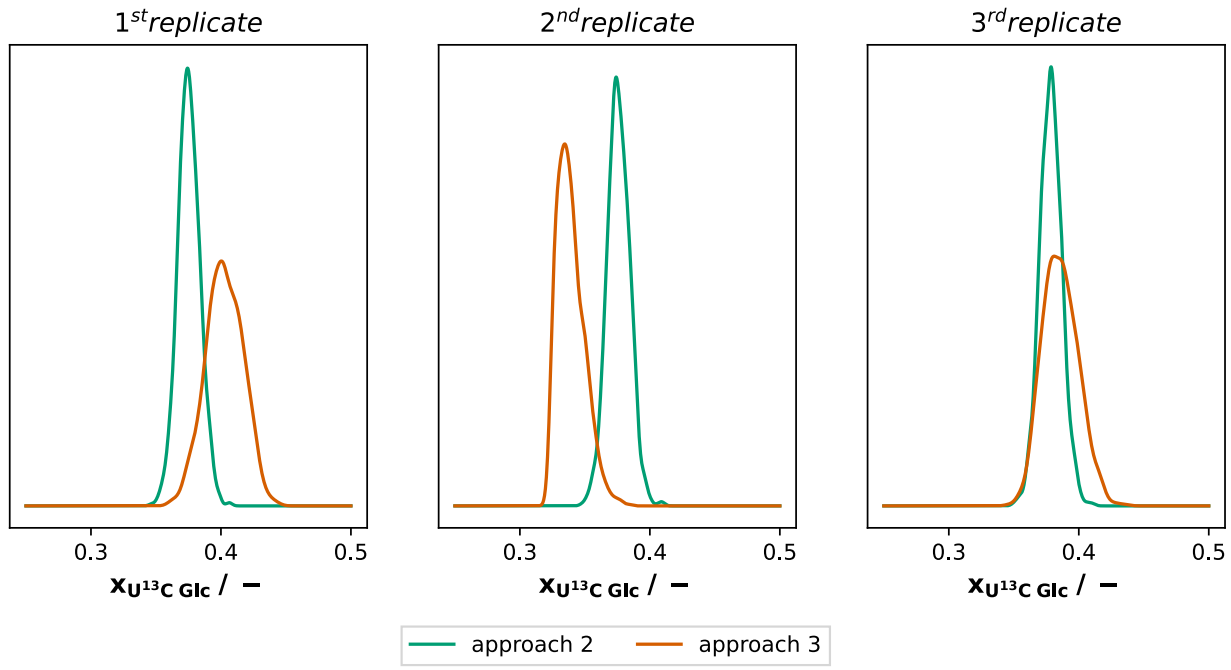


Figure 3.32: Based on the optimized parameters from 1000 MCMC samples for approaches 2 and 3, forward simulations were conducted to determine the relative abundance of uniformly labeled Glc after the pulse and by extension the substrate input labeling mixture.

These first observations have essentially been anticipated by the comparison between approaches 1 and 2, but since approach 3 retains the bioprocess model combined with the independent replicate handling, statements about bioprocess parameters can be made. The maximum growth rate and Gly export factor showed a measure of diversity across the replicates but are mostly congruent with the pooled results, albeit for the slightly underestimated variance when pooling. The maximum substrate uptake rate, however, is scattered much more severely for single replicates than when these parameters were defined as global. In an extreme outlier, the second replicate has a mean only slightly above 1600 mmol L_{cell}^{-1} h^{-1}. In the pertaining flux solution, this is mostly reflected by a much lower value for reaction v_4 leading across the system boundary towards the lower glycolysis and TCA cycle. This constitutes the most severe difference among replicates and approaches and it may be questioned whether this is a realistic phenotype.
With respect to the identifiability between approaches 2 and 3, it is basically unaltered. The posterior remains uninformative with regards to Cys and similarly broad distributions are obtained for EMPUP and Gly.
In consideration of the upcoming analysis of the larger model v1 and the at times poor expression of replicate variability with the pooled approach, it was decided to use the independent approach going forwards. This would also facilitate the process by lowering computation times.

3.5.7 Final evaluation of pool size estimation

Due to the upcoming comparison between the models v0 and v1 with their different scopes, the posterior predictive checks of approach 4 featuring v1 are presented first (figure 3.33).

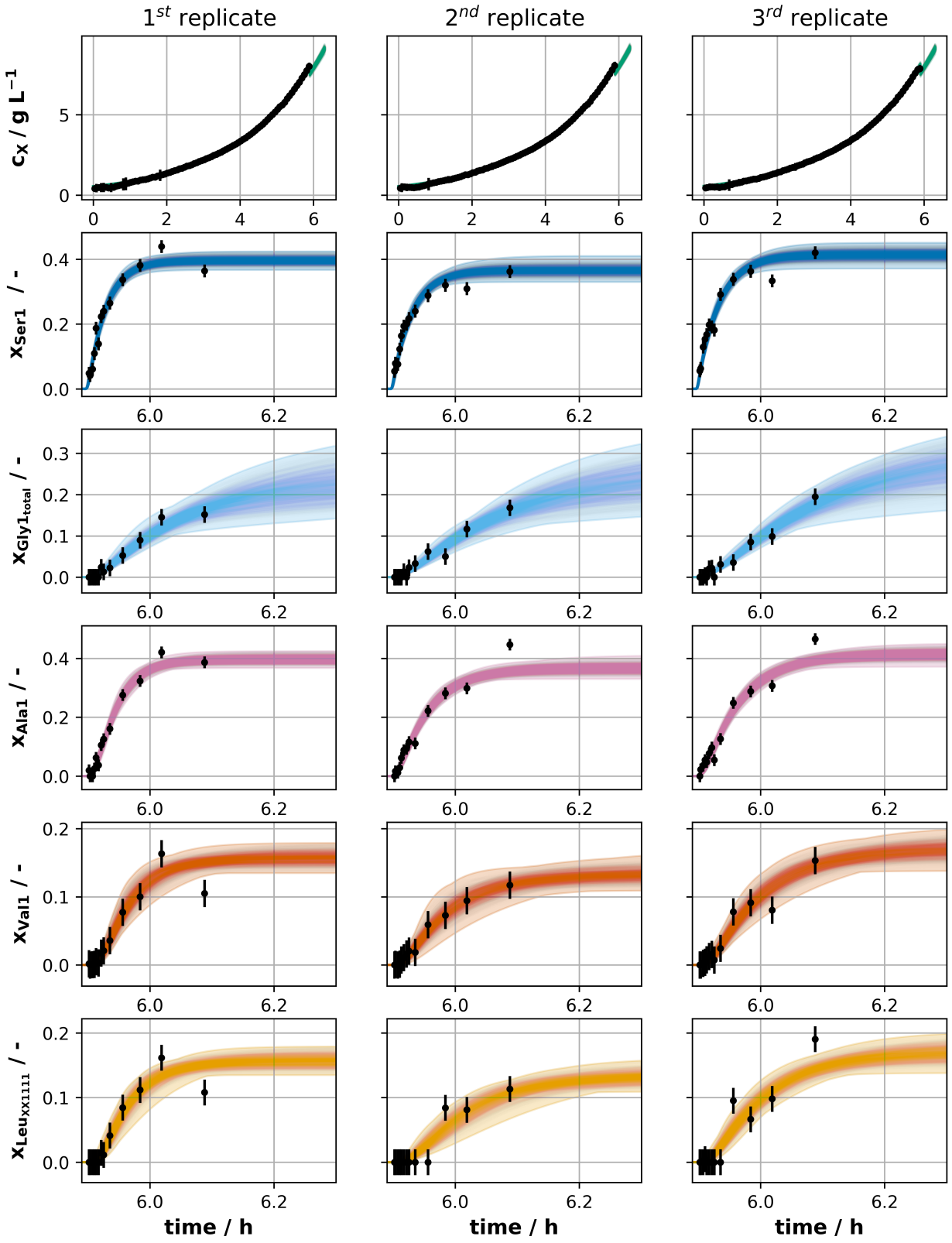


Figure 3.33: Posterior predictive checks computed by performing forward simulations of 1000 random MCMC samples acquired for approach 4.

Here, due to the inclusion of data from the Pyr-derived amino acids the previously discussed lower steady-state abundance of fully-labeled Ser is supported by analogue observations for Ala, Val, and Leu lending much more credibility to the observation as it is now more broadly supported by measurement data. Additionally, the newly added amino acids also share outlier data points with the original ones, in particular the second to last point of the first replicate and the last point of the third replicate.
Regarding the input labeling, the model expansion had a unifying effect, primarily by raising the predicted share of fully labeled Glc for the second replicate which thus converged closer towards the other replicates.

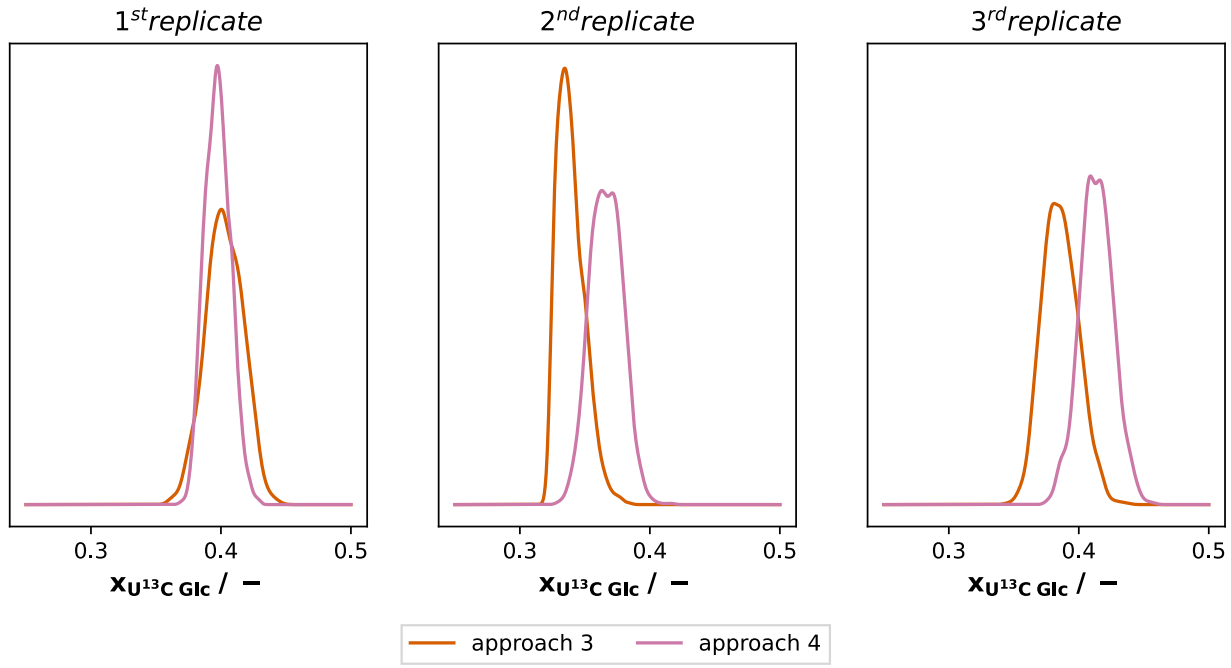


Figure 3.34: Based on the optimized parameters from 1000 MCMC samples for approaches 3 and 4, forward simulations were conducted to determine the relative abundance of uniformly labeled Glc after the pulse and by extension the substrate input labeling mixture.

Focusing on the amino acids included in both models first (figure 3.35), the posteriors for Cys which have been omitted previously are portrayed for this final and most relevant comparison to illustrate the insensitivity of the model towards Cys' pool size. For Gly, both v0 and v1 obtained near identical posteriors and thus pool sizes. The pool size of Ser was estimated higher by 1.5 mM - 2 mM in model v1, at least for the second and third replicate.

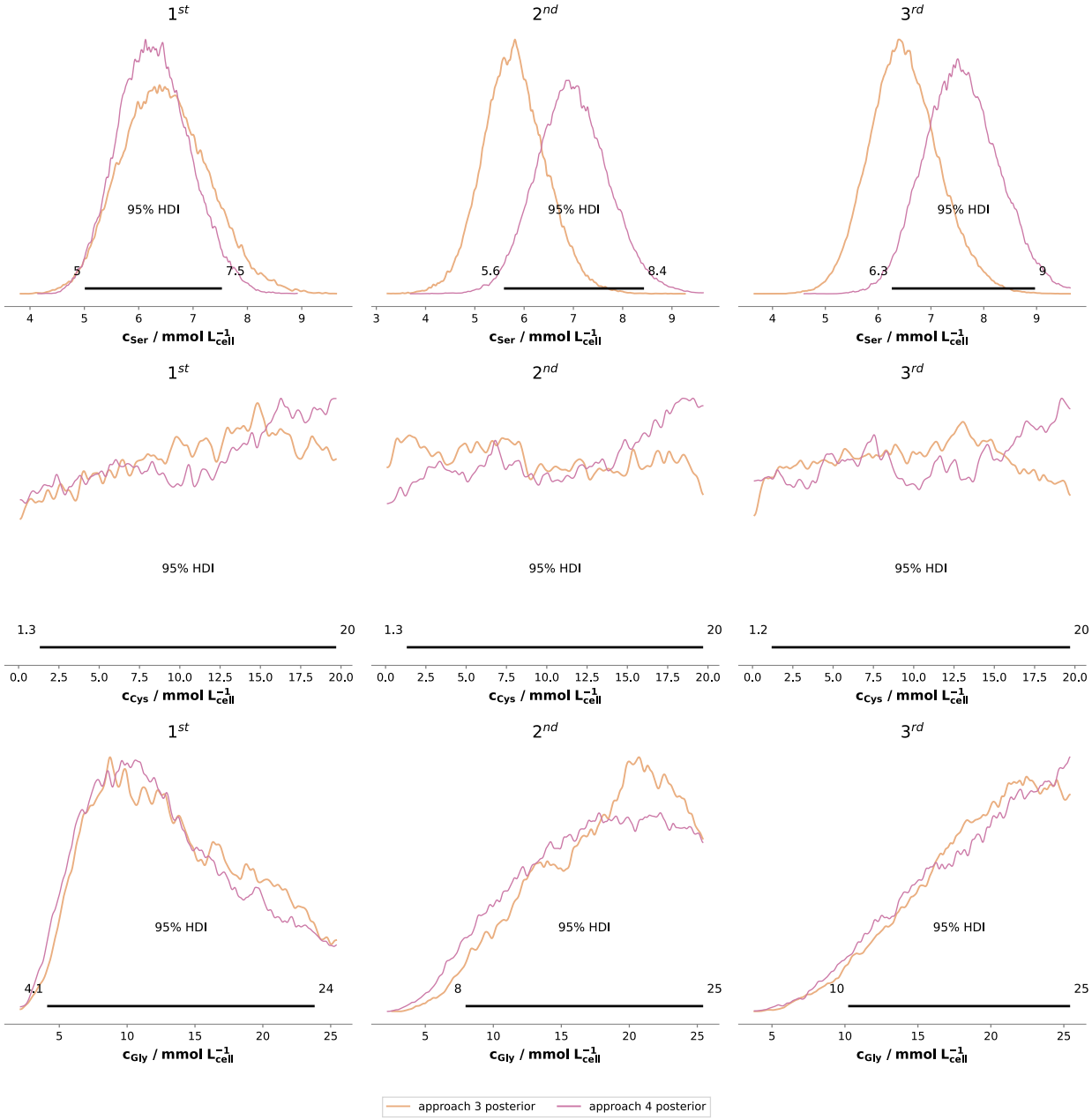


Figure 3.35: Marginal posterior distributions of Ser, Cys, and Gly pool sizes from approaches 3 and 4 to compare models v0 and v1.

Moving on to the Pyr-derived amino acids only present in v1, all three were unexpectedly well determined. While the first replicate underestimated all pools compared to the residual replicates, the totality of the probability mass of all replicates resided within about 20 mM for Ala and Val and within 10 mM for Leu. As the main goal here was to obtain a rough estimate, perhaps just an order of magnitude or a reasonably defined upper boundary for future modeling efforts, this result was much more certain than expected.

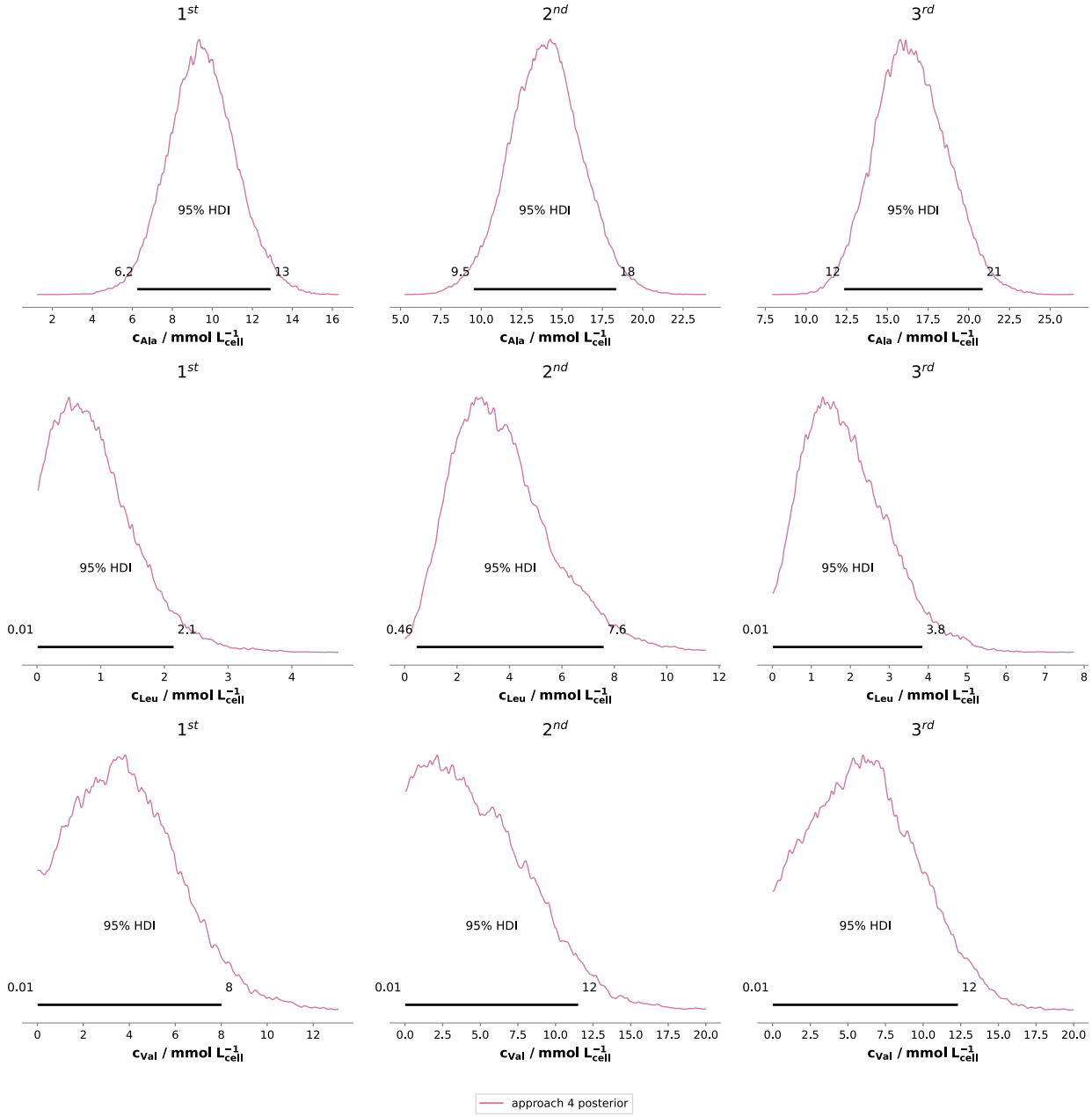


Figure 3.36: Marginal posterior distributions of Ala, Leu, and Val pool sizes from approach 4 featuring model v1.

The bioprocess was characterized in an equivalent manner by both models, in fact so much so that the posteriors for $\mu_{\max}$ and c_{X_0} overlap almost entirely (figure 3.37). As the priors are also portrayed, it can clearly be seen that the measurement data is informative resulting in updated and much narrower posteriors. The differences in $k_{\mathrm{scale}_{\mathrm{Gly,exp}}}$ are slightly more pronounced than for the other bioprocess parameters but remain minuscule with almost identical means and merely slight differences in uncertainty.

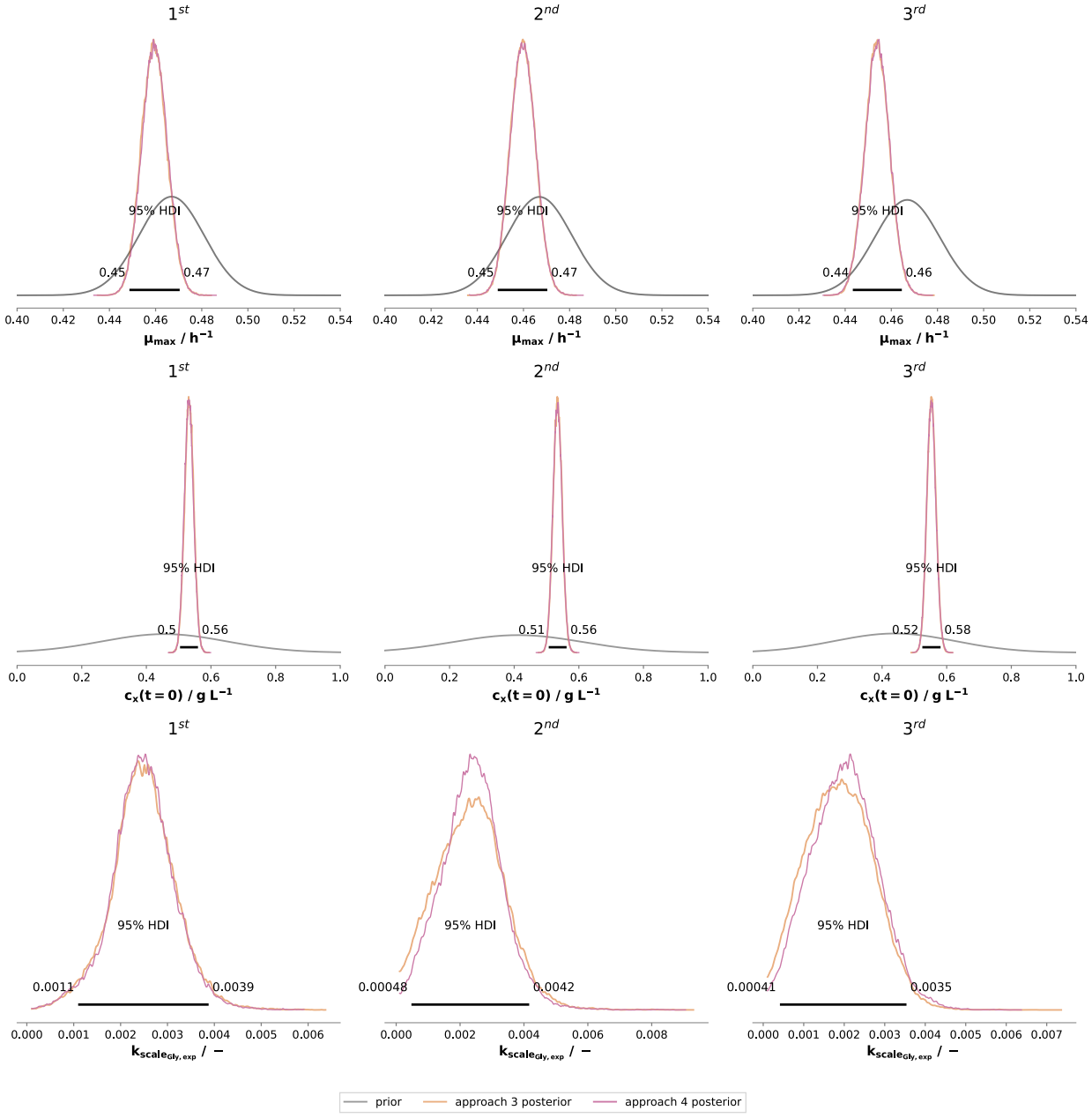


Figure 3.37: Marginal posterior distributions of growth-related parameters from approaches 3 and 4 to compare models v0 and v1.

To explain some of these findings, the underlying flux distribution has to be taken into account (figure 3.38). For the first replicate, both estimations are remarkably similar in absolute flux values across all shared reactions. For the second and third replicate, there is a significant offset regarding $v_{\mathrm{upt},S}$ and flux v_4, i.e. of the flow from substrate to the lower glycolysis. The fluxes towards Ser and Gly, however, exhibit roughly equal means and similar uncertainties so a lower level of abundance in the steady-state of one replicate was compensated by adjusting the respective pool size since the flux values are heavily restricted by design.

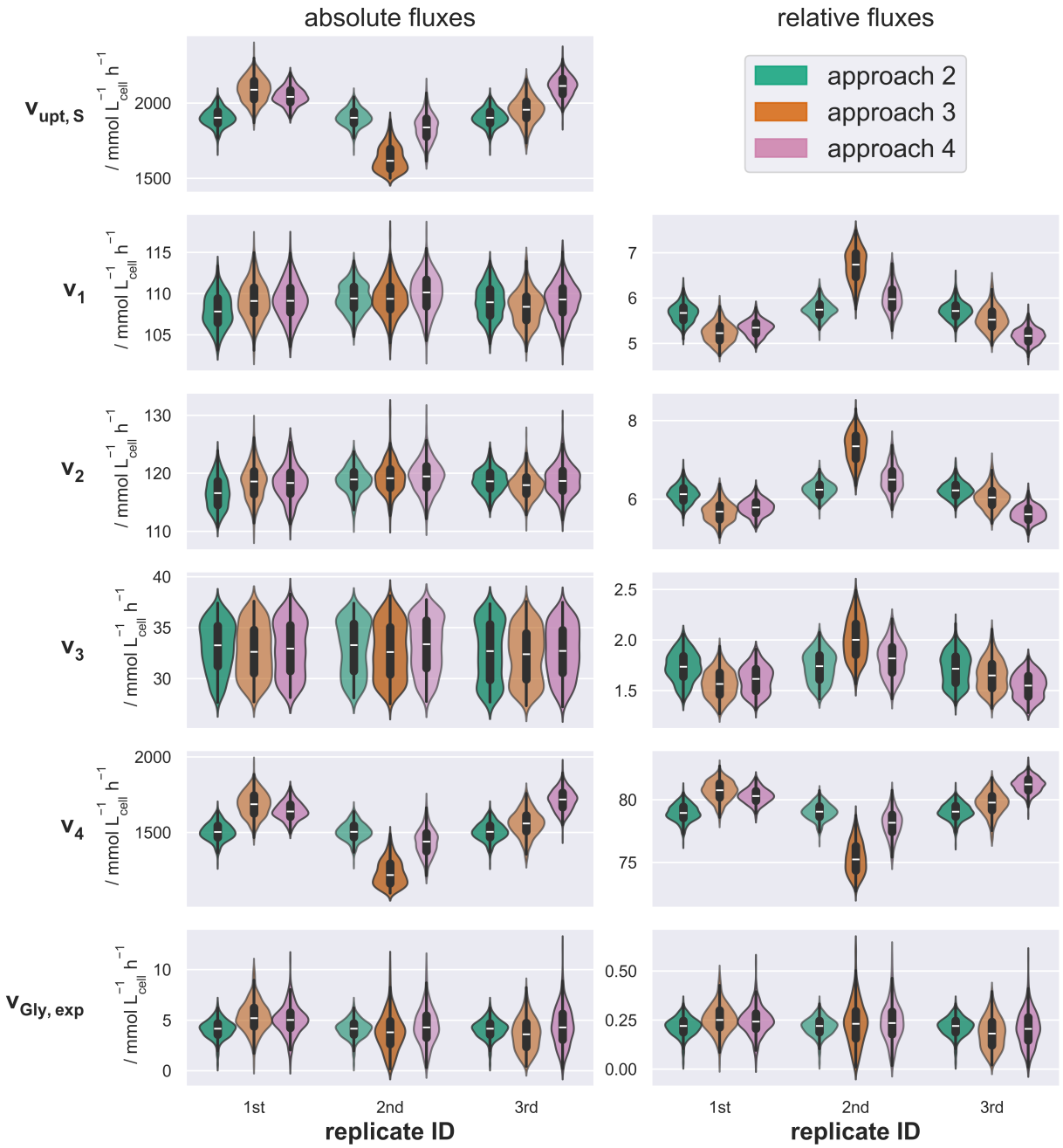


Figure 3.38: Depiction of distributions of absolute fluxes in mmol L^{-1}_{cell} h^{-1} and relative fluxes in % of the glucose uptake rate determined with models v0 and v1 subdivided by modelling approach and replicates.

The residual fluxes exclusive to model v1 are shown in figure 3.39. Here, too, divergences in steady-state abundances of e.g. Leu are mostly caused by the pool size estimations and only minor variations in fluxes across the replicates were observed. In accordance with the flux values from the upper glycolysis, v_5 towards Pyr and v_{10} towards AcCoA on the outside of the system boundaries are much lower in value than their replicate counterparts.

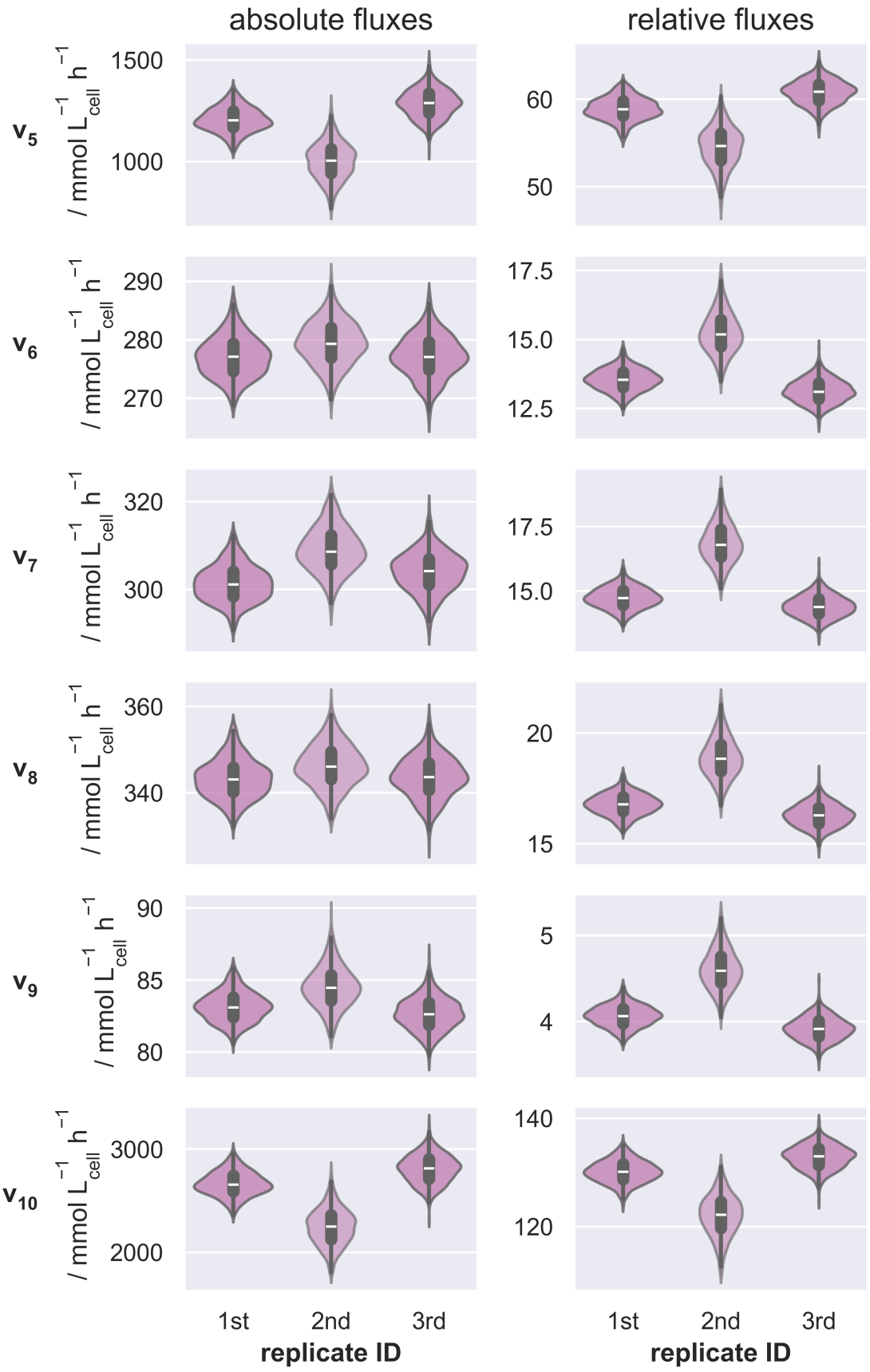


Figure 3.39: Depiction of distributions of absolute fluxes in mmol L^{-1}_{cell} h^{-1} and relative fluxes in % of the glucose uptake rate determined with model v1 subdivided by replicates. Since only approach 4 used model v1, only data from this approach is shown.

Since fluxomes of *C. glutamicum* WT have been published previously in literature, the estimated relative flux values can be compared to these counterparts to further evaluate the model. Using only approach 4 with model v1 as a reference, the relative reaction rate v_4 towards PEP of about 80 % has to be doubled since it is a uni-bi splitting reaction [3], i.e. PEP has a stoichiometric coefficient of 2. The resulting value of 160 % is comparable if slightly higher than comparative values from literature of 153.4 % [64] and 149.8 % [66]. The same is true for the flux v_{10} from Pyr toward the TCA cycle of about 130 % for replicates 1 and 3 and 120 % for replicate 2 compared to 122.3 % in the published fluxome [64]. In this case, replicate 2 is considerably closer to the reference value. A deviation is observed with respect to the flux from glycolysis towards Ser denominated here as v_1. As it is once again defined as a uni-bi splitting reaction, the flux of 10 % - 12 % is significantly higher than the reference value of 6.1 % [64]. It is relevant to state, though, that the reference fluxome does not include amino acid biosynthesis and merely features a reaction originating from Ser's precursor glyceraldehyde-3-phosphate across the system boundary. Since the cultivation conditions between shake flasks and microbioreactors may differ, slight deviations of the fluxes can realistically occur. Overall, the comparison with published fluxes mostly corroborates the realistic depiction of *C. glutamicum*'s flux distribution on Glc.

The next and final step, then, is the comparison of the resulting metabolite pool sizes to literature values (figure 3.40). For differences among replicates and replicate handling approaches as well as a reference to the estim8 results, refer to the previous sections.

It seems, the estimations most comparable to literature values have been obtained for Val, Ala, and to a lesser degree Leu. In contradistinction, the pool sizes of Ser and Gly have been overestimated relative to the publications and in a less pronounced way this extends to Leu and – depending on the study – Ala. The finding of Ser's and Gly's pool sizes can be connected to the previous observation of the relative high flux towards Ser. As the labeling time course is a function of the pool size and pertaining influx, a large flux might have been compensated by the model with a larger pool size. For Ala, the reference values from [203, 204] amounting to 12.5 mM and 10.4 mM, respectively, were located in the same range as the estimations with means of 9.5 mM, 14 mM, and 16.5 mM but other studies arrive at much lower quantities below 4 mM. In a similar fashion, in [203] a high intracellular concentration of 8 mM was reported for Val which is once again comparable to the estimates when considering their uncertainties beside the maximum a posteriori values of 3.9 mM, 5.1 mM, and 6.2 mM. The remaining studies determined Val concentrations around 1 mM to 2 mM. With regards to Leu, there were replicate-specific deviations resulting in a good agreement of the first and third replicates with literature values while the second replicate exhibited a much broader distribution with a comparatively high mean of 3.7 mM.

Additionally, pool sizes for glycolytic intermediates PEP and Pyr were on average estimated much larger than described in literature. However, the uncertainty on these estimates is very high for they were not detected via LC-MS/MS and thus no measurement data was available. In this case, the best use for the results would be to apply an upper boundary at around 50 mM when using this data for other modelling objectives instead of an infinite upper limit.

Combining these findings it may seem obvious at first glance to assume a systematic trend of the models to overestimate pool sizes yet there are some considerations to be aware of.

In general, the difficulty of measuring accurate pool sizes should be noted [17–19] as proven by

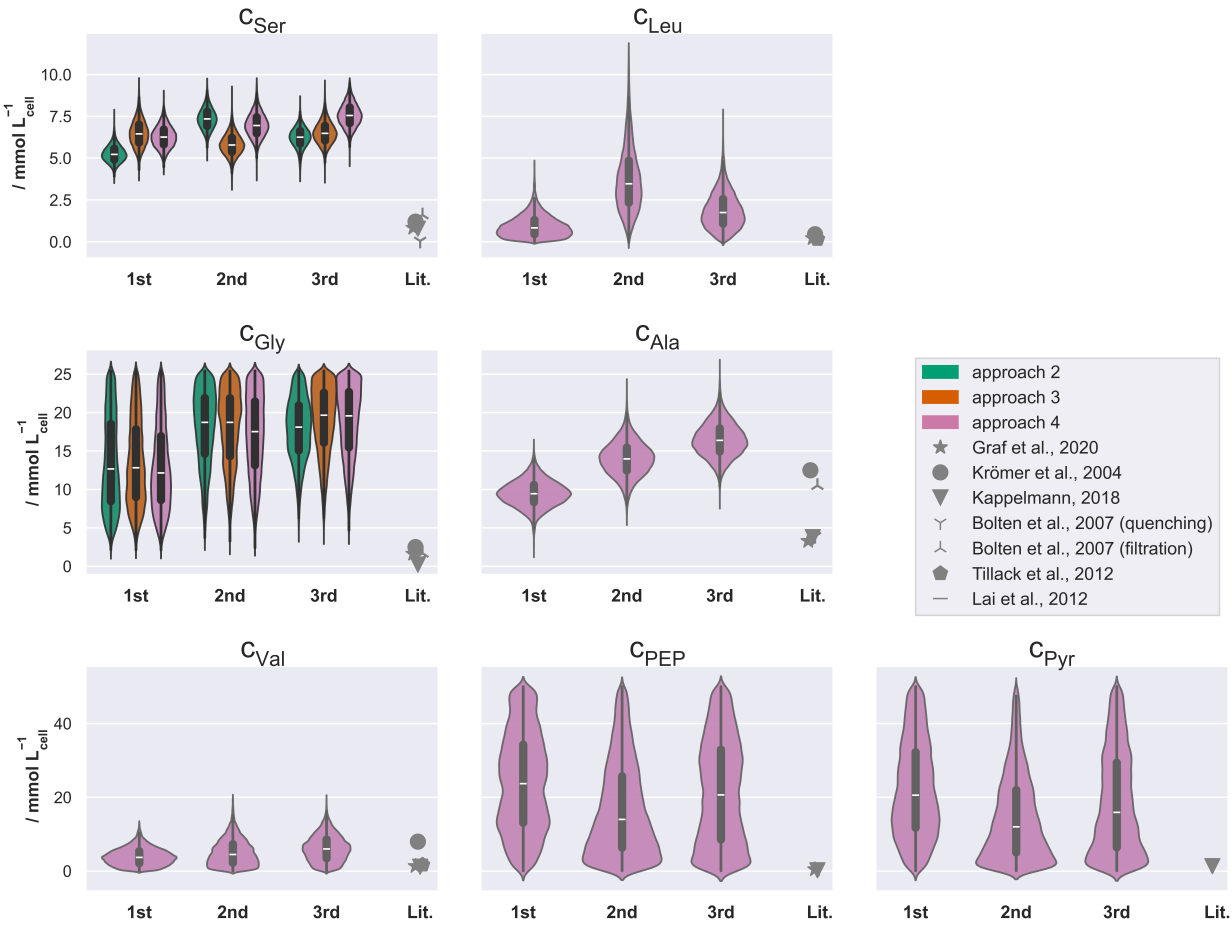


Figure 3.40: Estimated distributions for pool sizes with the MCMC pipeline using approaches 2 - 4. The portrayed literature values (Lit.) were taken from [66] (grey star), [203] (grey circle), [53] (grey triangle), [204] (grey downward and upward pointing three-lined stars for data generated with cold methanol quenching and rapid filtration, respectively), [17] (grey pentagon), and [205] (grey horizontal line).

the fact that the pertaining standard deviations are commonly very large relative to the determined mean. For example, for Gly one study [17] reported a pool size of 1292.55 µM ± 967.92 µM and another [205] of 3.3 mM ± 1.1 mM. After all and as previously remarked, the process requires the error-prone determination of multiple concentrations to eliminate the influence of the exometabolome and quenching-related systematic errors [17].

Furthermore, it poses quite the challenge to find literature data truly suitable for an apt comparison as none of the cultivations in the selected studies have been conducted in a microbioreactor and even obtaining a study with the same strain and measuring the relevant amino acids is non-trivial. In this case, [203] used *C. glutamicum* ATCC13287, a Lys producer derived from the WT, [17] *C. glutamicum* DM1800, [205] *C. glutamicum* SER-0, this time a Ser producer evolved from the WT, and only the remainder of the plotted studies actually contained data for the WT, i.e. *C. glutamicum* ATCC13032.

Perhaps the most important point, however, regards the utilized quenching methods. The astute reader may have noticed that values from two studies [203, 204] in particular generally featured much higher concentrations than the residual publications irrespective of the amino acid. This is the case as both used a rapid filtration method before incubation at high temperature for extraction.

The later study [204] even performed comparative measurements between cold methanol quenching and said filtration approach and found that filtration consistently resulted in much higher values. To highlight this, both results were plotted separately in figure 3.40 As this approach is much more closely related to the automated hot isopropanol quenching workflow [142] used in this thesis, it is sensible that the obtained values are more comparable, then. All other quoted studies used variations of the cold methanol quenching methods with different attempts and levels of effort directed towards correcting for metabolite leakage. It is not clear whether the whole-broth sampling approach with corrections for the exometabolome or the cold methanol quenching approach with its associated corrections is the superior option. With respect to the presented modelling approach, avenues to exclude an overestimation of intracellular pool sizes should be explored, e.g. by obtaining exometabolome data within the same experiment and perhaps via the inclusion of further extracellular pools although that would increase the dimensionality of the problem. In case of a substantial extracellular pool like for Gly, though, such a phenotype was noticed during model validation so it is uncertain whether the inclusion of further extracellular pools would in practice improve the accuracy of the obtained results.

Shifting the focus from the hard to gauge accuracy to precision, the results for all amino acids for which labeling data could be measured were unexpectedly positive with regards to uncertainty. Especially for Ser, the 95 % HDIs were located within a range of about 4.5 mM for all replicates and all four modelling approaches.

In light of the aforementioned systematic errors in place for pool size measurements [17–19], the presented pool size estimation can be evaluated as a success, since its results can at the very least be used to restrict pool sizes to a general order of magnitude if not identify them outright. With regards to actual practical applicability, by optimizing the sampling time points and reducing the number of replicates, one might sacrifice a selected number of wells for the purpose of generating data for pool size estimation while using the majority on e.g. an INST ILE or another separate experiment. Based on previous experiences with toy examples where even the knowledge of one pool size can suffice to perform an INST-MFA, restricting the pool sizes of up to six amino acids and up to three intermediates should fulfill the same purpose.

3.6 Case study II: INST ^{13}C-MFA on ethanol

This chapter is based partially on an upcoming publication co-first-authored by JN and Anton Stratmann (AS). JN designed the ILE with 1-^{13}C ethanol, conducted the automated experiment, LC-MS/MS analyses and TMID calculation. AS conducted the correction for natural isotope abundance. Model validation and expansion as described in 2.7.4 were performed by AS and JN. The Materials and Methods section pertaining to this chapter (2.7.4) was written by AS, Stephan Noack (SN), and JN. Optimizations and profile likelihood calculations were performed by AS. The estimation of extracellular rates for the INST ^{13}C-MFA was conducted by AS based on bioreactor data and by SN based on microbioreactor data. The biological evaluation of results as presented in this text and their visualization were authored by JN. The script for the visualization was based an earlier version by Martin Beyß.

^{13}C-MFA represents in a sense the culmination of ILEs and constitutes an important phenotyping technique in systems biology. Since the basic requirements of establishing a MSS, taking measurements for determining extracellular rates and metabolite labeling states, and performing metabolic quenching are fulfilled, the isotopically stationary variant can clearly be conducted with the automated workflow without requiring further developments or a proof of concept.
However, the same does not hold true for INST ^{13}C-MFA. Due to the inability to accurately measure pool sizes when using automated hot isopropanol quenching, a vital source of data is missing from the equation, hence reducing the number of measurements while increasing the system's degree of freedom. As the automated workflow is otherwise uniquely qualified for INST-ILEs due to its high degree of parallelization, it needs to be established whether or not an INST ^{13}C-MFA is possible with this innovative setup. A successful application of this technique would, after all, constitute both the first time to perform an INST ^{13}C-MFA at a microliter-scale and based on an automated experiment.
Regarding the object of this particular case study, an adaptive laboratory evolution experiment recently produced a mutant of *C. glutamicum* WT denominated as WT_ETH-evo exhibiting improved growth on ethanol as the sole carbon source [140]. Due to a mutation upstream of the *ald* gene interfering with the binding of the transcriptional regulator GlxR, an over-expression of the acetaldehyde dehydrogenase catalyzing the NAD-dependent oxidation of acetaldehyde to acetate was caused, thereby increasing the overall rate of ethanol degradation. As there was no published fluxome for *C. glutamicum* grown exclusively on ethanol, yet, this mutant was selected as the target for an attempt at INST ^{13}C-MFA using the automated ILE workflow.
This choice of application study has further advantages, though, aside from the novelty of a flux distribution on ethanol. Firstly, *C. glutamicum* is retained as a model organism. Secondly, there are previous results for *C. glutamicum* like an MFA on acetate, omics data on ethanol, and pool sizes on glucose which provide ample opportunity to contextualize and contrast results. Thirdly, the new mutant is investigated using fluxomics in addition to the already applied proteomics. Fourthly and finally, since ethanol is a 2-carbon molecule, the re-distribution of labeled C atoms in the CCM is limited and accordingly the informativeness of an isotopically stationary approach is, too. Hence, the substrate itself favors an INST approach which is in turn enabled by the automated workflow.

Regarding the design of the experiment, a traditional DoE simulation study as previously described in literature [206–208] was foregone due to the aforementioned limited tracer combinations offered by ethanol. Especially when planning an INST instead of an isotopically stationary labeling experiment, conducting such a study would be even more complex since it not only requires a reasonable approximation of a flux distribution (e.g. via flux balance analysis) and extracellular rates but metabolite pool sizes, as well. Also, it would require testing not only numerous combinations of 1-^{13}C, 2-^{13}C, and U^{13}C ethanol but an optimization of the time points for sampling. Instead, the latter were decided based on data from previous experiments with *C. glutamicum* which was readily available as a foundation for an informed judgement. In particular, the ethanol uptake rate of WT_ETH-evo in a lab-scale bioreactor experiment of 8.45 mmol g_X^{-1} h^{-1} suggested a slower uptake of carbon than on glucose with its expected uptake rate of roughly 4.5 mmol g_X^{-1} h^{-1} [195]. When relating these rates to the uptake of carbon atoms they amount to 16.9 C-mmol g_X^{-1} h^{-1} for ethanol and 27 C-mmol g_X^{-1} h^{-1} for glucose. Combining this observation with the INST labeling data from the proof of concept experiment with U^{13}C Glc and the WT, it was accordingly expected that the incorporation of labels would generally be slower, especially for metabolites pertaining or adjacent to the EMP and PPP. Thereby, a reasonably confident guess about informative time points could be made for the first ILE on ethanol. Compared to the time points sampled for the proof of concept experiment of 25 s, 30 s, 40 s, 50 s, 60 s, 70 s, 90 s, 120 s and 300 s after the pulse, for the present ILE on ethanol the delays 24 s, 35 s, 60 s, 120 s, 180 s, 1200 s, and 1800 s were chosen. The final point with a delay of 1800 s was selected in order to hopefully obtain one data point located in the isotopic steady-state of free amino acids.

All labeling time courses utilized for the INST ^{13}C-MFA are presented in the appendix (figures A7 and A8). Particularly for Gly, Ala, Val, Asp, L-homoserine (Hser), Thr, Met, Glu, Gln, and L-citrulline (Citr) the INST phase was sampled thoroughly. For amino acids which were either comparatively distant from ethanol's entry into the metabolic network or located downstream of large pools buffering the incorporation of labeled carbon atoms like e.g. Glu, only the beginning of the labeling dynamic was recorded. For example, Ser as a glycolysis-derived amino acid has a much slower dynamic on ethanol than on Glc. Accordingly, Gly which is one reaction downstream of Ser and features a larger extracellular pool exhibited an even more delayed and slower time course of label incorporation. The only included PPP-derived amino acid His showed barely any label incorporation during the observed time frame which was not unexpected. In these cases among others, the early sampled time points were rather uninformative. Even for the Pyr- and Asp-derived amino acids, the ISS was not nearly reached after 180 s so that the gap to the next data point of 1200 s turned out too large. An additional data point in this time frame or a reallocation of the present data points would certainly have proven advantageous.

Accordingly, harnessing the increased throughput of the automated workflow, follow-up experiments were designed with improved time points based on these results. To emphasize, while such iterative design is generally applied in many biological experiments, this is usually not the case for ILEs due to the previously discussed temporal and monetary burdens. Specifically, it was decided to include an additional time point by relinquishing the isotopically stationary experiment conducted in three biological replicates per condition, i.e. strain or tracer. Furthermore, the delays after 120 s were altered significantly. The data point at 180 s was pushed back to 540 s or

9 min and intended to be accompanied by samples after 14.5 min, 33 min, and a new final point at 45 min. Thereby, the previously unobserved time frame after 180 s was covered more evenly and a significantly later end point was sampled during the new ILEs.
Naturally, more labeling time courses than those included in the experiments had been detected and evaluated, hence the reasons for these exclusions ought to be discussed. Most prominently, Orn and Arg showed little to no label incorporation during the measured time frame while their direct neighbors Glu and Citr exhibited quite a fast dynamic with fully unlabeled mass traces of about 22 % and 40 %, respectively, after 1200 s. These paradoxical observations cannot be reconciled so the data showing the expected labeling dynamic were trusted over the complete absence of labeling implied by the Orn and Arg data. In case of the aromatic amino acids, their LC-MS/MS signals were either absent or had a comparably low intensity leading oftentimes to an increased effect of noise on the peak areas and accordingly fluctuations between biological replicates. Therefore, while L-phenylalanine (Phe) was detected, it was ultimately discarded due to its high variation. Finally, the double peak of Leu and Ile impaired the evaluation of especially the smaller Leu peaks. Due to the noisy signal, high variance time courses were recorded for Leu which seemed to plateau after 120 s – in other words, much earlier than any other amino acids and particularly of amino acids in the vicinity. Accordingly, only Ile's data was included for the flux estimation.
With respect to the results of the INST ^{13}C-MFA, the obtained flux map is shown in figure 3.41 and a full account of all fluxes was included in the appendix (table A1). It must be impressed upon the reader, however, that this is an intermediate result and may thus differ from the one presented in the upcoming publication mentioned at the beginning of this chapter. Due to the lack of previous fluxomes of *C. glutamicum* on ethanol, the inferred fluxes will be contrasted against the closest point of comparison available in literature, i.e. an isotopically stationary ^{13}C-MFA with the *C. glutamicum* WT on acetate [64]. Before doing so, it is important to note some further key differences between the two ^{13}C-MFAs. For the acetate MFA, the labeling states of CCM intermediates were observed and a more reduced metabolic network model was utilized. Amino acids were not included but their biosynthesis was implied by drain terms crossing the model boundaries. As the present INST ^{13}C-MFA used labeling data from free amino acids, the measured parameters used for the simulations are quite distinct. In the acetate case, the TCA cycle and EMP pathway were simplified but most crucially the Mal/OAA and PEP/Pyr pools were lumped in order to unify multiple anaplerotic reactions, thus simplifying the notoriously complex anaplerotic node. Since the absolute flux values are given in mU $mg_{protein}^{-1}$ without specification of the protein mass, only relative values will be contrasted.
Despite the similar entry of both substrates into the metabolic network, the first major deviation is constituted by the TCA cycle usage. On acetate, there was a higher relative flux into the TCA cycle of 76 % compared to 48 % via the citrate synthase reaction combining AcCoA and OAA to citrate (Cit). At the bifurcation originating from the combined Cit and isocitrate (Icit) pools, a ratio of glyoxylate shunt vs. continued TCA cycle usage of 24 % was observed on acetate compared to 75.5 % on ethanol. This difference is caused by a doubled flux of 36 % into the glyoxylate shunt on ethanol with a concomitant decrease of the TCA cycle flux toward AKG from 58 % to 11.8 %.

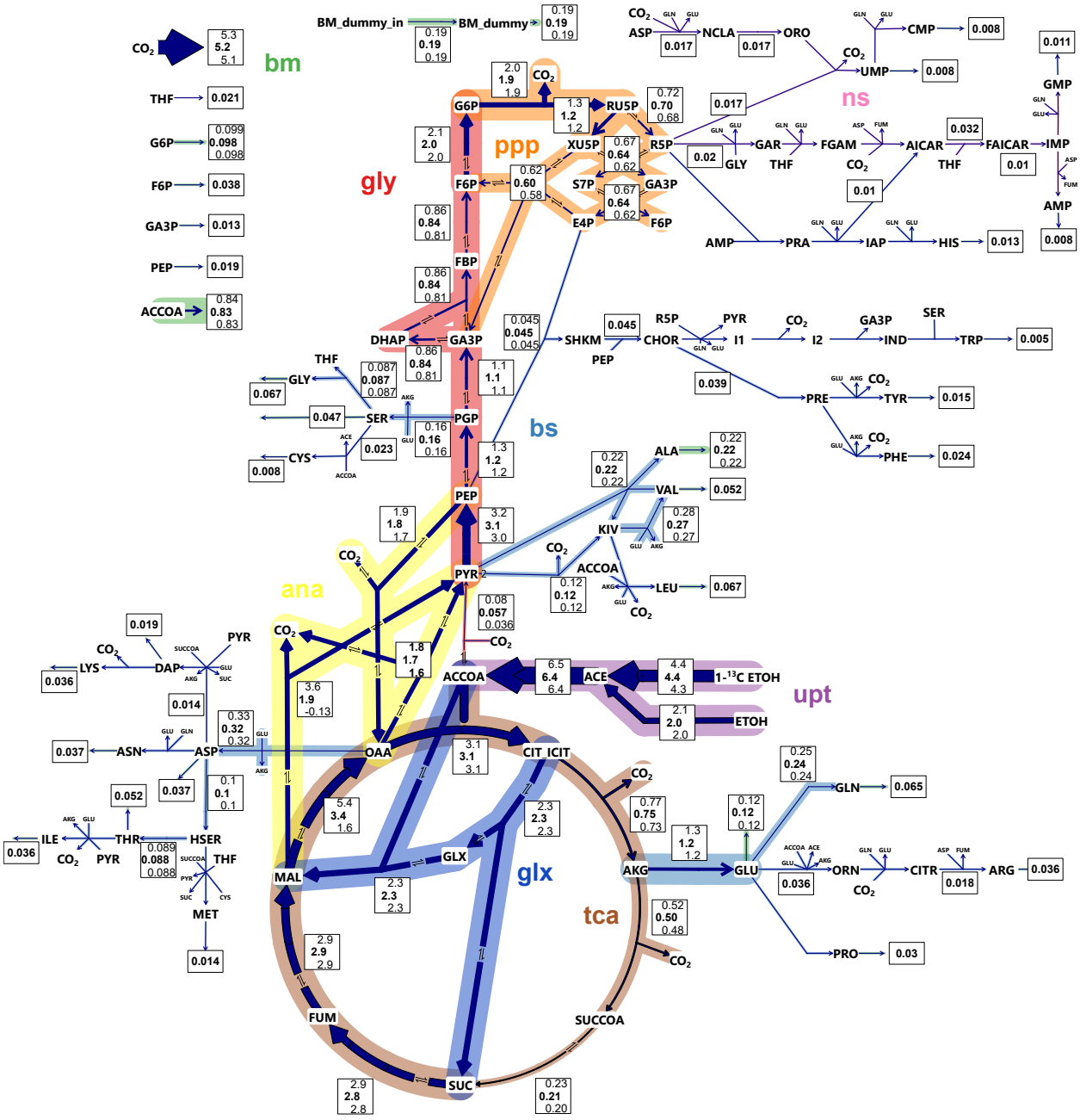


Figure 3.41: Flux map obtained by INST ^{13}C-MFA with *C. glutamicum* WT_ETH-evo grown on unlabeled ethanol and pulsed with 100 % 1-^{13}C ethanol during the mid-exponential growth phase. All flux values are given in mmol g_X^{-1} h^{-1} and the boxes contain from top to bottom the upper boundary of the 95 % CoI, net flux, and lower boundary of the 95 % CoI. Values above 1 were rounded to the first decimal, values below 0.1 to the third decimal and values in between to the second. The width of the arrows scales with a reaction's flux value and the width of the background colors pertaining to pathways relates to their usage. The portrayed pathways are glycolysis (gly), PPP (ppp), TCA cycle (tca), anaplerotic reactions (ana), glyoxylate shunt (glx), biosynthesis of amino acids (bs), biomass formation (bm), nucleotide synthesis (ns), and ethanol uptake (upt). In case multiple sequential reactions in linear pathways had the same flux value (usually preceding a biomass drain reaction), the redundant fluxes were omitted to preserve visual clarity.

This finding can be explained by the chemical characteristics of the substrates. While acetate and ethanol partially share a common degradation pathway, ethanol has a higher degree of reduction of 6 compared to acetate's 4 and accordingly is oxidized twice before the pathways converge. The

regeneration of the reduction equivalent NADH in these two reactions is likely a reason why the glyoxylate shunt activity is higher than the oxidative TCA cycle activity on ethanol since isocitrate dehydrogenase and α-ketoglutarate dehydrogenase both catalyse NAD-dependent oxidations, as well. As a side effect, the carbon dioxide evolution rate is much higher on acetate amounting to 481 % compared to the 82 % on ethanol as said reactions are decarboxylations. Such a loss of carbon atoms is also reflected in a much lower biomass yield of 0.29 g_{C_X} $g_{C_S}^{-1}$ on acetate compared to 0.41 g_{C_X} $g_{C_S}^{-1}$ on Glc [64] or the presently determined biomass yield on ethanol of 0.516 g_{CDW} g_S^{-1} or 0.40 g_{C_X} $g_{C_S}^{-1}$ (based on the biomass carbon content of 0.408 g_{C_X} g_{CDW}^{-1} [15]). Even so, the growth rate of 0.28 h^{-1} on acetate is considerably higher than the 0.19 h^{-1} determined on ethanol, although the high acetate concentration already impacted growth adversely at the utilized concentration [64].

There are, however, contradicting sources of data which need to be taken into account. Firstly, transcriptomics have uncovered a slightly lower expression of the isocitrate lyase gene *aceA* (*cg2560*) during growth on ethanol compared to acetate [62]. In the same publication, a near identical specific activity of the enzyme was observed in cell-free extracts. Accordingly, a large shift in the activity of AceA caused by disruptions of the transcriptome or proteome seems unlikely. The strain WT_ETH-evo investigated in this thesis, however, is not fully identical to the WT and while proteomics data [140] has shown no fold changes between the two with respect to AceA, the isocitrate dehydrogenase catalyzing the reaction from isocitrate to AKG exhibited a fold change of 0.38 for the mutant. It is unlikely that this suffices to explain the drastic difference of the TCA cycle usage observed on ethanol with WT_ETH-evo, especially since the protein concentration by itself does not permit direct statements about the pertaining flux [101] but it is possible that the WT would exhibit a ratio of glyoxylate shunt to TCA cycle usage closer to the acetate condition. Since no labeling data for glyoxylate is available, the closest data point to base this assumption on is the label incorporation into Glu which is derived directly from AKG. In an INST ILE of the WT grown on ethanol, the time course of said incorporation was slightly faster for the WT (figure A9) as observable by the offset of the unlabeled mass trace. This supports the claim of an exaggerated glyoxylate shunt activity of WT_ETH-evo relative to the WT while maintaining the original hypothesis of an altered TCA cycle activity during growth on ethanol relative to acetate based on the balance of reduction equivalents.

Due to the higher relative activity of the TCA cycle on acetate, the flow into gluconeogenesis and PPP are accordingly de-emphasized. The fluxes from PEP to GAP, GAP to F6P, and F6P to G6P amount to relative rates of 8 %, 3 %, and 5 % on acetate vs. 17 %, 13 %, and 32 % on ethanol, respectively. Nevertheless, the relative fluxes towards Ser are virtually identical with 2.59 % on acetate and 2.54 % on ethanol. Corresponding to the supply of G6P, entry into the PPP catalyzed by the 6-phosphogluconate dehydrogenase (Pgd) was determined to occur at a much higher rate of 30.6 % on ethanol compared to a mere 4 % on acetate.

In summation, growth on the more reduced substrate ethanol compared to acetate significantly alters the TCA cycle and glyoxylate shunt activity causing a more efficient growth characterized by a higher biomass yield, lower CER, and increased relative usage of gluconeogenesis and PPP. However, the adverse effect of ethanol on the growth rate compared to acetate is more pronounced as expressed in an even lower growth rate, especially with regards to the WT but also extending

to the improved mutant WT_ETH-evo.
Due to the lack of available pool size measurements, not all intracellular concentrations were identifiable by INST ^{13}C-MFA. Such pool sizes with 95 % CoIs approaching equivalence with the lower and upper boundaries restricting the solution space of the optimization, were deemed unidentifiable and accordingly only included in the table with all results in the appendix (table A2). The identifiable ones for which published pool sizes on Glc as the sole carbon source were found are additionally portrayed in figure 3.42. It is important to state that – similar to the results in section 3.5 – for all amino acids for which INST labeling data was included in the optimization, pool sizes were identified. Additionally, this is the case even for some intermediates, especially from the TCA cycle, for which no experimental labeling data had been obtained. A reason for this may be that most included labeling data belong to Glu-, Asp-, and Pyr-derived amino acids all of which are located in the close vicinity of the TCA cycle.
An expected difference between the two conditions is observed for glyoxylate as the glyoxylate shunt is known to be inactive on Glc and essential on ethanol. Accordingly, a pool size of 0 mM [53] and 0.015 mM $\pm$ 0.014 mM [17] were observed on Glc vs. 26.47 mM on ethanol. Similarly, the increased AcCoA pool is not unexpected as it constitutes the closest CCM intermediate to ethanol which may lead to an accumulation. Furthermore, the pool sizes of Ser and Gly are generally comparable but the MLEs are lower on ethanol which is in accordance with a higher relative flux towards Ser on Glc of 6.08 % vs. 2.54 %. As Ser is far apart from ethanol in the metabolic network, there is likely only a minimum flux required for providing precursors for growth. A major departure, however, is constituted by the manifold larger PEP and Ile pools on ethanol. With regards to PEP in particular, this node may constitute a bottleneck leading to such an inflated pool.
Evaluating the results of the present application study, an INST ^{13}C-MFA could be conducted based on an automated ILE at a microliter scale. Thereby, the ethanol catabolism of *C. glutamicum* was investigated on the level of the fluxome and an increased activity of the glyoxylate shunt in particular as well as the gluconeogenesis and the PPP was identified relative to other substrates entering at the level of AcCoA. For this first proof of concept fluxome and due to temporal restrictions, extracellular rates were estimated based on bioreactor data meaning the final version of the flux distribution may change slightly. However, the values of 0.19 h^{-1} for the growth rate and 6.4 mmol g_X^{-1} h^{-1} for the ethanol uptake rate portrayed in the flux map turned out to be close to those later estimated using the microbioreactor online data amounting to 0.186 h^{-1} $\pm$ 0.016 h^{-1} and 7.95 mmol g_X^{-1} h^{-1} $\pm$ 0.677 mmol g_X^{-1} h^{-1}. Therefore, a significant shift of the pathway usage by replacing these rates would be unexpected.
All in all, the successful application of INST ^{13}C-MFA greatly expands the scope of possible experiments accommodated by the automated ILE workflow.

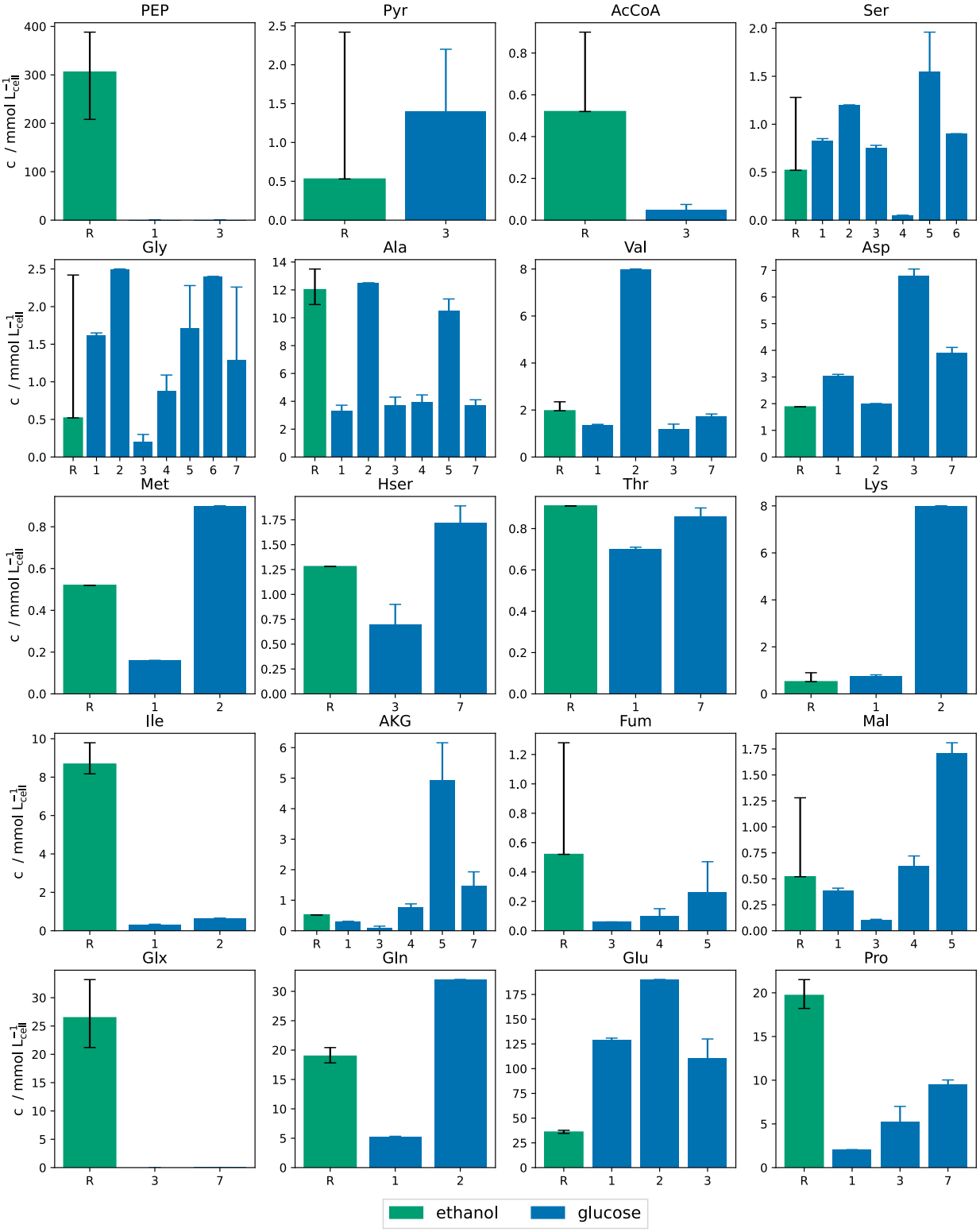


Figure 3.42: Metabolite pool sizes obtained by INST ^{13}C-MFA with *C. glutamicum* WT_ETH-evo grown on unlabeled ethanol and pulsed with 100 % 1-^{13}C ethanol during the mid-exponential growth phase. The present results with the pertaining 95 % CoI (R) are compared to literature values of *C. glutamicum* grown on Glc. Depending on the availability of data for a given metabolite, the references include [66] (1), [203] (2), [53] (3), [204] (4, cold methanol quenching data), [204] (5, rapid filtration data), [205] (6), and [17] (7).

3.7 Assembly and critical discussion of the overarching automated ILE workflow

The results of this dissertation detailed in the preceding sections all converge to form a new pipeline for ILEs presented in figure 3.43. From experimental work to data evaluation, numerous unit operations of this workflow have been successfully automated, thereby greatly increasing the potential throughput and contributing to standardization. The eponymous principles of automation, miniaturization, and parallelization were jointly applied to the experiments in particular and indeed led to the targeted reduction in cost and increase in throughput and walk-away time.
Walking through the automated ILE pipeline, the pre-experimental considerations remain mostly unchanged from the state of the art workflow. As the operation of automated platforms requires a Python script, this poses an additional task which was already automated as far as possible (see section 3.2), although customizing and testing the generated script are nonetheless required.
The experimental section was wholly changed. Transitioning from up to four parallel manually operated bioreactor cultivations at a liter-scale to up to 48 automated, parallelized, and miniaturized microbioreactor cultivations at a microliter-scale necessitated the move from cold methanol quenching to the newly established automated hot isopropanol quenching [142]. This procedure was rigorously validated (3.1.2), proven for INST ILEs (3.1.3), and applied to biological case studies (3.5 and 3.6). Since the sample processing in the form of centrifugation is included in the automated experiment, samples for detecting metabolite labeling states are immediately ready to use for LC-MS/MS analyses as e.g. no derivatization is required which would be the case for GC/MS analyses. Presupposing the justifiable assumption of a lack of oxygen limitation, it is even possible to increase the experimental throughput by sampling up to three times from the same well – depending on the available total volume. This is merely restricted by the need of an intermediary washing step of the LiHa pipettes taking roughly 30 s. Accordingly, the sampling time points need to be sufficiently spaced out to allow for that. Realistically, by coupling early and late sampling time points under consideration of this gap, it would be possible to decrease the number of necessary wells per INST ILE by half. Before utilizing this approach, it should be experimentally validated whether the removal of such a large fraction of the total volume does not impact cellular growth, though [209].
While the detection of metabolite labeling states is still conducted in an identical manner to the state of the art workflow (figure 1.2), the data processing up to the generation of TMIDs was altered significantly. Instead of relying on vendor software, the open source Python program `PeakPerformance` was developed in this thesis to perform peak fitting by Bayesian inference yielding peak parameters including uncertainty quantification. This innovative use of Bayesian statistics for chromatographic peak fitting proved not only advantageous with respect to automation, i.e. specifically to the degree of independence from user intervention, but also more accurate with regards to identifying signals as peaks. The most important enhancement, however, is the unprecedented quantification of measurement noise as a dedicated model parameter and the determination of uncertainty quantification for all parameters. Independent of whether `PeakPerformance` or the heretofore default of MultiQuant were employed, another Python program authored for this

workflow is used to calculate TMIDs, visualize the data, and structure it in a manner compatible to the correction for natural isotope abundance by the in-house developed meta-tool uNAC.
Regarding applications, it is self-evident that the simpler techniques such as quantitative labeling, isotopic profiling, and flux ratio analysis can be performed using this pipeline. All requirements for performing isotopically stationary ^{13}C-MFA such as measurement of extracellular substrate and product concentrations and obtaining labeling data during the ISS are fulfilled, as well. For particularly INST ^{13}C-MFA, this remained to be proven since hot isopropanol quenching does not accommodate the accurate measurement of metabolite pool sizes which are traditionally assumed to constitute essential data as the time course of label incorporation into a given metabolite pool is dependent on both the pertaining flux(es) and the pool size. Such proof was delivered by means of the biological use case of the first ever INST ^{13}C-MFA with a *C. glutamicum* strain on ethanol as the sole carbon source, as presented in section 3.6. To partially alleviate this drawback of hot isopropanol quenching and contribute an innovative way to gain insight into specific metabolite pool sizes using INST ^{13}C-labeling data, a new application was established for the estimation and uncertainty quantification of pool sizes pertaining to the amino acids Ser, Gly, Leu, Ala, and Val. This was realized by using a fully labeled substrate and evaluating all accrued data via a combination of a bioprocess model with a simplified ^{13}C-model (section 3.5). A parallelized Apache Airflow data pipeline was developed for this new application to enable its demanding simulations and make them accessible for less experienced users.
When critically discussing an attempt at an automated ILE pipeline and its merits, it is worthwhile to state a list of criteria that should ideally be fulfilled. According to a recent publication on the matter [210], such requirements would comprise the following items:

1. careful experimental design

This is of course a pre-requisite of experiments in general but ILEs in particular. Increasing the degree of automation for DoE was beyond the scope of this thesis but in order to increase the throughput even further, this point needs to be addressed.

2. parallelization of ^{13}C-ILEs

3. miniaturization of ^{13}C-ILEs

4. ability to maintain metabolic steady-state

5. permanent monitoring of growth

The aforementioned items were all fulfilled by relying on BioLector microbioreactor cultivations which facilitate conducting up to 48 batch experiments in parallel at a microliter-scale while online monitoring backscatter as a metric of growth. A (quasi-)metabolic steady-state can be assumed during the exponential growth phase of a batch experiment so this fundamental assumption of ^{13}C-MFA holds true.

6. rapid and automated sampling of labeled material

7. automated sample processing

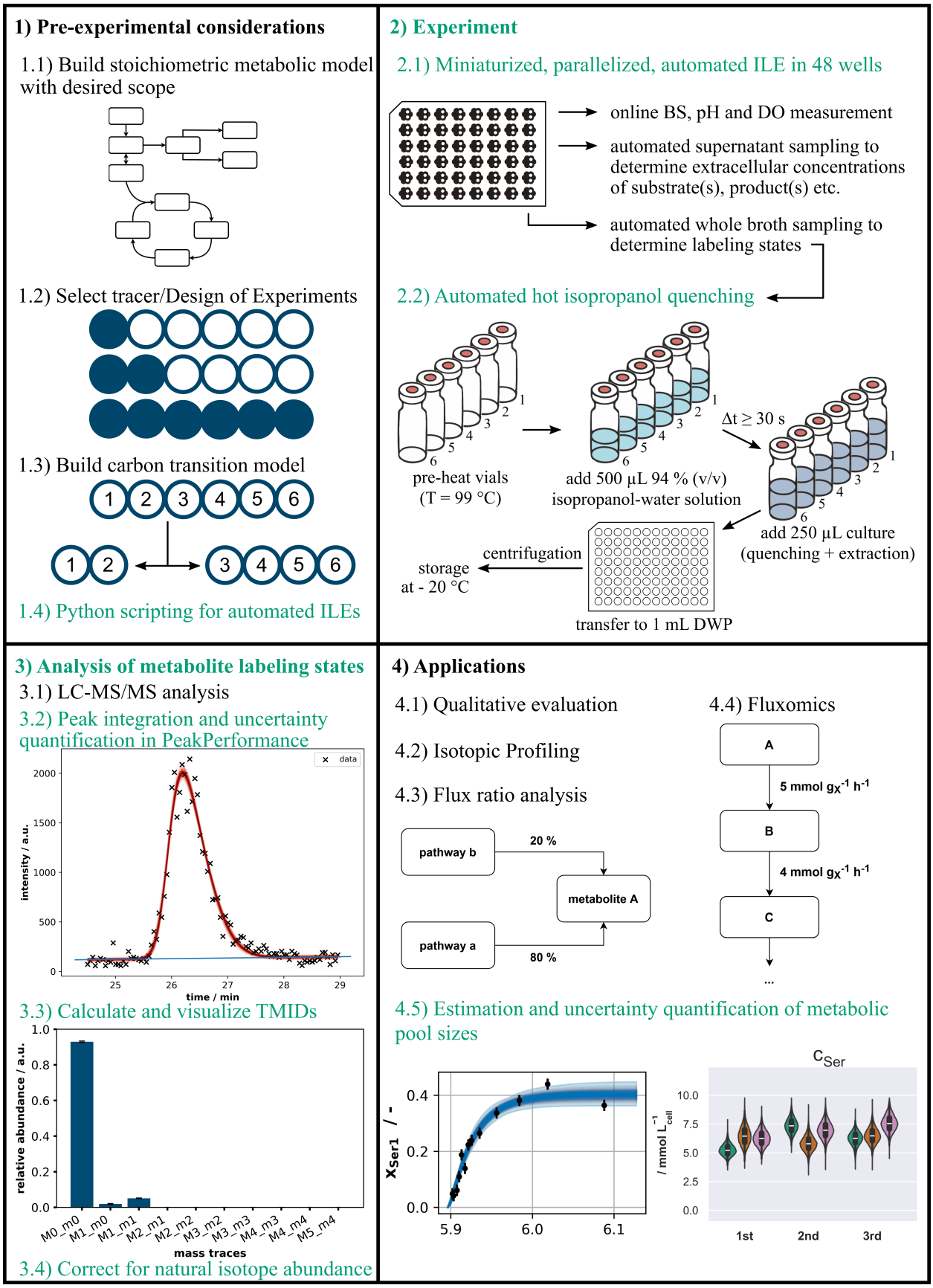


Figure 3.43: Automated ILE workflow as the culmination of the present thesis. This depiction invites a comparison to the state of the art ILE workflow shown in the introduction (figure 1.2). Due to the increased throughput and decreased costs enabling iterative experiments, the automated ILE workflow is presented as a cycle in contrast to the sequential nature of the state of the art workflow. Automated steps have been emphasized in green font color.

Since the robotic platforms utilized in this thesis were comprised of a liquid handling robot, a BioLector, and an automated centrifuge among other devices, samples were taken in an online data-dependent manner with subsequent quenching and removal of cells and cell debris by centrifugation. The resulting extracts were ready for LC-MS/MS analysis and potential dilutions before injection can be performed with a liquid handler.

8. fast and sensitive analytical technique due to low amount of biological material in samples

While the dilution inherent in automated isopropanol quenching is severe, the utilized LC-MS/MS method to detect amino acid labeling patterns is sufficiently sensitive for this purpose. Due to constraints with both available equipment and spacial limitations, an automated LC-MS/MS analysis directly following the sampling could not be realized in this thesis but this is not a technical but merely an infrastructural issue. In a high-throughput setting, however, a faster column method would be preferable and could diminish the allotted time for analysis by hours per FlowerPlate.

9. automated data extraction tool

Although there are avenues for further improvement given continuous development, `PeakPerformance` represents a significant step toward this goal. Even disregarding the innovations of taking measurement noise into account and performing uncertainty quantification by Bayesian statistics, the novel metrics for peak recognition drastically reduce the necessity for human supervision and especially manual action. Since the visual inspection and correction of peak data is cumbersome even at a low throughput, a major bottleneck was addressed with this new development.

10. more user-independent data interpretation tools

Depending on the application, this requires dedicated tools for a variety of purposes. Regarding the ones presented in this dissertation, `PeakPerformance` requires comparatively little human intervention once models and priors suited for a specific method have been established which tends to generally be the case for Bayesian methods. Furthermore, the pool size estimation workflow features a data pipeline which merely requires user input of the correct data assisted by a web interface guiding the user through this process. Naturally, when applying this technique to other organisms changes to the underlying model structure may be unavoidable. Additionally, issues with model convergence might occur and demand a user's attention. All things considered, however, this data pipeline is generally ready to use and offers access to computationally heavy MCMC simulation which the average user likely would not have otherwise.
If it is the goal to conduct high-throughput ^{13}C-MFA, a more user-independent solution remains to be created in future efforts.

11. integration of all steps into a pipeline

As the list shows, most but not all steps towards a comprehensive and integrated automated ILE pipeline were taken. The automated sections are without exception implemented in such a way that they connect to each other seamlessly. While this is obvious for the experimental work – the generated Python script leads to an experiment using quenching and yielding extracts

for LC-MS/MS analysis –, the later steps required active design decisions toward this end. For example, the peak integration with `PeakPerformance` returns an Excel file which can be interpreted by the software for data visualization and TMID calculation which in turn produces an Excel file containing correctly formatted results for natural isotope correction with uNAC. Merely such a level of connectivity between steps can increase the theoretically achievable throughput of the overall workflow significantly.

One factor setting apart ILEs from many other experiments is the traditionally high expenditure per experiment due to the cost of labeled substrate(s) which imposes a serious limitation on the possible specifications of a given experiment. Since this cost massively depends on the selected input labeling mixture, this burden center is uniquely susceptible to the choice of the experimenter. When performing DoE to rank different mixtures, the best result might include selectively labeled species which are commonly several times more expensive than their uniformly labeled counterparts. To demonstrate which dimensions this can take, the following hypothetical experimental costs for the automated and state of the art workflows are calculated. For the former, a filling volume of 800 µL per well, a pulse with labeled substrate of 1 % (v/v), and INST sampling at 8 time points in biological triplicates are assumed. For the latter, 1 L bioreactors are employed in biological duplicates and a 10 mL pulse with labeled substrate is allotted per replicate. Since this constitutes merely a rough estimate, the resulting numbers are intended only to give an impression of the orders of magnitude of the expected costs and between the two scenarios and should not be misinterpreted as certain. Due to the fluctuating prices of labeled substrates and the foreseeable increase in the costs of work and single use items, the absolute results are subject to continuous change.

$$c_{\text{automated}} \approx t_{\text{work}} r_h + c_{\text{substrate}} + c_{\text{FlowerPlate}} \quad = 10\ h\ 100\ \frac{€}{h} + 278.4\ € + 117\ € = \mathbf{1395.4}\ € \tag{3.27a}$$

$$c_{\text{manual}} \approx t_{\text{work}} r_h + c_{\text{substrate}} \quad = 20\ h\ 100\ \frac{€}{h} + 29000\ € = \mathbf{31000}\ € \tag{3.27b}$$

Clearly, having arrived at such a DoE result, the experimenter would have to reconsider and settle on the next best, economically viable alternative when using the state of the art workflow. The effects of the miniaturization inherent to the automated workflow, then, do not halt at merely causing a massive reduction in cost but functionally expand the economically feasible solution space of a DoE, thereby enabling usage of the best identified labeling mixtures and improving results.

Accordingly, as a consequence of the combined advantages of the automated workflow, namely the increased throughput at decreased costs, figure 3.43 presents the automated ILE workflow as a cycle in contradistinction to the sequence selected for the state of the art workflow (figure 1.2). This is meant to emphasize that an iterative design of ILEs is now realistically feasible and has indeed already been accomplished (see 3.6).

The preceding paragraphs were intended to clarify once again which of the limitations and drawbacks of the manual workflow have been addressed already and which new bottlenecks have thereby formed. The most pressing ones are now constituted by the modelling tasks, i.e. DoE and

data evaluation. While the analytical measurement of metabolite labeling states remains time-intensive and a faster method would need to be established to attain a high-throughput pipeline which can truly keep up with miniaturized cultivations, performing optimizations and statistical inferences for ^{13}C-MFA – especially INST – requires vastly more (simulation) time.

4 Outlook

When re-contextualizing the novel developments presented in this thesis and summarized in section 3.7 as the new status quo, one finds oneself at a bifurcation with regards to the future objective of the ILE workflow.

The development of `PeakPerformance` enables the possibility of realizing a data pipeline fully based on Bayesian statistics starting from peak analysis up to ^{13}C-MFA which has not been established previously. This would dramatically decrease the need for human intervention, at least after the necessary models have been built and validated. However, this approach effectively precludes a further increase in throughput due to the heavy computational load of the MCMC simulations employed for peak fitting and subsequent processes. It further increases the demand with regards to computing infrastructure as well as the energetic and thus monetary burden connected to such installations. Whereas the current ILE workflow is designed with connectivity between its sections in mind, these would have to be newly developed for a Bayesian variant and the complexity may increase if e.g. the distribution of the peak areas obtained by `PeakPerformance` are not normal-shaped since metrics such as the SSR require a mean value and standard deviation, i.e. a normally distributed error.

The alternative avenue is to further optimize the ILE workflow for increased throughput and automation, i.e. independence from the user, and aim to apply ^{13}C-MFA to a larger number of phenotyping experiments. With such a pipeline in place, high-throughput ^{13}C-MFA could even be performed as contract work for third parties – at least for some of the most important organisms –, not unlike the idea of a biofoundry.

Of course, a decision for or against one of these approaches is not necessarily unavoidable as these two visions can be pursued in separate follow-up projects. The immediate problems to tackle and actions required for the two, however, differ significantly.

Concluding this thesis, there obviously remains a large potential in ILEs in general and ^{13}C-MFA in particular. While considerable progress has been made in automation, experimentation, analytics, and simulation in the past 30 years, new applications of ILEs such as the estimation of metabolite pool sizes and novel variants of previously established techniques such as the automated, miniaturized INST ^{13}C-MFA can still be developed and improved upon. With the advent of autonomous experimentation and increasing independence from the user, the objectives of embedding ILEs in high-throughput workflows and expanding to routine application in industry are now more tenable than ever.

Literature

[1] “isotopes”. In: (2019). DOI: doi:10.1351/goldbook.I03331. URL: https://doi.org/10.1351/goldbook.I03331.

[2] J. Meija, T. B. Coplen, M. Berglund, W. A. Brand, P. De Bièvre, M. Gröning, N. E. Holden, J. Irrgeher, R. D. Loss, T. Walczyk, and T. Prohaska. “Isotopic compositions of the elements 2013 (IUPAC Technical Report)”. In: *Pure and Applied Chemistry* 88.3 (2016), pp. 293–306. DOI: doi:10.1515/pac-2015-0503. URL: https://doi.org/10.1515/pac-2015-0503.

[3] W. Wiechert, S. Niedenführ, and K. Nöh. “A Primer to ^{13}C Metabolic Flux Analysis”. In: *Fundamental Bioengineering*. Ed. by John Villadsen. Wiley-VCH Verlag GmbH & Co. KGaA, 2015. DOI: 10.1002/9783527697441.ch05.

[4] B. Schepaetz and S. Gurin. “The intermediary metabolism of phenylalanine labeled with radioactive carbon.” In: *Journal of Biological Chemistry* 180 (1949), pp. 663–673. ISSN: 1083-351X.

[5] R. E London. “^{13}C labeling in studies of metabolic regulation”. In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 20.4 (1988), pp. 337–383.

[6] W. Wiechert and A. A. De Graaf. “*In vivo* stationary flux analysis by ^{13}C labeling experiments”. In: *Metabolic Engineering* 54 (2006), pp. 109–154. DOI: 10.1007/BFb0102334.

[7] Y. Luo, H. Zang, Z. Yu, Z. Chen, A. Gunina, Y. Kuzyakov, J. Xu, K. Zhang, and P. C. Brookes. “Priming effects in biochar enriched soils using a three-source-partitioning approach: ^{14}C labelling and ^{13}C natural abundance”. In: *Soil Biology and Biochemistry* 106 (2017), pp. 28–35.

[8] M. Shahbaz, A. Kumar, Y. Kuzyakov, G. Börjesson, and E. Blagodatskaya. “Priming effects induced by glucose and decaying plant residues on SOM decomposition: a three-source ^{13}C/^{14}C partitioning study”. In: *Soil Biology and Biochemistry* 121 (2018), pp. 138–146.

[9] I. I. Stewart, T. Thomson, and D. Figeys. “^{18}O labeling: a tool for proteomics”. In: *Rapid Communications in Mass Spectrometry* 15.24 (2001), pp. 2456–2465.

[10] R. L. Westerman and L. T. Kurtz. “Priming effect of ^{15}N-labeled fertilizers on soil nitrogen in field experiments”. In: *Soil Science Society of America Journal* 37.5 (1973), pp. 725–727.

[11] M. Holtappels, G. Lavik, M. M. Jensen, and M. M. M. Kuypers. “^{15}N-labeling experiments to dissect the contributions of heterotrophic denitrification and anammox to nitrogen removal in the OMZ waters of the ocean”. In: *Methods in Enzymology*. Vol. 486. Elsevier, 2011, pp. 223–251.

[12] J. Kappelmann, M. Beyß, K. Nöh, and S. Noack. “Separation of ^{13}C- and ^{15}N-Isotopologues of Amino Acids with a Primary Amine without Mass Resolution by Means of O-Phthalaldehyde Derivatization and Collision Induced Dissociation”. In: *Anal Chem* 91.21 (2019), pp. 13407–13417. ISSN: 1520-6882 (Electronic) 0003-2700 (Linking). DOI: 10.1021/acs.analchem.9b01788. URL: https://www.ncbi.nlm.nih.gov/pubmed/31577133.

[13] U. Squin and A. I. Scott. “Carbon-13 as a Label in Biosynthetic Studies: Carbon-13 nuclear magnetic resonance spectroscopy is useful for the elucidation of biosynthetic pathways.” In: *Science* 186.4159 (1974), pp. 101–107.

[14] K. Sonntag, L. Eggeling, A. A. de Graaf, and H. Sahm. “Flux partitioning in the split pathway of lysine synthesis in *Corynebacterium glutamicum*: Quantification by ^{13}C-and ^{1}H-NMR spectroscopy”. In: *European Journal of Biochemistry* 213.3 (1993), pp. 1325–1331.

[15] A. Marx, A. A. de Graaf, and W. Wiechert. “Determination of the Fluxes in the Central Metabolism of *Corynebacterium glutamicum* by Nuclear Magnetic Resonance Spectroscopy Combined with Metabolite Balancing”. In: *Biotechnol Bioeng* 49 (1996).

[16] N. Kallscheuer, M. Vogt, J. Kappelmann, K. Krumbach, S. Noack, M. Bott, and J. Marienhagen. “Identification of the *phd* gene cluster responsible for phenylpropanoid utilization in *Corynebacterium glutamicum*”. In: *Applied Microbiology and Biotechnology* 100 (2016), pp. 1871–1881.

[17] J. Tillack, N. Paczia, K. Nöh, W. Wiechert, and S. Noack. “Error propagation analysis for quantitative intracellular metabolomics”. In: *Metabolites* 2.4 (2012), pp. 1012–30. ISSN: 2218-1989 (Print) 2218-1989 (Electronic) 2218-1989 (Linking). DOI: 10.3390/metabo2041012. URL: https://www.ncbi.nlm.nih.gov/pubmed/24957773.

[18] S. Noack and W. Wiechert. “Quantitative metabolomics: a phantom?” In: *Trends Biotechnol* 32.5 (2014), pp. 238–44. ISSN: 1879-3096 (Electronic) 0167-7799 (Linking). DOI: 10.1016/j.tibtech.2014.03.006. URL: https://www.ncbi.nlm.nih.gov/pubmed/24708998.

[19] N. Paczia, A. Nilgen, T. Lehmann, J. Gätgens, W. Wiechert, and S. Noack. “Extensive exometabolome analysis reveals extended overflow metabolism in various microorganisms.” In: *Microbial Cell Factories* 11 (2012).

[20] C. Wittmann, J. O. Krömer, P. Kiefer, T. Binz, and E. Heinzle. “Impact of the cold shock phenomenon on quantification of intracellular metabolites in bacteria”. In: *Analytical Biochemistry* 327.1 (2004), pp. 135–139.

[21] Q. Zhang, X. Zheng, Y. Wang, J. Yu, Z. Zhang, T. Dele-Osibanjo, P. Zheng, J. Sun, S. Jia, and Y. Ma. “Comprehensive optimization of the metabolomic methodology for metabolite profiling of *Corynebacterium glutamicum*”. In: *Appl Microbiol Biotechnol* 102.16 (2018), pp. 7113–7121. ISSN: 1432-0614 (Electronic) 0175-7598 (Linking). DOI: 10.1007/s00253-018-9095-1. URL: https://www.ncbi.nlm.nih.gov/pubmed/29876603.

[22] *MultiQuant*. Version 3.0.3. 2017.

[23] P. Millard, B. Delépine, M. Guionnet, M. Heuillet, F. Bellvert, and F. Létisse. “IsoCor: isotope correction for high-resolution MS labeling experiments”. In: *Bioinformatics* 35.21 (2019), pp. 4484–4487.

[24] P. Millard, B. Delépine, M. Guionnet, M. Heuillet, F. Bellvert, and F. Létisse. *IsoCor*. URL: https://github.com/MetaSys-LISBP/IsoCor/.

[25] C. Jungreuthmayer, S. Neubauer, T. Mairinger, J. Zanghellini, and S. Hann. “ICT: isotope correction toolbox”. In: *Bioinformatics* 32.1 (2016), pp. 154–156.

[26] D-*Glucose-$^{13}C_6$ (CAS 110187-42-3)*. https://www.scbt.com/de/p/d-glucose-13c6-110187-42-3. Web Page. Accessed: 22.06.2024.

[27] D-*Glucose-1,2-$^{13}C_2$*. https://www.sigmaaldrich.com/DE/de/product/aldrich/453188. Web Page. Accessed: 22.06.2024.

[28] S. Heux, J. Poinot, S. Massou, S. Sokol, and J. C. Portais. "A novel platform for automated high-throughput fluxome profiling of metabolic variants". In: *Metabolic Engineering* 25 (2014), pp. 8–19. ISSN: 1096-7184 (Electronic) 1096-7176 (Linking). DOI: 10.1016/j.ymben.2014.06.001. URL: https://www.ncbi.nlm.nih.gov/pubmed/24930895.

[29] B. E. Ebert and L. M. Blank. "Successful downsizing for high-throughput ^{13}C-MFA applications". In: *Methods Mol Biol* 1191 (2014), pp. 127–42. ISSN: 1940-6029 (Electronic) 1064-3745 (Linking). DOI: 10.1007/978-1-4939-1170-7_8. URL: https://www.ncbi.nlm.nih.gov/pubmed/25178788.

[30] A. Klingner, A. Bartsch, M. Dogs, I. Wagner-Döbler, D. Jahn, M. Simon, T. Brinkhoff, J. Becker, and C. Wittmann. "Large-Scale ^{13}C Flux Profiling Reveals Conservation of the Entner-Doudoroff Pathway as a Glycolytic Strategy among Marine Bacteria That Use Glucose". In: *Applied and Environmental Microbiology* 81 (2015).

[31] S. Leonelli and R. A. Ankeny. "What makes a model organism?" In: *Endeavour* 37.4 (2013), pp. 209–212. ISSN: 0160-9327.

[32] B. Müller and U. Grossniklaus. "Model organisms — a historical perspective". In: *Journal of Proteomics* 73.11 (2010), pp. 2054–2063. ISSN: 1874-3919.

[33] T. Hirasawa and H. Shimizu. "Glutamic acid fermentation: discovery of glutamic acid-producing microorganisms, analysis of the production mechanism, metabolic engineering, and industrial production process". In: *Industrial Biotechnology: Products and Processes*. Ed. by C. Wittmann and James C. Liao. Wiley-VCH Verlag GmbH & Co. KGaA, 2017. Chap. 11 - Glutamic Acid Fermentation, pp. 339–360.

[34] S. Kinoshita, S. Udaka, and M. Shimono. "Studies on the amino acid fermentation. Part I. Production of L-glutamic acid by various microorganisms". In: *J Gen Appl Microbiol* 3 (1957), pp. 193–205.

[35] Y. Izumi, I. Chibata, and T. Itoh. "Production and utilization of amino acids". In: *Angewandte Chemie International Edition in English* 17.3 (1978), pp. 176–183. ISSN: 0570-0833.

[36] J.-Y. Lee, Y.-A. Na, E. Kim, H.-S. Lee, and P. Kim. "The actinobacterium *Corynebacterium glutamicum*, an industrial workhorse". In: 26.5 (2016), pp. 807–822.

[37] V. F. Wendisch, J. M. P. Jorge, F. Pérez-García, and E. Sgobba. "Updates on industrial production of amino acids using *Corynebacterium glutamicum*". In: *World Journal of Microbiology and Biotechnology* 32 (2016), pp. 1–10. ISSN: 0959-3993.

[38] Precedence Research. *Glutamic Acid Market (By Application: Food & Beverages, Pharmaceuticals, Animal Feed, Others) - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2023-2032*. https://www.precedenceresearch.com/glutamic-acid-market. Published: Aug 2023, Accessed: 24.06.2024.

[39] Y. Tsuge and H. Matsuzawa. “Recent progress in production of amino acid-derived chemicals using *Corynebacterium glutamicum*”. In: *World Journal of Microbiology and Biotechnology* 37 (2021), pp. 1–13. ISSN: 0959-3993.

[40] J. Schneider and V. F. Wendisch. “Putrescine production by engineered *Corynebacterium glutamicum*”. In: *Applied Microbiology and Biotechnology* 88 (2010), pp. 859–868.

[41] S. Kind, W. K. Jeong, H. Schröder, O. Zelder, and C. Wittmann. “Identification and elimination of the competing N-acetyldiaminopentane pathway for improved production of diaminopentane by *Corynebacterium glutamicum*”. In: *Applied and Environmental Microbiology* 76.15 (2010), pp. 5175–5180.

[42] J. H. Shin, S. H. Park, Y. H. Oh, J. W. Choi, M. H. Lee, J. S. Cho, K. J. Jeong, C. J. Jeong, J. Yu, S. J. Park, and S. Y. Lee. “Metabolic engineering of Corynebacterium glutamicum for enhanced production of 5-aminovaleric acid”. In: *Microbial Cell Factories* 15 (2016), pp. 1–13.

[43] B. Blombach, T. Riester, S. Wieschalka, C. Ziert, J.-W. Youn, V. F. Wendisch, and B. J. Eikmanns. “*Corynebacterium glutamicum* tailored for efficient isobutanol production”. In: *Applied and Environmental Microbiology* 77.10 (2011), pp. 3300–3310.

[44] N. Kallscheuer and J. Marienhagen. “*Corynebacterium glutamicum* as platform for the production of hydroxybenzoic acids”. In: *Microbial Cell Factories* 17 (2018), pp. 1–13.

[45] J. Kalinowski, B. Bathe, D. Bartels, N. Bischoff, M. Bott, A. Burkovski, N. Dusch, L. Eggeling, B. J. Eikmanns, L. Gaigalat, A. Goesmann, M. Hartmann, K. Huthmacher, R. Krämer, B. Linke, A. C. McHardy, F. Meyer, B. Möckel, W. Pfefferle, A. Pühler, D. A. Rey, C. Rückert, O. Rupp, H. Sahm, V. F. Wendisch, I. Wiegäbe, and A. Tauch. “The complete *Corynebacterium glutamicum* ATCC 13032 genome sequence and its impact on the production of L-aspartate-derived amino acids and vitamins”. In: *Journal of Biotechnology* 104.1-3 (2003), pp. 5–25.

[46] M. Ikeda and S. Nakagawa. “The *Corynebacterium glutamicum* genome: features and impacts on biotechnological processes”. In: *Applied Microbiology and Biotechnology* 62 (2003), pp. 99–109.

[47] V. F. Wendisch, M. Bott, J. Kalinowski, M. Oldiges, and W. Wiechert. “Emerging *Corynebacterium glutamicum* systems biology”. In: *Journal of Biotechnology* 124.1 (2006), pp. 74–92. ISSN: 0168-1656.

[48] K. R. Kjeldsen and J. Nielsen. “In silico genome-scale reconstruction and validation of the *Corynebacterium glutamicum* metabolic network”. In: *Biotechnol Bioeng* 102.2 (2009), pp. 583–97. ISSN: 1097-0290 (Electronic) 0006-3592 (Linking). DOI: 10.1002/bit.22067. URL: https://www.ncbi.nlm.nih.gov/pubmed/18985611.

[49] H. Rönsch, R. Krämer, and S. Morbach. “Impact of osmotic stress on volume regulation, cytoplasmic solute composition and lysine production in *Corynebacterium glutamicum* MH20-22B”. In: *J Biotechnol* 104.1-3 (2003), pp. 87–97. ISSN: 0168-1656 (Print) 0168-1656 (Link-

ing). DOI: 10.1016/s0168-1656(03)00166-4. URL: https://www.ncbi.nlm.nih.gov/pubmed/12948632.

[50] C. Wittmann and A. A. De Graaf. “Metabolic Flux Analysis in *Corynebacterium glutamicum*”. In: *Handbook of Corynebacterium glutamicum*. Ed. by L. Eggeling and M. Bott. Taylor & Francis Group, 2005. Chap. 12, pp. 277–304.

[51] A. Yokota and N. D. Lindley. “Central Metabolism: Sugar Uptake and Conversion”. In: *Handbook of Corynebacterium glutamicum*. Ed. by L. Eggeling and M. Bott. Taylor & Francis Group, 2005. Chap. 10, pp. 215–240.

[52] R. Gerstmeir, V. F. Wendisch, S. Schnicke, H. Ruan, M. Farwick, D. Reinscheid, and B. J. Eikmanns. “Acetate metabolism and its regulation in *Corynebacterium glutamicum*”. In: *J Biotechnol* 104.1-3 (2003), pp. 99–122. ISSN: 0168-1656 (Print) 0168-1656 (Linking). DOI: 10.1016/s0168-1656(03)00167-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/12948633.

[53] J. Kappelmann. “Tandem-Mass-Spectrometry-Driven Investigation of the Anaplerotic Reactions in *Corynebacterium glutamicum*”. Dissertation. 2018. DOI: 10.18154/RWTH-2018-231115. URL: https://publications.rwth-aachen.de/record/751130/files/751130.pdf.

[54] U. Sauer and B. J. Eikmanns. “The PEP—pyruvate—oxaloacetate node as the switch point for carbon flux distribution in bacteria: We dedicate this paper to Rudolf K. Thauer, Director of the Max-Planck-Institute for Terrestrial Microbiology in Marburg, Germany, on the occasion of his 65th birthday”. In: *FEMS Microbiology Reviews* 29.4 (2005), pp. 765–794.

[55] J. Kappelmann, B. Klein, M. Papenfuss, J. Lange, B. Blombach, R. Takors, W. Wiechert, T. Polen, and S. Noack. “Comprehensive Analysis of *C. glutamicum* Anaplerotic Deletion Mutants Under Defined D-Glucose Conditions”. In: *Front Bioeng Biotechnol* 8 (2020), p. 602936. ISSN: 2296-4185 (Print) 2296-4185 (Linking). DOI: 10.3389/fbioe.2020.602936. URL: https://www.ncbi.nlm.nih.gov/pubmed/33553115.

[56] P. G. Peters-Wendisch, C. Kreutzer, J. Kalinowski, M. Pátek, H. Sahm, and B. J. Eikmanns. “Pyruvate carboxylase from *Corynebacterium glutamicum*: characterization, expression and inactivation of the *pyc* gene”. In: *Microbiology* 144 (1998), pp. 915–927.

[57] J. Kappelmann, W. Wiechert, and S. Noack. “Cutting the Gordian Knot: Identifiability of anaplerotic reactions in *Corynebacterium glutamicum* by means of ^{13}C-metabolic flux analysis”. In: *Biotechnol Bioeng* 113.3 (2016), pp. 661–74. ISSN: 1097-0290 (Electronic) 0006-3592 (Linking). DOI: 10.1002/bit.25833. URL: https://www.ncbi.nlm.nih.gov/pubmed/26375179.

[58] M. Mori and I. Shiio. “Pyruvate formation and sugar metabolism in an amino acid-producing bacterium, *Brevibacterium flavum*”. In: *Agricultural and Biological Chemistry* 51.1 (1987), pp. 129–138. ISSN: 0002-1369.

[59] M. Ikeda. “Sugar transport systems in *Corynebacterium glutamicum*: features and applications to strain development”. In: *Applied Microbiology and Biotechnology* 96 (2012), pp. 1191–1200. ISSN: 0175-7598.

[60] M. Cocaign, C. Monnet, and N. D. Lindley. “Batch kinetics of *Corynebacterium glutamicum* during growth on various carbon substrates: use of substrate mixtures to localise metabolic bottlenecks”. In: *Applied Microbiology and Biotechnology* 40 (1993), pp. 526–530. ISSN: 0175-7598.

[61] H. Dominguez, M. Cocaign-Bousquet, and N. D. Lindley. “Simultaneous consumption of glucose and fructose from sugar mixtures during batch growth of *Corynebacterium glutamicum*”. In: *Applied Microbiology and Biotechnology* 47 (1997), pp. 600–603. ISSN: 0175-7598.

[62] A. Arndt, M. Auchter, T. Ishige, V. F. Wendisch, and B. J. Eikmanns. “Ethanol catabolism in *Corynebacterium glutamicum*”. In: *J Mol Microbiol Biotechnol* 15.4 (2008), pp. 222–33. ISSN: 1660-2412 (Electronic) 1464-1801 (Linking). DOI: 10.1159/000107370. URL: https://www.ncbi.nlm.nih.gov/pubmed/17693703.

[63] D. J. Reinscheid, S. Schnicke, D. Rittmann, U. Zahnow, H. Sahm, and B. J. Eikmanns. “Cloning, sequence analysis, expression and inactivation of the *Corynebacterium glutamicum* pta-ack operon encoding phosphotransacetylase and acetate kinase”. In: *Microbiology* 145.2 (1999), pp. 503–513.

[64] V. F. Wendisch, A. A. De Graaf, H. Sahm, and B. J. Eikmanns. “Quantitative Determination of Metabolic Fluxes during Coutilization of two carbon sources: Comparative Analyses with *Corynebacterium glutamicum* during Growth on Acetate and/or Glucose”. In: *Journal of Bacteriology* 182.11 (2000), pp. 3088–3096.

[65] B. Eikmanns. “Central Metabolism: Tricarboxylic Acid Cycle and Anaplerotic Reactions”. In: *Handbook of Corynebacterium glutamicum*. Ed. by L. Eggeling and M. Bott. Taylor & Francis Group, 2005. Chap. 11, pp. 241–276.

[66] M. Graf, T. Haas, A. Teleki, A. Feith, M. Cerff, W. Wiechert, K. Nöh, T. Busche, J. Kalinowski, and R. Takors. “Revisiting the Growth Modulon of *Corynebacterium glutamicum* Under Glucose Limited Chemostat Conditions”. In: *Front Bioeng Biotechnol* 8 (2020), p. 584614. ISSN: 2296-4185 (Print) 2296-4185 (Electronic) 2296-4185 (Linking). DOI: 10.3389/fbioe.2020.584614. URL: https://www.ncbi.nlm.nih.gov/pubmed/33178676.

[67] D. J. Reinscheid, B. J. Eikmanns, and H. Sahm. “Malate synthase from *Corynebacterium glutamicum*: sequence analysis of the gene and biochemical characterization of the enzyme”. In: *Microbiology* 140.11 (1994), pp. 3099–3108. ISSN: 1350-0872.

[68] H. L. Kornberg and H. A. Krebs. “Synthesis of cell constituents from C_2-units by a modified tricarboxylic acid cycle”. In: *Nature* 179.4568 (1957), pp. 988–991.

[69] H. L. Kornberg. “The role and control of the glyoxylate cycle in *Escherichia coli*.” In: *Biochemical Journal* 99.1 (1966), p. 1.

[70] D. P. Clark and J. E. Cronan. “Two-carbon compounds and fatty acids as carbon sources”. In: *EcoSal Plus* 1.2 (2005), pp. 10–1128.

[71] M. Umakoshi, T. Hirasawa, C. Furusawa, Y. Takenaka, Y. Kikuchi, and H. Shimizu. “Improving protein secretion of a transglutaminase-secreting *Corynebacterium glutamicum* recombinant strain on the basis of ^{13}C metabolic flux analysis”. In: *Journal of Bioscience and Bioengineering* 112.6 (2011), pp. 595–601. ISSN: 1389-1723.

[72] A. Reiter, L. Herbst, W. Wiechert, and M. Oldiges. “Need for speed: evaluation of dilute and shoot-mass spectrometry for accelerated metabolic phenotyping in bioprocess development”. In: *Anal Bioanal Chem* 413.12 (2021), pp. 3253–3268. ISSN: 1618-2650 (Electronic) 1618-2642 (Linking). DOI: 10.1007/s00216-021-03261-3. URL: https://www.ncbi.nlm.nih.gov/pubmed/33791825.

[73] J. J. van Deemter, F. J. Zuiderweg, and A. Klinkenberg. “Longitudinal diffusion and resistance to mass transfer as causes of nonideality in chromatography”. In: *Chemical Engineering Science* 5 (1956), pp. 271–289.

[74] I. Langmuir. “The constitution and fundamental properties of solids and liquids. Part I. Solids”. In: *J. Am. Chem. Soc* 38.11 (1916), pp. 2221–2295.

[75] H. Schmidt-Traub, ed. *Preparative Chromatography of Fine Chemicals and Pharmaceutical Agents*. WILEY-VCH Verlag GmbH & Co. KGaA, 2005. ISBN: 3-527-30643-9.

[76] S. Banerjee and S. Mazumdar. “Electrospray ionization mass spectrometry: a technique to access the information beyond the molecular weight of the analyte”. In: *Int J Anal Chem* 2012 (2012). ISSN: 1687-8779 (Electronic) 1687-8760 (Print) 1687-8760 (Linking). DOI: 10.1155/2012/282574. URL: https://www.ncbi.nlm.nih.gov/pubmed/22611397.

[77] P. Kebarle and L. Tang. “From ions in solution to ions in the gas phase. The mechanism of electrospray mass spectrometry”. In: *Anal Chem* 65 (1993), pp. 972–986. DOI: 10.1002/mas.20247.

[78] A. J. Dempster. “A new Method of Positive Ray Analysis”. In: *Physical Review* 11.4 (1918), pp. 316–325. ISSN: 0031-899X. DOI: 10.1103/PhysRev.11.316.

[79] V. F. Lima, A. Erban, A. G. Daubermann, F. B. S. Freire, N. P. Porto, S. A. Candido-Sobrinho, D. B. Medeiros, M. Schwarzlander, A. R. Fernie, L. Dos Anjos, J. Kopka, and D. M. Daloso. “Establishment of a GC-MS-based ^{13}C-positional isotopomer approach suitable for investigating metabolic fluxes in plant primary metabolism”. In: *Plant J* (2021). ISSN: 1365-313X (Electronic) 0960-7412 (Linking). DOI: 10.1111/tpj.15484. URL: https://www.ncbi.nlm.nih.gov/pubmed/34486764.

[80] C. P. Long, J. Au, J. E. Gonzalez, and M. R. Antoniewicz. “^{13}C metabolic flux analysis of microbial and mammalian systems is enhanced with GC-MS measurements of glycogen and RNA labeling”. In: *Metabolic Engineering* 38 (2016), pp. 65–72. ISSN: 1096-7184 (Electronic) 1096-7176 (Print) 1096-7176 (Linking). DOI: 10.1016/j.ymben.2016.06.007. URL: https://www.ncbi.nlm.nih.gov/pubmed/27343680.

[81] T. D. Mark. “Fundamental aspects of electron impact ionization”. In: *International Journal of Mass Spectrometry and Ion Physics* 45 (1982), pp. 125–145. ISSN: 0020-7381.

[82] L. Konermann, E. Ahadi, A. D. Rodriguez, and S. Vahidi. “Unraveling the mechanism of electrospray ionization”. In: *Anal Chem* 85.1 (2013), pp. 2–9. ISSN: 1520-6882 (Electronic) 0003-2700 (Linking). DOI: 10.1021/ac302789c. URL: https://www.ncbi.nlm.nih.gov/pubmed/23134552.

[83] M. Wilm. “Principles of electrospray ionization”. In: *Mol Cell Proteomics* 10.7 (2011). ISSN: 1535-9484 (Electronic) 1535-9476 (Linking). DOI: 10.1074/mcp.M111.009407. URL: https://www.ncbi.nlm.nih.gov/pubmed/21742801.

[84] M. Yamashita and J. B. Fenn. “Negative Ion Production with the Electrospray Ion Source”. In: *J Phys Chem* 88 (1984), pp. 4671–4675.

[85] M. Yamashita and J. B. Fenn. In: *J Phys Chem* 88 (1984).

[86] A. G. Bailey. “The science and technology of electrostatic powder spraying, transport and coating”. In: *Journal of Electrostatics* 45 (1998), pp. 85–120.

[87] P Kebarle and U. H. Verkerk. “On the Mechanism of Electrospray Ionization Mass Spectrometry (ESIMS)”. In: *Electrospray and MALDI mass spectrometry: fundamentals, instrumentation, practicalities, and biological applications*. Ed. by Richard B Cole. John Wiley & Sons, 2011. Chap. 1. DOI: 10.1002/9780470588901.

[88] G. Taylor. “Disintegration of water drops in an electric field”. In: *Proc R Soc London A* 280.1382 (1964), pp. 383–397.

[89] Lord Rayleigh. In: *Phil Mag* 14.184 (1882).

[90] E. J. Davis and M. A. Bridges. “The Rayleigh limit of charge revisited. Light scattering from exploding droplets”. In: *Journal of Aerosol Science* 25.6 (1994), pp. 1179–1199.

[91] J. V. Iribarne and B. A. Thomson. “On the evaporation of small ions from charged droplets”. In: *The Journal of Chemical Physics* 64.6 (1976), pp. 2287–2294. ISSN: 0021-9606.

[92] J. P. Savaryn, T. K. Toby, and N. L. Kelleher. “A researcher’s guide to mass spectrometry-based proteomics”. In: *Proteomics* 16.18 (2016), pp. 2435–43. ISSN: 1615-9861 (Electronic) 1615-9853 (Print) 1615-9853 (Linking). DOI: 10.1002/pmic.201600113. URL: https://www.ncbi.nlm.nih.gov/pubmed/27553853.

[93] P. E. Miller and M. B. Denton. “The quadrupole mass filter: basic operating concepts”. In: *Journal of Chemical Education* 63.7 (1986), p. 617. ISSN: 0021-9584.

[94] C.S. Ho, C.W.K. Lam, M.H.M. Chan, R.C.K. Cheung, L.K. Law, L.C.W Lit, K.F. Ng, M.W.M. Suen, and H.L. Tai. “Electrospray Ionisation Mass Spectrometry: Principles and Clinical Applications”. In: *Clin Biochem Rev* 24 (2003).

[95] A. E. Cameron and D. F. Eggers. “An Ion "Velocitron"”. In: *Review of Scientific Instruments* 19.9 (1948), pp. 605–607. ISSN: 0034-6748 1089-7623. DOI: 10.1063/1.1741336.

[96] M. Balcerzak. “An Overview of Analytical Applications of Time of Flight-Mass Spectrometric (TOF-MS) Analyzers and an Inductively Coupled Plasma-TOF-MS Technique”. In: *Analytical Sciences* 19 (2003), pp. 979–989.

[97] M. Guilhaus. “Special feature: Tutorial. Principles and instrumentation in time-of-flight mass spectrometry. Physical and instrumental concepts”. In: *Journal of Mass Spectrometry* 30.11 (2005), pp. 1519–1532. ISSN: 1076-5174 1096-9888. DOI: 10.1002/jms.1190301102.

[98] B. A. Mamyrin. “Time-of-flight mass spectrometry (concepts, achievements, and prospects)”. In: *International Journal of Mass Spectrometry* 206 (2001), pp. 251–266.

[99] B. A. Mamyrin and D. V. Shmikk. “The linear mass reflectron”. In: *Sov Phys JETP* 49 (1979), pp. 762–764.

[100] W. Wiechert, M. Möllney, N. Isermann, M. Wurzel, and A. A. de Graaf. “Bidirectional reaction steps in metabolic networks: III. Explicit solution and analysis of isotopomer labeling systems”. In: *Biotechnology and Bioengineering* 66.2 (1999), pp. 69–85. ISSN: 0006-3592.

[101] S. Noack, R. Voges, J. Gatgens, and W. Wiechert. “The linkage between nutrient supply, intracellular enzyme abundances and bacterial growth: New evidences from the central carbon metabolism of *Corynebacterium glutamicum*”. In: *J Biotechnol* 258 (2017), pp. 13–24. ISSN: 1873-4863 (Electronic) 0168-1656 (Linking). DOI: 10.1016/j.jbiotec.2017.06.407. URL: https://www.ncbi.nlm.nih.gov/pubmed/28647528.

[102] J. Monod. “The growth of bacterial cultures”. In: *Annu. Rev. Microbiol.* (1949).

[103] J. Hemmerich, N. Tenhaef, W. Wiechert, and S. Noack. “pyFOOMB: Python framework for object oriented modeling of bioprocesses”. In: *Eng Life Sci* 21.3-4 (2021), pp. 242–257. ISSN: 1618-0240 (Print) 1618-0240 (Linking). DOI: 10.1002/elsc.202000088. URL: https://www.ncbi.nlm.nih.gov/pubmed/33716622.

[104] P. Fritzson, A. Pop, K. Abdelhak, A. Asghar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. D. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, L. Ochel, A. Palanisamy, V. Ruge, W. Schamai, M. Sjölund, B. Thiele, J. Tinnerholm, and P. Östlund. “The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development”. In: *Modeling, Identification and Control* 41.4 (2020), pp. 241–295. DOI: 10.4173/mic.2020.4.1.

[105] P. Fritzson, A. Pop, K. Abdelhak, A. Asghar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. D. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, L. Ochel, A. Palanisamy, V. Ruge, W. Schamai, M. Sjölund, B. Thiele, J. Tinnerholm, and P. Östlund. “The OpenModelica integrated environment for modeling, simulation, and model-based development”. In: Mic. 2022.

[106] D. Strohmeier and T. Latour. *estim8*. URL: https://pypi.org/project/estim8/.

[107] O. A. Martin. *Bayesian analysis with Python - Introduction to statistical modeling and probabilistic programming using PyMC3 and ArviZ (2nd edition)*. 2018.

[108] E.-J. Wagenmakers, M. Lee, T. Lodewyckx, and G. J. Iverson. “Bayesian versus Frequentist Inference”. In: *Bayesian evaluation of informative hypotheses*. Springer, New York, NY, 2008, pp. 181–207. DOI: https://doi.org/10.1007/978-0-387-09612-4_9.

[109] D. J. Venzon and S. H. Moolgavkar. “A Method for Computing Profile-Likelihood-Based Confidence Intervals”. In: *Applied Statistics* 37.1 (1988), pp. 87–94.

[110] A. Theorell, S. Leweke, W. Wiechert, and K. Nöh. “To be certain about the uncertainty: Bayesian statistics for ^{13}C metabolic flux analysis”. In: *Biotechnol Bioeng* 114.11 (2017), pp. 2668–2684. ISSN: 1097-0290 (Electronic) 0006-3592 (Linking). DOI: 10.1002/bit.26379. URL: https://www.ncbi.nlm.nih.gov/pubmed/28695999.

[111] O. A. Martin, R. Kumar, and J. Lao. “Bayesian Modeling and Computation in Python”. In: 2021. Chap. 1 - Bayesian Inference, pp. 1–30. ISBN: 9781003019169. DOI: 10.1201/9781003019169-1.

[112] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, and C. Yau. “Bayesian statistics and modelling”. In: *Nature Reviews Methods Primers* 1.1 (2021). ISSN: 2662-8449. DOI: 10.1038/s43586-020-00001-2.

[113] J. K. Kruschke. “Doing Bayesian Data Analysis (Second Edition): A Tutorial with R, JAGS, and Stan”. In: 2015. Chap. 5 - Bayes’ Rule, pp. 99–120. ISBN: 9780124058880. DOI: 10.1016/b978-0-12-405888-0.00005-2.

[114] G. E. P. Box. “Sampling and Bayes’ Inference in Scientific Modelling and Robustness”. In: *Royal Statistical Society. Journal. Series A: General* 143.4 (1980). DOI: https://doi.org/10.2307/2982063.

[115] A. Gelman. “A Bayesian Formulation of Exploratory Data Analysis and Goodness-of-fit Testing”. In: *International Statistical Review* 71.2 (2007), pp. 369–382. ISSN: 0306-7734 1751-5823. DOI: 10.1111/j.1751-5823.2003.tb00203.x.

[116] Erlis Ruli, Nicola Sartori, and Laura Ventura. “Marginal posterior simulation via higher-order tail area approximations”. In: *Bayesian Analysis* 9.1 (2014), pp. 129–146. DOI: 10.1214/13-BA851.

[117] B. Lambert. *A Student’s Guide to Bayesian Statistics*. 2018.

[118] B. Lambert. “A Student’s Guide to Bayesian Statistics”. In: 2018. Chap. 11 - Markov Chain Monte Carlo.

[119] A. Gelman, J. B. Carlin, H. S. Stern, David B. Dunson, A. Vehtari, and Donald B. Rubin. *Bayesian Data Analysis (3rd edition)*. 2021.

[120] R. M. Neal. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Report CRG-TR-93-1. University of Toronto, 1993.

[121] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. ISSN: 0021-9606 1089-7690. DOI: 10.1063/1.1699114.

[122] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. “Hybrid Monte Carlo”. In: *Physics Letters B* 195.2 (1987).

[123] R. M. Neal. “Handbook of Markov Chain Monte Carlo”. In: ed. by Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. Chapman & Hall / CRC Press, 2011. Chap. 5 - MCMC using Hamiltonian dynamics.

[124] M. D. Hoffmann and A. Gelman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: *Journal of Machine Learning Research* 15 (2014).

[125] A. Gelman and Donald B. Rubin. “Inference from Iterative Simulation Using Multiple Sequences”. In: *Statistical Science* 7.4 (1992).

[126] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner. “Rank-Normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion)”. In: *Bayesian Analysis* 16.2 (2021). ISSN: 1936-0975. DOI: 10.1214/20-ba1221.

[127] L. Wasserman. *All of statistics: A concise course in statistical inference*. Springer, 2013.

[128] O. Abril-Pla, V. Andreani, C. Carroll, L. Dong, C. J. Fonnesbeck, M. Kochurov, R. Kumar, J. Lao, C. C. Luhmann, O. A. Martin, M. Osthege, R. Vieira, T. Wiecki, and R. Zinkov. “PyMC: a modern, and comprehensive probabilistic programming framework in Python”. In: *PeerJ Comput Sci* 9 (2023), e1516. ISSN: 2376-5992 (Electronic) 2376-5992 (Linking). DOI: 10.7717/peerj-cs.1516. URL: https://www.ncbi.nlm.nih.gov/pubmed/37705656.

[129] R. D. Paul, J. F. Jadebeck, A. Stratmann, W. Wiechert, and K. Nöh. *hopsy - a methods marketplace for convex polytope sampling in Python*. 2023. DOI: 10.1101/2023.12.22.573091. URL: https://www.biorxiv.org/content/10.1101/2023.12.22.573091v1 (visited on 05/16/2024).

[130] K. Nöh, K. Gronke, B. Luo, R. Takors, M. Oldiges, and W. Wiechert. “Metabolic flux analysis at ultra short time scale: isotopically non-stationary ^{13}C labeling experiments”. In: *J Biotechnol* 129.2 (2007), pp. 249–67. ISSN: 0168-1656 (Print) 0168-1656 (Linking). DOI: 10.1016/j.jbiotec.2006.11.015. URL: https://www.ncbi.nlm.nih.gov/pubmed/17207877.

[131] J. Tillack, S. Noack, K. Nöh, A. Elsheikh, and W. Wiechert. “A Software Framework for Modeling and Simulation of Dynamic Metabolic and Isotopic Systems.” In: *Simul Notes Eur* 22.3-4 (2012), pp. 147–156.

[132] W. Wiechert and K. Nöh. “Quantitative metabolic flux analysis based on isotope labeling”. In: *Metabolic Engineering: Concepts and Applications*. Ed. by J. Nielsen, G. Stephanopoulos, and Sang Yup Lee. Vol. 13. WILEY-VCH GmbH, 2021. Chap. 3, pp. 73–136. ISBN: 9783527346622. DOI: 10.1002/9783527823468.

[133] W. Wiechert and A. A. de Graaf. “Bidirectional reaction steps in metabolic networks: I. Modeling and simulation of carbon isotope labeling experiments”. In: *Biotechnology and Bioengineering* 55.1 (1997), pp. 101–117. ISSN: 0006-3592.

[134] W. Wiechert. *Metabolische Kohlenstoff-Markierungssysteme: Modellierung, Simulation, Analyse, Datenauswertung*. Forschungszentrum Jülich, Zentralbibliothek, 1996.

[135] M. R. Antoniewicz, J. K. Kelleher, and G. Stephanopoulos. “Elementary metabolite units (EMU): a novel framework for modeling isotopic distributions”. In: *Metabolic Engineering* 9.1 (2007), pp. 68–86. ISSN: 1096-7176.

[136] M. Beyß, S. Azzouzi, M. Weitzel, W. Wiechert, and K. Nöh. “The Design of FluxML: A Universal Modeling Language for ^{13}C Metabolic Flux Analysis”. In: *Front Microbiol* 10 (2019), p. 1022. ISSN: 1664-302X (Print) 1664-302X (Linking). DOI: 10.3389/fmicb.2019.01022. URL: https://www.ncbi.nlm.nih.gov/pubmed/31178829.

[137] M. Weitzel. “High Performance Algorithms for Metabolic Flux Analysis”. Dissertation. 2010.

[138] S. B. Crown, C. P. Long, and M. R. Antoniewicz. “Integrated ^{13}C-metabolic flux analysis of 14 parallel labeling experiments in *Escherichia coli*”. In: *Metabolic Engineering* 28 (2015), pp. 151–158. ISSN: 1096-7176.

[139] W. Wiechert and M. Wurzel. “Metabolic isotopomer labeling systems: Part I: global dynamic behavior”. In: *Mathematical Biosciences* 169.2 (2001), pp. 173–205. ISSN: 0025-5564.

[140] L. Halle, N. Hollmann, N. Tenhaef, L. Mbengi, C. Glitz, W. Wiechert, T. Polen, M. Baumgart, M. Bott, and S. Noack. “Robotic workflows for automated long-term adaptive laboratory evolution: improving ethanol utilization by *Corynebacterium glutamicum*”. In: *Microb Cell Fact* 22.1 (2023), p. 175. ISSN: 1475-2859 (Electronic) 1475-2859 (Linking). DOI: 10.1186/s12934-023-02180-5. URL: https://www.ncbi.nlm.nih.gov/pubmed/37679814.

[141] C. Keilhauer, L. Eggeling, and H. Sahm. “Isoleucine Synthesis in *Corynebacterium glutamicum*: Molecular Analysis of the ilvB-ilvN-ilvC Operon”. In: *Journal of Bacteriology* 175.11 (1993).

[142] J. Nießer, M. F. Muller, J. Kappelmann, W. Wiechert, and S. Noack. “Hot isopropanol quenching procedure for automated microtiter plate scale ^{13}C-labeling experiments”. In: *Microb Cell Fact* 21.1 (2022), p. 78. ISSN: 1475-2859 (Electronic) 1475-2859 (Linking). DOI: 10.1186/s12934-022-01806-4. URL: https://www.ncbi.nlm.nih.gov/pubmed/35527247.

[143] M. Wellerdiek, D. Winterhoff, W. Reule, J. Brandner, and M. Oldiges. “Metabolic quenching of *Corynebacterium glutamicum*: efficiency of methods and impact of cold shock”. In: *Bioprocess Biosyst Eng* 32.5 (2009), pp. 581–92. ISSN: 1615-7605 (Electronic) 1615-7591 (Linking). DOI: 10.1007/s00449-008-0280-y. URL: https://www.ncbi.nlm.nih.gov/pubmed/19050933.

[144] S. Meinert, S. Rapp, K. Schmitz, S. Noack, G. Kornfeld, and T. Hardiman. “Quantitative quenching evaluation and direct intracellular metabolite analysis in Penicillium chrysogenum”. In: *Anal Biochem* 438 (2013), pp. 47–52. DOI: https://doi.org/10.1016/j.ab.2013.03.021.

[145] A. Feith, A. Teleki, M. Graf, L. Favilli, and R. Takors. “HILIC-Enabled ^{13}C Metabolomics Strategies: Comparing Quantitative Precision and Spectral Accuracy of QTOF High- and QQQ Low-Resolution Mass Spectrometry”. In: *Metabolites* 9.4 (2019). ISSN: 2218-1989 (Print) 2218-1989 (Electronic) 2218-1989 (Linking). DOI: 10.3390/metabo9040063. URL: https://www.ncbi.nlm.nih.gov/pubmed/30986989.

[146] *Microsoft Excel*. Version 2016. 2016.

[147] *Apache Airflow*. Version 2.6.3. 2023.

[148] *celery*. Version 5.3.1. 2023. URL: https://github.com/celery/celery/.

[149] F. Biscani and D. Izzo. “A parallel global multiobjective framework for optimization: pagmo”. In: *Journal of Open Source Software* 5.53 (2020), p. 2338. DOI: 10.21105/joss.02338. URL: https://doi.org/10.21105/joss.02338.

[150] L. Eggeling and M. Bott, eds. *Handbook of Corynebacterium glutamicum*. CRC Press, 2005. ISBN: 9781420039696. DOI: 10.1201/9781420039696.

[151] P. Droste, K. Nöh, and W. Wiechert. “Omix - A visualization tool for metabolic networks with highest usability and customizability in focus”. In: *Chemie Ingenieur Technik* 85.6 (2013), pp. 849–862. DOI: 10.1002/cite.201200234. URL: http://doi.wiley.com/10.1002/cite.201200234.

[152] M. Weitzel, K. Nöh, T. Dalman, S. Niedenführ, B. Stute, and W. Wiechert. “13CFLUX2—high-performance software suite for ^{13}C-metabolic flux analysis”. In: *Bioinformatics* 29.1 (2013), pp. 143–145.

[153] K. Nöh, S. A. Wahl, and W. Wiechert. “Computational tools for isotopically instationary ^{13}C labeling experiments under metabolic steady state conditions”. In: *Metabolic Engineering* 8.6 (2006), pp. 554–577. DOI: 10.1016/j.ymben.2006.05.006.

[154] A. Wächter and L. T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (Mar. 1, 2006), pp. 25–57. ISSN: 1436-4646. DOI: 10.1007/s10107-004-0559-y. URL: https://doi.org/10.1007/s10107-004-0559-y.

[155] A. Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmüller, and J. Timmer. “Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood”. In: *Bioinformatics* 25.15 (Aug. 1, 2009), pp. 1923–1929. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp358. URL: https://doi.org/10.1093/bioinformatics/btp358 (visited on 09/11/2023).

[156] K. J. Bent and A. G. Morton. “Amino Acid Composition of Fungi during Development in Submerged Culture.” In: *Biochemical Journal* 92 (1964).

[157] Michael Osthege. “Accelerated Bioprocess Research by Autonomous Experimentation and Bayesian Modeling”. Dissertation. 2023. DOI: 10.18154/RWTH-2024-04287. URL: https://publications.rwth-aachen.de/record/984915/files/984915.pdf.

[158] M. Osthege, N. Tenhaef, R. Zyla, C. Müller, J. Hemmerich, W. Wiechert, S. Noack, and M. Oldiges. “bletl - A Python package for integrating BioLector microcultivation devices in the Design-Build-Test-Learn cycle”. In: *Engineering in Life Sciences* 22.3-4 (2022), pp. 242–259. ISSN: 1618-0240 1618-2863. DOI: 10.1002/elsc.202100108.

[159] M. Osthege, N. Tenhaef, S. Noack, L. M. Helleckes, J. Nießer, A. Reiter, B. Geinitz, R. Hamel, L. Halle, C. Müller, and S. Schito. *JuBiotech/bletl: v1.4.1*. Version v1.4.1. Aug. 2023. DOI: 10.5281/zenodo.8220134. URL: https://doi.org/10.5281/zenodo.8220134.

[160] *Freedom EVOware*. URL: https://lifesciences.tecan.com/software-freedom-evoware.

[161] M. Osthege, L. M. Helleckes, J. Nießer, L. Halle, S. Noack, C. Kosonocky, C. Müller, and V. Steier. *JuBiotech/robotools: v1.8.0*. Version v1.8.0. Nov. 2023. DOI: 10.5281/zenodo.10210159. URL: https://doi.org/10.5281/zenodo.10210159.

[162] E. Kimball and J. D. Rabinowitz. "Identifying decomposition products in extracts of cellular metabolites". In: *Anal Biochem* 358.2 (2006), pp. 273–80. ISSN: 0003-2697 (Print) 0003-2697 (Linking). DOI: 10.1016/j.ab.2006.07.038. URL: https://www.ncbi.nlm.nih.gov/pubmed/16962982.

[163] W. Lu, X. Su, M. S. Klein, I. A. Lewis, O. Fiehn, and J. D. Rabinowitz. "Metabolite Measurement: Pitfalls to Avoid and Practices to Follow". In: *Annu Rev Biochem* 86 (2017), pp. 277–304. ISSN: 1545-4509 (Electronic) 0066-4154 (Linking). DOI: 10.1146/annurev-biochem-061516-044952. URL: https://www.ncbi.nlm.nih.gov/pubmed/28654323.

[164] S. Noack, K. Nöh, M. Moch, M. Oldiges, and W. Wiechert. "Stationary versus non-stationary ^{13}C-MFA: a comparison using a consistent dataset". In: *J Biotechnol* 154.2-3 (2011), pp. 179–90. ISSN: 1873-4863 (Electronic) 0168-1656 (Linking). DOI: 10.1016/j.jbiotec.2010.07.008. URL: https://www.ncbi.nlm.nih.gov/pubmed/20638432.

[165] J. Hiller, E. Franco-Lara, V. Papaioannou, and D. Weuster-Botz. "Fast sampling and quenching procedures for microbial metabolic profiling". In: *Biotechnol Lett* 29.8 (2007), pp. 1161–7. ISSN: 0141-5492 (Print) 0141-5492 (Linking). DOI: 10.1007/s10529-007-9383-9. URL: https://www.ncbi.nlm.nih.gov/pubmed/17479222.

[166] J. Hiller, E. Franco-Lara, V. Papaioannou, and D. Weuster-Botz. "Fast sampling and quenching procedures for microbial metabolic profiling". In: *Biotechnology Letters* 29 (2007), pp. 1161–1167.

[167] F. Lameiras, J. J. Heijnen, and W. M. van Gulik. "Development of tools for quantitative intracellular metabolomics of *Aspergillus niger* chemostat cultures". In: *Metabolomics* 11 (2015), pp. 1253–1264.

[168] L. M. Helleckes, M. Osthege, W. Wiechert, E. von Lieres, and M. Oldiges. "Bayesian calibration, process modeling and uncertainty quantification in biotechnology". In: *PLoS Computational Biology* 18.3 (2022), e1009223.

[169] L. M. Osthege M. and Helleckes and M. Siska. *JuBiotech/calibr8: v7.1.0*. Version v7.1.0. July 2023. DOI: 10.5281/zenodo.8109808. URL: https://doi.org/10.5281/zenodo.8109808.

[170] *Jinja2*. 2008. URL: https://github.com/pallets/jinja/.

[171] M. Osthege, L. M. Helleckes, J. Nießer, S. Noack, C. Kosonocky, C. Müller, and V. Steier. *JuBiotech/robotools: v1.4.0*. Version v1.4.0. Jan. 2023. DOI: 10.5281/zenodo.7568547. URL: https://doi.org/10.5281/zenodo.7568547.

[172] Jochen Nießer, Michael Osthege, Eric von Lieres, Wolfgang Wiechert, and Stephan Noack. “PeakPerformance - A tool for Bayesian inference-based fitting of LC-MS/MS peaks”. In: *Journal of Open Source Software* 9.104 (Dec. 2024). DOI: 10.21105/joss.07313. URL: https://joss.theoj.org/papers/10.21105/joss.07313.

[173] M. von Haugwitz. “Mass Spectrometric Data Processing for Metabolomics and Fluxomics: A Flexible Evaluation Framework with Quality Awareness”. Thesis. 2016. URL: https://publications.rwth-aachen.de/record/572757/files/572757.pdf.

[174] A. Theorell and K. Nöh. “Model Uncertainty Analysis for Metabolic Network Inference: A Case Study in Bayesian Model Averaging”. In: *IFAC-PapersOnLine* 51 (2018), pp. 124–125.

[175] PyMC Developers. *PyMC*. URL: https://github.com/pymc-devs/pymc.

[176] Adrian Seyboldt and PyMC Developers. *nutpie*. URL: https://github.com/pymc-devs/nutpie.

[177] R. Kumar, C. Carroll, A. Hartikainen, and O. A. Martin. “ArviZ a unified library for exploratory analysis of Bayesian models in Python”. In: *Journal of Open Source Software* 4.33 (2019). ISSN: 2475-9066. DOI: 10.21105/joss.01143.

[178] A. Azzalini. “A class of distributions which includes the normal ones”. In: *Scand J Statist* 12 (1985), pp. 171–178.

[179] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, M. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[180] J. K. Kruschke. *Doing Bayesian Data Analysis*. 1st Edition. 2010. ISBN: 9780123814852.

[181] S. Watanabe. “Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory”. In: *Journal of Machine Learning Research* 11 (2010), pp. 3571–3594.

[182] A. Vehtari, A. Gelman, and J. Gabry. “Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC”. In: *Statistics and Computing* 27.5 (2016), pp. 1413–1432. ISSN: 0960-3174 1573-1375. DOI: 10.1007/s11222-016-9696-4.

[183] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[184] W. McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by S. van der Walt and J. Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[185] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, Robert Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[186] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[187] The Matplotlib Development Team. *Matplotlib: Visualization with Python*. Version v3.9.0. May 2024. DOI: 10.5281/zenodo.11201097. URL: https://doi.org/10.5281/zenodo.11201097.

[188] U. W. Liebal, A. N. T. Phan, M. Sudhakar, K. Raman, and L. M. Blank. "Machine Learning Applications for Mass Spectrometry-Based Metabolomics". In: *Metabolites* 10.6 (2020). ISSN: 2218-1989 (Print) 2218-1989 (Electronic) 2218-1989 (Linking). DOI: 10.3390/metabo10060243. URL: https://www.ncbi.nlm.nih.gov/pubmed/32545768.

[189] A. D. Melnikov, Y. P. Tsentalovich, and V. V. Yanshole. "Deep Learning for the Precise Peak Detection in High-Resolution LC-MS Data". In: *Anal Chem* 92.1 (2020), pp. 588–592. ISSN: 1520-6882 (Electronic) 0003-2700 (Linking). DOI: 10.1021/acs.analchem.9b04811. URL: https://www.ncbi.nlm.nih.gov/pubmed/31841624.

[190] PyMC Developers. *PyTensor*. URL: https://github.com/pymc-devs/pytensor.

[191] *filelock*. 2014. URL: https://github.com/tox-dev/filelock.

[192] *docker*. 2023. URL: https://www.docker.com.

[193] R. J. Kleijn, W. A. van Winden, W. M. van Gulik, and J. J. Heijnen. "Revisiting the ^{13}C-label distribution of the non-oxidative branch of the pentose phosphate pathway based upon kinetic and genetic evidence". In: *FEBS J* 272.19 (2005), pp. 4970–82. ISSN: 1742-464X (Print) 1742-464X (Linking). DOI: 10.1111/j.1742-4658.2005.04907.x. URL: https://www.ncbi.nlm.nih.gov/pubmed/16176270.

[194] J. Kappelmann, B. Klein, P. Geilenkirchen, and S. Noack. "Comprehensive and accurate tracking of carbon origin of LC-tandem mass spectrometry collisional fragments for ^{13}C-MFA". In: *Anal Bioanal Chem* 409.9 (2017), pp. 2309–2326. ISSN: 1618-2650 (Electronic) 1618-2642 (Linking). DOI: 10.1007/s00216-016-0174-9. URL: https://www.ncbi.nlm.nih.gov/pubmed/28116490.

[195] S. Unthan, A. Grunberger, J. van Ooyen, J. Gatgens, J. Heinrich, N. Paczia, W. Wiechert, D. Kohlheyer, and S. Noack. "Beyond growth rate 0.6: What drives *Corynebacterium glutamicum* to higher growth rates in defined medium". In: *Biotechnol Bioeng* 111.2 (2014), pp. 359–71. ISSN: 1097-0290 (Electronic) 0006-3592 (Linking). DOI: 10.1002/bit.25103. URL: https://www.ncbi.nlm.nih.gov/pubmed/23996851.

[196] L. M. Helleckes, M. Osthege, W. Wiechert, E. von Lieres, and M. Oldiges. “Bayesian calibration, process modeling and uncertainty quantification in biotechnology”. In: *PLoS Comput Biol* 18.3 (2022), e1009223. ISSN: 1553-7358 (Electronic) 1553-734X (Print) 1553-734X (Linking). DOI: 10.1371/journal.pcbi.1009223. URL: https://www.ncbi.nlm.nih.gov/pubmed/35255090.

[197] A. Junghanns, T. Blochwitz, C. Bertsch, T. Sommer, K. Wernersson, A. Pillekeit, I. Zacharias, M. Blaesken, P.R. Mai, K. Schuch, C. Schulze, C. Gomes, and M. Najafi. *The Functional Mock-up Interface 3.0 - New Features Enabling New Applications*. Conference Paper. 2021. DOI: 10.3384/ecp2118117.

[198] M. Hucka, F. T. Bergmann, C. Chaouiya, A. Drager, S. Hoops, S. M. Keating, M. Konig, N. L. Novere, C. J. Myers, B. G. Olivier, S. Sahle, J. C. Schaff, R. Sheriff, L. P. Smith, D. Waltemath, D. J. Wilkinson, and F. Zhang. “The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 2 Core Release 2”. In: *J Integr Bioinform* 16.2 (2019). ISSN: 1613-4516 (Electronic) 1613-4516 (Linking). DOI: 10.1515/jib-2019-0021. URL: https://www.ncbi.nlm.nih.gov/pubmed/31219795.

[199] F. Maggioli, T. Mancini, and E. Tronci. “SBML2Modelica: integrating biochemical models within open-standard simulation ecosystems”. In: *Bioinformatics* 36.7 (2020), pp. 2165–2172. ISSN: 1367-4811 (Electronic) 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btz860. URL: https://www.ncbi.nlm.nih.gov/pubmed/31738386.

[200] J. F. Jadebeck, A. Theorell, S. Leweke, and K. Nöh. “HOPS: high-performance library for (non-)uniform sampling of convex-constrained models”. In: *Bioinformatics* 37.12 (2021), pp. 1776–1777. ISSN: 1367-4811 (Electronic) 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btaa872. URL: https://www.ncbi.nlm.nih.gov/pubmed/33045081.

[201] A. Theorell, J. F. Jadebeck, K. Nöh, and J. Stelling. “PolyRound: polytope rounding for random sampling in metabolic networks”. In: *Bioinformatics* 38.2 (2022), pp. 566–567. ISSN: 1367-4811 (Electronic) 1367-4803 (Print) 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btab552. URL: https://www.ncbi.nlm.nih.gov/pubmed/34329395.

[202] M. Osthege. *pytensor-federated*. URL: https://github.com/michaelosthege/pytensor-federated.

[203] J. O. Krömer, O. Sorgenfrei, K. Klopprogge, E. Heinzle, and C. Wittmann. “In-depth profiling of lysine-producing *Corynebacterium glutamicum* by combined analysis of the transcriptome, metabolome, and fluxome”. In: *J Bacteriol* 186.6 (2004), pp. 1769–84. ISSN: 0021-9193 (Print) 1098-5530 (Electronic) 0021-9193 (Linking). DOI: 10.1128/JB.186.6.1769-1784.2004. URL: https://www.ncbi.nlm.nih.gov/pubmed/14996808.

[204] C. J. Bolten, P. Kiefer, F. Letisse, J.-C. Portais, and C. Wittmann. “Sampling for Metabolome Analysis of Microorganisms”. In: *Anal Chem* 79 (2007), pp. 3843–3849.

[205] S. Lai, Y. Zhang, S. Liu, Y. Liang, X. Shang, X. Chai, and T. Wen. “Metabolic engineering and flux analysis of *Corynebacterium glutamicum* for L-serine production”. In: *Sci China Life Sci* 55.4 (2012), pp. 283–90. ISSN: 1869-1889 (Electronic) 1674-7305 (Linking). DOI: 10.1007/s11427-012-4304-0. URL: https://www.ncbi.nlm.nih.gov/pubmed/22566084.

[206] M. Möllney, W. Wiechert, D. Kownatzki, and A. A. de Graaf. “Bidirectional reaction steps in metabolic networks: IV. Optimal design of isotopomer labeling experiments”. In: *Biotechnology and Bioengineering* 66.2 (1999), pp. 86–103.

[207] K. Nöh, S. Niedenführ, Martin Beyß, and W. Wiechert. “A Pareto approach to resolve the conflict between information gain and experimental costs: multiple-criteria design of carbon labeling experiments”. In: *PLOS Computational Biology* 14.10 (2018), e1006533.

[208] M. Beyß, V. D. Parra-Peña, H. Ramirez-Malule, and K. Nöh. “Robustifying experimental tracer design for ^{13}C-Metabolic flux analysis”. In: *Frontiers in Bioengineering and Biotechnology* 9 (2021), p. 685323.

[209] J. Hemmerich, N. Tenhaef, C. Steffens, J. Kappelmann, M. Weiske, S. J. Reich, W. Wiechert, M. Oldiges, and S. Noack. “Less Sacrifice, More Insight: Repeated Low-Volume Sampling of Microbioreactor Cultivations Enables Accelerated Deep Phenotyping of Microbial Strain Libraries”. In: *Biotechnol J* 14.9 (2019), e1800428. ISSN: 1860-7314 (Electronic) 1860-6768 (Linking). DOI: 10.1002/biot.201800428. URL: https://www.ncbi.nlm.nih.gov/pubmed/30318833.

[210] S. Heux, C. Berges, P. Millard, J. C. Portais, and F. Letisse. “Recent advances in high-throughput ^{13}C-fluxomics”. In: *Curr Opin Biotechnol* 43 (2017), pp. 104–109. ISSN: 1879-0429 (Electronic) 0958-1669 (Linking). DOI: 10.1016/j.copbio.2016.10.010. URL: https://www.ncbi.nlm.nih.gov/pubmed/27838571.

Appendix

A1 Hot isopropanol quenching validation

The following figures A1, A2, and A3 exhibit the full results of the two validation experiments concerning the hot isopropanol quenching method as discussed in section 3.1.

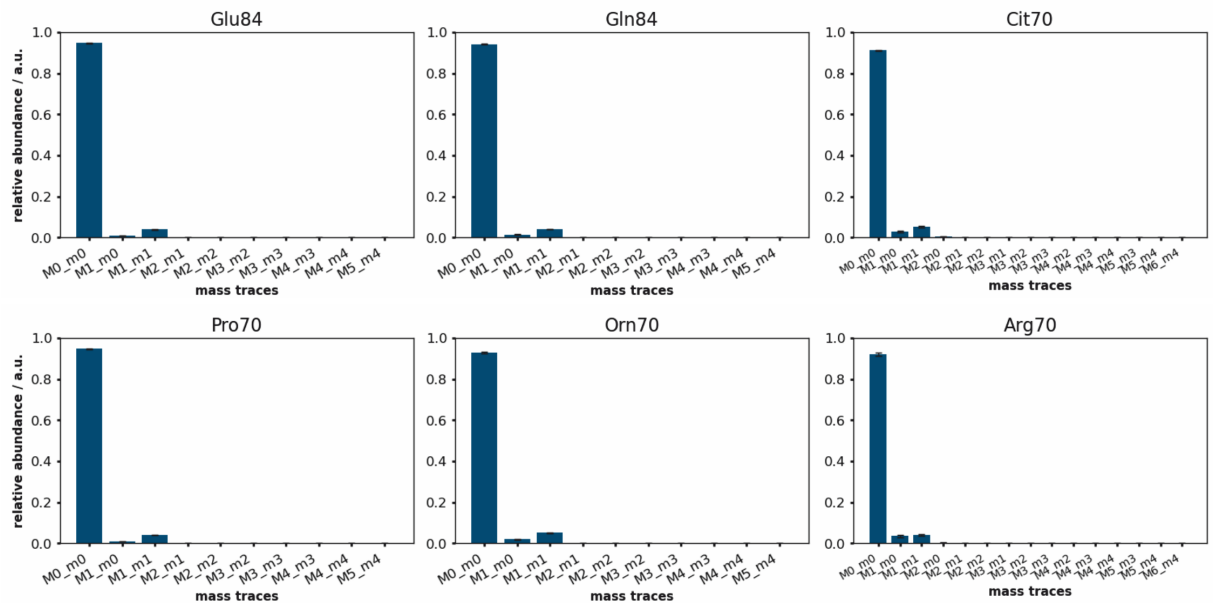


Figure A1: Results of the automated hot isopropanol validation experiment for Glu and Glu-derived amino acids.

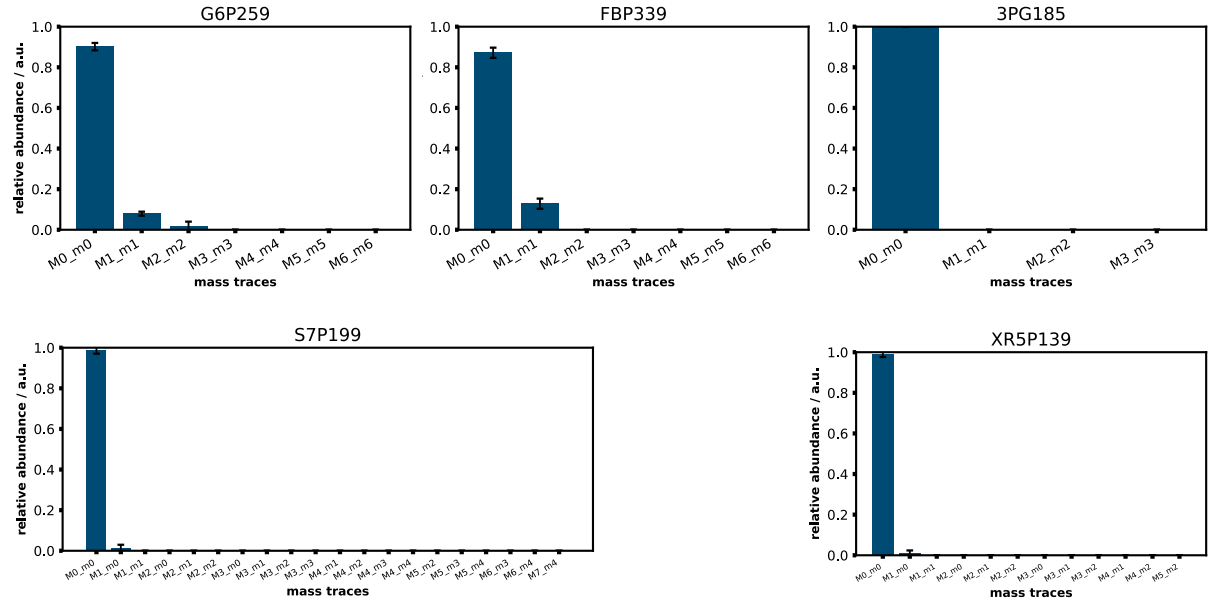


Figure A2: Results of the automated hot isopropanol validation experiment for free intermediates of EMP pathway and PPP.

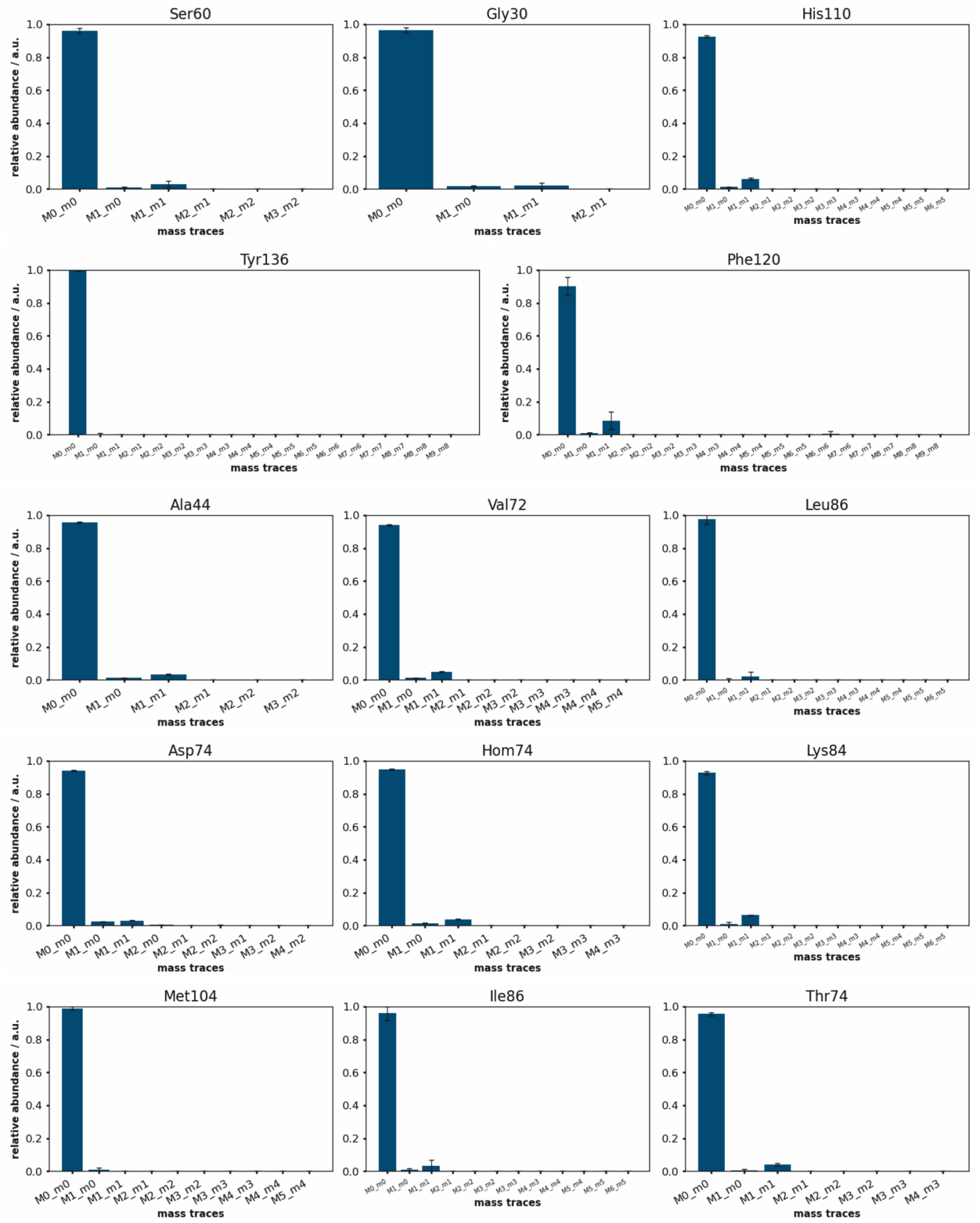


Figure A3: Results of the automated hot isopropanol validation experiment for all remaining measured free amino acids except the Glu-derived ones.

A2 New EvoWare pipetting commands implemented in robotools

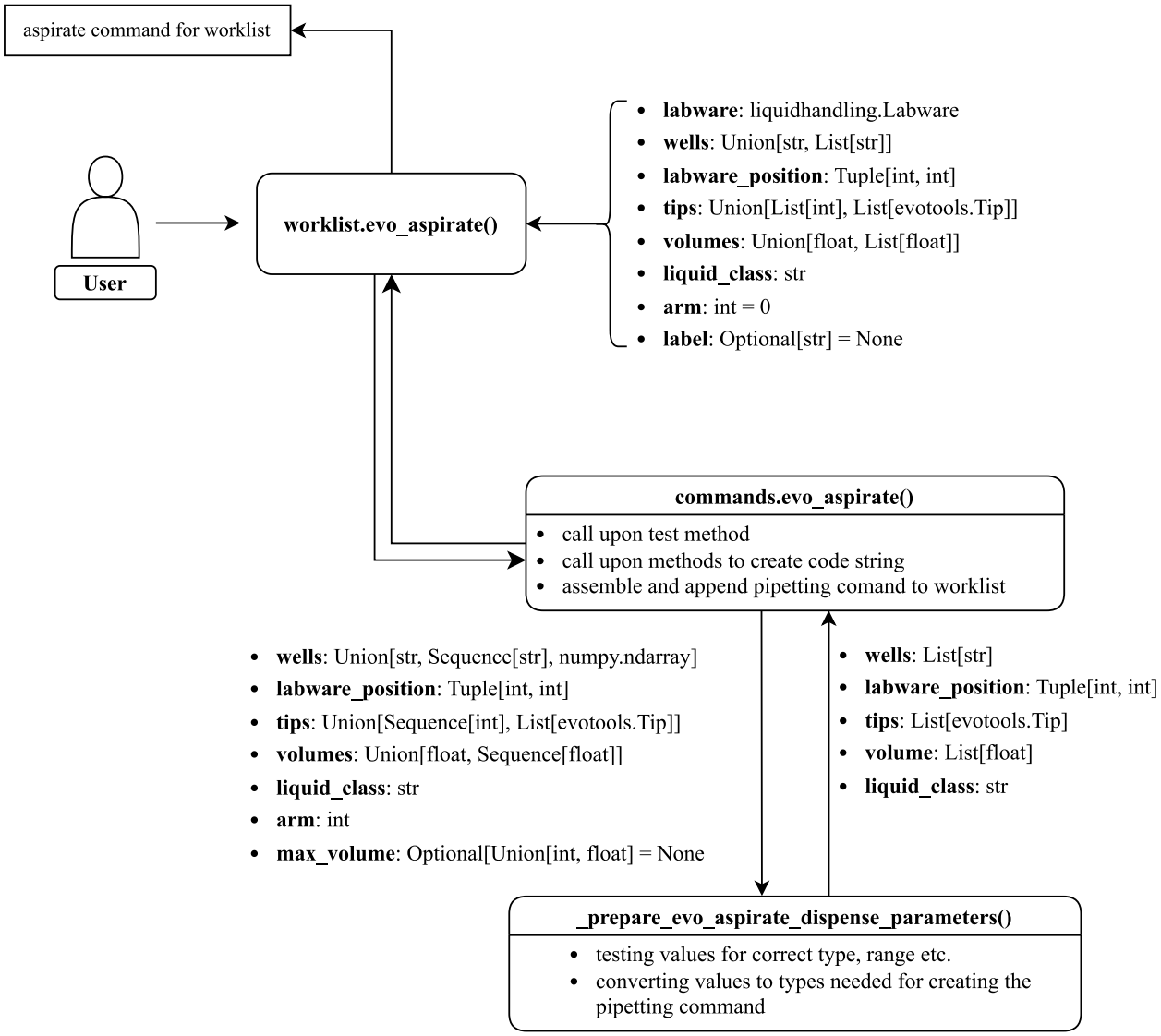


Figure A4: Implementation of the EVOware pipetting commands in the `robotools` Python package. While this figure is focused on aspirate commands, there are analogous methods for dispense commands.

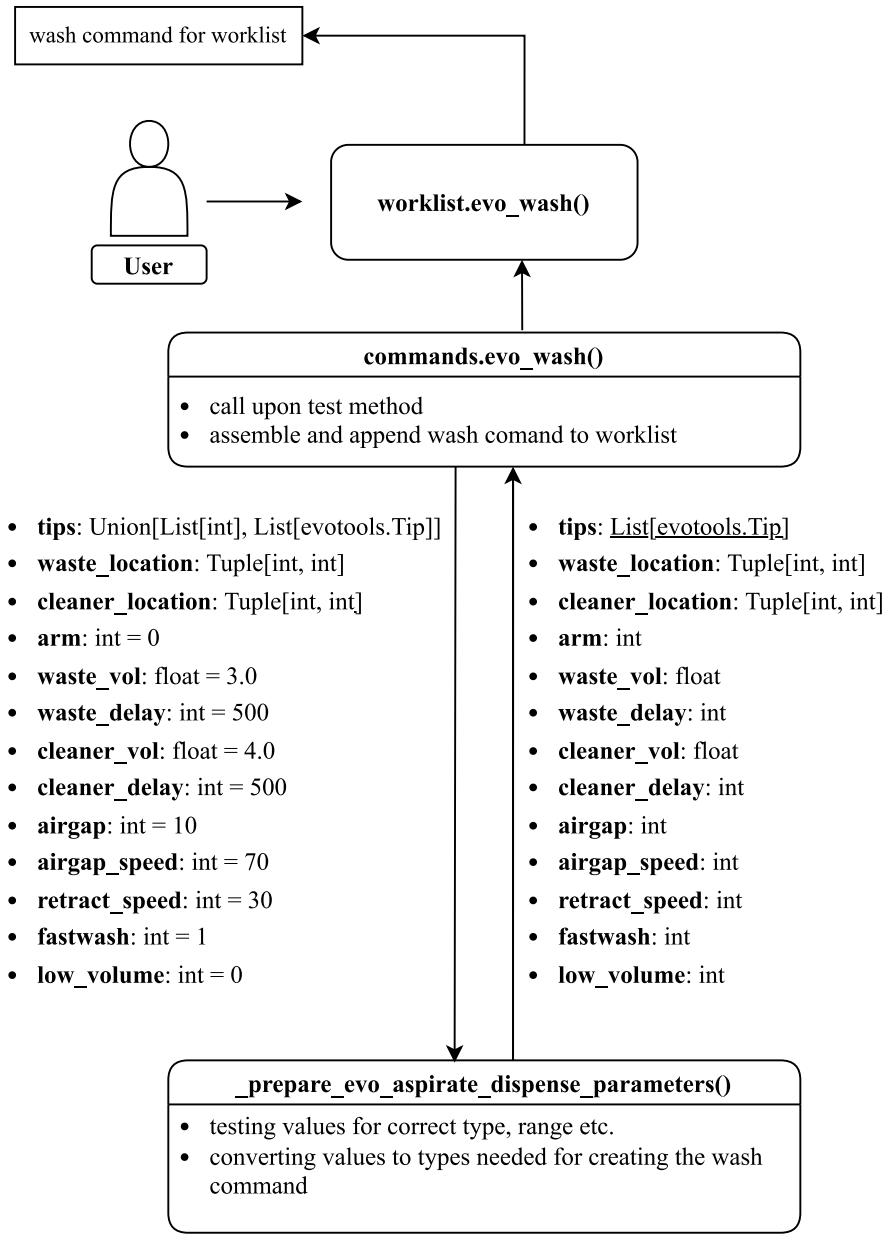


Figure A5: Implementation of the EVOware wash command in the `robotools` Python package.

A3 Height calculation of skew normal-shaped distributions in PeakPerformance

```
import numpy as np
import pytensor.tensor as pt


def delta_calculation(alpha):
    """
    Calculate the delta term included in several subsequent formulae.

    Parameters
    ----------
    alpha
        Skewness parameter of the skew normal distribution.
    """
    return alpha / (np.sqrt(1 + alpha**2))


def mue_z_calculation(delta):
    """
    Calculate the mue_z variable which is needed to compute a numerical approximation
    of the mode of a skew normal distribution.
    """
    return np.sqrt(2 / np.pi) * delta


def sigma_z_calculation(mue_z):
    """
    Calculate the sigma_z variable which is needed to compute a numerical approximation
    of the mode of a skew normal distribution.
    """
    return np.sqrt(1 - mue_z**2)


def skewness_calculation(delta):
    """Calculate the skewness of a skew normal distribution."""
    return (
        (4 - np.pi)
        / 2
        * ((delta * np.sqrt(2 / np.pi)) ** 3)
        / ((1 - 2 * delta**2 / np.pi) ** 1.5)
    )


def mode_offset_calculation(mue_z, skewness, sigma_z, alpha):
    """Calculate the offset between arithmetic mean and mode of a skew normal distribution."""
    # this formula originally contained the sign() function which led to an error due to usage
    # of pytensor variables -> use alpha/abs(alpha) instead for the same effect
    return (
        mue_z
```

```
        - (skewness * sigma_z) / 2
        - (alpha / abs(alpha)) / 2 * pt.exp(-(2 * np.pi) / abs(alpha))
    )


def mode_skew_calculation(loc, scale, mode_offset):
    """Calculate a numerical approximation of the mode of a skew normal distribution."""
    return loc + scale * mode_offset


def height_calculation(area, loc, scale, alpha, mode_skew):
    """
    Calculate the height of a skew normal distribution.
    The formula is the result of inserting time = mode_skew into the posterior.

    Parameters
    ----------
    area
        Area of the peak described by the skew normal distribution.
        (area between baseline and skew normal distribution)
    loc
        Location parameter of the skew normal distribution.
    scale
        Scale parameter of the skew normal distribution.
    alpha
        Skewness parameter of the skew normal distribution.
    mode_skew
        Mode of the skew normal distribution.

    Returns
    ----------
    mean
        Arithmetic mean of a skew normal distribution.
    """
    return area * (
        2
        * (1 / (scale * np.sqrt(2 * np.pi)) * pt.exp(-0.5 * ((mode_skew - loc) / scale) ** 2))
        * (0.5 * (1 + pt.erf(((alpha * (mode_skew - loc) / scale)) / np.sqrt(2))))
    )
```

Listing A1: Height calculation of skew normal-shaped models in `PeakPerformance`

A4 Modelica model implementations

A4.1 Modelica formulation of reduced sub-network model v0

```
within ;
model MPP_INST_v0

//makroskopic parameters
     Real V(start = V_0, fixed=true);          // [L]
     Real c_X(start = c_X0, fixed=true);       // [g_X/L]
     Real c_S0(start =  c_S0_0, fixed=true);   // [mmol / L]
     Real c_S1(start = 0, fixed=true);         // [mmol / L]
     Real c_S;                                 // [mmol / L]
     parameter Real c_S0_0 = 110.02;           // [mmol / L]
     // glucose labeling fractions
     Real x_S0;                                // [-]
     Real x_S1;                                // [-]
     // extracellular glycine
     Real c_Gly0_extra(start=c_Gly0_extra_0, fixed=true);   // [mmol / L]
     Real c_Gly1_extra(start=c_Gly1_extra_0, fixed=true);   // [mmol / L]
     Real c_Gly_extra;                                      // [mmol / L]
     // total glycine balance
     Real c_GLY_total;                         // [mmol / L]
     Real x_GLY1_total;                        // [mmol / L]
     // extracellular glycine labeling
     Real x_GLY1_extra;                        // [-]
     // feed parameters
     Real F;
     parameter Real tF = 5.896401709;          // [h]
     parameter Real dt = 3e-3;                 // [h]
     parameter Real VF = 0.05/1000;            // [mL] --> [L]
     // biomass specific cell volume
     parameter Real V_cell = 1.93/1000;        // [L_cell/g_X]
// kinetic paramameters
     Real mu;
     parameter Real mu_max = 0.635;            // [1/h]
     parameter Real K_mu_S = 0.00041;          // [mmol/L_cell]
     parameter Real v_upt_S_max = 2266.59;     // [mmol/L_cell/h]
     parameter Real K_S = 0.00085;             // [mmol/L_reactor]
// mass transport over cell boundaries
     Real v_upt_S;                             // [mmol/L_cell/h]
     Real v_Gly_exp;                           // [mmol/L_cell/h]
     parameter Real k_scale_gly_exp=0.00625;   // scaling glycine export rate to substrate import
     ↪  rate [-]
// intracellular states
     parameter Real c_EMPUP = 0.774;           // [mmol/L_cell]
     parameter Real c_SER = 4.807;             // [mmol/L_cell]
     parameter Real c_GLY_intra = 10.07;       // [mmol/L_cell]
     parameter Real c_CYS = 1.03;              // [mmol/L_cell]
//  intracellular labeling fractions
     Real x_EMPUP1(start = 0, fixed=true);      // [-]
```

```
    Real x_SER1(start = 0, fixed=true);        // [-]
    Real x_GLY1_intra(start = 0, fixed=true);  // [-]
    Real x_CYS1(start = 0, fixed=true);        // [-]
// intracellular fluxes
    Real v_1;                                  // [mmol/L_cell/h]
    Real v_2;                                  // [mmol/L_cell/h]
    Real v_3;                                  // [mmol/L_cell/h]
    Real v_4;                                  // [mmol/L_cell/h]
// biomass incorporation yield
  // lumped yields for EMPUP additinal accounting His, UMP, IMP
      Real Y_EMPup_X;
      parameter Real Y_G6P_X = 0.50877;        // [mmol/g_X]
      parameter Real Y_F6P_X = 0.1949;         // [mmol/g_X]
      parameter Real Y_GAP_X = 0.0676/2;       // [mmol/g_X]
      parameter Real Y_R5P_X = 0;              // [mmol/g_X]
      parameter Real Y_E4P_X = 0;              // [mmol/g_X]
      parameter Real Y_HIS_X = 0.066361905;              // [mmol/g_X]
      parameter Real Y_UMP_X = 0.086159479;               // [mmol/g_X]
      parameter Real Y_IMP_X = 0.10114561;                // [mmol/g_X]
      parameter Real Y_TRP_X = 0.027238;                  // [mmol/g_X]
      parameter Real Y_TYR_X = 0.07671;                   // [mmol/g_X]
      parameter Real Y_PHE_X = 0.126780952;               // [mmol/g_X]
      parameter Real Y_MET_X = 0.074780952;               // [mmol/g_X]
  // 3-PGA amino acids
      parameter Real Y_CYS_X = 0.0436;                  // [mmol/g_X]
      parameter Real Y_SER_X = 0.2427;                  // [mmol/g_X]
      parameter Real Y_GLY_X = 0.3491;                  // [mmol/g_X]

// initial values
    parameter Real V_0= 800 / 1e6;                      // [L]
    parameter Real c_X0 = 0.261;                        // [g_X/L]
    parameter Real c_Gly0_extra_0 =1e-10;               // [mmol/L]
    parameter Real c_Gly1_extra_0 =1e-10;               // [mmol/L]
    parameter Real c_S0_F = 0;                          // [mmol/L]
    parameter Real c_S1_F = 86.4/180.15*1000;           // [mmol/L]

equation
    //Extracellular mass balances
      // FedBactch feed
        //Substrate pulse
        if (time >= tF) and (time < tF + dt) then
          F = VF/dt;
        else
          F = 0;
        end if;
        //prevent negative substrate concentration
        if (c_S0 <= 0) then
          c_S0=0;
        end if;
        if (c_S1 <=0) then
          c_S1=0;
```

```
      end if;
      der(V) = F;

    // biomass
    mu = mu_max * c_S / (K_mu_S + c_S);
    der(c_X) = mu * c_X - F/V * c_X;

    // substrate
    v_upt_S = v_upt_S_max * c_S / (K_S + c_S); //[mmol/L_cell/h]
    der(c_S0) = F/V * c_S0_F - v_upt_S * V_cell * c_X * x_S0 - F/V * c_S0;
    der(c_S1) = F/V * c_S1_F - v_upt_S * V_cell * c_X * x_S1 - F/V * c_S1;
    c_S = c_S0 + c_S1;
    x_S0 = c_S0 / c_S;
    x_S1 = 1 - x_S0;

    // extracellular glycine
    v_Gly_exp=v_upt_S*k_scale_gly_exp ;
    der(c_Gly0_extra)= v_Gly_exp*c_X*V_cell*(1-x_GLY1_intra)-F/V*c_Gly0_extra;
    der(c_Gly1_extra)= v_Gly_exp*c_X*V_cell*x_GLY1_intra-F/V*c_Gly1_extra;
    c_Gly_extra = c_Gly0_extra + c_Gly1_extra;
    x_GLY1_extra = c_Gly1_extra / c_Gly_extra;

    //total glycine and labeling fractions
    c_GLY_total = c_GLY_intra*c_X*V_cell+c_Gly_extra;
    x_GLY1_total = (x_GLY1_intra*c_GLY_intra*c_X*V_cell+x_GLY1_extra*c_Gly_extra)/c_GLY_total;

  //intracellular balancing
    // biomass composition EMPUP
    Y_EMPup_X = Y_G6P_X + Y_F6P_X + Y_GAP_X + Y_R5P_X + Y_E4P_X+Y_HIS_X+Y_IMP_X+Y_UMP_X
    ↪  +Y_PHE_X+Y_TRP_X+Y_TYR_X;

    // flux balance
    0 = v_upt_S - (v_1 + v_4) - mu * Y_EMPup_X / V_cell - mu * c_EMPUP;
    0 = 2 * v_1 - (v_2 + v_3) - mu * (Y_SER_X+Y_TRP_X )/ V_cell - mu * c_SER;
    0 = v_2 - mu * (Y_GLY_X+Y_IMP_X ) / V_cell - mu * c_GLY_intra-v_Gly_exp;
    0 = v_3 - mu * (Y_CYS_X+Y_MET_X) / V_cell - mu * c_CYS;

    //labeling dynamics
    c_EMPUP * der(x_EMPUP1) = v_upt_S * x_S1 - (v_1 + v_4) * x_EMPUP1 - mu * Y_EMPup_X / V_cell
    ↪  * x_EMPUP1 - mu * c_EMPUP * x_EMPUP1;
    c_SER * der(x_SER1) = 2 * v_1 * x_EMPUP1 - (v_2 + v_3) * x_SER1 - mu * (Y_SER_X+Y_TRP_X ) /
    ↪  V_cell * x_SER1  - mu * c_SER * x_SER1;
    c_GLY_intra * der(x_GLY1_intra) = v_2 * x_SER1 - mu * (Y_GLY_X+Y_IMP_X ) / V_cell *
    ↪  x_GLY1_intra  - mu * c_GLY_intra * x_GLY1_intra-v_Gly_exp*x_GLY1_intra;
    c_CYS * der(x_CYS1)      = v_3 * x_SER1 - mu * (Y_CYS_X+Y_MET_X ) / V_cell * x_CYS1  - mu *
    ↪  c_CYS * x_CYS1;

  annotation (experiment(StopTime =7));
end MPP_INST_v0;
```

Listing A2: Modelica code, i.e. model formulation, of reduced metabolic sub-network model v0.

A4.2 Modelica formulation of reduced sub-network model v1

```
within ;
model MPP_INST_v1_1

//makroskopic parameters
     Real V(start = V_0, fixed=true);          // [L]
     Real c_X(start = c_X0, fixed=true);       // [g_X/L]
     Real c_S0(start =  c_S0_0, fixed=true);   // [mmol / L]
     Real c_S1(start = 0, fixed=true);         // [mmol / L]
     Real c_S;                                 // [mmol / L]
     parameter Real c_S0_0 = 110.02;           // [mmol / L]
     // glucose labeling fractions
     Real x_S0;                                // [-]
     Real x_S1;                                // [-]
     // extracellular glycine
     Real c_Gly0_extra(start=c_Gly0_extra_0, fixed=true);   // [mmol / L]
     Real c_Gly1_extra(start=c_Gly1_extra_0, fixed=true);   // [mmol / L]
     Real c_Gly_extra;                                      // [mmol / L]
     // total glycine balance
     Real c_GLY_total;                         // [mmol / L]
     Real x_GLY1_total;                        // [mmol / L]
     // extracellular glycine labeling
     Real x_GLY1_extra;                        // [-]
     // feed parameters
     Real F;
     parameter Real tF = 5.896401709;          // [h]
     parameter Real dt = 3e-3;                 // [h]
     parameter Real VF = 0.05/1000;            // [mL] --> [L]
     // biomass specific cell volume
     parameter Real V_cell = 1.93/1000;        // [L_cell/g_X]
/// kinetic paramameters
     Real mu;
     parameter Real mu_max = 0.635;            // [1/h]
     parameter Real K_mu_S = 0.00041;          // [mmol/L_cell]
     parameter Real v_upt_S_max = 2266.59;     // [mmol/L_cell/h]
     parameter Real K_S = 0.00085;             // [mmol/L_reactor]
// mass transport over cell boundaries
     Real v_upt_S;                             // [mmol/L_cell/h]
     Real v_Gly_exp;                           // [mmol/L_cell/h]
     parameter Real k_scale_gly_exp=0.00625;   // scaling glycine export rate to substrate import
     ↪  rate [-]
/// intracellular states

     parameter Real c_EMPUP = 0.774;
     parameter Real c_PEP = 2.33;
     parameter Real c_PYR = 9.43;
     parameter Real c_SER = 4.807;
     parameter Real c_GLY_intra = 10.07;
     parameter Real c_CYS = 1.03;
     parameter Real c_ALA=20.69;
     parameter Real c_KIV = 0.136;
```

```
    parameter Real c_VAL =6.826;
    parameter Real c_LEU =3.57;

//  intracellular labeling fractions
    Real x_EMPUP1(start = 0, fixed=true);
    Real x_PEP1(start = 0, fixed=true);
    Real x_PYR1(start = 0, fixed=true);
    Real x_SER1(start = 0, fixed=true);
    Real x_GLY1_intra(start = 0, fixed=true);
    Real x_CYS1(start = 0, fixed=true);
    Real x_ALA1(start = 0, fixed=true);
    Real x_KIV1(start = 0, fixed=true);
    //Real x_KIV11010(start = 0, fixed=true);
    //Real x_KIV00101(start = 0, fixed=true);
    Real x_VAL1(start = 0, fixed=true);
    //Real x_VAL11010(start = 0, fixed=true);
    //Real x_VAL00101(start = 0, fixed=true);
    Real x_LEUXX1111(start = 0, fixed=true);

// intracellular fluxes
    Real v_1;
    Real v_2;
    Real v_3;
    Real v_4;
    Real v_5;
    Real v_6;
    Real v_7;
    Real v_8;
    Real v_9;
    Real v_10;

// biomass incorporation yield

    // lumped yields for EMPUP
    Real Y_EMPup_X;
    parameter Real Y_G6P_X = 0.50877;                // [mmol/g_X]
    parameter Real Y_F6P_X = 0.1949;                 // [mmol/g_X]
    parameter Real Y_GAP_X = 0.0676/2;               // [mmol/g_X]
    parameter Real Y_R5P_X = 0;                      // [mmol/g_X]
    parameter Real Y_E4P_X = 0;                      // [mmol/g_X]
    parameter Real Y_HIS_X = 0.066361905;            // [mmol/g_X]
    parameter Real Y_UMP_X = 0.086159479;            // [mmol/g_X]
    parameter Real Y_IMP_X = 0.10114561;             // [mmol/g_X]
    parameter Real Y_TRP_X = 0.027238;               // [mmol/g_X]
    parameter Real Y_TYR_X = 0.07671;                // [mmol/g_X]
    parameter Real Y_PHE_X = 0.126780952;            // [mmol/g_X]
    parameter Real Y_MET_X = 0.074780952;            // [mmol/g_X]

   // key metabolites
    parameter Real Y_PEP_X = 0.0975;                 // [mmol/g_X]
    parameter Real Y_PYR_X = 0;                      // [mmol/g_X]
```

```
    parameter Real Y_CYS_X = 0.0436;                    // [mmol/g_X]
    parameter Real Y_SER_X = 0.2427;                    // [mmol/g_X]
    parameter Real Y_GLY_X = 0.3491;                    // [mmol/g_X]

    parameter Real Y_ALA_X = 1.146434349;               // [mmol/g_X]
    parameter Real Y_VAL_X = 0.2704;                    // [mmol/g_X]
    parameter Real Y_LEU_X = 0.347657143;               // [mmol/g_X]

// initial values
  //macroscopic
    parameter Real V_0= 800 / 1e6;
    parameter Real c_X0 = 0.261;
    parameter Real c_Gly0_extra_0 =1e-10;
    parameter Real c_Gly1_extra_0 =1e-10;
    parameter Real c_S0_F = 0;                            // [mmol/L_reactor]
    parameter Real c_S1_F = 86.4/180.15*1000;              // [mmol/L_pulse]

equation
    //Extracellular mass balances
      // FedBactch feed
        //Substrate pulse
        if (time >= tF) and (time < tF + dt) then
          F = VF/dt;
        else
          F = 0;
        end if;
        //prevent negative substrate concentration
        if (c_S0 <= 0) then
          c_S0=0;
        end if;
        if (c_S1 <=0) then
          c_S1=0;
        end if;
        der(V) = F;

       // biomass
       mu = mu_max * c_S / (K_mu_S + c_S);
       der(c_X) = mu * c_X - F/V * c_X;

       // substrate
       v_upt_S = v_upt_S_max * c_S / (K_S + c_S); //[mmol/L_cell/h]
       der(c_S0) = F/V * c_S0_F - v_upt_S * V_cell * c_X * x_S0 - F/V * c_S0;
       der(c_S1) = F/V * c_S1_F - v_upt_S * V_cell * c_X * x_S1 - F/V * c_S1;
       c_S = c_S0 + c_S1;
       x_S0 = c_S0 / c_S;
       x_S1 = 1 - x_S0;

       // extracellular glycine
       v_Gly_exp=v_upt_S*k_scale_gly_exp ;
       der(c_Gly0_extra)= v_Gly_exp*c_X*V_cell*(1-x_GLY1_intra)-F/V*c_Gly0_extra;
```

```
      der(c_Gly1_extra)= v_Gly_exp*c_X*V_cell*x_GLY1_intra-F/V*c_Gly1_extra;
      c_Gly_extra = c_Gly0_extra + c_Gly1_extra;
      x_GLY1_extra = c_Gly1_extra / c_Gly_extra;

      //total glycine and labeling fractions
      c_GLY_total = c_GLY_intra*c_X*V_cell+c_Gly_extra;
      x_GLY1_total = (x_GLY1_intra*c_GLY_intra*c_X*V_cell+x_GLY1_extra*c_Gly_extra)/c_GLY_total;


  //Intracellular balancing
    // biomass composition EMPUP
    Y_EMPup_X = Y_G6P_X + Y_F6P_X + Y_GAP_X + Y_R5P_X + Y_E4P_X+Y_HIS_X+Y_IMP_X+Y_UMP_X
    ↪  +Y_PHE_X+Y_TRP_X+Y_TYR_X;

    // flux Balance
    0 = v_1 - 0.5 * (v_2+v_3+(Y_SER_X +  Y_TRP_X ) /V_cell * mu + mu* c_SER);
    0 = v_2 - (v_Gly_exp + (Y_GLY_X+Y_IMP_X)/V_cell * mu + mu * c_GLY_intra );
    0 = v_3 - ((Y_CYS_X+Y_MET_X )/V_cell * mu + mu * c_CYS);
    0 = v_4 - (v_upt_S - v_1 - Y_EMPup_X/V_cell * mu - mu * c_EMPUP);
    0 = v_5 - (v_4 *2 - v_upt_S - mu * c_PEP - Y_PEP_X/V_cell * mu);
    0 = v_6 - (Y_ALA_X/V_cell * mu + mu * c_ALA );
    0 = v_7 - 2 *(-v_6 + v_8 + v_9 + mu * c_KIV);
    0 = v_8 - (v_6 + Y_VAL_X/V_cell * mu + mu * c_VAL);
    0 = v_9 - (Y_LEU_X/V_cell * mu + mu * c_LEU);
    0 = v_10 - (v_upt_S + v_5 - v_7 - v_6 - mu * c_PYR);

    //labeling dynamics
    c_EMPUP * der(x_EMPUP1) = v_upt_S * x_S1 - (v_1 + v_4) * x_EMPUP1 - mu * Y_EMPup_X / V_cell *
    ↪  x_EMPUP1 - mu * c_EMPUP * x_EMPUP1;
    c_SER * der(x_SER1) = 2 * v_1 * x_EMPUP1 - (v_2 + v_3) * x_SER1 - mu * (Y_SER_X+ Y_TRP_X ) /
    ↪  V_cell * x_SER1  - mu * c_SER * x_SER1;
    c_GLY_intra * der(x_GLY1_intra) = v_2 * x_SER1 - mu * (Y_GLY_X+Y_IMP_X) / V_cell *
    ↪  x_GLY1_intra  - mu * c_GLY_intra * x_GLY1_intra-v_Gly_exp*x_GLY1_intra;
    c_CYS * der(x_CYS1)      = v_3 * x_SER1 - mu * (Y_CYS_X+Y_MET_X ) / V_cell * x_CYS1  - mu *
    ↪  c_CYS * x_CYS1;
    c_PEP * der(x_PEP1)      = 2 * v_4 * x_EMPUP1 - (v_upt_S + v_5)* x_PEP1 - Y_PEP_X/V_cell * mu
    ↪  * x_PEP1 - mu * c_PEP * x_PEP1;
    c_PYR * der(x_PYR1)      = (v_upt_S + v_5) * x_PEP1 - (v_7 + v_6 + v_10)* x_PYR1 - mu * c_PYR
    ↪  * x_PYR1;
    c_ALA * der(x_ALA1)      = v_6 * x_PYR1 - Y_ALA_X/V_cell * mu * x_ALA1 - mu * c_ALA * x_ALA1;
    c_KIV * der(x_KIV1)      =  v_7/2 * x_PYR1^2 + v_6 * x_VAL1 - (v_8 + v_9) * x_KIV1 - mu *
    ↪  c_KIV * x_KIV1;
    c_VAL * der(x_VAL1)      = v_8 * x_KIV1 - v_6 * x_VAL1 - Y_VAL_X/V_cell * mu * x_VAL1 - mu *
    ↪  c_VAL * x_VAL1;
    c_LEU * der(x_LEUXX1111) = v_9 * x_KIV1- Y_LEU_X/V_cell * mu * x_LEUXX1111 - mu * c_LEU *
    ↪  x_LEUXX1111;

  annotation (experiment(StopTime =7));
end MPP_INST_v1_1;
```

Listing A3: Modelica code, i.e. model formulation, of reduced metabolic sub-network model v1.

A5 Benchmarking the MCMC pipeline versus estim8

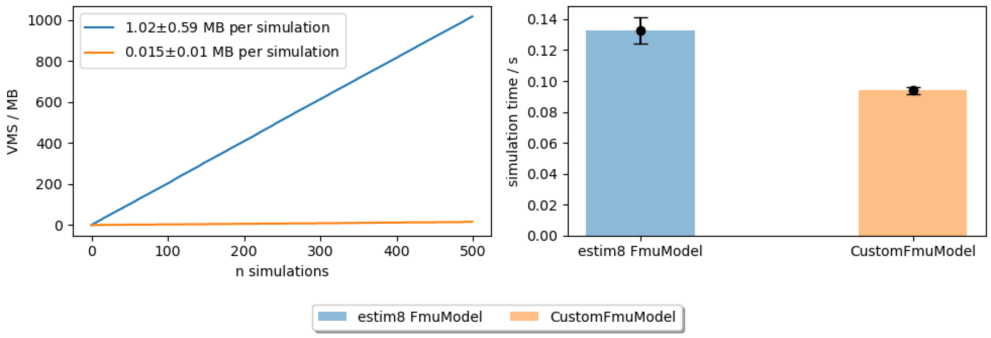


Figure A6: Benchmarking the MCMC pipeline for bioprocess modelling against estim8 in terms of required virtual memory size (VMS) and simulation time.

A6 Full results of the ethanol ^{13}C-INST MFA with *C. glutamicum* WT_EtOH-Evo

The following figures A7 and A8 depict the simulated vs. measured mass traces. The full results of the flux and pool size estimations including 95 % CoI are exhibited in the separate tables A1 and A2, respectively.

The optimization procedure to solve the inverse problem and obtain the presented flux distribution included the mass traces for Met as raw data but since the model simulation predicted a much more significant labeling enrichment for the M1_m1 mass trace, a workaround was applied. The M1_m1 and M2_m1 mass traces were lumped which has been emphasized in A8 by plotting this joint curve in a separate diagram where it is labeled as "M+1" referring to the increased mass of the product ion. All remaining mass traces have been plotted in the diagram to its left and are labeled according to their mass trace.

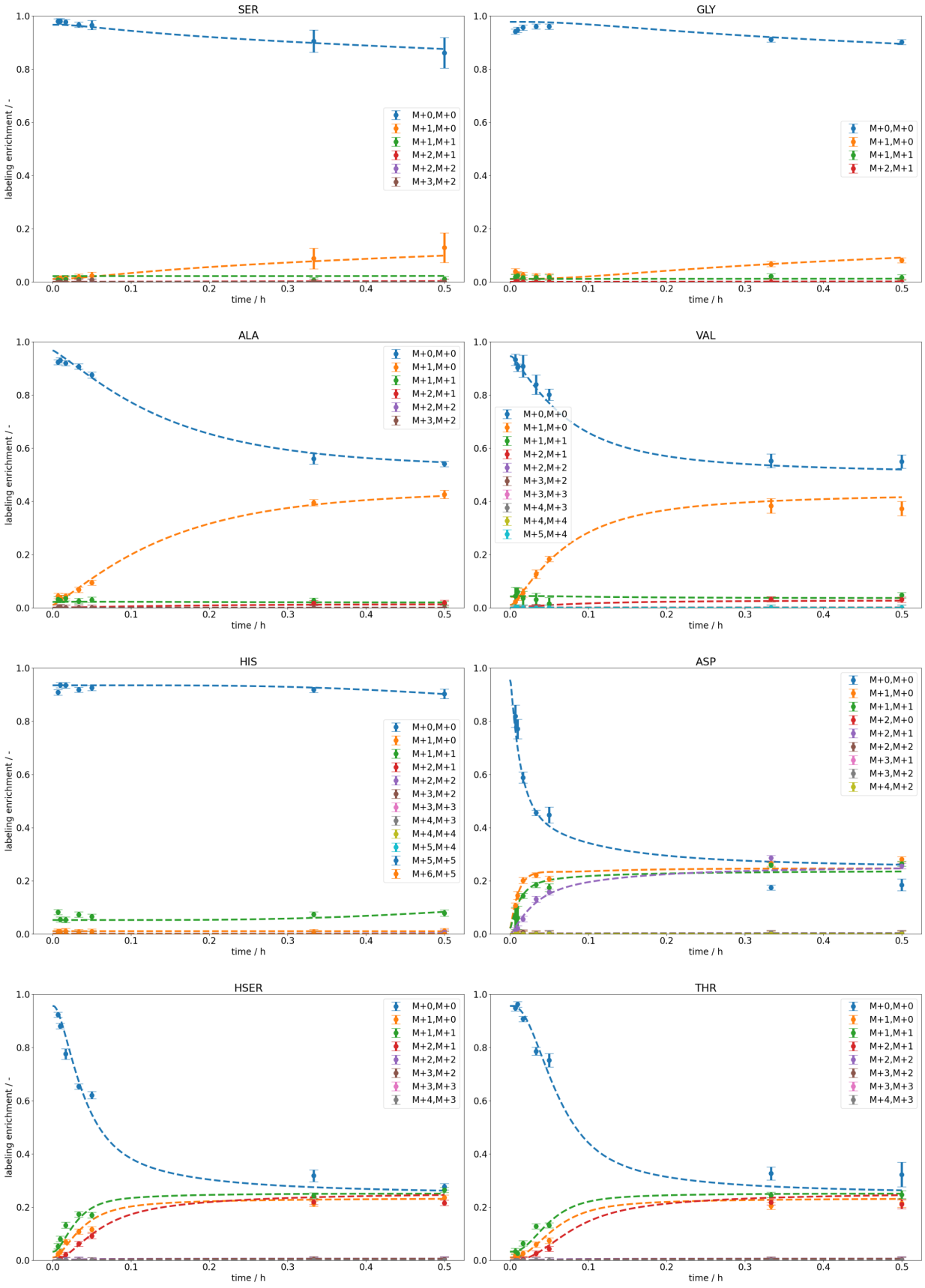


Figure A7: Simulated vs. measured mass traces obtained during the optimization procedure for the INST ^{13}C-MFA with *C. glutamicum* WT_EtOH-Evo using a 1-^{13}C ethanol tracer. The points pertain to the experimental measurements in biological triplicates and the dashed lines to the simulations. Portrayed are from left to right and top to bottom L-serine, L-glycine, L-alanine, L-valine, L-histidine, L-aspartate, L-homoserine, and L-threonine.

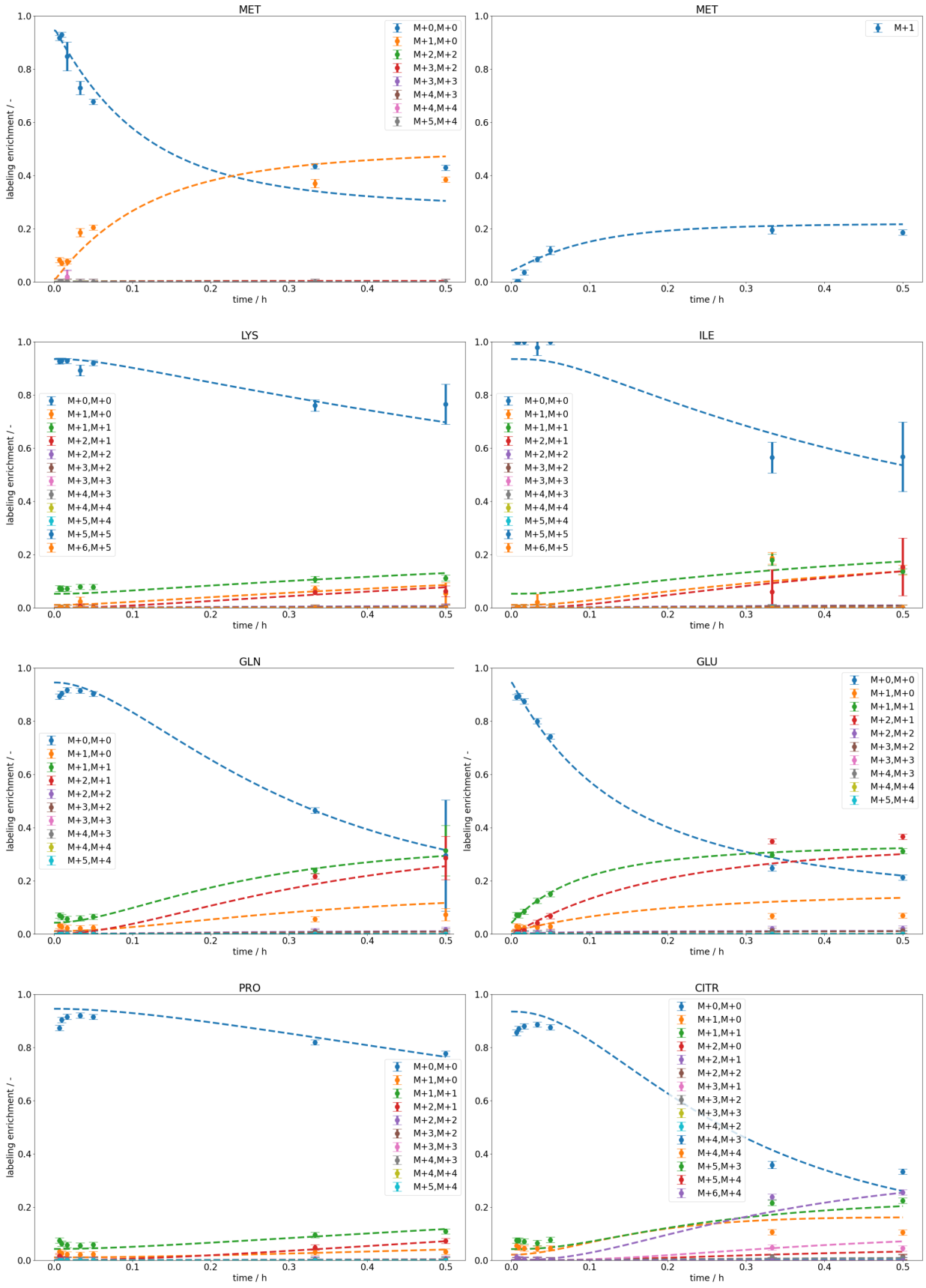


Figure A8: Simulated vs. measured mass traces obtained during the optimization procedure for the INST ^{13}C-MFA with *C. glutamicum* WT_EtOH-Evo using a 1-^{13}C ethanol tracer. The points pertain to the experimental measurements in biological triplicates and the dashed lines to the simulations. Portrayed are from left to right and top to bottom L-methionine, L-lysine, L-isoleucine, L-glutamine, L-glutamate, L-proline, and L-citrulline.

Figure A9 compares the label incorporation of *C. glutamicum* WT_ETH-evo and the WT in order to make a qualitative statement about the likely pathways usage with regards to the TCA cycle.

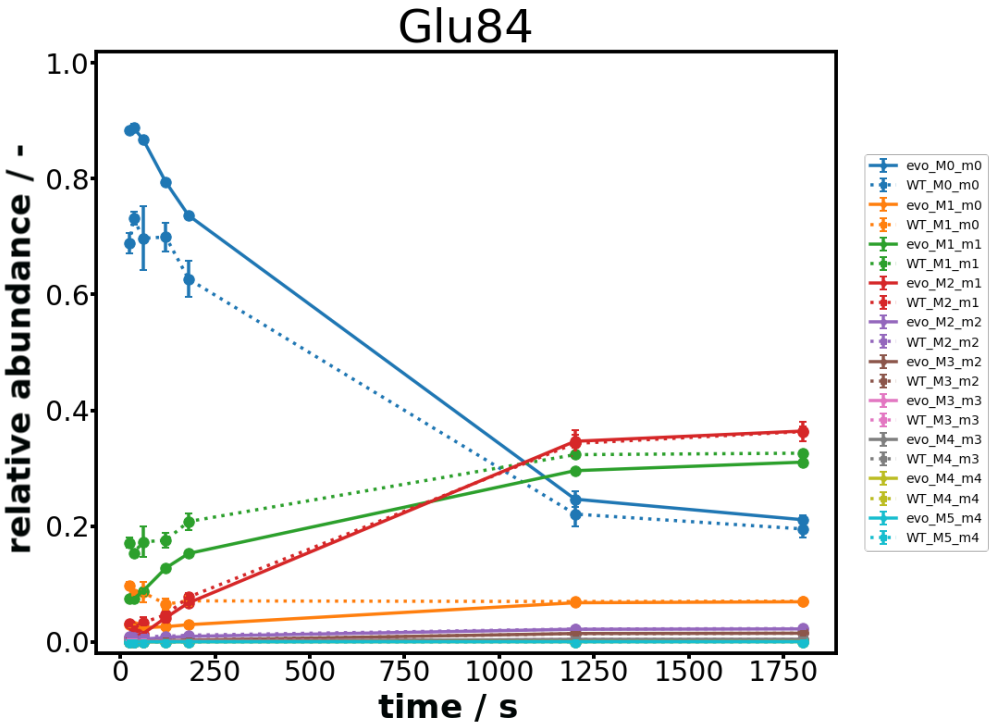


Figure A9: Label incorporation into Glu of *C. glutamicum* WT_ETH-evo (line) and the WT (dashed line) grown on unlabeled ethanol and spiked with 1-^{13}C ethanol during the mid-exponential growth phase. The dynamic is comparable but the WT exhibits a slight offset indicating a faster incorporation of labeled carbon atoms.

Table A1: List of maximum likelihood estimators of absolute and relative flux values as well as lower and upper bounds of the 95 % CoIs from the ethanol ^{13}C-INST MFA with *C. glutamicum* WT_EtOH-Evo. Absolute values are given in mmol g_X^{-1} h^{-1} and relative values in % of the total ethanol uptake rate. The ".n" suffix denominates net fluxes and ".x" exchange fluxes. "EtOH_U" refers to unlabeled ethanol and "EtOH" to 1-^{13}C ethanol.

reaction	MLE (abs)	95% CoI lb	95% CoI ub	MLE (rel)	95% CoI lb	95% CoI ub
CS.n	3.076	3.055	3.111	48.38	48.05	48.93
ENO.n	-1.239	-1.261	-1.217	-19.49	-19.82	-19.14
ENO.x	5.230	4.083	6.803	82.24	64.21	106.98
FBA.n	-0.835	-0.858	-0.811	-13.13	-13.50	-12.76
FBA.x	37.078	5.205	99.939	583.04	81.85	1571.53
Fum.n	2.898	2.872	2.935	45.56	45.16	46.16
Fum.x	36.340	21.470	99.938	571.44	337.62	1571.52
GAPDH.n	-1.077	-1.099	-1.055	-16.94	-17.28	-16.59
GAPDH.x	38.645	26.304	66.087	607.69	413.63	1039.21
ICD.n	0.752	0.730	0.775	11.83	11.49	12.18
ICL.n	2.324	2.303	2.346	36.54	36.21	36.89
ICL.x	0.000	0.000	0.061	0.00	0.00	0.95
MALS.n	2.324	2.303	2.346	36.54	36.21	36.89
MALS.x	0.003	0.001	0.177	0.04	0.01	2.79
ME.n	-1.856	-3.624	0.130	-29.18	-56.98	2.04
ME.x	0.001	0.001	0.099	0.01	0.01	1.56
MQO_MDH.n	3.366	1.585	5.351	52.93	24.93	84.14
MQO_MDH.x	49.737	33.903	99.902	782.12	533.12	1570.95
ODHC.n	0.497	0.476	0.519	7.82	7.48	8.17
PCx_ODX.n	-1.720	-1.822	-1.630	-27.05	-28.66	-25.63
PCx_ODX.x	0.001	0.001	0.094	0.01	0.01	1.47
PDHC.n	-0.057	-0.080	-0.036	-0.90	-1.25	-0.56
PDHC.x	0.000	0.000	0.021	0.00	0.00	0.33
PEPCk_PEPCx.n	-1.755	-1.856	-1.664	-27.59	-29.19	-26.17
PEPCk_PEPCx.x	0.010	0.001	0.104	0.15	0.01	1.64
PFK.n	-0.835	-0.857	-0.813	-13.13	-13.47	-12.78
PFK.x	21.058	0.993	99.923	331.13	15.61	1571.28
PGD.n	1.945	1.877	2.010	30.58	29.52	31.61
PGI.n	2.043	1.975	2.108	32.12	31.06	33.15
PGI.x	12.903	0.001	99.915	202.90	0.01	1571.16
PK_PPS.n	-3.102	-3.204	-3.011	-48.78	-50.38	-47.35
PK_PPS.x	0.610	0.422	0.904	9.59	6.64	14.22
PTA.n	6.418	6.418	6.474	100.93	100.93	101.81
RPE.n	1.245	1.200	1.289	19.58	18.88	20.27
RPE.x	10.205	0.001	99.912	160.48	0.01	1571.11
RPI.n	0.699	0.677	0.721	10.99	10.65	11.34
RPI.x	7.787	0.001	99.910	122.44	0.01	1571.08
SCS___1.n	0.214	0.203	0.225	3.37	3.20	3.54
SCS___1.x	0.152	0.136	0.170	2.39	2.14	2.68
SCS___2.n	0.214	0.203	0.225	3.37	3.20	3.54
SCS___2.x	0.152	0.136	0.170	2.39	2.14	2.68
SQO.n	2.821	2.795	2.859	44.36	43.96	44.95
SQO.x	38.998	22.336	99.940	613.24	351.24	1571.56
TAL.n	0.645	0.622	0.668	10.14	9.77	10.51
TAL.x	33.167	0.001	99.935	521.54	0.01	1571.47
TKT1.n	0.645	0.623	0.667	10.14	9.79	10.48

reaction	MLE (abs)	95% CoI lb	95% CoI ub	MLE (rel)	95% CoI lb	95% CoI ub
TKT1.x	23.761	0.001	99.926	373.65	0.01	1571.32
TKT2.n	0.600	0.578	0.622	9.44	9.09	9.79
TKT2.x	40.785	0.001	99.942	641.34	0.01	1571.58
TPI.n	-0.835	-0.857	-0.813	-13.13	-13.47	-12.78
TPI.x	41.337	5.712	99.943	650.03	89.82	1571.59
bmACCOA.n	0.835	0.835	0.844	13.13	13.13	13.27
bmALA.n	0.221	0.221	0.223	3.48	3.48	3.51
bmAMP.n	0.008	0.008	0.008	0.13	0.13	0.13
bmARG.n	0.036	0.036	0.036	0.57	0.57	0.57
bmASN.n	0.037	0.037	0.037	0.58	0.58	0.58
bmASP.n	0.037	0.037	0.037	0.58	0.58	0.58
bmCMP.n	0.008	0.008	0.008	0.13	0.13	0.13
bmCYS.n	0.008	0.008	0.008	0.13	0.13	0.13
bmDAP.n	0.019	0.019	0.019	0.30	0.30	0.30
bmF6P.n	0.038	0.038	0.038	0.59	0.59	0.59
bmG6P.n	0.098	0.098	0.099	1.54	1.54	1.55
bmGAP.n	0.013	0.013	0.013	0.21	0.21	0.21
bmGLN.n	0.065	0.065	0.065	1.03	1.03	1.03
bmGLU.n	0.124	0.124	0.124	1.94	1.94	1.96
bmGLY.n	0.067	0.067	0.067	1.06	1.06	1.06
bmGMP.n	0.011	0.011	0.011	0.18	0.18	0.18
bmHIS.n	0.013	0.013	0.013	0.20	0.20	0.20
bmILE.n	0.036	0.036	0.036	0.57	0.57	0.57
bmLEU.n	0.067	0.067	0.067	1.06	1.06	1.06
bmLYS.n	0.036	0.036	0.036	0.56	0.56	0.56
bmMET.n	0.014	0.014	0.014	0.23	0.23	0.23
bmPEP.n	0.019	0.019	0.019	0.30	0.30	0.30
bmPHE.n	0.024	0.024	0.024	0.38	0.38	0.38
bmPRO.n	0.030	0.030	0.030	0.48	0.48	0.48
bmSER.n	0.047	0.047	0.047	0.74	0.74	0.74
bmTHR.n	0.052	0.052	0.052	0.82	0.82	0.82
bmTRP.n	0.005	0.005	0.005	0.08	0.08	0.08
bmTYR.n	0.015	0.015	0.015	0.23	0.23	0.23
bmUMP.n	0.008	0.008	0.008	0.13	0.13	0.13
bmVAL.n	0.052	0.052	0.052	0.82	0.82	0.82
bsAICAR___1.n	0.010	0.010	0.010	0.15	0.15	0.15
bsAICAR___2.n	0.010	0.010	0.010	0.15	0.15	0.15
bsALA.n	0.221	0.221	0.223	3.48	3.48	3.51
bsAMP___1.n	0.010	0.010	0.010	0.16	0.16	0.16
bsAMP___2.n	0.010	0.010	0.010	0.16	0.16	0.16
bsARG___1.n	0.018	0.018	0.018	0.28	0.28	0.28
bsARG___2.n	0.018	0.018	0.018	0.28	0.28	0.28
bsASN.n	0.037	0.037	0.037	0.58	0.58	0.58
bsASP.n	0.324	0.324	0.329	5.10	5.10	5.17
bsCHOR.n	0.045	0.045	0.045	0.70	0.70	0.70
bsCITR.n	0.036	0.036	0.036	0.57	0.57	0.57

reaction	MLE (abs)	95% CoI lb	95% CoI ub	MLE (rel)	95% CoI lb	95% CoI ub
bsCMP.n	0.008	0.008	0.008	0.13	0.13	0.13
bsCYS.n	0.023	0.023	0.023	0.36	0.36	0.36
bsDAP___1.n	0.014	0.014	0.014	0.21	0.21	0.21
bsDAP___2.n	0.014	0.014	0.014	0.21	0.21	0.21
bsDAP___3.n	0.014	0.014	0.014	0.21	0.21	0.21
bsDAP___4.n	0.014	0.014	0.014	0.21	0.21	0.21
bsFAICAR.n	0.032	0.032	0.032	0.51	0.51	0.51
bsFGAM.n	0.020	0.020	0.020	0.31	0.31	0.31
bsGAR.n	0.020	0.020	0.020	0.31	0.31	0.31
bsGLN.n	0.245	0.245	0.247	3.85	3.85	3.89
bsGLU.n	1.248	1.248	1.266	19.63	19.63	19.91
bsGLY.n	0.087	0.087	0.087	1.37	1.37	1.38
bsGMP.n	0.011	0.011	0.011	0.18	0.18	0.18
bsHIS.n	0.013	0.013	0.013	0.20	0.20	0.20
bsHOM.n	0.103	0.103	0.103	1.61	1.61	1.62
bsI1.n	0.005	0.005	0.005	0.08	0.08	0.08
bsI2.n	0.005	0.005	0.005	0.08	0.08	0.08
bsIAP.n	0.013	0.013	0.013	0.20	0.20	0.20
bsILE.n	0.036	0.036	0.036	0.57	0.57	0.57
bsIMP.n	0.032	0.032	0.032	0.51	0.51	0.51
bsIND.n	0.005	0.005	0.005	0.08	0.08	0.08
bsKIV.n	0.119	0.119	0.121	1.88	1.88	1.90
bsLEU.n	0.067	0.067	0.067	1.06	1.06	1.06
bsLYS.n	0.036	0.036	0.036	0.56	0.56	0.56
bsMET.n	0.014	0.014	0.014	0.23	0.23	0.23
bsNCLA.n	0.017	0.017	0.017	0.26	0.26	0.26
bsORN.n	0.036	0.036	0.036	0.57	0.57	0.57
bsORO.n	0.017	0.017	0.017	0.26	0.26	0.26
bsPHEo.n	0.024	0.024	0.024	0.38	0.38	0.38
bsPRA.n	0.013	0.013	0.013	0.20	0.20	0.20
bsPRE.n	0.039	0.039	0.039	0.62	0.62	0.62
bsPRO.n	0.030	0.030	0.030	0.48	0.48	0.48
bsSER.n	0.162	0.162	0.163	2.55	2.55	2.57
bsSHKM.n	0.045	0.045	0.045	0.70	0.70	0.71
bsTHR.n	0.088	0.088	0.089	1.39	1.39	1.39
bsTRP.n	0.005	0.005	0.005	0.08	0.08	0.08
bsTYR.n	0.015	0.015	0.015	0.23	0.23	0.23
bsUMP.n	0.017	0.017	0.017	0.26	0.26	0.26
bsVAL.n	0.274	0.274	0.276	4.30	4.30	4.34
dummy.n	0.193	0.193	0.195	3.04	3.04	3.06
effTHF.n	0.021	0.021	0.021	0.32	0.32	0.32
excCO2.n	5.205	5.138	5.305	81.85	80.80	83.43
feedEtOH.n	4.353	4.311	4.393	68.45	67.78	69.08
feedEtOH_U.n	2.006	1.950	2.063	31.55	30.67	32.44
mu.n	0.193	0.193	0.195	3.04	3.04	3.06

Table A2: List of maximum likelihood estimators of pool sizes with lower and upper bounds of the 95 % CoIs from the ethanol ^{13}C-INST MFA with *C. glutamicum* WT_EtOH-Evo. All values were originally determined in mmol g_X^{-1} and converted to mM via the cellular volume of 0.00193 L_{cell} g_X^{-1} [49, 164].

name	MLE / mmol / g_X	95% CoI lb	95% CoI ub	MLE / mM	95% CoI lb	95% CoI ub
ACCOA	0.001	0.001	0.002	0.52	0.52	0.90
ACE	0.001	0.001	0.002	0.58	0.58	0.96
AICAR	0.014	0.002	0.749	7.23	0.94	388.23
AKG	0.001	0.001	0.001	0.52	0.52	0.52
ALA	0.023	0.021	0.026	12.03	10.95	13.50
AMP	0.096	0.002	0.749	49.91	0.90	388.27
ARG	0.032	0.002	0.749	16.36	1.01	388.24
ASN	0.013	0.002	0.749	6.76	0.91	388.23
ASP	0.004	0.004	0.004	1.88	1.88	1.88
CHOR	0.014	0.002	0.749	7.26	0.94	388.23
CITR	0.006	0.006	0.006	3.04	3.04	3.04
CIT_ICIT	0.001	0.001	0.002	0.52	0.52	0.90
CMP	0.065	0.002	0.749	33.48	1.03	388.25
CO2	0.750	0.706	0.750	388.50	365.77	388.50
CYS	0.046	0.002	0.749	24.03	0.89	388.25
DAP	0.054	0.047	0.064	27.96	24.10	33.24
DHAP	0.315	0.002	0.749	163.14	0.84	388.16
E4P	0.711	0.002	0.749	368.37	0.88	388.28
F6P	0.729	0.002	0.749	377.78	0.89	388.26
FAICAR	0.016	0.002	0.749	8.51	1.02	388.23
FBP	0.732	0.002	0.749	379.21	0.89	388.31
FGAM	0.031	0.002	0.749	15.87	1.00	388.24
FUM	0.001	0.001	0.002	0.52	0.52	1.28
G6P	0.735	0.002	0.749	380.68	0.89	388.11
GA3P	0.666	0.002	0.749	345.08	0.85	388.26
GAR	0.070	0.002	0.749	36.39	0.80	388.26
GLN	0.037	0.034	0.039	18.98	17.83	20.43
GLU	0.069	0.067	0.073	35.97	34.58	37.69
GLX	0.051	0.041	0.064	26.47	21.20	33.19
GLY	0.001	0.001	0.005	0.52	0.52	2.42
GMP	0.092	0.002	0.749	47.69	0.89	388.27
HIS	0.001	0.001	0.003	0.52	0.52	1.66
HSER	0.002	0.002	0.002	1.28	1.28	1.28
I1	0.048	0.002	0.749	25.07	0.90	388.25
I2	0.010	0.002	0.749	5.02	0.80	388.23
IAP	0.001	0.001	0.003	0.52	0.52	1.66
ILE	0.017	0.016	0.019	8.68	8.17	9.79
IMP	0.033	0.002	0.749	16.97	1.03	388.24
IND	0.007	0.002	0.749	3.75	0.92	388.23
KIV	0.001	0.001	0.005	0.52	0.52	2.42
LEU	0.064	0.002	0.749	33.08	1.03	388.25
LYS	0.001	0.001	0.002	0.52	0.52	0.90
MAL	0.001	0.001	0.002	0.52	0.52	1.28
MET	0.001	0.001	0.001	0.52	0.52	0.52
NCLA	0.057	0.002	0.749	29.55	0.97	388.25
OAA	0.001	0.001	0.002	0.52	0.52	1.28

name	MLE / mmol / g_X	95% CoI lb	95% CoI ub	MLE / mM	95% CoI lb	95% CoI ub
ORN	0.001	0.001	0.001	0.52	0.52	0.52
ORO	0.028	0.002	0.749	14.26	0.95	388.24
PEP	0.590	0.402	0.749	305.95	208.12	388.28
PGP	0.009	0.002	0.586	4.92	0.79	303.55
PHE	0.051	0.002	0.749	26.49	0.92	388.25
PRA	0.001	0.001	0.003	0.52	0.52	1.66
PRE	0.074	0.002	0.749	38.21	0.81	388.26
PRO	0.038	0.035	0.042	19.71	18.21	21.51
PYR	0.001	0.001	0.005	0.53	0.53	2.42
R5P	0.090	0.002	0.749	46.83	0.88	388.27
RU5P	0.726	0.002	0.749	376.18	0.88	388.21
S7P	0.009	0.002	0.749	4.90	0.79	388.23
SER	0.001	0.001	0.008	0.53	0.53	3.94
SHKM	0.107	0.002	0.749	55.41	0.95	388.28
SUC	0.001	0.001	0.002	0.52	0.52	1.28
SUCCOA	0.001	0.001	0.001	0.52	0.52	0.52
THF	0.020	0.002	0.749	10.30	0.82	388.23
THR	0.002	0.002	0.002	0.91	0.91	0.91
TRP	0.030	0.002	0.749	15.65	0.99	388.24
TYR	0.020	0.002	0.749	10.57	0.83	388.23
UMP	0.017	0.002	0.749	8.91	0.78	388.23
VAL	0.004	0.004	0.005	1.97	1.97	2.35
XU5P	0.722	0.002	0.749	374.25	0.88	388.15