Absolute quantification of peptide products in *in vitro* digestions: A computational approach using Bayesian inference

Dissertation

for the award of the degree
“Doctor rerum naturalium”
of the Georg August University of Göttingen

within the doctoral programm IMPRS Genome Science
of the Georg August University School of Science (GAUSS)

submitted by
Sarah Henze
from Seesen, Germany

Göttingen, Germany, May 2022

Thesis Advisory Committee

Dr. Juliane Liepe (Reviewer)
Research group Quantitative and Systems Biology,
Max Planck Institute for Multidisciplinary Sciences, Göttingen

Prof. Dr. Peter Sollich (2nd Reviewer)
Research group Non-equilibrium Statistical Physics,
Institute for Theoretical Physics, University of Göttingen

Prof. Dr. Lutz Walter
Department of Primate Genetics, German Primate Center, Göttingen

Further members of the Examination Board

Prof. Dr. Henning Urlaub
Research group Bioanalytical Mass Spectrometry,
Max Planck Institute for Multidisciplinary Sciences, Göttingen
and Institute for Clinical Chemistry, University Medical Center, Göttingen

Prof. Dr. Tim Beißbarth
Department of Medical Bioinformatics, University Medical Center, Göttingen

Dr. Johannes Soeding
Research group Computational Biology,
Max Planck Institute for Multidisciplinary Sciences, Göttingen

Date of the oral examination: 04 July 2022

The fundamental problem of scientific progress,
and a fundamental one of everyday life,
is that of learning from experience.

– Harold Jeffreys, Theory of Probability, 1961 [1].

Abstract

Insight into enzyme specificities and dynamics is central to understanding biochemical processes. The peptide products generated by purified proteases in *in vitro* digestions are often identified by mass spectrometry measurements. However, these provide only relative quantification and to obtain absolute quantities, laborious titration of synthetic peptide equivalents is required. Our aim is to develop a method to convert MS ion signals to concentrations for many peptide products computationally without further experimental effort. To achieve this, a conversion parameter for each digestion product needs to be estimated. We present an algorithm named Quantification of Peptides using Bayesian inference (QPuB), which works on the principle of mass conservation. It employs Bayesian statistical inference in an adaptive, population-based Markov chain Monte Carlo sampling scheme to estimate the conversion factors. This approach allows to quantify the underlying uncertainty in the form of full posterior distributions of the estimated parameters. We calibrated the algorithm on synthetic noise-free datasets mimicking the dynamics of real proteases. For low-informative data causing parameter non-identifiability, we propose strategies to enable successful inference. We show that QPuB is able to infer the conversion factors for up to 45 peptides with high accuracy and precision. Although the algorithm still requires further development, we believe that QPuB could become a useful quantification tool to the field of peptidomics.

Keywords: Label-free quantification, absolute quantification, *in vitro* digestion, peptidomics, mass spectrometry, Bayesian inference, Markov chain Monte Carlo, Differential Evolution.

Acknowledgments

“Which is more important,” asked Big Panda, “the journey or the destination?”
“The company.” said Tiny Dragon. – James Norbury, The company.

I would like to thank Dr. Juliane Liepe for giving me the opportunity to do my PhD with her group and for her supervision and support. Thank you for introducing me to the field of Bayesian inference, for the chance to widen my scientific horizon and for the possibility to attend numerous conferences.

I want to express my gratitude to my thesis committee members Prof. Dr. Peter Sollich and Prof. Dr. Lutz Walter for their valuable advice and input to the project and their moral support and encouragement over the years. Furthermore, I want to thank Prof. Dr. Henning Urlaub, Prof. Dr. Tim Beißbarth and Dr. Johannes Soeding for agreeing to be part of my extended examination board.

I am thankful to Prof. Dr. Henning Urlaub for providing access to the mass spectrometry facilities, to Dr. Michele Mishto and his lab in London, as well as Artem Mansurkhodzhaev for providing input and data for QPuB.

I would like to acknowledge the funding of the International Max Planck Research School for Genome Science. A warm thank you to the coordinators Henriette Irmer, Katja Lidschreiber and Frauke Bergmann for their support and to all the people who fill the program with life.

I thank the current and former members of the Quantitative and Systems Biology group for the interdisciplinary and fun working environment. Special thanks to Artem and Wai Tuck for their patient explanations on work in the lab. Many thanks also to the Proteomics group for fruitful discussions over lunch and showing me around the mass spectrometry lab. I highly recognise

the input and discussion with Étienne Morice and Alexander Wulf, you have really helped me develop a deeper understanding of the important aspects of my work.

I want to express my gratitude to everyone who proof-read parts of this thesis and provided helpful corrections and suggestions. Thank you (in the order of the chapters) to Nyet Cheng Chiam, Alexander Wulf, Andreas Linden (also for testing the QPuB package), Artem Mansurkhodzhaev, Wai Tuck Soh, Hanna Rötschke, Lara Köpping, Yehor Horokhovskyi, John Cormican, Étienne Morice, Guus Bertens, Linda Olsthoorn, Ronja Markworth, and Kristian Blom.

Thank you to Niklas Korsbo for fruitful conversations at conferences and introducing me to useful julia packages. I want to thank Gerrit Wellecke for last minute help on LaTeX emergencies and to Cheng and Wai Tuck for pushing me on the last meters.

Special thanks to Annika and Max, my fellows in misery, for pushing and pulling each other through the rocky time of writing. Finally, to my family and friends for reminding me of a life outside the office, for the appreciated distractions and the unconditional encouragement.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Systems biology is the endeavour to understand dynamic biological processes using mathematical and computational approaches. Through the development of models, simulations and data analysis, complex interactions can be analysed and differences under different conditions recognised [2]. The key players of biological processes are **proteins**. Their function is regulated by many factors, from synthesis through inhibition and enhancement to degradation. Proteins can be degraded by other proteins, a type of enzymes referred to as **proteases**. Upon binding, the so-called substrate is disassembled into smaller fragmental products down to their individual components [3]. This reaction can be simplified as

$$\text{Enzyme} + \text{Substrate} \rightarrow \text{Enzyme} + \text{Products}.$$

The degradation products are called **peptides**. These can be functional and their activity depends on environmental conditions and is subject to stimuli. The large-scale study of peptides is called **peptidomics**. Understanding their role in the cell dynamics is of increasing interest in fundamental as well as medical research. For example, proteases and peptides play an important part in the adaptive immune system. Inside human body cells, specific proteases named proteasomes are responsible for the degradation of proteins. Following the antigen processing and presentation pathway, some of the degradation products are presented at the cell surface and recognised by immune cells. If the presented peptides are of viral origin, an immune response can be triggered [4]. Proteases as well as peptides can be useful candidates in drug development for a variety of diseases, including diabetes, cardiovascular diseases and cancer [5, 6, 7].

In response towards environmental conditions, protein expression or certain reactions are usually not simply switched on or off, but it rather results in a gradual change in protein and peptide concentration. Consequently, the research interest experiences a shift from not only identifying but also quantifying them. **Quantification** therefore became crucial for the holistic description of the protein dynamics. *In vitro* experiments provide a useful analysis alongside *in cellulo* and *in vivo* studies [8]. Unfortunately, with today's techniques, a direct observation of peptide concentrations in a large complex sample is not possible. Instead, related quantities can be measured to indirectly infer the peptide amounts [9]. **Mass spectrometry** provides an automated, sensitive and high-throughput technology for protein and peptide identification. A protein is digested *in vitro* by a protease and the resulting mixture of peptides is measured over time in a mass spectrometer. The sample is ionized and subjected to an electric field, which causes the ions to be deflected according to their mass and charge. From the induced signal intensities, the peptide identity can be inferred [10]. For absolute quantification of the peptides, a subsequent titration of synthetic peptide equivalents has to be performed. By loading different amounts of the peptide and measuring the corresponding signal response, a linear relationship between the two quantities can be observed [11, 12, 13, 14, 15, 16]:

$$\text{signal} = \text{response factor} * \text{amount}$$

Through calculation of the respective response factor specific to each peptide, the peptide amounts in the digest can be inferred. However, this procedure is laborious and time-consuming, therefore not feasible for large scale peptidomics analyses. Due to the linearity, the signal intensities can also be used directly for relative quantification. It is possible to compare the intensities of the same peptide between samples under different experimental conditions, to draw qualitative conclusions about the effect. However, the behaviour of the peptide ions in the mass spectrometer also depends on peptide properties other than mass, which makes a comparison of intensities of different peptides among one another impossible [17, 18, 19, 20]. Another way to achieve absolute quantification is to add a known isotope-labelled or unlabelled standard to the sample for every peptide of interest. This allows to approximate the product amount by comparison to the known amount injected [21]. This technique requires additional experimental effort, which becomes infeasible if many peptide products must be quantified. With the vast amount of data generated by today's mass spectrometry hardware, the necessity for powerful computational approaches increases.

Over the years, different tools improving the accuracy of standard label-based and label-free techniques have been developed. However, many approaches still require labelling [22, 23], only focus on relative quantification [24, 25], or aim at absolute quantification of proteins rather than peptides [8, 26, 27, 28, 29]. Attempts have been made to predict the response factor based on the physico-chemical properties of the analyte [30, 31, 32, 33]. However, to our knowledge, there are no publications solving the problem of automated, label-free, absolute quantification of peptide products without further experimental effort. Another approach, if direct or indirect measurement of parameters of a system is difficult, is **parameter estimation** using analytical or computational approaches. A model is proposed, which links the quantity of interest with the observable quantity. The optimal parameter values minimise the deviation between the predicted model output and the experimentally measured data, which is defined by an objective function [34, 35, 36]. In 2002, Peters et al. [37] followed this approach and published their **mass balance method** for response factor estimation. Making use of the linear relationship they aimed to derive the response factor (or signal conversion coefficient, as they call it) for each peptide product from the kinetic signal data. The main principle they applied is the fact that in a closed system of *in vitro* digestions the mass conservation holds at every point in time. All the amount of substrate degraded must be equal to the amount of peptides produced, no mass should be lost and no additional mass can be generated. With this, they relied on the assumption that all peptides are detected and identified and that no mass loss is happening over the course of the experiment. They defined an optimization problem of finding the set of conversion factors that minimises the deviation of the amount of peptides produced from the amount of substrate degraded. Subsequently, from the estimated conversion factors the peptide concentrations of every peptide products can be calculated using only the conversion factor of the substrate, which needs to be measured through titration. Their approach significantly reduces the experimental effort needed for reliable quantification of all major peptides in a sample. Peter's mass balance method has been applied [38] and further developed by Mishto et al. [39]. In 2012, they published **QME**, **Q**uantification with **M**inimal **E**ffort. They introduced some changes to the optimisation problem and used the downhill simplex method to solve it. Comparison of their results obtained by the QME method to those measured via titration indicated agreement.

Using the concept of mass conservation provides a reliable way to estimate the conversion factors of many peptide products and to infer the peptide amounts without the need of cumbersome peptide titrations, sample labelling or introduction of internal standard peptides. However, the QME tool was tuned specifically for a certain analysis on a particular instrument and has since not been applied often and was never properly benchmarked. A drawback of their implementation is its dependence on instrument features of the mass spectrometer used, which need to be provided by the user obtained through laborious calibration. A larger difficulty is faced for samples analysed with modern mass spectrometers. The high accuracy of recent hardware leads to an almost-complete identification of the set of products and large and complex peptide mixtures are common for many applications. A high-dimensional problem like this can become challenging to solve with QME. Another disadvantage of optimisation methods is that single point estimates do not convey the uncertainty that still comes with the systematic and random errors in the mass spectrometry measurements. A better computational approach is required.

1.1 Aim and approach of this thesis

In this thesis, we present **QPuB**, **Q**uantification of **P**eptides **u**sing **B**ayesian inference. Building on the ideas of Peters et al. [37] and Mishto et al. [39], we use the linear relationship between signal intensities and amounts as well as the concept of mass conservation to convert MS ion peak areas to absolute peptide amounts with no or little further experimental effort. Our framework does not require instrument specific settings and provides an easy-to-use analysis tool. For the estimation of the conversion factors, instead of an optimisation routine, we employ likelihood-based Bayesian inference in a Markov chain Monte Carlo scheme. Bayesian inference is a statistical technique used to combine prior knowledge about the parameters and gained knowledge through experimental data to obtain educated estimates. The quality of the estimates is assessed through the definition of an objective function, called likelihood. In this case, it controls the deviation from mass balance. Using an iterative sampling scheme, parameter values are proposed and accepted or rejected based on their likelihood [40]. In this way, instead of finding a sole minimizer of the objective function, a full distribution of possible values for every conversion factor is obtained. This conveys a measure for the uncertainty in the estimate. The underlying implementation is an adaptive population-based algorithm, called Differential Evolution Markov chain suitable for large dimensions, developed by Ter Braak and Vrugt et al. [41, 42, 43, 44, 45]. Using the inferred conversion factor distributions, distributions of normalised signal intensities for all peptide products can be calculated, which allow for relative quantification between different peptides. If, in addition, the experimentally measured conversion factor of the substrate is provided, distributions for the amounts of all peptide products are returned, enabling absolute quantification. Our algorithm is calibrated and benchmarked on noise-free simulated data. If sufficient information about the parameters is conveyed by the data, QPuB is able to reliably infer the correct conversion factors with high accuracy and precision for up to 50 products, and possibly more. It has the potential to add a useful and user-friendly tool to the pool of methods for absolute quantification in peptidomics.

1.2 Structure of this thesis

The remainder of this thesis is structured as follows. Following chapter 1, the second chapter introduces the relevant biological background of quantitative peptidomics including a description of tandem mass spectrometry. In the third chapter, we describe key concepts of Bayesian inference and Markov chain Monte Carlo schemes, in particular the underlying Differential Evolution Markov chain algorithm. We provide an introduction to theoretical considerations to assess parameter identifiability of a mathematical system and explain the benchmarking framework applied in this thesis. The fourth chapter presents the implementation of the QPuB package. In Chapter 5, we calibrate and benchmark the QPuB algorithm using noise-free simulated data and discuss its performance. We show examples where parameter inference is successful and the peptides' conversion factors can be estimated with high precision and accuracy. We also present examples, where the provided data is not sufficient to enable successful inference. Two strategies to overcome these limitations are presented. Chapter 6 provides a discussion of the results, the limitations and an outlook. Lastly, we conclude this thesis with a summary.

2 | Experimental background

The central dogma of molecular biology describes how genetic information flows inside a cell. It was first stated by Francis Crick in 1957 [46]. The proteome is the entire set of proteins expressed from the genome. The word ”proteome” was coined in 1995 by Wilkins et al., as a fusion of the words ”protein” and ”genome” [47, 48, 49]. With the human genome containing around 21 000 genes [50, 51], the number of unique proteins is estimated to be a few millions, considering alternative splicing, polymorphisms and post-translational modifications [52]. Only an estimated fraction of 35% of all proteins is conserved between cell types, the so-called housekeeping proteins [53] Protein expression is cell type dependent and varies in time as a response to environmental conditions, stress or diseases [54]. It can fluctuate several orders of magnitude from a few copies per cell up to several millions [55, 56]. Proteins have vital functions within cells, ranging from providing structure over signalling and transporting cargo to catalyzing metabolic reactions. The proteins of interest in this thesis are a special kind of enzymes, which degrade other proteins into smaller fragments, so called proteases [3].

2.1 Protein chemistry

Proteins are large biomolecules that consist of one or more chains of amino acid residues, folded into a three-dimensional structure [57, 3]. **Amino acids** are organic molecules, consisting of a central carbon atom with an amino and a carboxylic group attached. Structure, properties and function of the amino acid are defined by a side chain (see Fig. 2.1A). The genetic information of a cell can encode twenty canonical amino acids with different properties [57, 3]. Amino acids are linked to each other via peptide bonds. Through a condensation reaction, a covalent bond between the nitrogen atom of the amino group of one amino acid and the carbon atom of the carboxylic group of a second amino acid is formed. A connection of at least two amino acid residues is called a **peptide** (see Fig. 2.1B) [3]. **Proteins** are folded polypeptides consisting of ten or more amino acid residues with a molecular weight of more than 10 000 Da, but definitions vary [57]. The amino acid sequence is the unique primary structure of a protein. Folding and coiling by forming hydrogen bonds across the peptide backbone results in the secondary structure. The tertiary structure is the unique three-dimensional shape of the molecule and determines the protein's identity [57, 3]. Chemical changes to the amino acid chain after synthesis are called **post-translational modifications** (PTM). Common examples are phosphorylation, oxidation or methylation, among others. These modifications have a major impact on the protein's function [58].

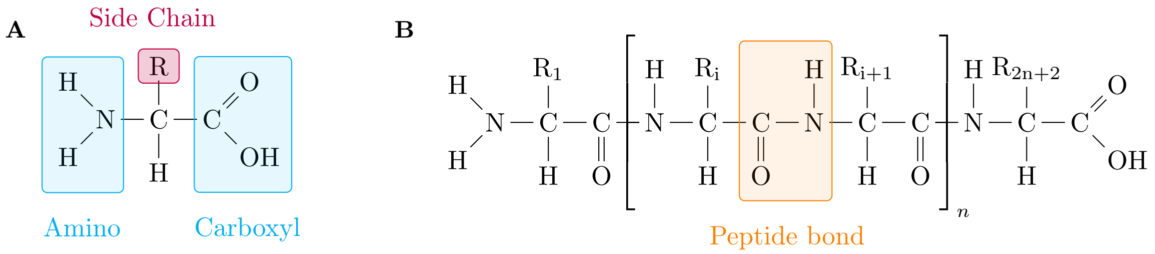


Fig. 2.1: General structural formulas of amino acids and polypeptides. (A) General structural formula of an amino acid with amino ($-NH_2$) and carboxylate ($-COOH$) functional groups and the side chain R specific to the amino acid. **(B)** General structural formula of a polypeptide of length 2n+2. Amino acid residues are connected via peptide bonds. The end finishing with an amino group is called N-terminus, the other end with the carboxylate group is called C-terminus [3].

2.2 Protease dynamics

Proteins can be categorised by their structure or their function. The protein class of interest in this thesis are enzymes, which act as **biological catalysts** [3]. They bind to other molecules (substrates) and accelerate their structural conversion into different molecules (products) by lowering the activation energy of the reaction. The enzyme itself remains unchanged in this reaction. Their structure has a particular conformation called the active site, which contains a binding site to bind the substrate and a catalytic site where the catalytic reaction occurs. Depending on the kind of reaction the enzymes catalyse, they are categorised into different classes. Hydrolases, for example, break chemical bonds by addition of a water molecule [3]. A special kind of hydrolases are **proteases** (also called peptidases). They break down proteins by hydrolysing their peptide bonds (proteolysis) [3]. The half life of a peptide bond can normally be years and proteases increase the rate of the hydrolysis by multiples [59]. We distinguish exopeptidases, which hydrolyse N-terminal peptide bonds, cleaving off single amino acid residues, and endopeptidases, which break peptide bonds along the protein backbone, releasing shorter peptide fragments. When a protein is fragmented by an enzyme into smaller chains of amino acids, we speak of an **enzymatic digest** (see Fig. 2.3) [3]. The mathematical framework of enzymatic digests will be explained in Section 2.2.3.

2.2.1 Trypsin

Trypsin is the most commonly used protease in MS-based proteomics [60]. It was discovered by the German physiologist Wilhelm Kühne in 1876 [61]. Acting as a digestive enzyme in the duodenum, trypsin breaks down proteins into smaller peptides to facilitate absorption. Trypsin is routinely applied in bottom-up proteomics to achieve a controlled digest of proteins for analysis [10]. Its advantage lies in its specificity: It mainly hydrolyses the peptide bond C-terminally of the basic amino acids arginine and lysine [60]. This results in highly regular protein fragments with a length of about 10 amino acid residues, depending on the amino acid composition of the substrate [62].

2.2.2 Proteasome

The proteasome is a multi-catalytic endopeptidase. It degrades damaged proteins inside cells and prevents old proteins from accumulating. In addition, it plays an important role in the adaptive immune system by processing peptide antigens in preparation for display by major histocompatibility complex class I (MHC) proteins on the surface of cells. Its existence was hypothesised in 1977 by Joseph Etlinger and Alfred L. Goldberg [64] and proven in 1978 by Avram Hershko, Ahron Ciechanover and Irwin Rose [65] (Nobel price in 2004 [66]). In 1994, the structure of the proteasome was solved [67]. In eukaryotic cells, there are two types of proteasomes with different **structures**, called the 20S and the 26S proteasome (see Fig. 2.2A). The 26S proteasome is a large protein complex consisting of a cylindrical 20S core particle capped with a regulatory 19S particle on at least one side [68, 69]. The core is composed of four rings of seven subunits each stacked into a barrel-like shape creating a chamber. The two outer rings (α-subunits) maintain the structure, the two inner rings (β-subunits) contain three to seven active sites on their inner surface [70, 69]. Based on the isoforms of these catalytic subunits, we can distinguish different types of

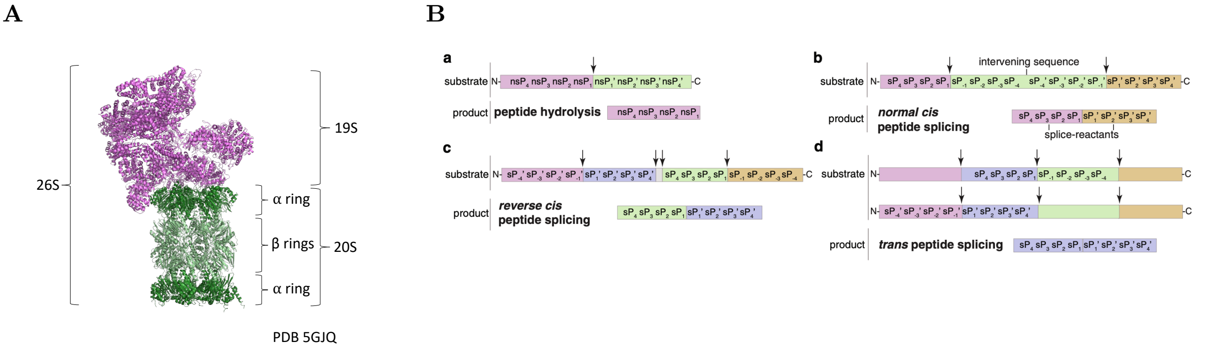

A
26S
19S
α ring
β rings
20S
α ring
PDB 5GJQ
B
a
substrate N- $nsP_4\ nsP_3\ nsP_2\ nsP_1$ $nsP_1'\ nsP_2'\ nsP_3'\ nsP_4'$ -C
product **peptide hydrolysis** $nsP_4\ nsP_3\ nsP_2\ nsP_1$
b
intervening sequence
substrate N- $sP_4\ sP_3\ sP_2\ sP_1$ $sP_{-1}\ sP_{-2}\ sP_{-3}\ sP_{-4}\ sP_{-4}'\ sP_{-3}'\ sP_{-2}'\ sP_{-1}'$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$ -C
product ***normal cis* peptide splicing** $sP_4\ sP_3\ sP_2\ sP_1$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$
splice-reactants
c
substrate N- $sP_{-4}'\ sP_{-3}'\ sP_{-2}'\ sP_{-1}'$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$ $sP_4\ sP_3\ sP_2\ sP_1$ $sP_{-1}\ sP_{-2}\ sP_{-3}\ sP_{-4}$ -C
product ***reverse cis* peptide splicing** $sP_4\ sP_3\ sP_2\ sP_1$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$
d
substrate N- $sP_4\ sP_3\ sP_2\ sP_1$ $sP_{-1}\ sP_{-2}\ sP_{-3}\ sP_{-4}$ -C
N- $sP_{-4}'\ sP_{-3}'\ sP_{-2}'\ sP_{-1}'$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$ -C
product ***trans* peptide splicing** $sP_4\ sP_3\ sP_2\ sP_1$ $sP_1'\ sP_2'\ sP_3'\ sP_4'$

Fig. 2.2: Structure of the 26S proteasome and schematic of proteasome-generated peptide products. (A) Cryo-EM structure of the human 26S proteasome. Image generated using PyMOL version 2.4. **(B)** Schematic of proteasome-generated peptide products. Figure reprinted with permission from [63].

20S proteasomes. Together with different types of regulatory particles, the proteasome complexes can have different dynamics and cleavage preferences [69]. The **mechanism** by which the 26S proteasome identifies and degrades target proteins is called ubiquitin-dependent degradation. Target proteins are tagged with the molecular label ubiquitin for degradation. This allows the regulatory 19S unit of the proteasome to recognise the protein as substrate. After deubiquitination, it partially unfolds the protein and transfers it to the inside of the 20S chamber. Upon binding to the active sites, degradation of the substrate occurs through hydrolysis of the peptide bonds. The different β-subunits have slightly different substrate specificities, resulting in the preferred proteolysis after different amino acid residues. Shorter peptides are released into the cytosol [66, 69]. The 20S proteasome can also exist freely. It makes up around half the proteasome pool in a cell [71]. Missing the 19S regulatory unit for ubiquitin recognition, it can perform ubiquitin-independent protein degradation [72].

Compared to tryptic digests, protein degradation by proteasomes is more diverse, resulting in a **peptide repertoire** that is very large. A special mechanism of proteasomes increases the pool of possible peptide products even more: In addition to canonical peptide bond hydrolysis resulting in proteasomal cleavage products (PCP), the proteasome has the ability to re-ligate created peptide products in a process called proteasome-catalysed peptide splicing (PCPS). This phenomenon was first described by Vigneron et al. in 2004 [73], and confirmed and further investigated by many [74, 75, 76, 77]. Two non-contiguous fragments, called splice-reactants, can be fused to form a proteasome-generated spliced product (PSP) with a sequence not contiguously present in the parental molecule [73, 78, 79]. Although PCPS can occur via condensation [80], splicing via transpeptidation [73] is probably predominant [79]. Three types of PSPs are currently distinguished (Fig. 2.2B). When both fragments stem from the same substrate molecule, the product is denoted *cis*-PSP. Normal *cis*-PCPS follows the orientation from N- to C-terminus of the parental protein, whereas reverse *cis*-PCPS ligates fragments in the reverse order. When the fragments stem from two distinct substrates, the product is called *trans*-PSP [74, 75, 39, 81].

2.2.3 Michaelis–Menten enzyme kinetics

Enzyme kinetics is a field in biochemistry that describes the catalysed processing of substrates by enzymes and quantitatively investigates the change of reactions rates under different conditions [82]. The experimental and theoretical base for it was laid in the early 20th century and is still standard today. Leonor Michaelis and Maud L. Menten built on the work of many others, most importantly Victor Henri [83], when they published their findings on initial-rate methods for steady-state enzyme-catalysed reactions in 1913 [84]. Their ideas were further developed by George E. Briggs and John B.S. Haldane [85], who generalised the concept to the Henri–Michaelis–Menten kinetics that is widely used [86].

The model assumes *in vitro* enzymatic reactions under controlled, well-mixed conditions. The number of molecules should be great enough so that the distribution of reactants can be assumed to be a continuum. The stochastic behaviour can then be approximated by deterministic dynamics, which are easier to analyse. The initial amount of the substrate in the solution should be much larger than the amount of enzyme [87, 3]. The major finding of Michaelis and Menten was, that the binding of the substrate to the enzyme can be split into two stages: a rapid reaction forming a low-energy enzyme-substrate complex, and a slower reaction catalysing the hydrolysis of the substrate into product fragments [87]. Denoting the enzyme by E, the substrate by S, the enzyme-substrate complex by ES and the generated products by P_1 and P_2, an elementary **enzymatic hydrolysis reaction** can be approximated as follows [87]:

$$E + S \underset{k_{\text{off}}}{\overset{k_{\text{on}}}{\rightleftharpoons}} ES \xrightarrow[k_{\text{cat}}]{} E + P_1 + P_2. \tag{2.1}$$

The constants k_{on} and k_{off} are called forward rate and reverse rate respectively, and k_{cat} is the catalytic rate constant. The reverse catalytic step can be neglected under small product concentrations [87, 3]. Depending on the specificity of the enzyme of interest, different reactions yielding distinct peptide products P_d $(d = 1, \dots, D)$ can be combined.

This deterministic reaction is commonly represented as a system of ordinary differential equations (ODE). The law of mass action states that the rate of a reaction is proportional to the product of the concentrations of the reactants. This gives the following set of **coupled Michaelis–Menten ODEs** for the concentrations of the reactants over time ($[X]$ denotes the concentration of reactant X, t denotes time) [87]:

$$\frac{\mathrm{d}[S]}{\mathrm{d}t}(t) = -k_{\text{on}}[S](t)[E](t) + k_{\text{off}}[ES](t) \tag{2.2a}$$

$$\frac{\mathrm{d}[ES]}{\mathrm{d}t}(t) = \quad k_{\text{on}}[S](t)[E](t) - (k_{\text{off}} + k_{\text{cut}})[ES](t) \tag{2.2b}$$

$$\frac{\mathrm{d}[E]}{\mathrm{d}t}(t) = -k_{\text{on}}[S](t)[E](t) + (k_{\text{off}} + k_{\text{cut}})[ES](t) \tag{2.2c}$$

$$\frac{\mathrm{d}[P_1]}{\mathrm{d}t}(t) = \quad k_{\text{cat}}[ES](t) \tag{2.2d}$$

$$\frac{\mathrm{d}[P_2]}{\mathrm{d}t}(t) = \quad k_{\text{cat}}[ES](t) \tag{2.2e}$$

together with the initial concentrations at the start of the reaction $[S](0)$, $[E](0)$, $[ES](0)$, $[P_1](0)$, $[P_2](0)$. An illustration of the resulting concentration kinetics over time is sketched in Fig. 2.3.

In an *in vitro* reaction compartment, mass balance must be conserved. The amount of free and bound enzyme involved in the reaction must be kept constant and the amount of substrate can only be converted into product mass. This leads to the following **conservation conditions** [87]:

$$[E](t) + [ES](t) = [E](0) \tag{2.3a}$$

$$[S](t) + [ES](t) + [P_1](t) + [P_2](t) = [S](0). \tag{2.3b}$$

To date, no analytical solution of the system has been derived. However, for given kinetic parameters and initial conditions of the reactants, numerical solutions can be calculated [87], as described in Sec. 4.8.1.

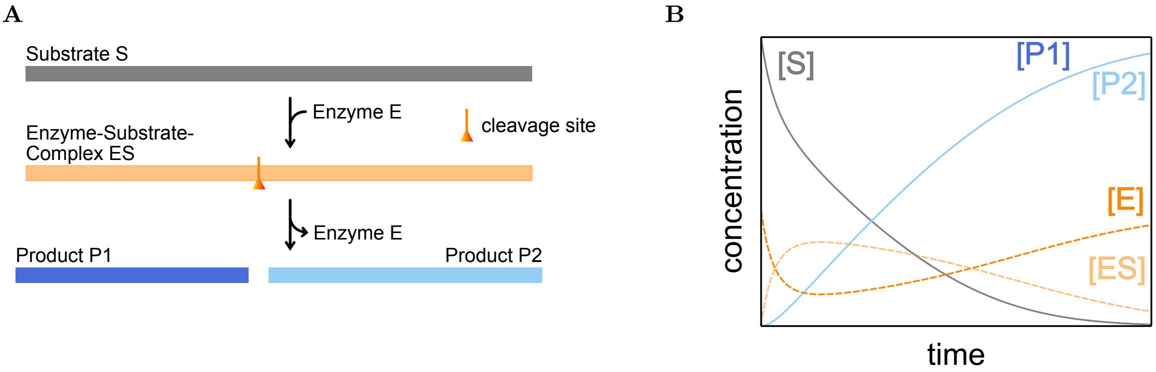


Fig. 2.3: Schematic of a Michaelis–Menten kinetics. (A) Cleavage pattern. Hydrolysis of substrate S by enzyme E into products P_1 and P_2, through an intermediate enzyme-substrate complex ES. In the following, the N-terminal and C-terminal cleavage products created by a hydrolysis reaction will be denoted *sibling peptides*. **(B) Concentration kinetics.** Concentrations of all reactants over time. By design both products have the same concentration and therefore appear as one. [Figure (B) modified from U+003F – Own work, CC0, `https://commons.wikimedia.org/w/index.php?curid=15943986`]

2.3 Mass spectrometry-based peptidomics

Proteomics is the large-scale investigation of the proteome. Its aim is the identification of the protein sequences, their role in reactions and their abundances in order to understand their functions in the biological system [88]. A field adjacent to proteomics is **peptidomics**, the large-scale study of peptides [89]. Protein analysis techniques have evolved dramatically over the past seventy years. Early approaches involved manual chemical analyses of few proteins that were cumbersome and time-consuming [57]. One of the earliest methods for **protein identification** was the Edman degradation, where the protein is sequenced by repeatedly cleaving off the N-terminal amino acid residue and identifying it [90, 91]. Nowadays, current instruments allow to analyse even low amounts of complex protein mixtures in short time [89]. A high-throughput proteomic technology often applied is **mass spectrometry** (MS). In the ”top-down” approach, intact proteins are purified and measured in MS, whereas in the ”bottom-up” approach, the proteins are subjected to enzymatic digestion (usually using specific proteases such as trypsin) and the generated peptides are analysed in MS. From the identified peptides the parental protein

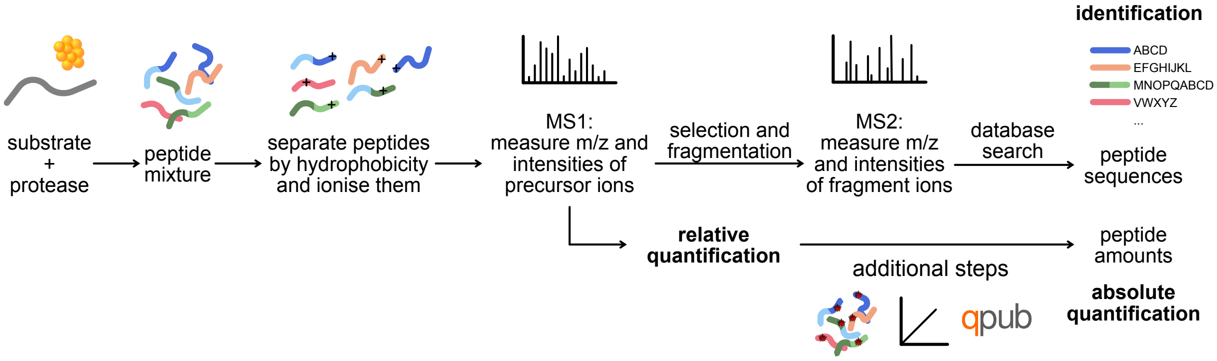


Fig. 2.4: Schematic of an MS-based peptide identification and quantification pipeline. A polypeptide is digested *in vitro* by a protease (Sec. 2.2). The peptide mixture is measured in liquid chromatography tandem mass spectrometry. Resulting mass spectra are used to identify the peptide sequences via database search (Sec. 2.3.2). Based on the signal intensities, a label-free relative quantification between same peptides is possible. For absolute quantification, additional steps involving labelling techniques or titrations are required (Sec. 2.3.3). QPuB provides a computational alternative to these experimental efforts. [Figure inspired by [20].]

can be identified [10]. Mass spectrometry can also be used for peptidomics research. In a bottom-up approach, a protein is *in vitro* digested by a protease of interest and the peptides in the sample are identified. Automated high-throughput technologies accompanied with computational pipelines allow for rapid analysis of large and complex peptide mixtures [92].

2.3.1 Tandem mass spectrometry

Mass spectrometry (MS) is a laboratory technique to measure the mass-to-charge ratios of ions. Its invention can be dated back to the beginning of the last century, when Joseph John Thomson and Francis Wiliam Aston built the first mass spectrographs for measuring isotopes of chemical elements [93, 94, 95, 96, 97, 98, 99, 100]. A sample of particles of interest is ionised and exposed to an electric or magnetic field, which deflects them based on their mass-to-charge-ratio. Today, mass spectrometry is a technique indispensable for the identification and structural determination of unknown compounds in a wide range of applications, one important field of which is peptide and protein identification [92]. A schematic of an MS-based peptide identification pipeline is shown in Fig. 2.4.

A mass spectrometer mainly consists of three components: an ion source to ionise the sample, a mass analyser to accelerate and select the ions, and a detector to measure the resulting currents. Usually, the sample intended for MS analysis is a complex compound mixture. This makes it difficult to distinguish between similar masses. To avoid overcrowding at the inlet of the mass spectrometer, the mass spectrometer can be coupled on-line with **column chromatography** [101, 92]. With this, separation of the compounds by physico-chemical properties is achieved, before they are introduced to the ion source. Today's standard is high performance liquid chromatography (HPLC), which separates the peptides by e.g. their hydrophobicity. The separation of the peptide mixture eluted over retention time can be visualised in an ion chromatogram [102, 10]. The first component of a mass spectrometer is the **ion source**, which ionises the sample and transfers it into gas phase. There are a variety of techniques to achieve ionisation. One of the first was electron ionisation, where the sample is bombarded with electrons. Applying this method to peptides would cause these fragile molecules to break. A frequently used soft ionisation method without breaking the peptide bonds is called electrospray ionisation [103, 104, 105] (Nobel price in 2002 [106]). Here, the liquid solvent containing the peptides is electrically dispersed into a fine aerosol, usually by applying a positive voltage. Charged droplets are formed which continuously decrease in size due to evaporation of the solvent until they become unstable and disintegrate into smaller droplets. Eventually, the charge is transferred onto the peptide molecule inside the droplet, forming peptide ions of variable charge. The nebulised beam is subsequently led into the vacuum chamber of the mass spectrometer [10]. The core of the mass spectrometer is the **mass analyser**. It applies an electro-magnetic field that accelerates the peptide ions and deflects them according to their mass-to-charge ratio. The degree of deflection is recorded by a **detector**. A common mass analyser for proteomics experiments is the orbitrap [107, 108], which has an innovative barrel-like shape that combines a mass analyser and a detector in one. Application of a negative current causes the ions to orbit around an inner electrode. The resulting image current is detected by an outer electrode. Depending on the mass-to-charge-ratio of the ions, different frequencies are measured and transformed into signal intensities using Fast Fourier Transformation [107]. The orbitrap is a high-throughput mass analyser that offers high mass accuracy, high resolving power and a wide mass range [108], which makes it particularly useful for analysing complex peptide mixtures. The unique feature of tandem mass spectrometers is their ability to perform multiple cycles of MS analysis, separated by a fragmentation step [10]. In the first round, the **survey scan**, the mass-to-charge ratios of the intact peptide ions are measured and the induced ion signal intensities are detected. A common graphical representation is the mass spectrum (Fig. 2.4) [10]. Using a second mass analyser as a mass filter, precursor ions can be selected for fragmentation based on their intensity, charge or mass-to-charge ratio. Two approaches can be distinguished. In data-dependent acquisition (DDA) a fixed number of ions is selected, commonly the top N ions based on their intensities in the survey scan. Data-independent acquisition (DIA), on the other hand, forwards all peptide ions for fragmentation [10]. There is a variety of methods for **fragmentation** of precursor ions in mass spectrometry. A common technique is collision-induced dissociation (CID) [92]. A variant used nowadays in orbitrap tandem mass spectrometers is higher-energy C-trap dissociation (HCD) [109]. Selected precursor ions travel to the collision cell, which is filled with an inert gas. Collision of the accelerated peptide ions with the gas atoms causes them to break stochastically at any chemical bond, releasing smaller fragment ions. The charges of the peptide ion are transferred to the fragments. Fragments without charge are neutral and remain undetected in the following MS cycle. In HCD, the main break point is the peptide bond. After fragmentation, the resulting fragment ions undergo a second round of MS analysis, the **product-ion scanning**. The measured mass-to-charge ratios and corresponding ion signal intensities of each detected fragment can be displayed in a tandem mass spectrum [10]. In total, the MS output provides information about the mass-to-charge ratio and the respective ion signal intensities of the ionised peptides and their fragment ions.

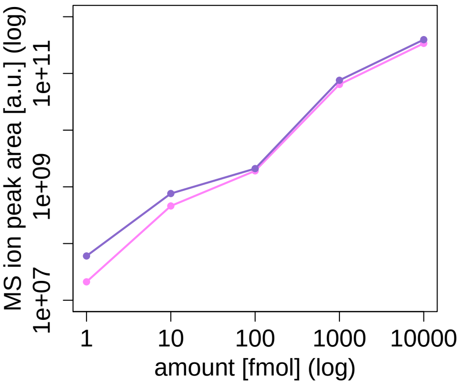
2.3.2 Peptide identification

From the mass spectrum of the precursor ions, the mass and charge of the peptides can be deduced by making use of the natural occurrence of carbon isotopes inside the peptide molecules. Sequence information can be reconstructed from the mass spectrum of the fragment ions based on the molecular masses of the amino acid residues [10]. In the past, researchers had to decipher that information from the MS data manually. Nowadays, there is a wide variety of software available for peptide and protein identification. They can be grouped into two classes. Algorithms for sequence **database search** identify peptides by comparing the experimentally obtained mass spectra with theoretical mass spectra derived from sequences in a database. Popular proprietary software for protein identification via database search are Mascot [110, 111] and SEQUEST [112]. A freeware alternative is MaxQuant [113]. An alternative approach is ***de novo* sequencing** [114]. Here, the peptide sequence is actively inferred from the MS/MS data. As opposed to the database search, which can only identify peptides that match existing sequences, the *de novo* approach is able to identify novel sequences as well. Common software applying a *de novo* algorithm is PEAKS (proprietary) [115]. Both approaches aim at identifying the peptide sequence from the tandem mass spectra and provide confidence scores for their assignments [116].

2.3.3 Peptide quantification

Tandem mass spectrometry does not directly provide quantitative information about the analysed sample. There is a variety of methods that permit quantification of proteins and peptides. According to what kind of information is provided, we distinguish relative and absolute quantification. Relative quantification is achieved by comparing the amounts of the same peptides or proteins between samples, providing a quantitative ratio or a relative change (”x-fold increase”). Absolute quantification, on the other hand, yields exact peptide numbers or concentrations within a sample and allows to compare abundances of different peptides [117].

Fig. 2.5: Linear relation between amount and signal intensity. Calibration curve obtained via titration of a synthetic peptide equivalent of the hydrolysis product KRAS 5–14 G12V [generated based on data published in [118]]. The ion signal intensity was measured for different amounts of peptide over a range of 0 to 10 pmol. The background signal detected at 0 fmol was subtracted from the other measurements in the plot. The relationship between the loaded amount and the resulting ion signal intensity is a linear one.



2.3.3.1 Titration of synthetic peptide equivalents

After performing an MS analysis of a sample of peptides, absolute quantities of peptides of interest can be obtained by subsequent titration of synthetic peptide equivalents. A crucial requirement is to use the same instrument and setup as for the initial measurement. For a range of peptide amounts the corresponding signal intensity is measured by MS and plotted in a calibration curve (Fig. 2.5). For a range of analyte concentrations, the so-called linear dynamic range, a linear relationship between the amount and the intensity can be observed [11, 12, 13, 14, 15, 16]. Using this correlation, the MS1 signal intensities from the analysis of the peptide mixture can be translated into absolute amounts for each titrated peptide.

2.3.3.2 Label-based quantification

Other MS-based quantification approaches can be broadly classified into two classes: label-based and label-free. Label-based quantification relies on incorporating stable isotopes into the molecules themselves or a tag molecule. Three types of labelling techniques can be distinguished [21, 117]. In metabolic labelling, heavy amino acids are incorporated into the protein *in vivo*. A prominent technique is Stable Isotope Labeling of Amino acids in Cell culture (SILAC, [119]). In chemical labelling, labelled tags are linked to the amino acid side chains of the peptides prior to digestion. Common approaches to be mentioned are isotope-coded affinity tags (ICAT, [120]) and tandem mass tags (TMT, [121]). In enzymatic labelling, the proteins are digested in presence of heavy water. All of the above methods only provide relative quantification. Absolute quantification can be achieved by adding a known amount of internal standard peptide (synthetised containing isotopes) to the sample. This way, relative quantification of sample peptides to an absolute value can be realised (AQUA, [122]). In summary, label-based techniques provide accurate means for peptide quantification that are, however, quite costly and require additional steps in sample preparation. Additionally, with most approaches, only a limited number of peptides can be quantified within a sample and only a limited number of samples can be compared [21, 117].

2.3.3.3 Label-free quantification

Techniques for peptide quantification that do not require labelling of samples are called label-free (Label-free quantification, LFQ). Because of their simplicity, they are of high demand in the proteomics field. The samples of interest are analysed sequentially without the need for additional manipulation like labelling. Label-free techniques are usually divided into two groups. The first method uses the mass spectrum of the precursor peptide ions and **compares the ion intensities** of the same peptide in different samples [123, 124]. Because of the linear relationship between the peptide amounts and their ion current described above, signal intensities are a relative measure for the ion abundance. The great accuracy of today's mass spectrometers allows the use of the mass spectra of the peptide ions for relative quantification between the same peptide in multiple samples [125], provided that identical experimental and MS conditions were used and the precursor masses and retention times were identical. Over the years, many algorithms were developed to improve intensity-based LFQ [28, 126, 127, 17]. The second method relies on **counting the mass spectra of the fragment ions** to infer protein abundance [124]. If a peptide is more abundant in one sample than in the other, then, statistically, there will be more fragments observed in MS2. Therefore, the more tandem mass spectra are assigned for the peptide of interest, the more abundant it should be [24]. This forms the basis for a fast and easy procedure to compare peptide amounts between samples [128]. In the simplest form, the number of tandem mass spectra assigned to the same peptide/protein is counted [128]. Over the years, many strategies for improving the performance have emerged [129, 25, 124, 128]. Studies have demonstrated the intensity-based method to be slightly superior to the spectral counting approach, especially for low-intensity peptides. Yet, other studies have shown an equal performance. Combining both approaches was observed to improve label-free quantification. In summary, label-free quantification offers a simple alternative to label-based techniques. Its ease of use with no additional work required for accurate and robust relative quantification of samples with any size and complexity makes it a popular technique, that is by now widely accepted. Unfortunately, absolute quantification of peptides in a sample is not directly possible with label-free techniques [124, 128].

2.4 MS-based peptidomics for enzyme kinetics

For the absolute quantification of the peptide products in an enzymatic digest over time the following procedure is commonly applied (Fig. 2.4). The protease of interest is incubated with the substrate of interest in a controlled environment to elicit digestion. After fixed time intervals, the reaction is stopped for a part of the mixture and the current state of the peptide product population is analysed by tandem mass spectrometry. MS measurements are performed in multiple biological and technical replicates. Multiple digestions under same biological conditions account for biological variability. Multiple measurements of the same sample with same technical settings account for technical measurement error. A Mascot database search is performed for peptide identification. A relative quantification of the peptide products under different biological conditions is possible via comparing ion signal intensities of same peptides (LFQ). For absolute quantification, titrations of selected synthetic peptide equivalents can be performed to obtain the concentrations over time. Titrations are performed in multiple technical replicates [39, 118].

3 | Theoretical methods background

The goal of this work is parameter estimation based on observed data. Bayesian inference in a Markov Chain Monte Carlo scheme offers a convenient tool to achieve this. Bayesian inference is a technique that combines previous knowledge on the unknown parameters with new knowledge obtained through experimental data. In the framework of Markov Chain Monte Carlo, this combined knowledge is continuously updated until the optimal distribution of parameter values is found. In particular, in this thesis, an adaptive and population-based Differential Evolution Markov chain algorithm will be applied. This chapter presents the theoretical background for this approach. The concept of parameter identifiability is introduced and the benchmarking framework to review the performance of an algorithm is explained.

3.1 Bayesian inference

In statistics, mainly two schools of thought are distinguished. Frequentist statistics defines probability using frequencies and relative proportions. The Bayesian approach, on the other hand, defines probability as 'a degree of rational belief' [130]. This work will use a Bayesian point of view. Bayesian inference is a method to make an educated decision using statistical knowledge to a problem at hand. It applies Bayes' theorem to combine experimental sample information with previous knowledge on the data to update the probability of a hypothesis of interest [131]. The gist of this perspective is that instead of returning only an optimum point estimate, it returns a distribution of possible values, which accounts for uncertainty in the data [40].

3.1.1 Bayes' Theorem

The heart of Bayesian statistics is Bayes' theorem (more correctly Bayes-Price theorem). It provides a simple formula to calculate the probability of an event conditional on an event that previously occurred [131]. A typical phrasing of the theorem today is the following:

Theorem 3.1 (Discrete form of Bayes' Theorem [131])**.** *Let A and $B_1, \ldots, B_n$ be random variables, the B_i being disjoint with $P(\bigcup_{i=1}^{n} B_i) = 1$, i.e. one of the events B_i is certain to occur. Let $P(\cdot)$ be the probability of an event, $P(\cdot|\cdot)$ denotes the conditional probability of an event given the occurrence of the second. Then*

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{P(A)} \qquad (j = 1, \ldots, n), \tag{3.1}$$

with $P(A) = \sum_{i=1}^{n} P(A|B_i)P(B_i)$.

3.1.2 Model-based Bayesian inference

Oftentimes, the Bayesian approach is used for model-based parameter estimation. Bayes' theorem shows, how the degree of belief in a hypothesis can be updated, when new evidence becomes available [40]. The hypothesis of interest then deals with the question, whether an estimated set of parameter values does explain the observed data.

Theorem 3.2 (Continuous form of Bayes' Theorem [131, 40, 132])**.** *Let $\theta \in \Theta \subseteq \mathbb{R}^n$ be the unknown parameter vector of a model. Denote the collected data by the continuous random variable $X = (X_1, \ldots, X_M) \in \mathcal{X} \subseteq \mathbb{R}^n$ with independent observations X_m. Let the variable x be a particular realisation of X. Let p be the probability density function of x. Then*

$$p(\theta|x) = \frac{p(\theta)\, p(x|\theta)}{p(x)}, \tag{3.2}$$

where $p(x) = \int_\Theta p(x|\vartheta)\, p(\vartheta)\, \mathrm{d}\vartheta$.

The quantity of interest, $p(\theta|x)$, is called posterior probability distribution and represents the probability that the hypothesis is true after evidence through new relevant data is gathered. The probability of the hypothesis, $p(\theta)$, before new data is observed, is called prior probability. The likelihood function $p(x|\theta)$ is defined as the conditional probability of the observable quantity x given the unobserved variable θ, and shows to what extent the data supports the hypothesis. The denominator $p(x) \neq 0$ is the total probability of the data, irrespective of the parameters, the marginal distribution [132].

3.1.3 The prior distribution

The prior distribution $p(\theta)$ (or prior, for short) is the subjective knowledge about the hypothesis before data is observed, i.e. the initial belief. It can be based on information from the past, like data collected from former simulations or experiments, or on experience of an expert.

A prior distribution is called proper if its integral is finite:

$$\int_{\Theta} p(\theta)\,\mathrm{d}\theta < \infty. \tag{3.3}$$

Depending on their information content, different types of priors are distinguished [132]. A prior is called uninformative or non-informative, if no or very little information about the parameters is known. A non-informative prior is the most objective prior that can be defined [131], attempting to impart no information about the parameters of interest, e.g. by an unbounded uniform distribution. In this case, the posterior is driven mainly by the information conveyed by the data. If prior knowledge about the parameters is available, an informative prior can be defined. Here often a normal distribution is used, with mean equal to the believed value and a narrow standard deviation. In most cases, it should be possible to provide at least a weakly informative prior, i.e. a normal distribution with a wide standard deviation or a uniform prior with bounds:

$$p : \Theta \to \mathcal{U}([a, b]).$$

3.1.4 The likelihood function

All relevant experimental information about a parametrised model can be summarised in the sampling distribution, the so-called likelihood function [131]. It quantifies the agreement of the model output with the experimental data [35]. The larger the likelihood, the more likely are the parameters to describe the observed data [131]. The likelihood function (or simply likelihood, for short) is defined as the conditional probability density $p(x|\theta)$ of the observed data x given the parameters θ, considered as a function of θ.

For independent experimental data $x_m(t)$ ($m = 1, \dots, M$, $t = 1, \dots, T$), the likelihoods of the individual observations multiply:

$$p(x|\theta) = \prod_{m=1}^{M} \prod_{t=1}^{T} p(x_m(t)|\theta). \tag{3.4}$$

Commonly, measurement noise is assumed to be normally distributed according to $\mathcal{N}(0, \sigma)$. In this case, the difference between the model output $y_m(t, \theta)$ under the estimated parameter values θ and the observed experimental data $x_m(t)$ can be described by the following likelihood function [35]:

$$p(x|\theta) = \prod_{m=1}^{M} \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{\big(x_m(t) - y_m(t, \theta)\big)^2}{2\sigma^2} \right). \tag{3.5}$$

The goal of the inference is to obtain parameters, which result in model outputs as close to the observed data as possible, i.e. that maximise the likelihood function. In a frequentist setting, the best fit yields a point estimate – the maximum likelihood estimate – of the parameters [35]. In many cases however, the likelihood is difficult to calculate. Here, sampling methods provide a convenient way to circumvent this problem. We will introduce the popular class of Markov Chain Monte Carlo methods in Section 3.2.

3.1.4.1 The log-likelihood

In practice, it is often convenient to use the logarithm of the likelihood function, log $p(x|\theta)$. Maximizing the log-likelihood is equivalent to maximizing the likelihood, because it is a strictly increasing function. The logarithm simplifies the calculations in Eqn. (3.5):

$$\log\, p(x|\theta) = -\sum_{m=1}^{M}\sum_{t=1}^{T}\log\Big(\frac{1}{\sqrt{2\pi\sigma^2}}\Big)\frac{\big(x_m(t)-y_m(t,\theta)\big)^2}{2\sigma^2}, \tag{3.6}$$

therewith increasing the speed and improving the numerical accuracy when dealing with small probabilities [].

3.1.5 The marginal likelihood function

The marginal likelihood function (also referred to as integrated likelihood, prior predictive distribution, evidence or model evidence), is the distribution of the observed data marginalised over the parameters [40]:

$$p(x) = \int_{\Theta} p(x|\vartheta)\, p(\vartheta)\, \mathrm{d}\vartheta. \tag{3.7}$$

In Bayes' theorem (3.2), it serves as a normalizing constant that guarantees that the integral of the posterior probability density over all values of θ is equal to 1.

3.1.6 The posterior distribution

According to Bayes' theorem (3.2), the prior probability and the likelihood function are combined to form the posterior probability distribution of θ given x (or posterior, for short). It summarises all the information available, i.e. the previous beliefs and relevant sample information, into a final perception about the parameters [131]. If the prior distribution is proper, then the posterior distribution is also proper, i.e. the integral is finite. This ensures that a sampling approach can be used to approximate the posterior [35].

Analogous to the maximum likelihood estimate, the parameter that maximises the posterior is called maximum a posteriori estimate [35]. However, a considerable advantage of the Bayesian approach is the generation of a full distribution of estimated values for the parameters, involving uncertainty. If a single parameter value is desired, the posterior distribution can be summarised using the mode, the median, or the mean [131]. In addition, the uncertainty of the estimate can be measured using the standard deviation of the distribution [131]. It provides a feeling for the confidence that can be laid in the prediction.

3.1.7 Bayesian model comparison

The inference process provides a mean to update the prior density to the posterior density through the likelihood function. From the inferred posterior distributions for the parameters, one can now conclude a decision for the underlying hypothesis. A comparison of two competing hypotheses can be achieved by comparing their respective posteriors. Let $\tilde{\theta}$ and $\bar{\theta}$ be the parameter sets to be compared. If

$$\frac{p(\tilde{\theta}|x)}{p(\bar{\theta}|x)} > 1, \tag{3.8}$$

then parameter set $\tilde{\theta}$ with the larger posterior probability is considered the better choice [40]. This reasoning is the basis for the decision step in the MCMC algorithm, see Sec. 3.2.1.2.

3.1.7.1 Sidenote on the marginal likelihood

Marginal likelihoods are often difficult to calculate. However, it is independent of θ, and therefore identical for all possible hypotheses considered. It does not change the outcome of a comparison of probabilities of different hypotheses:

$$\frac{p(\tilde{\theta}|x)}{p(\bar{\theta}|x)} \overset{(3.2)}{=} \frac{p(x|\tilde{\theta})\,p(\tilde{\theta})\,p(x)}{p(x|\bar{\theta})\,p(\bar{\theta})\,p(x)} = \frac{p(x|\tilde{\theta})\,p(\tilde{\theta})}{p(x|\bar{\theta})\,p(\bar{\theta})} \tag{3.9}$$

and can hence be omitted. This simplifies the posterior calculations in the Bayesian inference [40]:

$$p(\theta|x) \propto p(\theta)\,p(x|\theta). \tag{3.10}$$

3.2 The Markov Chain Monte Carlo method

In many applications, the model is too complex to obtain the posterior distribution in closed form by analytical methods. In such situations, sampling approaches can be applied to approximate the posterior. One class of sampling techniques are Markov chain Monte Carlo methods (MCMC for short). Over the years, a wide variety of different algorithms has been developed. One of the earliest processes in this class is the Metropolis–Hastings algorithm [132]. In this work, we will use an advanced Metropolis–Hastings based algorithm that employs an adaptive population-based approach, namely Differential Evolution Markov Chain.

3.2.1 Basic principle

The idea of MCMC algorithms in Bayesian inference is to approximate the posterior distribution by a sampling distribution. The earliest and simplest MCMC version is the Metropolis–Hastings algorithm. The main idea can be summarised as follows. For every parameter to be estimated, a sequence of samples is drawn in a particular manner and evaluated by calculating the corresponding Bayesian posterior. By either rejecting or accepting the proposed values, the emerging Markov chain iterates through the state space, converging to a limiting distribution that represents the target posterior.

3.2.1.1 Markov chain theory

In the sampling scheme, a discrete stochastic process for the parameters is created: $\{\theta(i) \in \Theta \, : \, i \in \mathbb{N}_0\}$, where Θ is the state space and i denotes the number of samples created, also referred to as iterations. In practice, the number of iterations will be finite $i = 1, \ldots, N \in \mathbb{N}_0$.

Definition 3.1 (Markov chains [133, 134])**.** A discrete-time stochastic process fulfills the Markov property, if the conditional probability of the future state $\theta(i+1)$ given the past $\theta(0), \ldots, \theta(i)$ depends only on the current state $\theta(i)$:

$$p(\theta(i+1)|\theta(0), \ldots, \theta(i)) = p(\theta(i+1)|\theta(i)) \tag{3.11}$$

A process that satisfies the Markov property is called Markov process or Markov chain.

The distribution of the first state $\theta(0)$ is called the initial distribution. The conditional probability $p(\theta(i+1)|\theta(i))$ of arriving at state $\theta(i+1)$ following the current position $\theta(i)$ is also called transition probability.

For the purpose of parameter estimation, the Markov chain should sample the whole state space over the course of iterations and eventually stabilise at the posterior distribution.

Definition 3.2 (Stationarity [134])**.** A Markov chain is stationary (or at equilibrium), if the distribution of any part of the chain does not depend on i:

$$p(\theta(i+1), \ldots, \theta(i+k)|\theta(i)) = p(\theta(j+1), \ldots, \theta(j+k)|\theta(j)) \quad \forall i, j, k \in \mathbb{N}_0 \tag{3.12}$$

A theorem that ensures, that the Markov chain reaches its stationary distribution given certain conditions is the Ergodic theorem.

Theorem 3.3 (Ergodic theorem [135, 136]). *Let $\{\theta(i)\}$ be a Markov chain with stationary distribution $\pi(\theta)$. If it is ergodic, i.e. if it is*

- *time-homogeneous (its transitions do not depend on iteration time),*
- *irreducible (every point in the state space can potentially be reached regardless of the starting point),*
- *aperiodic (it does not oscillate between states),*

then, for large iteration number, the samples will be close to the stationary distribution:

$$p(\theta(i)|\theta(0)) \xrightarrow[i\to\infty]{} \pi(\theta) \qquad \forall\, \theta, \theta(0) \in \Theta \tag{3.13}$$

Whether a Markov chain has a stationary distribution depends on the way it is constructed. In the following section, the Metropolis–Hastings sampling algorithm is presented, which ensures, that the posterior distribution is a stationary distribution of the Markov chain generated. Any advanced sampling scheme using the Metropolis–Hastings idea at its core needs to make sure, that the way it generates the Markov chain, guarantees ergodicity. Then, by the Ergodic theorem, the chains will converge to the posterior.

3.2.1.2 The Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm [137, 138, 139] is the most general algorithm of all the MCMC variants [136]. Its principle is to simulate a Markov chain that covers the state space and whose stationary distribution is the target distribution. To achieve this, it pursues a trial-and-error strategy by proposing samples that are either accepted or rejected based on experience by taking into account the current value. The concept of a simple MCMC sampler is illustrated in Fig. 3.1 and a summary of the algorithm is shown in Alg. 1 using pseudo code notation [140, 134].

1. Initialisation. The Markov chain is initialised by (randomly) picking start values from the state space, which in Bayesian inference is the prior distribution:

$$\theta(0) \sim p(\theta). \tag{3.14}$$

The algorithm then proceeds in an iterative manner using discrete time steps $i = 1, \ldots, N$.

2. Proposal step. Transitions from one state $\theta(i)$ to the next state $\theta(i+1)$ are generated using the proposal distribution $g(\theta(i+1)|\theta(i))$. In every iteration a candidate $\zeta \in \Theta$ for the next state is suggested according to $g(\zeta|\theta(i))$. The easiest form of proposal function would be a small perturbation of the current value resulting in a random walk: $q(\zeta|\theta(i)) = \theta(i) + \epsilon$ [136]. The selection of the proposal function influences the efficiency of the sampling algorithm [35].

3. Metropolis step. For the proposed candidates, the posterior probability $p(\zeta|x)$ is calculated and compared to the posterior value of the current state of the chain $p(\theta(i)|x)$ by calculating the Metropolis–Hastings acceptance probability:

$$\alpha(\zeta, \theta(i)) = \min\big\{\frac{p(\zeta|x)}{p(\theta(i)|x)}\,\alpha_{\text{sym}}, 1\big\}, \tag{3.15}$$

$$\text{where } \alpha_{\text{sym}} = \frac{g(\theta(i)|\zeta)}{g(\zeta|\theta(i))}. \tag{3.16}$$

The factor α_{sym} vanishes if the proposal is symmetric. The ratio of the two posteriors comprises whether the proposed candidate values are more likely to explain the observed data x than the previous value.

4. Acceptance–rejection step. If the new value better explains the data, then it is always accepted and the chain moves to this new state. If the candidate is rejected, the chain remains at its current location. The crux of the algorithm is the occasional ”downhill” transition, when a candidate, which has a lower posterior, may also be accepted with a certain probability. This is implemented by comparing α to a uniform random number $u \sim \mathcal{U}([0, 1])$:

$$\theta(i+1) = \begin{cases} \zeta & \text{if } u \leq \alpha & (3.17\text{a}) \\ \theta(i) & \text{if } u > \alpha & (3.17\text{b}) \end{cases}$$

5. Convergence. Over iteration time, the Markov chain slowly transfers the parameter values from the prior distribution to the stationary target posterior distribution.

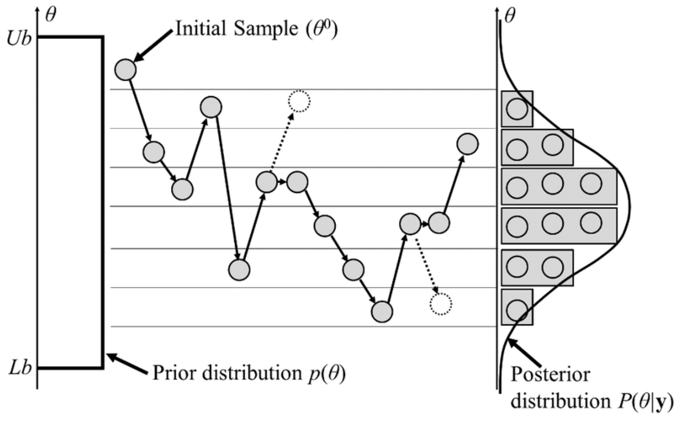

θ
Ub
Lb
Initial Sample (θ^0)
Prior distribution $p(\theta)$
θ
Posterior distribution $P(\theta|\mathbf{y})$

Fig. 3.1: Illustration of the MCMC concept. Starting from an initial sample, the Markov chain evolves by proposing candidate values from the prior range in a particular fashion, that depends on the underlying proposal function. These candidates are successively accepted (grey dots) or rejected (white dots). After sufficiently many iterations, the accepted samples approximate the posterior distribution. [This figure is taken from [141], Creative Commons 4.0.]

Algorithm 1 The Metropolis–Hastings algorithm. The MH algorithm is the simplest Markov Chain Monte Carlo method. It iteratively transfers the prior parameter distributions into posterior distributions taking into account the observed data.

```
Input: Observed data $x$
Output: Estimations of the parameters $\theta$

  Sample start values $\theta(0) \sim p(\theta)$                    ▷ 1. Initialisation

  for iterations $i = 1, \ldots, N$ do
      Propose candidate $\zeta$ using proposal function $g$          ▷ 2. Proposal step
      Calculate $\alpha$ according to (3.15) using data $x$          ▷ 3. Metropolis step

      if uniform($[0, 1]$) $\leq \alpha$ then                          ▷ 4. Acc./Rej. step
          Accept candidate and move to new state (3.17a)
      else
          Reject candidate and stay at current state (3.17b)
      end if
  end for
```

3.2.2 Practical considerations

When running MCMC simulations, several decisions in algorithm settings and chain handling have to be made. In this section, we will only name a few. A central question for example is where to start the sampling process and when to stop.

Number of parallel chains. The classical MCMC frameworks generate a single Markov chain. Other algorithms require the run of several chains in parallel. The opinions on which approach is beneficial deviate. While some promote the use of a single long chain [142, 143], others recommend running multiple interconnected chains [144].

Starting values. There is no universal rule on how to find a good starting point for the Markov chain(s) [145]. It can be randomly sampled from the prior distribution, or informed by previous simulations. Optimisation can be used to approximate the global mode of the target distribution wherever possible, which can then be used as initial value [146]. Different runs of a simulation should be started from different initial points, to check the reliability of the estimates and to eliminate the possibility of getting stuck in local modes [145]. The same applies to running multiple interconnected chains, where the starting points of the parallel chains should be dispersed [45, 145]. After a sufficient number of iterations, the chain(s) should become independent of the particular choice of the starting point [142].

Warm-up period. The number of iterations the chain needs to explore the space before it reaches its equilibrium is called burn-in. These early samples are usually neglected in the calculation of the summary statistics of the posterior distribution. Recommendations on what fraction to discard differ widely, from as little as 1-2% [142] up to 50% [144].

Thinning. A strategy often used to reduce autocorrelations in a Markov chain is discarding samples and only keeping every k-th observation. However, since this procedure wastes a lot of information, some researchers recommend to only apply it if computational issues (storage, cost of function evaluations) call for it [147, 148, 146, 149].

Acceptance rate. The fraction of accepted samples can be calculated, which allows to monitor the sampler performance. An acceptance rate of 15-30% usually indicates good mixing [136, 44].

3.2.3 Stopping rules

The Markov chain has reached its stationary distribution when further samples do not change the estimate [45]. Although theoretically it can be assessed whether a chain will eventually converge, in practice it is difficult to determine when the sequence is representative of the target distribution [150, 151, 133, 145]. This section presents some considerations on the duration of the sampling process.

3.2.3.1 Stopping after a fixed number of iterations

Running a fixed number of iterations relies on trial-and-error and is a trade-off between speed and accuracy. Premature termination will probably lead to inaccurate inference. Running a simulation for more iterations will increase the accuracy [145]. However, running simulations longer than actually needed unnecessarily uses time and resources [146].

3.2.3.2 Stopping after visual inspection

Visualisation is an important tool in assessing Markov chain Monte Carlo performance [152]. The results for each inferred parameter are usually investigated in two ways: the traceplots of the chain(s) and the resulting posterior distributions. A **traceplot** of a parameter shows the parameter values over iteration time. It represents the movement of the chain in the state space, also referred to as mixing. It may include the burn-in samples to visualise the transition from starting values to target distribution. A well-mixing chain should look like a “hairy caterpillar” [146]. From any arbitrary starting point, it should explore the whole state space and after some time become stationary around some mean value. As long as the traceplot shows trends, convergence has not been achieved. When multiple parallel chains are used in a simulation, they should sample in agreement with each other, once converged [44]. The distributions of the parameters are typically depicted in the form of **histograms**, which, for sufficiently many samples, should approximate the posterior density. From this, the most likely parameter value is directly apparent, together with its uncertainty in the form of the standard deviation. The parameter estimates can be provided in the form of summary statistics [40]. Burn-in samples are typically not included.

3.2.3.3 Stopping using convergence criteria

Since the early 1990s, a large variety of convergence criteria has been developed, as reviewed in [146, 153, 150, 154]. There are methods for single chain MCMC simulations, as well as convergence criteria for multiple chains. Some only give lower bounds for the required number of iterations, others provide concrete stopping rules [150]. The preferred way to establish convergence is via theoretical considerations [153]. However, most approaches are too conservative or just too complex in practice [153, 150]. As a result, empirical tools are more commonly applied to assess convergence, or the lack thereof. A popular diagnostic was developed by Gelman and Rubin in 1992 [144].

The **Gelman–Rubin convergence criterion** can be applied to any multi-chain MCMC algorithm output [150]. Convergence is diagnosed through comparison of the parameter variances within the chains and between the chains. A quantitative measure, called potential scale reduction factor, is defined as

$$\hat{R} = \sqrt{\frac{V}{W}}, \tag{3.18}$$

where W is an estimate of the within-sequence variance and V is an estimate of the mixture-of-sequences variance, calculated using W and the between-sequence variance estimate. Gelman and Rubin argue that starting from an overdispersed initial distribution, after a finite number of iterations, the numerator will dominate the denominator, meaning $\hat{R}$ will approach 1. Once it is sufficiently close to 1 for every parameter, this suggests that their distributions are close to the target distribution [144]. A threshold often used is $\hat{R} < 1.2$ [155].

3.2.3.4 Limitations

The general sentiment is that no single method provides the ultimate convergence diagnostic. Some approaches may lead to premature termination, different techniques may even contradict each other and results should be interpreted with caution. No method can guarantee that a finite Markov chain is representative of the limit distribution [150]. Nevertheless, they provide a valuable control of the algorithm's progress. Visual inspection alone can be quite subjective and unreliable [40, 45]. In high-dimensional inference problems, it is not practical to investigate all of the traceplots and posterior distributions. Easier, preferably numerical summaries of the convergence properties are required [156]. In summary, it is advisable to use a mixture of approaches with different properties, together with appropriate visualisation and experience [153, 157, 150, 151].

3.2.4 Advanced algorithms

The usefulness of MCMC methods in Bayesian inference has led to the development of a multitude of different algorithms by researchers in different fields [151]. Although existing theory proves convergence of well-constructed Markov chains to the target posterior, in practice the rate of convergence can be very slow. An issue often addressed is the question of orientation and scaling of the proposal to achieve faster convergence. Proposal schemes with an automated adjustment are called **adaptive methods**. While the problem of choosing an appropriate direction was solved in early adaptive schemes proposed by Gilks and Roberts in 1994 [158], the problem of choosing an optimal scale remained [44]. The disadvantage of classical approaches is the fact that the proposal step size is fixed, hampering efficiency. The scale factor needs to be carefully tuned before running the algorithm. If the chosen step size is too small, then most proposals will be accepted, but the chain will move only slowly towards the limiting distribution. If the chosen step size is too large, then most samples will be rejected, also resulting in slow convergence [43]. It would be beneficial to make larger jumps in the beginning to reach a region of high likelihood faster, and to decrease the step size as the chain approaches more precise estimates. An automatic adaptation of the proposal step size allows to overcome these difficulties [43]. Haario et al. proposed a sampling scheme utilising information from past states of the chain to adapt the proposal distribution [159, 160]. Many variations of the adaptive algorithms have been published with the goal to improve efficiency [161]. They can be grouped in two classes: single- and **multiple-chain methods**. In a multi-chain approach, the chains are regarded as members of a population that can learn from each other by taking into consideration the current states of the other chains. The idea was taken from a class of global optimisation algorithms called evolutionary algorithms. Here, a population of individuals evolves over generation time through reproduction, mutation and selection based on an evaluation of their fitness. The first evolutionary MCMC methods were developed by Liang et al. in 2000 [162, 163]. The population-based algorithm in focus of this thesis is the Differential Evolution Markov chain algorithm.

3.2.4.1 The Differential Evolution Markov chain algorithm

The Differential Evolution Markov Chain (DE-MC) is an adaptive population-based MCMC algorithm that takes care of both choosing an optimal direction as well as an appropriate scale in the proposal step [41]. It is a merger of the evolutionary algorithm called Differential Evolution (DE) and an adaptive MCMC scheme. Differential Evolution (DE) is a population-based global optimisation approach developed by Storn and Price in 1997 [164, 165]. It works on the principle, that any population member can improve using the information contained in the population as a whole [166]. A genetic algorithm is applied to update a population of states through mutation, crossover and selection.

The first attempt to combine the Differential Evolution method with an MCMC scheme was conducted by Strens et al. [167, 166]. Independently, in 2006 Cajo J.F. ter Braak integrated the essential ideas of DE with MCMC for Bayesian inference [41]. A flaw of the algorithm was the impractical need of a large number of parallel chains. In a follow-up joint work with Jasper A. Vrugt in 2008 [42], improvements were made to circumvent this using an archive of past states. In addition, the resulting DE-MC$_{\text{(ZS)}}$ algorithm increases the variety of update directions by introducing a second type of proposal. Alternatively, the DREAM algorithm (Differential Evolution Adaptive Metropolis, [43]) was developed in 2009, decreasing the number of chains required by implementing a self-adaptive randomised subspace sampling, that introduces even more proposal directions and decreases autocorrelation in the samples. A combination of the above approaches was developed as DREAM$_{\text{(ZS)}}$ [168] in 2012, which allows efficient sampling of target distributions of up to one hundred parameters with as few as three parallel chains. The algorithm was published in slightly different versions on diverse programming platforms [168, 44, 45]. The DREAM framework has experienced many applications across a multitude of research fields [44].

A summary of the DREAM$_{\text{(ZS)}}$ algorithm is shown in Alg. 2 using pseudo code. In accordance with the implementation of [44, 45], we present the algorithm with only a single chain pair in the calculation of the proposal ($\delta = 1$). In the following, the main steps are explained in detail. The original notation was moderately modified to match the notation of this thesis. Let $c = 1, \dots, C$ denote the number of parallel chains (default $C = 3$), $d = 1, \dots, D$ the dimension of the parameter vector and $i = 1, \dots, N$ iteration time. Let the current state of the Markov chain of parameter d in chain c at iteration i be given by $\theta_d^{(c)}(i)$. In DREAM$_{\text{(ZS)}}$, the current state is provided as a population matrix X of dimension $C \times D$:

$$X = \begin{pmatrix} \theta_1^{(1)}(i) & \cdots & \theta_D^{(1)}(i) \\ \vdots & \ddots & \vdots \\ \theta_1^{(C)}(i) & \cdots & \theta_D^{(C)}(i) \end{pmatrix}. \tag{3.19}$$

Algorithm 2 The Differential Evolution Markov Chain algorithm. The DE-MC algorithm applies an adaptive population-based sampling scheme with two kinds of proposals using information of past states to increase efficiency [41, 42, 43, 44, 45].

Input: Observed data x
Output: Estimations of the parameters θ

```
Initialise fake archive $Z \sim p(\theta)$                                  ▷ 1. Initialisation
Initialise start values $X \sim p(\theta)$

while not GR converged do
    for every chain $c = 1, \ldots, C$ do
        if uniform($[0, 1]$) $\geq p_\gamma$ then                          ▷ 2. Proposal step
            Propose candidate $\zeta^{(c)}$ according to snooker update (3.23)
        else
            Propose candidate $\zeta^{(c)}$ according to parallel direction update (3.21)
        end if
        Optional: (adaptive) crossover step

        Calculate $\alpha$ according to (3.26) using data $x$             ▷ 3. Metropolis step
        if uniform($[0, 1]$) $\leq \alpha$ then                            ▷ 4. Acc./Rej. step
            Accept candidate and move to new state (3.27a)
        else
            Reject candidate and stay at current state (3.27b)
        end if
    end for
end while
```

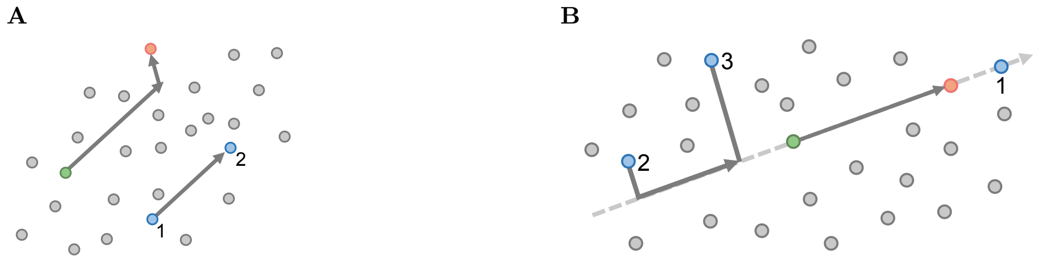


Fig. 3.2: Proposal scheme of the DREAM$_{\text{(ZS)}}$ algorithm. (A) Parallel direction update. **(B)** Snooker update. The grey dots denote the past states of all chains. Green is the current state of a chain to be updated. Blue are the anchor points drawn from the archive used to generate the proposal candidate (orange). [Figure adapted from [42].]

1. Initialisation. Since the algorithm requires sampling from past states, an initial "fake" archive Z of dimension $Z_0 \times D$ is generated randomly by sampling the elements from the prior:

$$Z = \begin{pmatrix} Z[1,1] & \cdots & Z[1,D] \\ \vdots & \ddots & \vdots \\ Z[Z_0,1] & \cdots & Z[Z_0,D] \end{pmatrix}, \quad \text{where } Z[z,d] \sim p(\theta). \tag{3.20}$$

The initial vector population is also chosen randomly $X[c,d] \sim p(\theta)$ and should be distributed over the entire prior range so that multiple modes can be found, if they exist [44].

2. Proposal step. In what they call the "mutation" step, a candidate $\zeta^{(c)}$ for each of the chains is proposed in either of two ways: via a parallel direction update or an occasional snooker update.

In the **parallel direction update**, for every chain, a pair of points $Z[z_1,], Z[z_2,]$ are randomly chosen from the archive of past states without replacement. The weighted difference of those is added to the current point. In this way, diversity is introduced and superior vectors in the population are allowed to influence others. This takes care of the problem of finding the appropriate scale and orientation. The parallel direction proposal is defined as follows [45]:

$$\zeta^{(c)} = X[c,] + \gamma(d)(1+e_1)\Big(Z[z_1,] - Z[z_2,]\Big) + e_2 \qquad \forall c = 1,\dots,C. \tag{3.21}$$

Here, the function γ is defined as

$$\gamma(d) = \begin{cases} 1 & \text{if } u_\gamma \geq p_\gamma, \text{ where } u_\gamma \sim \mathcal{U}([0,1]) \\ \frac{2.38\beta_0}{\sqrt{2d}} & \text{else} \end{cases} \tag{3.22}$$

and denotes the jump rate that controls the acceptance rate. It can be tuned through the factor β_0 [41, 44], in most applications they use $\beta_0 = 1$. Every few iterations ($p_\gamma = 10$ in [41], $p_\gamma = 5$ in [43, 168, 44, 45]), a gamma jump of 1 is introduced, that allows the chain to leap between modes in a multi-modal limiting distribution. In [41] it is suggested to use a gamma jump of 0.98 instead. The random constants $e_1 \sim \mathcal{U}^D([-b_1, b_1])$ and $e_2 \sim \mathcal{N}^D(0, b_2)$ (with b_1, b_2 small) are noise factors introduced to account for randomisation and to guarantee ergodicity, respectively. Since this proposal is symmetric, the Metropolis acceptance probability below simplifies with $\alpha_{\text{sym}} = 1$.

To increase the variety of possible updates, in 10% of the iterations, a **snooker update** with adaptive step size is performed. Here, for every chain, three points $Z[z_1,], Z[z_2,], Z[z_3,]$ are sampled from the archive without replacement. A line is drawn through the current point $X[c,]$ and $Z[z_1,]$. The other two points are orthogonally projected onto that line. The difference of those projected points is then used to generate a proposal in a new direction [45]:

$$\zeta^{(c)} = X[c,] + \gamma_s(1+e_1)\Big(Z_\perp[z_2,] - Z_\perp[z_3,]\Big) + e_2 \tag{3.23}$$

$$= X[c,] + \gamma_s(1+e_1)\frac{(Z[z_2,] - Z[z_3,])(X[c,] - Z[z_1,])}{(X[c,] - Z[z_1,]) \cdot (X[c,] - Z[z_1,])}(X[c,] - Z[z_1,]) + e_2 \tag{3.24}$$

The jump rate here is sampled uniformly around a value of 1.7 as $\gamma_s \sim \mathcal{U}([1.2, 2.2])$ [42]. The symmetry factor in the acceptance probability in the snooker case is defined differently in the author's implementations [42, 44, 45], this definition is taken from [42]:

$$\alpha_{\text{sym}} = \frac{||\zeta - Z[z_1,]||^{D-1}}{||X[c,] - Z[z_1,]||^{D-1}}. \tag{3.25}$$

To ensure that the parameters stay within the prior range specified, boundary handling is applied. Vrugt et al. suggest different options, of which only the "folding" option maintains detailed balance. In this case, if a proposal lands outside the prior range, it will re-enter the range on the opposite bound [44].

3. Metropolis step. The decision step whether the candidate should be replaced by the proposal is called "selection". It follows the same acceptance probability as the Metropolis–Hastings algorithm (Eqn. (3.15)):

$$\alpha(\zeta^{(c)}, \theta^{(c)}(i)) = \min\Big\{1, \frac{p(\zeta^{(c)}|x)}{p(\theta^{(c)}(i)|x))}\alpha_{\text{sym}}\Big\}, \tag{3.26}$$

where α_{sym} as defined by the parallel or snooker proposal step, respectively.

4. Acceptance–rejection step. The DREAM$_{(\text{ZS})}$ algorithm follows the usual acceptance rule of the MH algorithm (Eqn. (3.17a)):

$$\theta(i+1) = \begin{cases} \zeta & \text{if } u \leq \alpha \quad (3.27a) \\ \theta(i) & \text{if } u > \alpha \quad (3.27b) \end{cases}$$

4A. Crossover step. In higher dimensions, it can make sense to update only a subset of the candidate vector. This further broadens diversity in the population and increases efficiency [41]. This efficiency can be further enhanced by implementing a self-adaptive version of this subspace sampling [43].

4B. Archive update. The current state X is appended to the archive Z. This happens every few iterations only, achieving a thinning of the chain [42].

5. Convergence. Vrugt et al. [44] use the Gelman–Rubin criterion for its power and robustness. The proof showing that the algorithm produces Markov chains that converge to a stationary distribution equal to the target posterior distribution is given in [42, 43, 168].

3.3 Parameter identifiability

It would be desirable to know before running a long simulation, whether the inference has the potential of being successful. A fundamental question is whether the parameters of interest are in fact estimable [34]. Knowledge about a physical system is often limited and the amount of data observed is incomplete and noisy. As a result, uncertainty accumulates due to model choice and experimental errors, and transfers to uncertainty in the parameter estimates [34]. Has the right type of data been collected for the model? Has a sufficient amount of data been collected? Does the likelihood function have a maximum and is it unique? Can the parameters be uniquely identified? If not, what are sources of uncertainty? How does the parameter uncertainty affect the problem at hand? How does it affect the inference performance? What can be done to improve the situation? [169, 34] This section introduces the main concepts of parameter identifiability and methods to analyse it.

3.3.1 Definition of identifiability

A parameter is said to be identifiable, if it is possible to infer its value from the data. That means, if two parameter values are equally likely to describe the data, they must be equal. In other words, the maximal likelihood is attained by a unique parameter value.

Definition 3.3 (Parameter identifiability [169])**.** The parameter θ is called globally (or uniquely) identifiable, if the function $\theta \mapsto p(\theta|x)$ is injective:

$$p(\theta_1|x) = p(\theta_2|x) \Rightarrow \theta_1 = \theta_2 \qquad \forall \theta_1, \theta_2 \in p(\theta) \tag{3.28}$$

It is only locally identifiable, if this relation holds only in a neighbourhood of values. It is called non-identifiable, if it is neither globally nor locally identifiable. In the case of local identifiability, multiple solutions to the parameter estimation problem exist. If the parameter is locally as well as globally non-identifiable, an infinite number of solutions might exist [34]. It should be noted, that although a parameter is theoretically identifiable, the inference in practice can be difficult [170].

In the Bayesian setting, the concept of identifiability is not so clearly defined and subject to discussion. A much quoted opinion was expressed by Lindley, who said that non-identifiability does not cause real difficulties in the Bayesian approach [171]. Indeed, an advantage of the Bayesian approach is, that models that are non-identifiable in the frequentist theory can become identifiable in the Bayesian theory through the definition of a suitable prior [172]. Identifiability is then a problem of the likelihood [173]. But even if the likelihood is non-identifiable, a posterior distribution for every parameter can always be obtained. It is just a question whether this estimate is at all meaningful [171], since all information comes solely from the prior and no additional information is gained through the data. In this case, the MCMC sampling can encounter difficulties with convergence and the result can be misleading.

3.3.2 Diagnosing non-identifiability

There are some symptoms that indicate non-identifiability [34]:

- The parameters cannot be uniquely identified, even for noise-free data.
- Inference results in broad posterior distributions with large parameter uncertainty.
- The posterior is equal to the prior, no information gain through the likelihood/ the data
- Different parameter values yield the same likelihood value, low sensitivity of the likelihood towards change in parameter values
- High correlation between the parameters
- Different runs of the inference with different or even the same initialisation yield different parameter estimates
- Model output is inaccurate and predictions not possible.

3.3.3 Sources of non-identifiability

Realising that an identifiability problem exists is one thing. The question is what causes this non-identifiability. Two sources of parameter non-uniqueness can be distinguished. The first reason, why parameters might not be (uniquely) inferable, could be due to the choice of the model concept or the model equations. This is called **structural non-identifiability**. Underdetermined or ill-posed model equations can be singled out theoretically before any data is observed [34]. The other reason for non-identifiability is data-dependent and called **practical non-identifiability**. On the one hand, an insufficient amount of data might be supplied or the wrong kind of data at all. Also unsuitable initial or boundary conditions can be an issue [34]. On the other hand, the supplied data can in principle be suitable, but contains errors or noise [34]. Careful experimental design is a start for improvement [87].

3.3.4 Identifiability analysis

There are different approaches to analyse the identifiability of a model. **Analytical methods** investigate the structure of the model equations before any data is observed and typically require advanced mathematical techniques [174]. If a model a priori turns out to be structurally non-identifiable, it will obviously be practically non-identifiable as well. However, if the analytical method assures structural identifiability, this does not imply practical identifiability [34]. Practical identifiability can only be assessed using **data-based methods** [35]. Investigation of the likelihood function for local and global maxima and the curvature around them is commonly done via visualisation of the response towards different parameter values. A problem is globally identifiable, if the response surface shows a unique maximal peak. Multiple distinct peaks with the same maximal likelihood function values (global optima) make the problem locally identifiable but globally non-identifiable. Flat surfaces may occur when combinations of parameter values result in the same likelihood values, indicating parameter interaction [34]. In this case, the problem is locally and globally non-identifiable. Visualisation provides a practical approach to understand the shape of the response surface, but in higher dimensions visualisation becomes tricky. A systematic approach is required [34].

The flatness of the likelihood surface due to parameter correlation can be investigated using the covariance matrix. Flat directions are characterised by large eigenvalues thereof. The corresponding eigenvectors specify the direction. For a gaussian likelihood, instead of looking at the covariance matrix Σ, the Hessian (or curvature matrix) $\mathcal{K}$ of the likelihood can be analysed. It is the negative inverse of the covariance $\mathcal{K} = -\Sigma^{-1}$ and hence, large eigenvalues of Σ correspond to small eigenvalues of $\mathcal{K}$. The Hessian of a function is the matrix of the second-order partial derivatives of the function. The **Hessian of a normal likelihood** is therefore

$$\mathcal{K}_{ij}(x) = \frac{\partial^2 \mathcal{L}(x|\theta)}{\partial\theta_i \partial\theta_j}, \tag{3.29}$$

which is independent of the parameters θ.

Let $\lambda_1, \ldots, \lambda_D$ be the eigenvalues of $\mathcal{K}$ and $w^{(\lambda_1)}, \ldots, w^{(\lambda_D)}$ the corresponding eigenvectors. Normalisation of the eigenvalues by the sum of all eigenvalues allows to quantify the information carried by the respective direction. We define the **proportion of variance explained** by eigenvector $w^{(\lambda_d)}$ as [175]

$$\mathcal{I}(w^{(\lambda_d)}) = \frac{\lambda_d}{\sum_{d=1}^{D} \lambda_d}. \tag{3.30}$$

Low variance explained, resulting from small eigenvalues, means high correlation and therefore a flat likelihood. To find out which parameters are responsible for this low information, the eigenvectors are themselves normalised by the sum of their elements. We define the parameter contribution to direction $w^{(\lambda_d)} = (w_1^{(\lambda_d)}, \ldots, \ldots w_D^{(\lambda_d)})$ by

$$\mathcal{C}(w^{(\lambda_d)}) = \frac{w^{(\lambda_d)}}{\sum_{i=1}^{D} w_i^{(\lambda_d)}}. \tag{3.31}$$

In multi-dimensional likelihoods, there might be multiple directions, in which the surface is flat. Let $\mathcal{D}$ be the set of indices of vanishing eigenvalues and $w^{(\lambda_d)}$ with $d \in \mathcal{D}$ the corresponding eigenvectors. We denote the **subspace spanned by the low-informative eigenvectors** $W_{\mathcal{D}}$ and it is determined by the outer product of these vectors with themselves:

$$W_{\mathcal{D}} = \sum_{d \in \mathcal{D}} w^{(\lambda_d)} w^{(\lambda_d)\mathsf{T}} \tag{3.32}$$

The diagonal $\mathrm{diag}(W_{\mathcal{D}})$ of this matrix gives the **contribution of parameters to the low-informative space**. Sorting the elements in decreasing order identifies the parameters which are least restricted by the data and will probably run into problems in the inference. In summary, the identifiability of the parameters given the data is quantified by the degree to which they contribute to possibly flat directions [176].

3.3.5 Dealing with non-identifiability

Once non-identifiability is diagnosed and the causes are detected, the question is, how to deal with it. This depends on the level of output uncertainty induced and the purpose and context of the model. Even if the parameter estimates are inaccurate, the model output can still be tolerable. If not, the researcher can try to improve the model structure or the data collected. A thorough model analysis can be performed to adjust model equations and assumptions or the objective function can be reconsidered. Some parameters might be possible to eliminate or to replace by inferable relations between parameters instead, certain parameter values could be fixed. In the data collection process different biological conditions could be analysed or the recording technique could be changed to result in more precise measurements. However, in practice the right data can be hard or even impossible to obtain. In the inference, settings can be altered (like the prior distribution) or the method used can be changed to a more effective algorithm [34, 169, 35].

3.4 Performance evaluation

A crucial component of algorithm development is performance evaluation. One way to achieve this is to run the algorithm in a controlled setting on data where the true solution is known and can be compared to the model outcome. The parameters leading to the model outcomes, however, are not necessarily observed and a good model fit does not allow to draw conclusions about the estimated parameters as such [145].

3.4.1 Simulated data

A common practice in parameter inference is the generation of synthetic data that mimics the data obtained by real-world experiments in structure and size. The model under investigation is used to simulate data from a particular set of parameter values. The algorithm can then be applied to the simulated data, pretending the parameter values are unknown. Subsequently, the inferred parameter values can be compared to the correct parameter values used to create the data. For single point estimates, the difference of the inferred value to the true value should be minimal. For Bayesian inference, the true parameters should be covered by the uncertainty of the posterior distribution. This offers an objective assessment of the algorithm performance. [87, 145]

In general, a single simulated dataset will not be meaningful for performance evaluation. This simulation framework provides a cheap and rapid way to create as much data as needed for algorithm validation. Different conditions can be simulated and the effect of measurement noise in the data can be studied. It can help to build the model, to understand the influence of parameters across the parameter space and to confirm the validity of the inference method applied. In the case of Bayesian inference, it can be used to check whether the observed data provides information beyond the prior information. On the other hand, this procedure also provides opportunity to demonstrate limitations of the algorithm to deepen understanding of the model and method used. If the algorithm can be calibrated to perform reproducibly and reliably on the synthetic data, it can be applied to real data. There is, however, no guarantee that the performance will be comparable. [87, 145]

See Section 4.8.1 for details on how simulated data for QPuB is generated.

3.4.2 Error measurement

We will use precision, accuracy and repeatability to evaluate the algorithm performance. Various definitions of these terms exist. Descriptions of how we define the terms in this thesis are given below. An illustration of the difference between precision and accuracy is depicted in Fig. 3.3.

The **precision** quantifies the experimental uncertainty of the inference. It can be expressed by numerical exactness in terms of decimal digits, or by how close a series of estimated values of a parameter are to each other. In Bayesian inference, a fit is said to have high precision, if the standard deviation of the obtained distribution around the inferred mean is small [177]. For noise-free data, the parameters should be inferred with minimal uncertainty. For noisy data, clearly the uncertainty will be larger depending on the level of noise.

The **accuracy** evaluates the correctness of the estimate, i.e. how close the estimated values are to the correct values of the parameters. In case of Bayesian posterior parameter distributions, usually the mean, the median or the maximum a posteriori probability estimate (the mode(s)) is used. The inference results are said to have high accuracy, if the deviation from the true values is small [177].

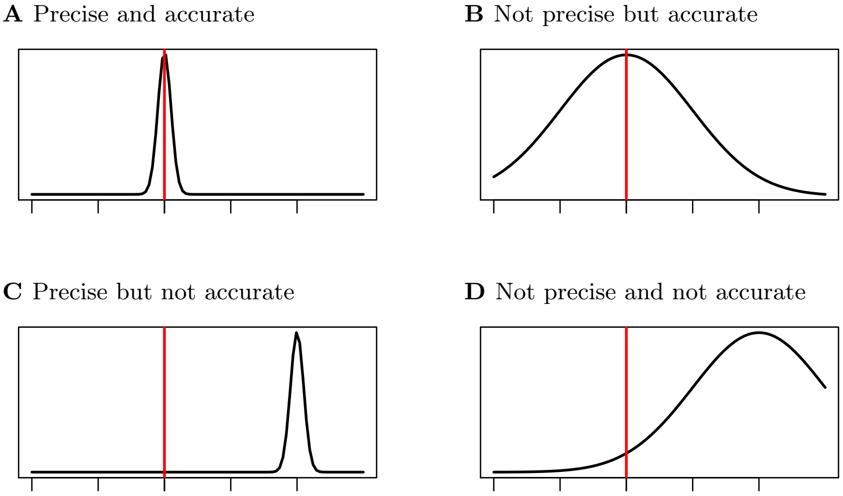


Fig. 3.3: Inference errors: precision vs. accuracy. This figure illustrates the difference between precise and accurate posterior estimates. An estimate is precise, if the standard deviation is small. An estimate is accurate, if the mean is equal to the correct value (vertical red line). Accuracy can therefore only be assessed if the true value is known, e.g. in model calibration using simulated data. **(A)** The optimal posterior distribution is both precise and accurate. **(B)** In case of large uncertainty in the data, the estimate can be of low precision but still accurate. **(C)** An estimate which is precise but not accurate is the worst outcome. **(D)** An estimate which is not precise and not very accurate is not optimal, but still acceptable if the correct value is at least covered by the distribution. The definition of what is "sufficiently precise and accurate" is up to the user. [Figure inspired by Fig. 4.1 in [177].]

Two kinds of error sources can be distinguished: random error and systematic error. Random error is an effect of statistical variability and affects the algorithm precision. Systematic error, on the other hand, is an effect of statistical bias and affects the accuracy. While the random error is directly evident from the inference results, the systematic error can only be assessed through the help of simulated data with known true values. [177] The deviation of an estimate $\hat{\theta}$ to the true value θ^* can be represented in different forms [177]. The absolute error is defined as the absolute value of the difference between the estimated and the true value:

$$\epsilon = |\theta^* - \hat{\theta}| \qquad \text{(absolute error)} \qquad (3.33a)$$

Dividing the absolute error by the magnitude of the true value (provided $\theta^* \neq 0$), gives the relative error:

$$\eta = \frac{\epsilon}{|\theta^*|} \qquad \text{(relative error)} \tag{3.33b}$$

which can also be specified in percent:

$$\delta = 100\% \times \eta \qquad \text{(percent error)} \tag{3.33c}$$

Ideally, inference results have both high accuracy and high precision. The average estimate should coincide with the true value, i.e. the error should be zero. If this cannot be achieved (due to noise or missing data), at least the correct value should be covered by the posterior distribution with tight standard deviations.

Repeatability is a measure of the reliability of an inference. Repeated measurements should yield the same results, when all conditions are kept constant [177]. The inference is repeated on the same data, applying the same computational setup, same algorithm settings and same conditions over a short period of time. Experimental measurements should be repeated a few times for statistical significance. This is usually not a problem for computer simulations, where it is easy to repeat an inference an arbitrary number of times. The variability that arises can be reported using the standard deviation of the collection of posterior means obtained. Let $n = 1, \dots, N_{\text{rep}}$ be the number of repeated inference runs with estimated means μ_n, then

$$\sigma_\mu = sd(\{\mu_1, \dots, \mu_{N_{\text{rep}}}\}). \tag{3.34}$$

A value of 0 is ideal, but unrealistic in an inference scheme due to the inherent randomness. It is up to the user to decide on his acceptable threshold.

4 | Implementation and Benchmarking framework

We propose a software package called QPuB (Quantification of Peptides using Bayesian inference). It is based on the ideas of Peters et al. [37] and Mishto et al.[39] and adapts their notations. The goal is to computationally derive the amounts of all peptide products based on the MS signal intensities and only the titration of the substrate. The quantity that links the two is called conversion factor. The estimation of the conversion factors for every peptide product is achieved using Bayesian inference in an MCMC scheme. In this chapter, the main components and settings of the QPuB framework are described. A schematic of the QPuB pipeline is shown in Fig. 4.1 and the QPuB algorithm is summarised in Alg. 3 using pseudo code. The QPuB code was implemented in R version 4.0.3 (2020-10-10) [178] using RStudio [179], implementing the following packages [180, 181, 182, 183].

4.1 Notation

Digestion. The substrate or the substrate sequence is denoted by S. Let D be the total number of peptide products generated by digestion of the substrate by the enzyme. The individual peptides are referred to by P_d, where $d = 1, \ldots, D$. Let T denote the total number of measuring points in digestion time and let $t = 0, \ldots, T$ be the counter. The total number of amino acid residues in the substrate sequence is denoted by A and the amino acid position in the substrate is counted by $a = 1, \ldots, A$.

Kinetics. The amounts of the peptide products in the loading volume over digestion time are symbolised by $c_d(t)$ (for 'concentration'):

$$C : [0, D] \times [0, T] \to \mathbb{R}_+, \tag{4.1}$$

$$C[d, t] = c_d(t). \tag{4.2}$$

By convention, the index $d = 0$ refers to the substrate, so $c_0(t)$ denotes the amount of the substrate over time. Similarly, $s_d(t)$ stands for the measured MS signal intensities over time:

$$S : [0, D] \times [0, T] \to \mathbb{R}_+, \tag{4.3}$$

$$S[d, t] = s_d(t). \tag{4.4}$$

Parameters. The conversion factors that link the MS signals and the amounts for all peptide products are typified by the vector $v = (v_1, \ldots, v_D)$. They are the unknown parameters of interest to be inferred by the algorithm. Along with that, a nuisance parameter σ is also estimated, which denotes the standard deviation of the normal distribution in the likelihood function. Hence, the parameter set becomes

$$\theta = (v_1, \ldots, v_D, \sigma), \tag{4.5}$$

which has length $D + 1$.

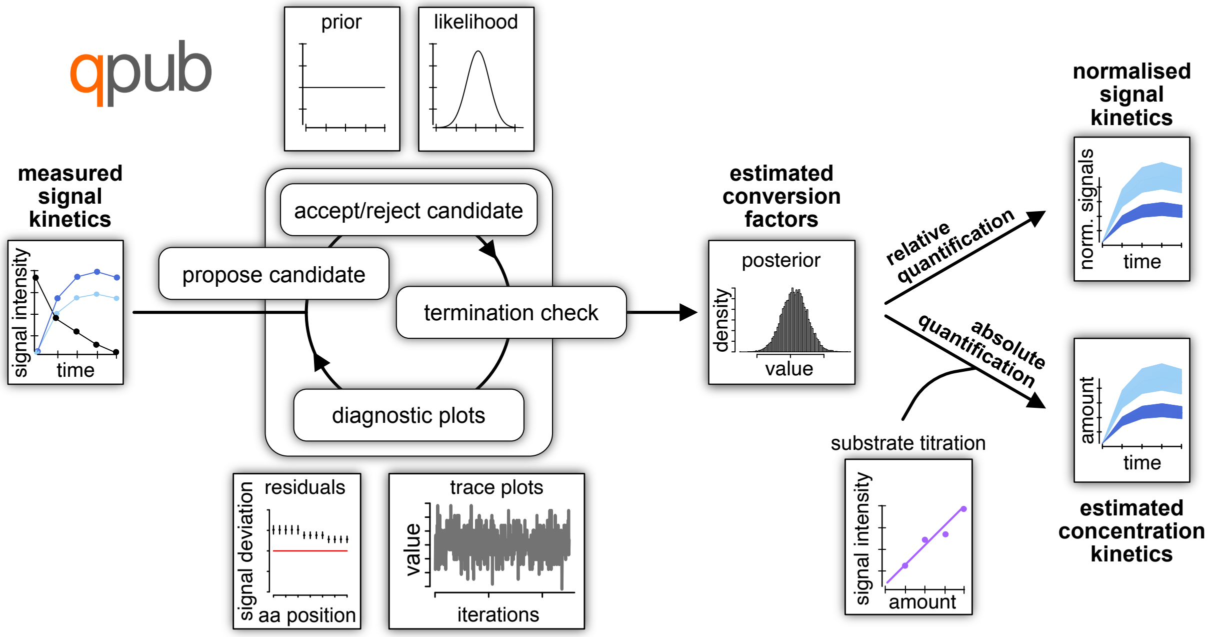


Fig. 4.1: Schematic of the QPuB package. QPuB takes MS signal intensities over digestion time as input and returns normalised signal intensities or absolute concentrations, if the substrate titration is provided. The core module is an algorithm based on Bayesian inference using an iterative Metropolis–Hastings sampling scheme. Graphical output includes residual plots to investigate fulfilment of the mass balance condition, the full parameter posterior distributions and the final kinetic plots of the peptide amounts.

Algorithm 3 The QPuB algorithm. QPuB employs the DREAM$_{(ZS)}$ algorithm with three parallel chains.

```
Input: Signal intensities
Input: Slope of the substrate titration
Output: Posterior distributions of the conversion factors
Output: Estimations of the peptide amounts

  Data pre-processing

  Initialisation of the "fake" archive $\theta^{(c)}(1 : M_0) \sim p(\theta)$        ▷ 1. Initialisation
  Initialisation of the chains $\theta^{(c)}(M_0 + 1) \sim p(\theta)$
  Initialisation of the proposal function
  Initialisation of the convergence diagnostics

  for $i = 1, \dots, N$ do
      for every chain $c = 1, 2, 3$ do
          if uniform([0, 1]) > 0.1 then                                              ▷ 2. Proposal step
              if uniform([0, 1]) ≤ 0.1 then
                  $\gamma = 0.98$                                                    ▷ Gamma jump
              else
                  $\gamma = \frac{2.38}{\sqrt{2D+2}}$
              end if
              Propose candidate $\zeta^{(c)}$ according to parallel direction update (3.21)
          else
              $\gamma = \text{uniform}([1.2, 2.2])$
              Propose candidate $\zeta^{(c)}$ according to snooker update (3.23)
          end if

          Calculate $\alpha$ according to (3.26) using data                         ▷ 3. Metropolis step
          if uniform([0, 1]) ≤ $\alpha$ then                                         ▷ 4. Acc./Rej. step
              Accept candidate and move to new state (3.27a)
          else
              Reject candidate and stay at current state (3.27b)
          end if
      end for

      Create diagnostic plots                                                        ▷ Diagnostics
      Create convergence diagnostics
  end for

  Combine all chains excl. 50% burn-in                                               ▷ Post-processing
  Create summary statistics
  Calculate peptide amounts using (4.17)
```

4.2 Model assumptions

The main assumptions that QPuB is based on are the following:

1. For every peptide, the concentration is in linear relation to its signal intensity.
2. During an *in vitro* enzymatic digestion, the amount of amino acids in the solution remains constant: no amino acids are lost or created in the process of transforming the substrate into products (mass conservation).

4.2.1 Linear correlation

As described in Sec. 2.3.3.1, Fig. 2.5, the relationship between the loaded substance amounts and the measured MS signal intensities is linear in the pmol range relevant for enzymatic *in vitro* digestions:

$$s_d = m_d c_d + n_d \quad \forall d = 0, \ldots, D, \tag{4.6}$$

where m_d is the slope and n_d the intercept of the line. Subtracting the background signal, which is the signal at time 0, from all time points yields $n_d = 0$. The slopes of the peptide titration curves are multiples of the substrate slope: $m_d = \rho_d m_0$ with $\rho_d \in \mathbb{R}$. Using this observation and rearranging the equations results in

$$c_0 = \frac{1}{m_0} s_0 =: v_0 \, s_0, \tag{4.7}$$

$$c_d = \frac{1}{\rho_d \, m_0} s_d =: v_d \, s_d. \tag{4.8}$$

The inverse of the slope is called conversion factor. Every substance has its individual conversion factor which depends on its physico-chemical properties [37] and is instrument specific and constant in time.

The conversion factor of the substrate is now normalised to $v_0 = 1$. The peptide's conversion factors are all relative to the substrate.

4.2.2 Mass conservation

The second principle that we make use of is the law of mass conservation. Since we consider a closed system of *in vitro* digestions, we know that each amino acid present in the solution at time $t = 0$ is still present at any time $t > 0$. We assume the ideal case that all peptide products are detected in the MS analysis. Formally, the sum of amino acids of the products should be equal to the amount of substrate degraded since the beginning, for every amino acid position and for every time point:

$$\sum_{d=1}^{D} c_d(t) b_{da} = c_0(0) - c_0(t) =: \Delta c_0(t) \quad \forall a, t. \tag{4.9}$$

The factor $b_{da} \in [0, 1]$ is the probability that the product d contains amino acid a. The amount of substrate degraded since time $t = 0$ is $\Delta c_0(t)$. Now, substituting the expression for c_d according to equation (4.8), we have

$$\sum_{d=1}^{D} v_d s_d(t) b_{da} - \Delta c_0(t) = 0 \quad \forall a, t. \tag{4.10}$$

This mass conservation will build the basis for our likelihood function (see Sec. 4.4.2).

4.3 Input and pre-processing

Mandatory input for the algorithm includes the measured MS signal intensity kinetics of the substrate and the peptide products, as well as the titration measurements of the substrate.

4.3.1 MS signal intensities

QPuB takes as data input a table containing the amino acid sequences of the substrate and all detected and identified peptide products in one letter code and their measured MS ion peak area values over digestion time. Usually, the protein digest is documented over a digestion time of a few hours with regular intermediate measurements. Depending on the protein digest of interest, the number of peptide products can vary from just a few to a few thousands. Multiple measurement replicates can be provided.

4.3.2 Data preparation

Before the inference is run, the input is controlled and certain characteristics of the input data are extracted. To infer where the peptide products originate from in the substrate, the product sequence is aligned to the substrate sequence, taking non-uniqueness into consideration. If a product can be created by the process of simple cleavage, then it is assumed to be a cleavage product. If the peptide sequence cannot be found in the linear sequence of the parental molecule, the peptide is identified as a proteasome-generated spliced product.

Example. Assume a substrate with sequence ABCDEFGHABCD and the products with sequences DEFGH and DEFABC, respectively. We number the amino acids in the substrate consecutively $a = 1, \ldots, 12$. Since the sequence of peptide 1 is a direct subsequence of the substrate sequence, it is identified as cleavage peptide stemming from the substrate positions 4 to 8. On the other hand, the sequence of peptide 2 cannot be directly aligned to the substrate sequence and therefore is identified as spliced peptide, originating from ligating the stretch between positions 4 and 6 to either the subsequence 1 to 3 or the subsequence 9 to 11.

Using the above position information, the position probability matrix $b \in \mathbb{R}^{D \times A}$ is defined. It serves to identify which peptide products contain which amino acids to be able to control mass balance. It contains the probabilities of the amino acid $a = 1, \dots, A$ of the substrate being used in the production of peptide $d = 1, \dots, D$. The entry b_{da} is defined by

$$b_{da} = \frac{1}{N_a}, \tag{4.11}$$

where N_a is the number of possible origins the amino acid at position a has in the substrate.

Example. In the example above, the frequencies of the single amino acids in the substrate sequence are $N_{1:12} = [2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2]$. Consider the product $i = 1$, the cleaved 5mer DEFGH with positions 4 to 8. According to Eqn. (4.11), the positions 4 to 8 then get the matrix entries $b_{14} = b_{15} = b_{16} = b_{17} = b_{18} = \frac{1}{1} = 1$, all the other positions have probability 0, therefore $b_{1:} = [0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0]$. Product 2, on the other hand, does not have a unique position code assigned. The amino acid A, for example, could stem either from position 1 or 9 in the substrate, making $N_1 = N_9 = 2$. We consider both possibilities equally likely. Hence the probability of A in the peptide 2 stemming from position 1 in the substrate is $b_{21} = \frac{1}{2}$ by Eqn. (4.11). In total, the second row of the position probability matrix reads $b_{2:} = [0.5, 0.5, 0.5, 1, 1, 1, 0, 0, 0.5, 0.5, 0.5, 0]$.

In order to control mass balance, the amount of products generated must be compared to the amount of substrate degraded. Instead of referring to the mass change since the beginning of the digest, the difference between one time point and the next is calculated and the dataset of signal differences is denoted by an apostrophe S' with elements $s'_d(t) := s_d(t) - s_d(t-1)$ and analogously the matrix of differences in concentration over time C' with $c'_d(t) := c_d(t) - c_d(t-1)$ for all peptides $d = 1, \dots, D$.

Finally, all signals are scaled by 100 times the mean order of magnitude of the substrate to avoid high orders of magnitude and potential numerical issues arising from it.

4.3.3 Substrate titration

The initial amount of substrate loaded $[S](0)$ in the *in vitro* digest under consideration must be provided. Additionally, the titration information of the substrate in the form of loaded substrate amounts and corresponding measured signal intensities are required. If the titration data is provided by the user, we fit it using a linear model (4.6) $s_0 = m_0 c_0 + n_0$. With the obtained slope and intercept, the peptide amounts can be retrieved after the inference.

4.4 Bayesian setup

In Section 3.1 the main components of a Bayesian inference scheme were described. This section presents the concrete definitions used in QPuB.

4.4.1 The prior distribution

The prior probability distribution expresses the initial belief about the unknown parameters, before any data is observed. To be unbiased, the QPuB algorithm uses a weakly informative flat prior, namely a uniform one:

$$p(\theta) = \sum_{d=1}^{D+1} \theta_d, \tag{4.12}$$

where

$$\theta_d \sim \begin{cases} \mathcal{U}(10^{-4}, 10^4) & \text{for } d = 1, \ldots, D \text{ (conversion factors),} \\ \mathcal{U}(10^{-20}, 10^4) & \text{for } d = D+1 \text{ (sigma).} \end{cases}$$

The range for the conversion factors is chosen like this, because experimental observations suggest that the conversion factors span eight orders of magnitude, and because the factor for the substrate is set to 1 by default.

4.4.2 The likelihood function

The likelihood in QPuB is based on the law of mass conservation and defined using the objective function (4.9). The likelihoods for the independent observations over the amino acid positions a and time points t and measurement replicates r multiply (see Eqn. (3.4)). In this thesis, we deal with noise-free data only, therefore the number of replicates is 1 and can be omitted in the equation. For convenience, we are using a log-likelihood $\mathcal{L}(S'|\theta) = \log p(S'|\theta)$. Considering a normally distributed deviation

$$\sum_{d=1}^{D} v_d s'_d(t) b_{da} - c'_0(t) \sim \mathcal{N}(0, \sigma),$$

the log-likelihood can be defined as

$$\mathcal{L}(S'|\theta) = -\frac{1}{2\sigma^2} \sum_{t=1}^{T} \sum_{a=1}^{A} \Big(\sum_{d=1}^{D} v_d s'_d(t) b_{da} - c'_0(t) \Big)^2 - \frac{AT}{2} \log(2\pi\sigma^2). \tag{4.13}$$

4.4.3 The posterior distribution

The posterior distribution is calculated using Bayes' theorem (Eqn. (3.2)). As only ratios of posterior densities are considered in the metrolopis step and the marginal likelihood therefore cancels itself out (3.1.7), it suffices to calculate the product of the prior and the likelihood (Eqn. (3.10)). Since a log-likelihood is used, this simplifies to a calculation of

$$p(\theta|S') = \exp(\log(p(\theta)) + \mathcal{L}(S'|\theta))). \tag{4.14}$$

4.5 MCMC setup

The posterior can be approximated using an MCMC scheme (Sec. 3.2). QPuB employs a variant of the DREAM$_{\mathrm{(ZS)}}$ algorithm (Sec. 3.2.4.1). Convergence is assessed graphically and using the Gelman–Rubin criterion (Sec. 3.2.3).

4.5.1 Iterative sampling scheme

QPuB is based on a mixture of descriptions from the publications of the authors of the DE-MC family [41, 42, 43, 168, 44, 45], and their MATLAB and Python implementations as available online, as well as an independent R implementation by Florian Hartig [184]. A list of variable settings used in QPuB is provided in Tab. 4.1.

Table 4.1: DREAM$_{\mathrm{(ZS)}}$ settings used in QPuB.

setting	value
number of chains C	3
length of "fake" archive Z_0	10(D+1)
number of proposal pairs δ	1
snooker update probability p_{snooker}	0.1
gamma tuner β_0	1
gamma jump probability p_γ	0.1
proposal noise b_1	0.2
proposal noise b_2	0
boundary handling	fold
crossover rate CR	1
thinning	1000
burn-in	50%

Instead of having two separate objects of the current state and the archive of past states, QPuB defines a joint chain matrix $\theta_d^{(c)}(i)$ of dimension $N \times (3 * (D+1))$ containing all past and present states of the N chains over iteration time as rows:

$$\begin{array}{ccccc|ccccc|ccccc} \multicolumn{5}{c}{\text{chain 1}} & \multicolumn{5}{c}{\text{chain 2}} & \multicolumn{5}{c}{\text{chain 3}} \\ v_1 & v_2 & \cdots & v_D & \sigma & v_1 & v_2 & \cdots & v_D & \sigma & v_1 & v_2 & \cdots & v_D & \sigma \\ \hline \multicolumn{5}{c|}{\cdots} & \multicolumn{5}{c|}{\cdots} & \multicolumn{5}{c}{\cdots} \end{array} \tag{4.15}$$

1. Initialisation. The "fake" archive is initialised with a length of $Z_0 = 10\,(D+1)$ by sampling from the prior. The chains are by default initialised from values dispersed over the prior range, namely one chain from the lower bound, one chain from the upper bound, and the third chain from the middle point of the prior range:

$$\theta_d^{(c)}(1:Z_0) \sim p(\theta) \qquad \text{for } c = 1, \ldots, C \tag{4.16a}$$

$$\theta_d^{(1)}(Z_0+1) = 1 \times 10^{-4} \qquad \text{for } d = 1, \ldots, D \tag{4.16b}$$

$$\theta_d^{(1)}(Z_0+1) = 1 \times 10^{-20} \qquad \text{for } d = D+1 \tag{4.16c}$$

$$\theta_d^{(2)}(Z_0+1) = 1 \times 10^{4} \qquad \text{for } d = 1, \ldots, D+1 \tag{4.16d}$$

$$\theta_d^{(3)}(Z_0+1) \approx 0.5 \times 10^{4} \qquad \text{for } d = 1, \ldots, D+1 \tag{4.16e}$$

2. Proposal step. The proposal employs a mixture of 90% parallel direction update and 10% snooker update. The symmetry factor was implemented with an exponent of $D+1$. The gamma jump distance is set to 0.98. The crossover step is not implemented in the current version of QPuB.

The **Metropolis step** and the **Acc./Rej. step** are performed like in the original algorithm.

4.5.2 Diagnostic output

Over the course of iterations, diagnostic output is produced to track the performance of the algorithm. **Trace plots** and **posterior histograms** are generated (Sec. 3.2.3). In addition, the mass balance is illustrated in **residual plots** (Fig. 4.1). Using the current conversion factor values in the current iteration, the estimated peptide amounts are calculated and compared to the amount of substrate degraded. For every pair of consecutive time points and for every amino acid position, the plot shows the degree to which the current parameter estimates fulfil the mass balance condition. Over the course of the iteration, the product residual distributions should converge towards the amount of substrate digested. The residuals are summarised by the mean total residuals per chain.

4.5.3 Convergence criterion

The QPuB run is terminated after a fixed number of iterations defined by the user. Graphical portrayal of the Markov chains (Sec. 4.5.2) is created and the potential scale reduction factor defined by the Gelman–Rubin diagnostic (3.18) is calculated excluding the nuisance parameter σ. The authors of the DREAM$_{(\mathrm{ZS})}$ algorithm use $\hat{R} < 1.2$ as a threshold to diagnose convergence. The chains were usually run longer than actually necessary to assess how precise the estimates could potentially become.

4.6 Output and post-processing

After the inference, QPuB returns the full posterior distributions of the estimated parameters. A generous burn-in fraction of 50% of the samples is discarded. Since after convergence, all chains sample from the same stationary distribution, the chains can be combined into a single object. From this, the estimated conversion factors are used to calculate estimated peptide amounts.

4.6.1 Estimated conversion factors

To obtain a measure for the best guess as well as the uncertainty of the estimate, the following overall summary statistics are calculated for the distributions of the conversion factors of all peptide products and the nuisance parameter σ: mean and median, standard deviation, 5% and 95% quantiles, minimum and maximum.

4.6.2 Estimated peptide amounts

From the distributions of the conversion factors, the peptide amounts can be obtained. Let m_0 be the slope and n_0 be the intercept of the substrate calibration curve as obtained in Sec. 4.3.3. Using the median $\bar{v}_d$ of the parameter distribution of peptide d, the median amount $\bar{c}_d(t)$ of product d can be calculated from the input signal intensities for every point in time:

$$\bar{c}_d(t) = \frac{\bar{v}_d s_d(t)}{m_0} - n_0. \tag{4.17}$$

We proceed accordingly for the other quantiles of the conversion factor distributions: The 5% quantile of the conversion factor distribution is used to obtain the 5% quantile of the corresponding concentration distribution; likewise for the 95% quantiles. The result is an estimated concentration kinetics for every peptide product of the digestion including uncertainty.

If the substrate titration was not provided by the user, only normalised signals are returned: $\bar{c}_d(t) = \bar{v}_d s_d(t)$. These can be used for relative quantification between the different peptide products but they are not absolute amounts. This is already a considerate advantage of the QPuB pipeline. Common label-free quantification only allows for relative quantification of the same peptides between samples.

4.7 Identifiability analysis

Before the inference is run, we can check whether the dataset of interest contains sufficient information for a potentially successful inference. Using the theory described in Section 3.3, we identify difficulties and indications for improvement. The curvature matrix of the dataset is determined and its eigenvectors and eigenvalues are calculated. As seen in Section 3.3.4, the curvature matrix is defined as the second derivative of the function of interest, in this case the likelihood function. The matrix of second partial derivatives of the log-likelihood (4.13) is then proportional to

$$\mathcal{K}_{ij}(v) = \frac{\partial^2 \mathcal{L}(S'|v)}{\partial v_i \partial v_j} \propto \sum_{t=1}^{T} \sum_{a=1}^{A} s'_i(t) s'_j(t) b_{ia} b_{ja}. \tag{4.18}$$

Note that the curvature matrix of our likelihood is independent of the parameters v. This allows us to calculate it using the data before the inference to assess parameter identifiability (Sec. 3.3).

4.8 Benchmarking framework

As described in Section 3.4.1, to be able to validate the model and calibrate the algorithm, synthetic data needs to be constructed. Even the most complex biochemical processes can usually be divided into elementary binding reactions [87]. For the creation of data, we will make the simplifying assumption, that the proteases used in the enzymatic digest follow the Michaelis–Menten reaction kinetics (Sec. 2.2.3). Complex details in the reaction steps will be neglected. Using the MM model equations (2.2a), data is simulated for benchmarking of the QPuB algorithm performance. The benchmarking framework of Section 3.3 will be explained for the specific case of QPuB.

4.8.1 Creation of simulated data for QPuB

The concentration kinetics of enzymatic digests of hypothetic substrates into products was simulated following several different cleavage patterns combined with different reaction rates to achieve a great variety of datasets. Using the concentrations and predetermined conversion factors, the signal kinetics was calculated. For evaluation of the algorithm performance, QPuB is applied to the simulated signal intensity data. The inferred conversion factors can then be compared to the true known values used to create the data. In addition, the calculated concentration kinetics can be compared to the true kinetics (see Fig. 4.2).

Simple enzymatic reactions were considered yielding one or two peptide products. Generated products may continue to be digested by acting as substrates to the enzyme themselves. The digestion pattern can be described by $j = 1, \dots, J$ reactions.

The initial concentration of the reactants were set to the following values:

$$[S](0) = 200\,\text{pmol}, \tag{4.19a}$$

$$[E](0) = 0.05\,\text{pmol}, \tag{4.19b}$$

$$[ES](0) = 0\,\text{pmol}, \tag{4.19c}$$

$$[P_d](0) = 0\,\text{pmol} \qquad \text{for all } d = 1, \dots, D. \tag{4.19d}$$

The digestion time was set to 4 h with half-hourly measurements. The binding and unbinding rates were sampled from discrete intervals for every reaction j:

$$k^j_{\text{on}} \in \{x \in \mathbb{Z} \mid 1 \leq x \leq 100\} \tag{4.20a}$$

$$k^j_{\text{off}} \in \{x \in \mathbb{Z} \mid 500 \leq x \leq 1000\}. \tag{4.20b}$$

The ranges were chosen arbitrarily to yield suitable digestion kinetics over time. The catalytic rate constant was modified to depend on the concentration of one of the products P (by default the N-terminal cleavage product):

$$k^j_{\text{cat}} = \frac{k^j}{1 - \frac{p^j[P](t)}{[S](0)}}, \tag{4.21}$$

where k^j was sampled from an interval of positive integers $k^j \in \{x \in \mathbb{Z} \mid 1 \leq x \leq 500\}$ and p^j is a factor sampled from $p^j \in \{0.1, 0.2, 0.3, \dots, 1\}$.

With these specifications, we simulated a digestion according to the Michaelis–Menten equations (2.2a), yielding numerical concentration kinetics of all reactants. We then removed the complexes from the system, by adding the concentrations of the complexes to the concentrations of the respective substrates to ensure mass balance.

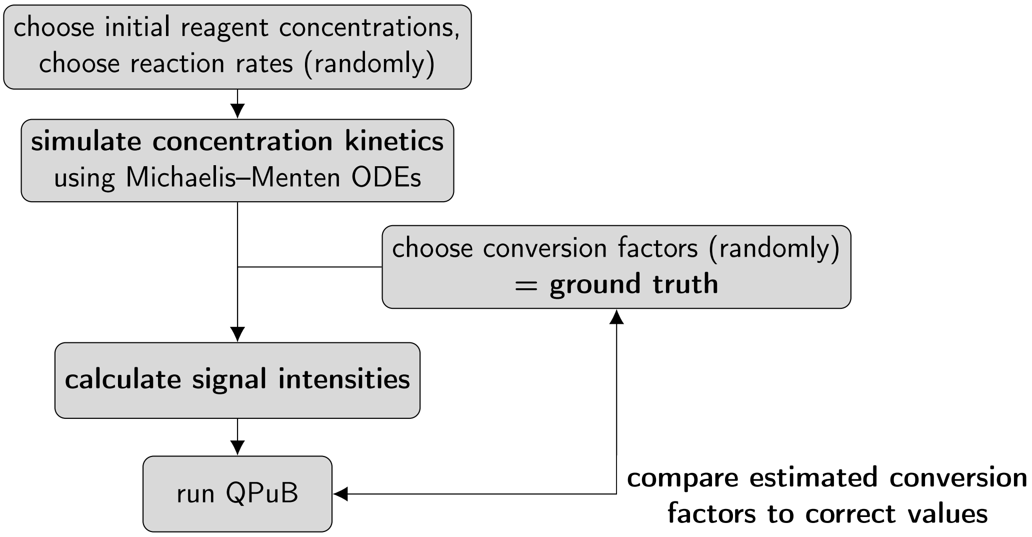

choose initial reagent concentrations, choose reaction rates (randomly)
simulate concentration kinetics using Michaelis–Menten ODEs
choose conversion factors (randomly) = ground truth
calculate signal intensities
run QPuB
compare estimated conversion factors to correct values

Fig. 4.2: Schematic of *in silico* data simulation and model validation. Simulating data allows comparison of the inference results to the true underlying parameters for algorithm validation.

The conversion factors for all products were sampled from the prior distribution or a subinterval thereof. The signal intensities were calculated according to the principle of linear relation (4.8) and multiplied by a large factor to match a realistic order or signal magnitude:

$$s_i(t) = \frac{c_i(t)}{v_i} \cdot 10^{10}. \tag{4.22}$$

This procedure yields the exact signal intensity kinetics of the substrate and all peptide products over time. For more realistic data, noise was introduced to the system. We used multiplicative noise drawn from a normal distribution with different standard deviations:

$$s_i(t) = \frac{c_i(t)}{v_i} \cdot 10^{10} \cdot \nu \qquad \text{where } \nu \sim \mathcal{N}(1, \sigma_{\text{noise}}). \tag{4.23}$$

Several replicates of measurements were generated.

These simulations were implemented in the Julia programming language version 1.6.3 [185] using the *DiffEqBiological* package for chemical reaction models [186] and the *OrdinaryDiffEq* package for solving ODEs [187], as well as *Latexify* for convenient LaTeX formatting [188].

4.8.2 Performance evaluation

We use the criteria described in Section 3.4 to evaluate QPuB performance. The **precision** is evaluated using the standard deviations of the posterior distributions. **Accuracy** is assessed through comparison of the estimated conversion factor median to their correct values and reported in the form of percent error (Eqn. (3.33c)). The mean over all peptides will be reported:

$$\delta_v = \frac{1}{D} \sum_{d=1}^{D} 100 \cdot \frac{|v_d^* - \hat{v}_d|}{v_d^*}, \tag{4.24}$$

where $\hat{v}_d$ denotes the posterior median and v_d^* the true value of conversion factor d. Since the researcher is more interested in the inferred peptide concentrations rather than the conversion factor *per se*, the deviation of the median inferred concentrations to the true underlying concentrations is computed. **Repeatability** is quantified by the standard deviation of the mean across runs (Eqn. (3.34)).

5 | Application to simulated data

To investigate the performance of the algorithm and to calibrate the model, we applied QPuB to perfect simulated data without noise. We consider *in silico* datasets with differing complexity, generated as explained in 4.8.1. We assume that all products are detected and that mass balance over the time span of the kinetics is satisfied. Since the underlying true solution is known, the inference outcome can be compared to it. We compare the conversion factors, that were estimated from the simulated signal intensities, to the true values used to obtain the data. We also compare the therewith calculated concentrations to the true ones.

In Section 5.1, a detailed description of the QPuB pipeline is given on the basis of the simplest example possible. It serves as a proof of concept and illustrates the individual steps in the algorithm. After that, in Section 5.2 and Section 5.3, two examples are shown in comparison to demonstrate two scenarios that can happen when applying QPuB to a dataset. In Example 1 the parameter inference is successful. In Example 2, on the other hand, insufficient information is conveyed to successfully infer the parameters. To solve this issue, two different strategies to increase the information content of the system are applied. Finally, in Section 5.4 a more complicated dataset is presented, where the strategies proposed before coalesce and demonstrate a successful inference on 46 parameters.

5.1 Example 0

We begin with a miniature example of the smallest digest possible, with only two peptide products. This example serves to illustrate the important steps in the QPuB pipeline. We show that QPuB successfully infers the correct conversion factors for both peptide products with high precision and accuracy.

5.1.1 The data

For simplicity, we assume, that the amino acid sequence of the substrate S is the Latin alphabet. The enzyme E binds the substrate and cleaves it after amino acid position 12, releasing peptide products P_1

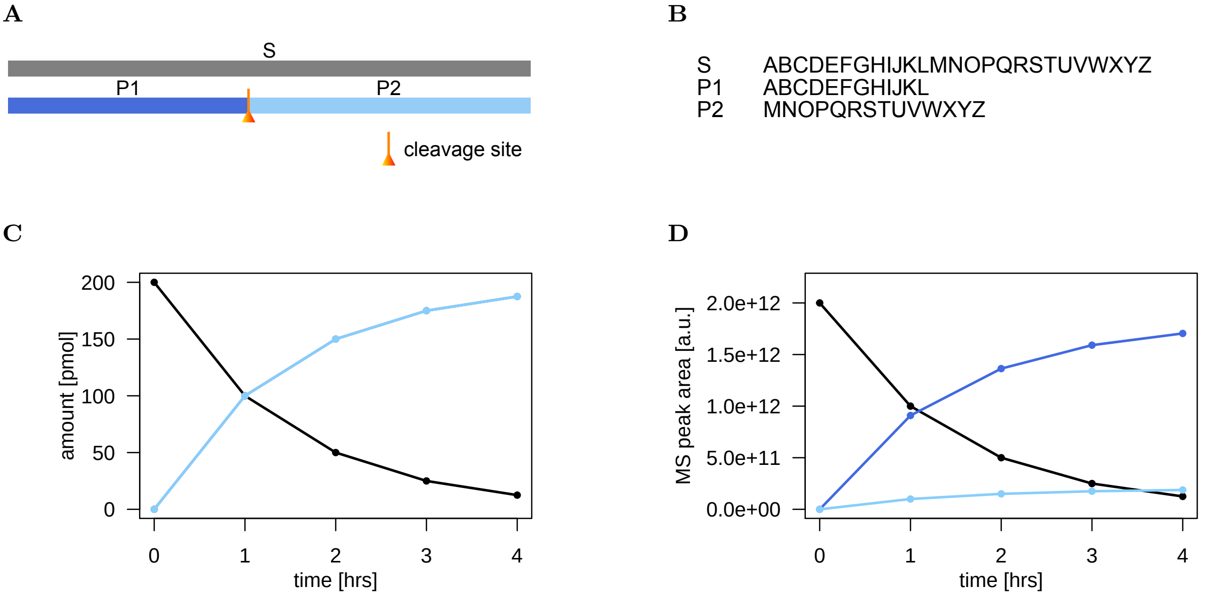


Fig. 5.1: Schematic of the cleavage pattern of Ex. 0 and the resulting kinetics. (A-B) Digestion pattern. The schematic depicts the digest of a 26-mer substrate by a specific endopeptidase into two peptide products (numbered accordingly). The enzyme hydrolyses the peptide bond at position 12, creating products 1 and 2. (B) shows the corresponding amino acid sequences. **(C) Kinetics of reactant amounts.** Depicted are the amounts of the substrate and the two peptide products over the digestion time, as simulated. Colors as in (A). Note that by construction, the amounts of the sibling peptides must be equal and are therefore not distinguishable in the plot. **(D) Kinetics of MS signal intensities.** Depicted are the MS signal intensities of the substrate and the peptide products over digestion time, as computed from (C) and the conversion factors shown in Tab. 5.1B. Colors as in (A). This serves as QPuB input. For numerical values of the kinetics see Tab. 5.1. Note that peptides with equal concentrations but different conversion factors will have different intensities. A comparison of different peptides based on their intensities is therefore not suited.

Table 5.1: Numerical values used in Ex. 0. (A) Kinetics of the reactant amounts. Amounts in [pmol] for the substrate and all peptide products over a digestion time of 4 hours, as simulated. **(B) Conversion factors.** Conversion factors of all peptide products, systematically chosen. The conversion factor of the substrate is set to 1 by default. **(C) Kinetics of MS signal intensities.** MS signal intensities in [a.u.] for the substrate and all peptide products over a digestion time of 4 hours, as calculated from (A) and (B) using Eqn. (4.8). Values were rounded to two digits precision for printing.

(A) Amounts [pmol]

Reactant	0h	1h	2h	3h	4h
S	200	100	50	25	12.5
P_1	0	100	150	175	187.5
P_2	0	100	150	175	187.5

(B) Conversion factors

Param	Value
v_0	1
v_1	1.1
v_2	10

(C) MS signal intensities [a.u.]

Reactant	0h	1h	2h	3h	4h
S	2.00×10^{12}	1.00×10^{12}	5.00×10^{11}	2.50×10^{11}	1.25×10^{11}
P_1	0.00	9.09×10^{11}	1.36×10^{12}	1.59×10^{12}	1.70×10^{12}
P_2	0.00	1.00×10^{11}	1.50×10^{11}	1.75×10^{11}	1.88×10^{11}

and P_2 (see Fig. 5.1A). For the purpose of illustration, we choose simple numerical values. The following initial conditions for the substrate and the products are used: $[S](0) = 200\,\text{pmol}$, $[E](0) = 0.05\,\text{pmol}$, $[ES](0) = [P_1](0) = [P_2](0) = 0\,\text{pmol}$, followed by a steady decay of the substrate with a consequent increase of the peptide product pair, as seen in Tab. 5.1A and Fig. 5.1C. The conversion factors were chosen as in Tab. 5.1B. Using the linear relation (4.8), the signal kinetics were calculated over a time course of 4 hours and shown in Tab. 5.1C and Fig. 5.1D. This serves as QPuB input. Note that peptides with equal concentrations but different conversion factors will have different intensities. A direct comparison of different peptides based on their intensities is therefore not suited.

5.1.2 Results

5.1.2.1 Data preparation

To be able to identify the mass balance of the system, QPuB first matches the sequence of the peptide products to the substrate using sequence alignment. The resulting substrate accessions are:

reactant	type	start	end
S	-	1	26
P_1	cleavage	1	12
P_2	cleavage	13	26

The position probability matrix looks as follows:

$$b = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}, \tag{5.1}$$

with a switch of values after the 12th element (cleavage position) in each row.

The signal differences between two consecutive time points were calculated. To avoid numerical issues, the data was devided by 1e10:

$$S' = \begin{pmatrix} 0 & 100.00 & 50.00 & 25.00 & 12.50 \\ 0 & 90.90 & 45.10 & 23.00 & 11.00 \\ 0 & 10.00 & 5.00 & 2.50 & 1.25 \end{pmatrix}. \tag{5.2}$$

5.1.2.2 Identifiability analysis

In QPuB, first the parameter identifiability is investigated using the curvature matrix. This gives an indication whether the data is able to sufficiently constrain the conversion factor values for a unique inference. The context of this will become apparent in Example 2. In this example, the matrix is

$$\mathcal{K}(S') = \begin{pmatrix} 1593.75 & 0 \\ 0 & 224984.38 \end{pmatrix} \tag{5.3}$$

and has the following eigenvectors and eigenvalues

$$\lambda_1 = 224984.375 \qquad \lambda_2 = 1593.75 \tag{5.4}$$

$$w_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad w_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \tag{5.5}$$

which results in a partitioning of information with the majority contained in the first component:

$$\mathcal{I} = \begin{pmatrix} 0.993 \\ 0.007 \end{pmatrix}. \tag{5.6}$$

5.1.2.3 Inference results

The goal is to obtain estimates for the conversion factors v_1 of product P_1 and v_2 of product P_2 respectively, as well as the additional nuisance parameter σ of the likelihood function. We initialise three parallel chains, starting from the dispersed initial values. The chains explore the prior range over the course of 1 million iterations and converge to the posterior distribution. The traceplots of the chains are shown in Fig. 5.2C. The Gelman–Rubin scale reduction factors for the conversion factors are close enough to 1 to diagnose **convergence**: $\hat{R}_1 \approx 0.99$ and $\hat{R}_2 \approx 0.99$. The estimated values result in perfect **mass balance**, as shown in the residual plots in Fig. 5.2A and the numerical mean residuals (chain 1: 2.36×10^{-14}, chain 2: 2.50×10^{-14}, chain 3: 2.36×10^{-14}). The resulting **posterior distributions** in the form of densities of the conversion factors and the nuisance parameter are depicted in Fig. 5.2B. A summary statistics is provided in Tab. 5.2.

Table 5.2: Statistics of the posterior distributions in Ex. 0. The burn-in of the chains was discarded and the three chains were combined for the summary statistic. For the conversion factors, their true value (denoted by an asterisk) was subtracted from the mean to visualise the accuracy. Values are rounded to two digits precision for printing.

Parameter	mean	sd
$v_1 - v_1^*$	1.47×10^{-14}	2.35×10^{-11}
$v_2 - v_2^*$	0.00	4.93×10^{-15}
σ	2.16×10^{-13}	1.53×10^{-14}

Comparing the inferred **conversion factors** to the true values, we see that for peptide 2 the true value coincides perfectly with the median of the distribution. For peptide 1, the true value lies at the lower end of the posterior distribution, which may look not very accurate at first sight, however based on the high precision we judge this estimate to be very successful. The deviation from the true value is reported as percent error as defined in Eqn. (3.33c): $\delta_1 = 1.33 \times 10^{-15}\%$ and $\delta_2 = 0\%$. For larger number of peptides, it is convenient to report the deviation as the mean over the individual deviations as defined in Eqn. (4.24), which in this case is $\delta_v = 6.06 \times 10^{-14}\%$. Subsequently, the **peptide amounts are calculated** from the conversion factor distributions using the Eqn. (4.22). The resulting amounts over time are depicted in Fig. 5.2D. Due to the highly precise estimates, the standard deviations of the **concentrations** are very small and invisible in the plot (see Fig. 5.2D).

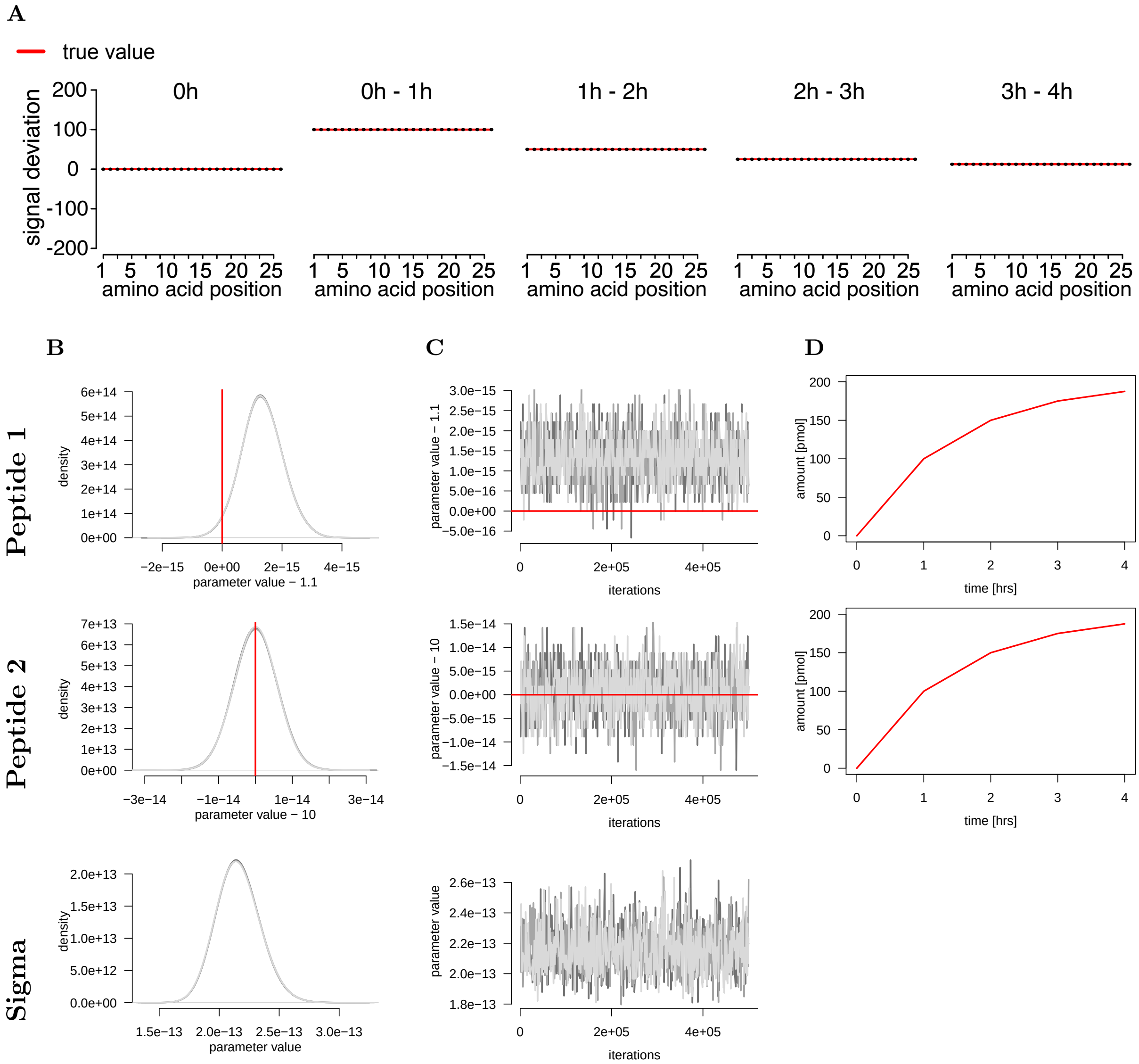


Fig. 5.2: Inference results of Ex. 0. Caption on next page.

5.1.3 Summary

As a proof of concept and to introduce the working principle of QPuB an example with the simplest cleavage pattern with only two peptide products was analysed. QPuB is able to infer the correct values with high accuracy and precision. This suggests, that the framework is a promising approach in the endeavour of conversion factor estimation. Other datasets of the same pattern yielded an estimate of sigma as low as e-98 if its lower prior bound was adjusted accordingly, resulting in a conversion factor estimate with no variation in the chain at all (standard deviation 0).

Although this example would be easily solvable by hand, for problems with higher dimensions this will quickly become cumbersome or impossible.

Fig. 5.2: Inference results of Ex. 0. Figure on page 59. **(A) Residual plots.** The residual plots show whether the estimate satisfies the mass balance condition. The red solid horizontal lines represent the difference in ion peak areas of substrate degraded between two successive time points. The dots represent the median of inferred amount of every amino acid position of all products together as inferred by QPuB. The range between the 5% and 95% quantile are indicated by vertical lines, which, because the conversion factor distributions (see B) are very narrow, are not visible here. With the correctly inferred conversion factor values the dotted black line coincides with the continuous red line. Only the residual plots of one chain is shown, the others are qualitatively similar. **(B) Marginal posterior conversion factor distributions.** Depicted are the posterior distributions of the estimated conversion factors of every peptide product and the nuisance parameter in the form of densities derived from histograms of the frequencies of the parameter values over iteration time. The first 50% samples were excluded as burn-in. Three densities are shown for the three parallel Markov chains run by the algorithm in shades of grey, but hardly distinguishable in the plot. To visualise the small spread of the distributions, the correct conversion factor value was subtracted on the x-axis for the conversion factors, not sigma. The correct conversion factor value is marked in red, which is known by construction of the *in silico* dataset. **(C) Traceplots of the Markov chains.** Shown are the parameter values over iteration time. For the conversion factors, the true value was subtracted again. The three parallel Markov chains are plotted in shades of grey. The chains were thinned by a value of 1000 and 50% burn-in samples were excluded. The true value is marked in red. **(D) Distributions of the estimated peptide amounts.** Depicted are the amounts of all peptide products over time, as derived from the distributions of the estimated conversion factors in (B) using Eqn. (4.8). The three chains without burn-in were combined into one for the calculation. The solid line indicates the median of the combined distribution and the shaded area (invisible) illustrates the confidence range. The lower bound is calculated using the 5% quantile of the parameter distribution, and the upper bound using the 95% quantile respectively. Because the conversion factor distributions are very narrow, the confidence ranges are not visible. The peptide product amounts obtained by QPuB perfectly coincide with the true values (red).

5.2 Example 1

This example is an extension of the previous example, with the same simple substrate following a cleavage pattern that results in six peptide products. We will show, in less detail, how QPuB successfully infers the correct conversion factors for all peptide products with high precision and accuracy. For comparison, in the similar Example 2 of the next section this will not be the case.

5.2.1 The data

The substrate considered in this example is the same substrate as in the example before. This time, we assume, that in addition to the production of peptides 1 and 2, the substrate can also be cleaved at a second position, releasing two more products, of which one can itself act as a substrate to the protease, setting free another two products. In total, this digestion pattern leads to six different peptide

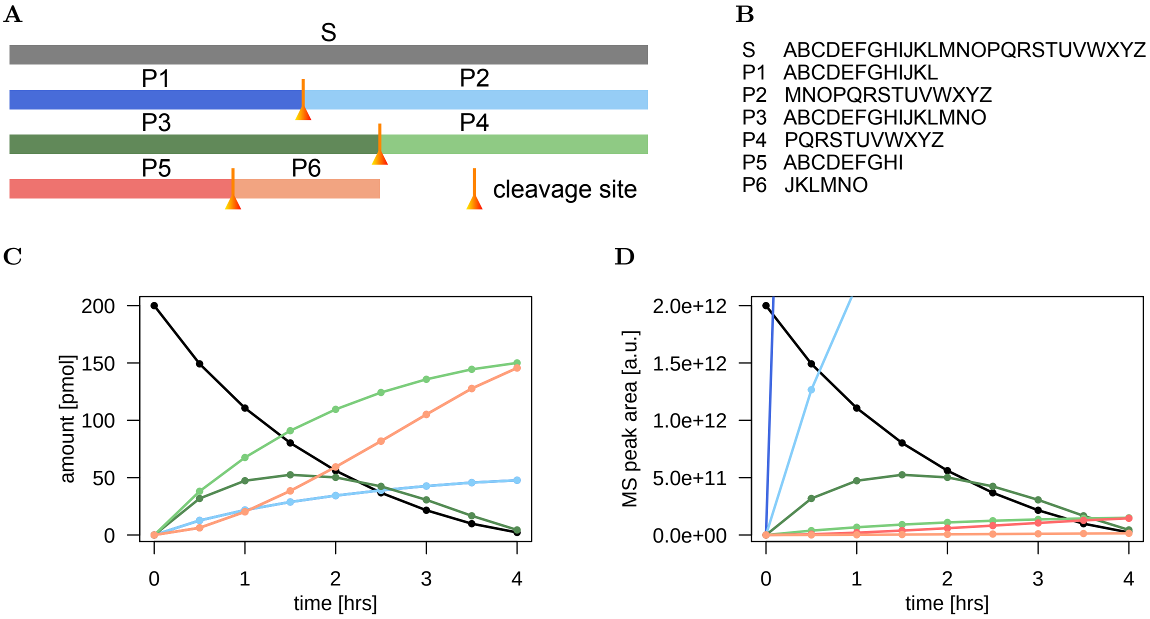


Fig. 5.3: Schematic of the cleavage pattern of Ex. 1 and the resulting kinetics. (A) Digestion pattern. The schematic depicts the digest of a 26-mer substrate by a specific endopeptidase into six peptide products (numbered accordingly). The enzyme hydrolyses the peptide bonds at positions 12 and 15, creating products 1 and 2 or 3 and 4, respectively. Peptide 3 serves as a substrate to the enzyme itself, with a cleavage site at position 9, creating products 5 and 6. All other products will not be further digested. **(B) Amino acid sequences.** Sequences of the substrate and the generated products. **(C) Concentration kinetics.** Depicted are the amounts of the substrate and all peptide products over the digestion time, as simulated. Colors are as in (A). The amounts of the darkblue and the lightblue peptides P_1 and P_2 are equal by construction. **(D) Kinetics of MS signal intensities.** Depicted are the MS signal intensities of the substrate and all peptide products over the digestion time, as computed from the amounts in (B) using the conversion factors in Tab. 5.3B. This serves as QPuB input. For numerical values see Table 5.3. Note that the most abundant lightgreen peptide P_4 in (A) does not necessarily have the highest signal intensity.

products (see Fig. 5.3A). We will refer to products directly derived from the substrate as first-generation products (P_1, P_2, P_3, P_4) and to products derived by further digestion of one of the products by second-generation products (P_5, P_6). The concentration kinetics for this cleavage pattern was simulated using the julia scripts described in Section 4.8.1. The initial conditions of the reagents were set as follows: $[S](0) = 200\,\text{pmol}$, $[E](0) = 0.05\,\text{pmol}$, $[ES](0) = [P_d](0) = 0\,\text{pmol}$ for all $d = 1, \ldots, 6$. The reaction rates were randomly sampled and a simulated reaction kinetics was selected (Tab. 5.3A and Fig. 5.3C). Since we are considering noise-free data, only one replicate is produced. Conversion factors were chosen with different orders of magnitude as in Table 5.3B. Using the concentration kinetics and the conversion factor values, signal kinetics were obtained (Tab. 5.3C and Fig. 5.3D). These signal intensities were used as input to QPuB.

Table 5.3: Numerical data used in Ex. 1. (A) Concentration kinetics. Amounts in [pmol] for the substrate and all peptide products over a digestion time of 4 hours. **(B) Conversion factors.** Conversion factors of all peptide products, systematically chosen. The conversion factor of the substrate is set to 1 by default. **(C) Kinetics of MS signal intensities.** MS signal intensities in [a.u.] for the substrate and all peptide products over a digestion time of 4 hours, as calculated from (A) and (B) using Eqn. (4.8). Shown are only the hourly timepoints, in the inference we also used the half-hourly measurements. Values in (A) and (C) are rounded to two digits precision for printing.

(A) Concentrations [pmol]

Reactant	0h	1h	2h	3h	4h
S	200.00	110.67	56.06	21.56	2.23
P_1	0.00	21.77	34.38	42.69	47.72
P_2	0.00	21.77	34.38	42.69	47.72
P_3	0.00	47.33	50.22	30.65	4.40
P_4	0.00	67.57	109.56	135.75	150.05
P_5	0.00	20.24	59.33	105.11	145.65
P_6	0.00	20.24	59.33	105.11	145.65

(B) Conversion factors

Param	Value
v_0	1
v_1	0.01
v_2	0.1
v_3	1
v_4	10
v_5	10.01
v_6	100

(C) MS signal intensities [a.u.]

Reactant	0h	1h	2h	3h	4h
S	2.00×10^{12}	1.11×10^{12}	5.61×10^{11}	2.16×10^{11}	2.23×10^{10}
P_1	0.00	2.18×10^{13}	3.44×10^{13}	4.27×10^{13}	4.77×10^{13}
P_2	0.00	2.18×10^{12}	3.44×10^{12}	4.27×10^{12}	4.77×10^{12}
P_3	0.00	4.73×10^{11}	5.02×10^{11}	3.06×10^{11}	4.40×10^{10}
P_4	0.00	6.76×10^{10}	1.10×10^{11}	1.36×10^{11}	1.50×10^{11}
P_5	0.00	2.02×10^{10}	5.93×10^{10}	1.05×10^{11}	1.46×10^{11}
P_6	0.00	2.02×10^{9}	5.93×10^{9}	1.05×10^{10}	1.46×10^{10}

Table 5.4: Statistics of Ex. 1. Inferences were run for 2e6 iterations, burn-in was discarded and the three chains were combined into one. To visualise the accuracy, the true conversion factor value was subtracted for all conversion factors. Values were rounded to two digits precision. **(A) Posterior statistics of one run.** Statistics of the posterior distributions of all parameters in Ex. 2. **(B) Repeatability.** The inference on the dataset was repeated 100 times. In none of the runs the chains were converged to a stationary distribution. This table reports the mean of the posterior means across runs and the corresponding standard deviations.

(A) Single run

Parameter	mean	sd
$v_1 - v_1^*$	-1.49×10^{-15}	6.21×10^{-13}
$v_2 - v_2^*$	-1.30×10^{-14}	6.65×10^{-13}
$v_3 - v_3^*$	3.00×10^{-14}	4.94×10^{-14}
$v_4 - v_4^*$	2.01×10^{-13}	4.37×10^{-13}
$v_5 - v_5^*$	3.00×10^{-13}	6.14×10^{-10}
$v_6 - v_6^*$	2.00×10^{-12}	4.24×10^{-12}
σ	1.44×10^{-13}	6.97×10^{-15}

(B) Repeatability

Parameter	mean	sd
$v_1 - v_1^*$	-1.47×10^{-15}	1.32×10^{-16}
$v_2 - v_2^*$	-1.29×10^{-14}	1.15×10^{-15}
$v_3 - v_3^*$	2.86×10^{-14}	4.95×10^{-15}
$v_4 - v_4^*$	2.04×10^{-13}	4.18×10^{-14}
$v_5 - v_5^*$	2.93×10^{-13}	4.52×10^{-14}
$v_6 - v_6^*$	1.98×10^{-12}	4.01×10^{-13}
σ	1.47×10^{-13}	1.41×10^{-14}

5.2.2 Results

QPuB was applied to this dataset and achieved to estimate the conversion factors of all six peptide products with high precision and accuracy. The underlying Markov chains converged over iteration time. The mass balance requirement is satisfied to a high degree. The marginal posterior distributions of the estimated conversion factors have a unique maximum and a very narrow standard deviation. Comparison of the medians to the correct values used to create the *in silico* data shows high agreement. The kinetics of the peptide amounts derived from the conversion factors estimated are in accordance with the underlying ground truth. Simulation results were highly repeatable.

Simulations were run with 2 million iterations. The Markov chains explored the prior range and settled in a region of highest probability, with good mixing. All three parallel chains converged to roughly the same values (Fig.5.4C). Also the Gelman–Rubin diagnostics suggests **convergence**, since the scale reduction factor of every parameter is close to 1 ($\underset{d=1,\dots,D}{\text{mean}}(\hat{R}_d) \approx 1.004$). In an example with full mass information and without noise in the data, the **mass balance** requirement can be perfectly satisfied. The residual plots are depicted in Figure 5.4A. For every amino acid position, the total mass over all peptides containing this amino acid position is calculated from the conversion factor distributions. The medians are in perfect agreement with the amount of substrate degraded for every point in digestion time. The spread of each distribution is so small that they are invisible in the plots. The means of the residuals per chain over all amino acid positions and all time points are small (chain 1: -5.52×10^{-15}, chain 2: -7.42×10^{-15}, chain 3: -6.92×10^{-15}). Figure 5.4B shows the **posterior distributions of the conversion factors** for every chain with good agreement between the chains. The distributions have a unique maximum and a very narrow standard deviation, showing high precision of the estimate. See Tab. 5.4A for numerical values. The results were validated by comparing the estimates of the conversion factors to the correct conversion factors used to create the dataset. The **estimation accuracy of the conversion factors** in this example is very high. In Figures 5.4B-C the true values are indicated by red lines. Unfortunately, the true values are outside the posterior ranges, but the deviation is so small that we judge the estimates to be highly satisfactory. For comparability, the deviation between the estimation median and the true value is given as the relative error in percent (see Sec.3.4.2). The largest percent error has peptide 1 with $1.36 \times 10^{-11}\%$, all other deviations are below that. The mean deviation over all peptides is $\delta_v = 7.36 \times 10^{-12}\%$. Finally, the **peptide product amounts** are calculated from the estimated conversion factors (Fig. 5.4D). Since the standard deviation of the conversion factor posteriors are so small, the uncertainty in the calculated amounts is invisible in the plots. We end up with distributions of peptide amounts over time which highly coincide with the true curves. The relative error in the parameter estimates transfers to the error in amount estimates. The mean deviation of the estimate to the true value over all time points is $\delta_c = 7.59 \times 10^{-12}\%$. The inference on the same dataset with the same settings was **repeated** 100 times. The mean and standard deviation of the posterior means of every conversion factor across runs are shown in Tab. 5.4B.

Fig. 5.4: Inference results of Ex. 1. Figure on page 65. **(A) Residual plots.** The residual plots show whether the estimate satisfies the mass balance condition. The red solid horizontal lines represent the difference in ion peak areas of substrate degraded between two successive time points. The dots represent the median of inferred amount of every amino acid position of all products together as inferred by QPuB. The range between the 5% and 95% quantile are indicated by vertical lines, which, because the conversion factor distributions (see B) are very narrow, are not visible. With the correctly inferred conversion factor values the dotted black line coincides with the continuous red line. Only the residual plots of one chain for every second time point is shown, the others are qualitatively similar. **(B) Marginal posterior conversion factor distributions.** Depicted are the posterior distributions of the estimated conversion factors of every peptide product in the form of densities derived from histograms of the frequencies of the parameter values over iteration time. Three densities are shown for the three parallel Markov chains run by the algorithm. The first 50% samples were excluded as burn-in. To visualise the small spread of the distributions, the correct conversion factor value was substracted on the x-axis. The correct conversion factor value is marked in red, which is known by construction of the *in silico* dataset. **(C) Traceplots of the Markov chains.** Shown are the parameter values over iteration time. The three parallel Markov chains are plotted in shades of grey. The chains were thinned by a value of 1000 and 50% burn-in samples were excluded. The true value is marked in red. **(D) Distributions of the estimated peptide amounts.** Depicted are the amounts of all peptide products over time, as derived from the distributions of the estimated conversion factors in (A) using Eqn. (4.8). The three chains were combined into one for the calculation. The solid line indicates the median of the combined distribution and the shaded area (invisible) illustrates the confidence range. The lower bound is calculated using the 5% quantile of the parameter distribution, and the upper bound using the 95% quantile respectively. Because the conversion factor distributions are very narrow, the confidence ranges are not visible. The peptide product amounts obtained by QPuB perfectly coincide with the true values (red).

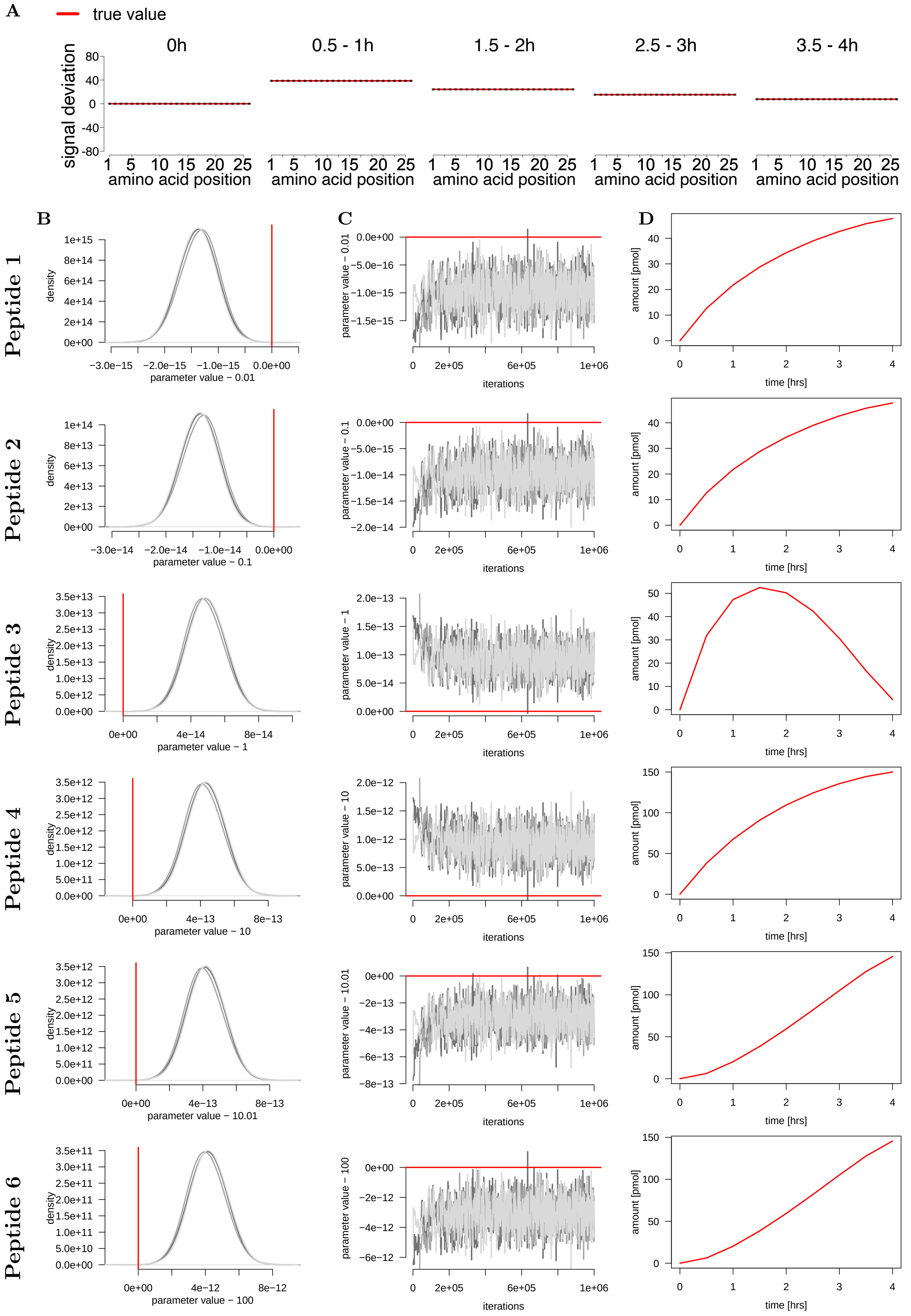


Fig. 5.4: Inference results of Ex. 1. Caption on page 64.

5.2.3 Summary

We presented an example, where QPuB successfully inferred the correct conversion factors and the calculated kinetics of amounts coincided with the true kinetics. Due to the perfect data without noise, inference results are subject to very small uncertainty, exhibited in the form of narrow standard deviations.

Many different cleavage patterns with differing kinetics were investigated. Ranging from very broad cleavage patterns with only first order products, to very deep patterns where all products are further digested, and many variants inbetween including interconnected patterns with peptide splicing. In the majority of datasets with low number of products (<20) were identifiable. Here, the performance of QPuB is highly precise, accurate and repeatable. However, datasets exist where this is not the case, as we will see in the next section.

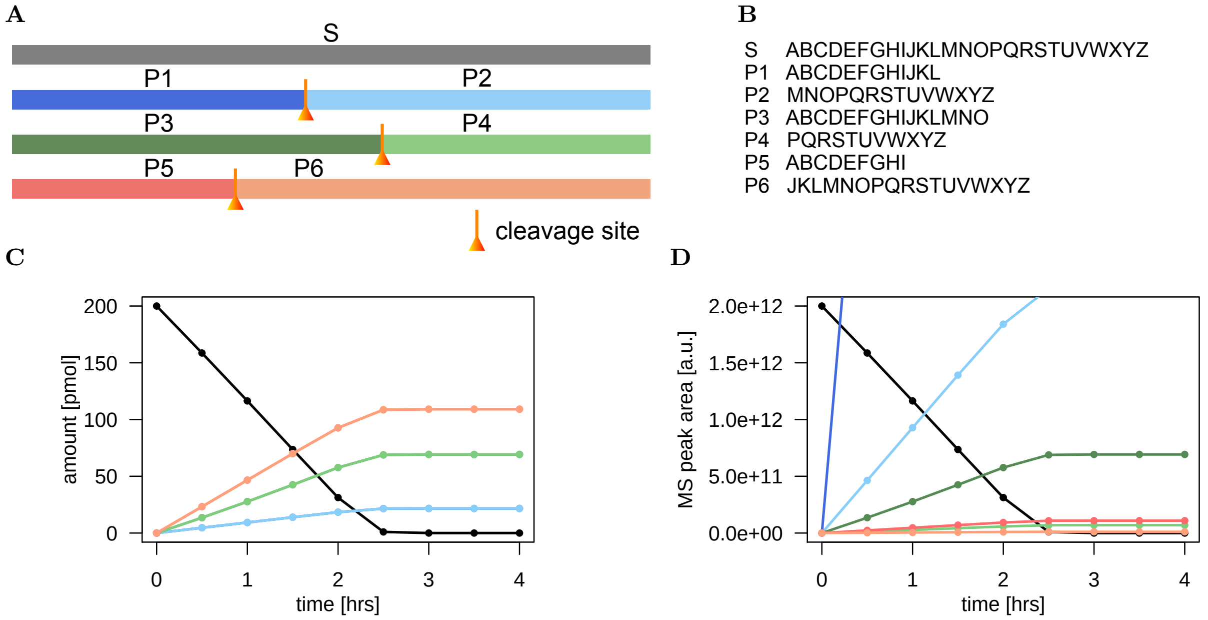


Fig. 5.5: Schematic of the cleavage pattern of Ex. 2 and the resulting kinetics. (A) Digestion pattern The schematic depicts the digest of a 26-mer substrate by a specific endopeptidase into six peptide products (numbered accordingly). The enzyme hydrolyses the peptide bonds at positions 12, 15 and 9, creating product pairs 1 and 2, 3 and 4, and 5 and 6, respectively. **(B) Amino acid sequences** Sequences of the substrate and the generated products. **(C) Concentration kinetics** Depicted are the amounts of the substrate and all peptide products over the digestion time, as simulated. Colors are as in (A). By construction, "sibling" peptides are equally abundant. **(D) Kinetics of MS signal intensities** Depicted are the MS signal intensities of the substrate and all peptide products over the digestion time, as computed from the amounts in (B) using the conversion factors in Tab. 5.5B. This serves as QPuB input. For numerical values see Tab. 5.5.

Table 5.5: Numerical data used in Ex. 2. (A) Concentration kinetics. Amount in [pmol] for the substrate and all peptide products over a digestion time of 4 hours. **(B) Conversion factors.** Conversion factors of all peptide products, systematically chosen. The conversion factor of the substrate is set to 1 by default. **(C) Kinetics of MS signal intensities.** MS signal intensities in [a.u.] for the substrate and all peptide products over a digestion time of 4 hours, as calculated from (A) and (B) using Eqn. (4.8). Shown are only the hourly time points, in the inference we also used the half-hourly measurements. Values in (A) and (C) are rounded to two digits precision for printing. *[Remark: The author realised too late, that the data simulation in this example yielded negative substrate amounts, due to numerical issues of the ODE solver. However, the negative value is so small, that the mass balance is not affected much. In similar datasets with the same cleavage pattern but different numerical values, the results of this section are qualitatively the same.]*

(A) Amounts [pmol]

Reactant	0h	1h	2h	3h	4h
S	200.00	116.42	31.25	0.00	-3.36×10^{-8}
P_1	0.00	9.28	18.40	21.65	21.65
P_2	0.00	9.28	18.40	21.65	21.65
P_3	0.00	27.68	57.67	69.23	69.23
P_4	0.00	27.68	57.67	69.23	69.23
P_5	0.00	46.63	92.67	109.12	109.12
P_6	0.00	46.63	92.67	109.12	109.12

(B) Conversion factors

Parameter	Value
v_0	1
v_1	0.01
v_2	0.1
v_3	1
v_4	10
v_5	10.01
v_6	100

(C) MS signal intensities [a.u.]

Reactant	0h	1h	2h	3h	4h
S	2.00×10^{12}	1.16×10^{12}	3.13×10^{11}	1.35×10^{6}	-335.79
P_1	0.00	9.28×10^{12}	1.84×10^{13}	2.16×10^{13}	2.16×10^{13}
P_2	0.00	9.28×10^{11}	1.84×10^{12}	2.16×10^{12}	2.16×10^{12}
P_3	0.00	2.77×10^{11}	5.77×10^{11}	6.92×10^{11}	6.92×10^{11}
P_4	0.00	2.77×10^{10}	5.77×10^{10}	6.92×10^{10}	6.92×10^{10}
P_5	0.00	4.66×10^{10}	9.26×10^{10}	1.09×10^{11}	1.09×10^{11}
P_6	0.00	4.66×10^{9}	9.27×10^{9}	1.09×10^{10}	1.09×10^{10}

5.3 Example 2

This is an example that shows difficulties in the parameter inference. The same substrate as in Example 1 is digested with a slightly different cleavage pattern, resulting in six slightly different peptide products that follow different kinetics. The information contained in the data is too low for QPuB to identify the parameters reliably. We will show two ways to increase the information content of the system to enable successful inference.

5.3.1 The data

The same substrate as in Example 1 is digested with a slightly different cleavage specificity. The same four first order products are produced as before, but none of the products is further digested. Instead, a third cleavage site in the substrate sequence is introduced, resulting in six first order peptide products in total (see Fig. 5.5A). For comparability, we used the same initial conditions and reaction rates for the three reactions as in Example 1. The obtained concentration kinetics is shown in Fig. 5.5C, with numerical values in Tab. 5.5A. The same conversion factor values were defined as before (Tab. 5.5B). The resulting signal kinetics is provided in Tab. 5.5C and depicted in Fig. 5.5D.

5.3.2 Results: Insufficient information

We show that the convergence of the Markov chains to the correct values fails, although mass balance is achieved. A multitude of possible parameter combinations can explain the data. The information content of the dataset is determined using the identifiability approach from Sec.3.3. We will show, that the data does not contain sufficient information for all parameters to be identifiable. A general rule of thumb for the inferability will be proposed.

5.3.2.1 Inference results

After 2 million iterations, the single Markov chains are **not converged** yet. The Gelman–Rubin diagnostic does not suggest convergence ($\underset{d=1,\dots,D}{\text{mean}}(\hat{R}_d) \approx 3.2$) and visually the chains are still in the sampling process, far from any limiting distribution. The three chains are not in agreement with each other (Fig. 5.6C). However, the **mass balance** condition seems satisfied, as the residuals look perfect (Fig. 5.6A). The means of the residuals per chain are small, but not as small as in Example 1 (chain 1: 1.76×10^{-4}, chain 2: 1.47×10^{-4}, chain 3: 9.27×10^{-5}). This observation suggests the existence of multiple solutions and the inability of QPuB to find the particular one that was used to create the data.

Since the chains are not converged to a normal distribution, it is not very reliable to calculate a summary statistics and to compare the statistics to the correct values. Nonetheless, the statistics are shown in Tab. 5.6A. From the current state of the chains, the **estimation accuracy of the conversion factors** is investigated. Large deviation of the median to the correct values is observed. The deviation in percent is as high as 329.18% for peptide 1 and gives a mean deviation over all peptides of $\delta_v = 134.51\%$. For data without noise, this level of uncertainty is disappointing. The combined chains were subsequently used to calculate the peptide amounts. The **estimated concentration kinetics** show broad ranges of uncertainty (Fig. 5.6D). Due to the linear relationship between conversion factor and concentration, the high error in the conversion factors directly translates to a low **estimation accuracy of the peptide amounts**: $\delta_c = 134.72\%$. The simulation was **repeated** 100 times. Every run stopped after 2e6 iterations returns Markov chains which are currently sampling a different part of the prior space (results not shown). In all cases, the posterior distributions have not reached a limiting distribution. In most cases, the chains are sampling in the correct order of magnitude. In some cases, the chains sample in unison. In total, the mean across runs deviates from the true values with a standard deviation indicating high uncertainty (Tab. 5.6B).

Table 5.6: Statistics of Ex. 2 with insufficient information. Inferences were run for 2e6 iterations, burn-in was discarded and the three chains were combined into one. Values were rounded to three digits precision. **(A) Posterior statistics of one run.** Statistics of the posterior distributions of all parameters in Ex. 2. **(B) Repeatability.** The inference on the dataset was repeated 100 times. In none of the runs the chains were converged to a stationary distribution. This table reports the mean of the posterior means across runs and the corresponding standard deviations.

(A) Single run

Parameter	mean	sd
v_1	0.043	0.004
v_2	0.430	0.042
v_3	1.055	0.007
v_4	10.555	0.072
v_5	3.115	0.887
v_6	31.114	8.859
σ	0.005	0.001

(B) Repeatability

Parameter	mean	sd
v_1	0.031	0.014
v_2	0.313	0.142
v_3	1.036	0.024
v_4	10.358	0.239
v_5	5.553	2.972
v_6	55.476	29.687
σ	0.004	0.002

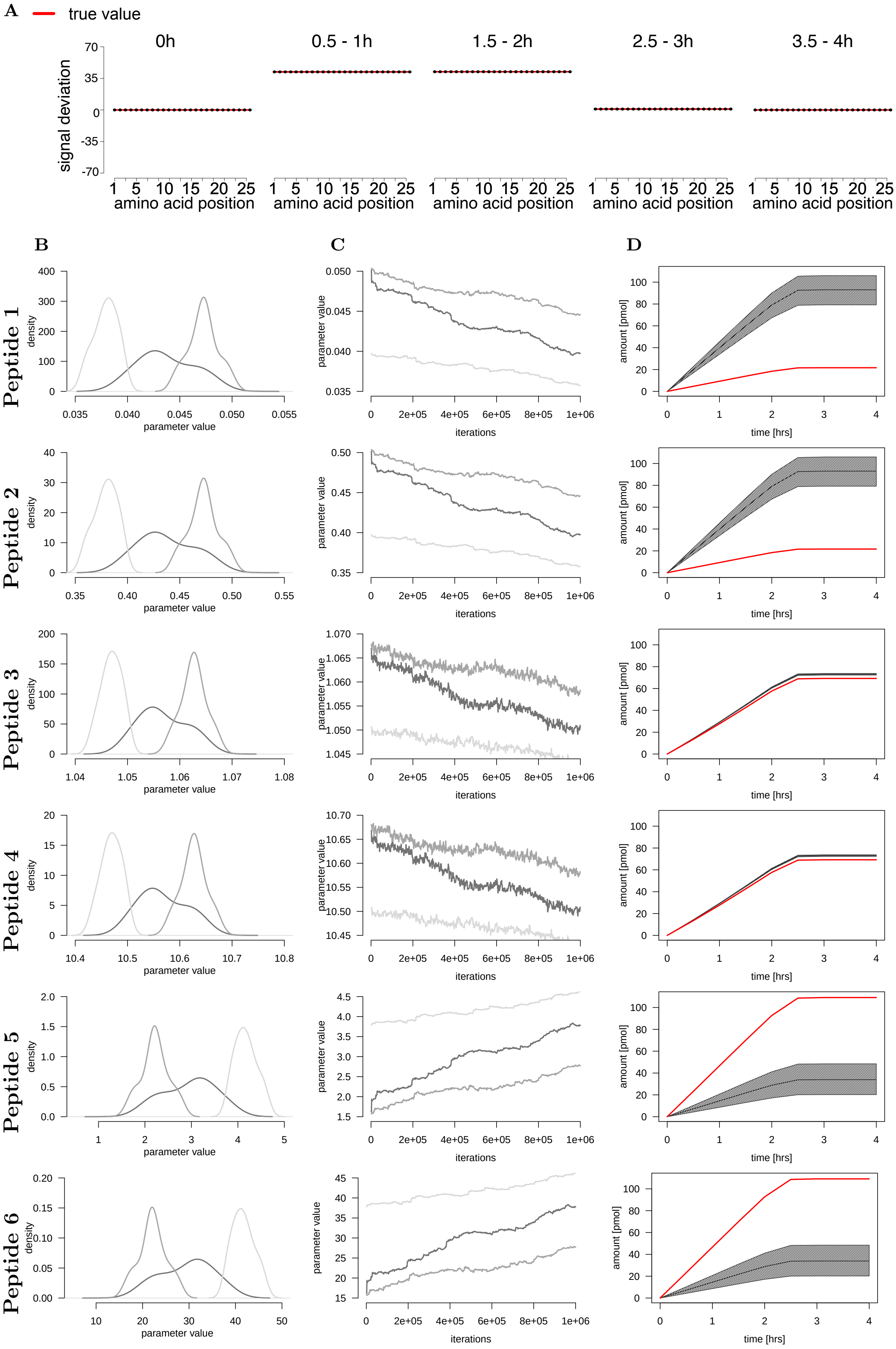

A
true value
0h
0.5 - 1h
1.5 - 2h
2.5 - 3h
3.5 - 4h
signal deviation
70
35
0
-35
-70
1 5 10 15 20 25
amino acid position
B
C
D
Peptide 1
Peptide 2
Peptide 3
Peptide 4
Peptide 5
Peptide 6
density
parameter value
iterations
amount [pmol]
time [hrs]
0.035 0.040 0.045 0.050 0.055
0.35 0.40 0.45 0.50 0.55
1.04 1.05 1.06 1.07 1.08
10.4 10.5 10.6 10.7 10.8
0 2e+05 4e+05 6e+05 8e+05 1e+06
0 1 2 3 4

Fig. 5.6: Inference results of Ex. 2 with insufficient information. Figure on page 70. **(A) Residual plots.** The residual plots show whether the estimate satisfies the mass balance condition. The red solid horizontal lines represent the difference in ion peak areas of substrate degraded between two successive time points. The dots represent the median of inferred amount of every amino acid position of all products together as inferred by QPuB. The range between the 5% and 95% quantile are indicated by vertical lines, which, because the conversion factor distributions (see (B)) are very narrow, are not visible. With inferred conversion factor values which satisfy mass balance, the dotted black line coincides with the continuous red line. Only the residual plots of one chain for every second time point is shown, the others are qualitatively similar. **(B) Marginal posterior conversion factor distributions.** Depicted are the posterior distributions of the estimated conversion factors of every peptide product in the form of densities derived from histograms of the frequencies of the parameter values over iteration time. Three densities are shown for the three parallel Markov chains run by the algorithm. The first 50% samples were excluded as burn-in. A unique normal stationary distribution has not been reached in the iteration time run. The correct conversion factor values are not in range of the plots. **(C) Traceplots of the Markov chains.** Shown are the parameter values over iteration time. The three parallel Markov chains are plotted in shades of grey. The chains were thinned by a value of 1000 and 50% burn-in samples were excluded. Stationarity of the chains has not been reached. The true value of each parameter is not in the y-range plotted. **(D) Distributions of the estimated peptide amounts.** Depicted are the amounts of all peptide products over time, as derived from the distributions of the estimated conversion factors in (A) using Eqn. (4.8). The three chains were combined into one for the calculation. The solid line indicates the median of the distribution and the shaded area illustrates the confidence range. The lower bound is calculated using the 5% quantile of the parameter distribution, and the upper bound using the 95% quantile respectively. The peptide product amounts obtained by QPuB do not cover the true kinetics (red).

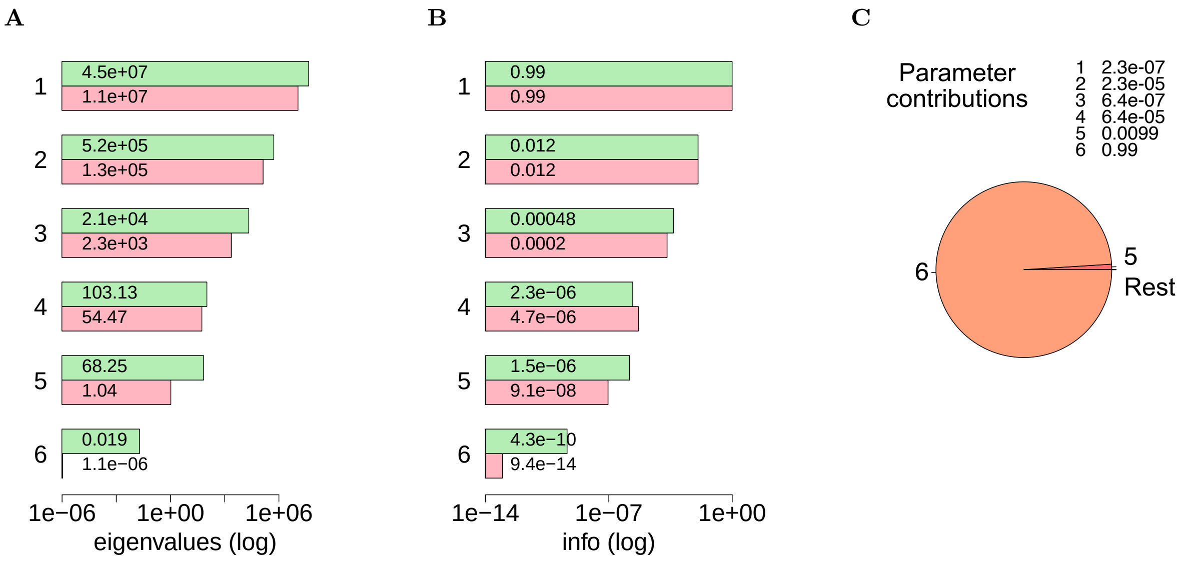


Fig. 5.7: Identifiability analysis of Ex.2. (A) Eigenvalues. The plot visualises the six eigenvalues of the curvature matrix of the likelihood for the dataset of Ex. 2 (red) compared to the eigenvalues of Ex. 1 (green). Note the difference in the last eigenvalue 6. **(B) Information content.** Depicted are the normalised eigenvalues, i.e. the proportion of variance explained by the corresponding eigenvector. Note the difference in the last component. **(C) Parameter contribution to the subspace of low-informative eigenvectors.** In this example, the subspace is spanned by only one eigenvector. The largest contribution has parameter 6 belonging to peptide 6. Values were rounded to two significant digits for printing.

5.3.2.2 Identifiability analysis

The identifiability analysis gives insight to the issue at hand. The curvature matrix of the likelihood function is calculated together with its eigenvalues and eigenvectors. A normalisation gives the information content of the different principal components. Figure 5.7 shows the eigenvalues and information contents of Ex. 1 and Ex. 2. A comparison reveals a particular difference: the last eigenvector in Ex. 1 carries more information than in Ex. 2:

Example	lowest info
1	4.3×10^{-10}
2	9.4×10^{-14}

The results indicate that somewhere between these two values lies a threshold under which the inference will not succeed anymore. The corresponding eigenvalue is too close to zero, implying too large variance in the data to sufficiently constrain the parameters. After systematic investigation of a multitude of datasets, we propose the following rule of thumb:

Heuristic principle 5.1 (Information threshold for successful inference)**.** QPuB struggles to converge to the correct conversion factor values in reasonable iteration time, if the curvature matrix reveals eigenvectors carrying an information content of the order of e-12 or lower.

The information contained in this dataset does not suffice for the inference of the correct values. Due to correlations, a multitude of possible solutions satisfying mass balance exists. Investigation of the curvature matrix reveals that there exists a threshold of information under which the parameter contributing most to the space of low-informative directions is not sufficiently constrained. The heuristic principle 5.1 is proposed, stating that the threshold for most datasets seems to be of the order of e-12. In the following, we will present two strategies to increase the information content of the system sufficiently to make a successful inference possible.

5.3.3 Strategy 1: Additional information through peptide titrations

In this section, a procedure is developed to provide more information to the system in the form of additional peptide titrations. Recall that peptide ion signal intensities can be measured for synthetic peptide equivalents, as described in Sec. 2.3.3.1. From the calibration curve, the conversion factor can be calculated. This experimentally derived conversion factor can then be supplied to the QPuB inference. This reduces the number of parameters to be estimated and adds constraints to the remaining peptides.

We will test this concept in the *in silico* framework by fixing the conversion factor of a peptide to its correct value, used to generate the data.

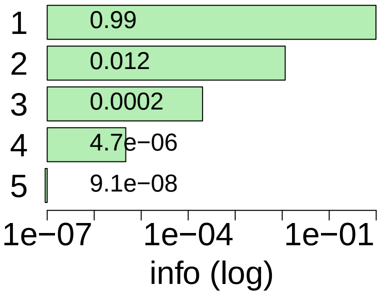
5.3.3.1 The additional data

Figure 5.7 shows that parameter 6 contributes most to the low-informative subspace of the curvature matrix and therefore makes a good candidate for this experiment. Its value is fixed to its correct value $v_6 = 100$ and the parameter vector to be estimated reduces its dimension by one: $\theta = (v_1, v_2, v_3, v_4, v_5, \sigma)$.

5.3.3.2 Identifiability analysis

In the insufficiently constrained case, an infinite combination of parameter values yields the same likelihood value. By fixing the value of one of the parameters, this uncertainty is taken away. The least-informative eigenvector is removed from the system, and the now least-informative eigenvector carries sufficient information to pass the rule of thumb. The eigenvalues are large enough to be distinguishable from zero, resulting in all remaining parameters to be identifiable. Figure 5.8 shows the normalised eigenvalues after setting parameter 6 to its correct value. According to the heuristic 5.1, the now lowest information is sufficiently large to enable a successful inference.

Fig. 5.8: Identifiability analysis of Ex. 2 after peptide titration. Depicted are the normalised eigenvalues mirroring the information content when fixing parameter 6 to its true value $v_6 = 100$. Dropping one dimension, a little shift in the remaining values is observed. The last normalised eigenvalue is now above the threshold e-12 of the proposed heuristic.



5.3.3.3 Inference results

Indeed, the system becomes identifiable and the chains converge to a stationary distribution close to the correct values. After running the fixed number of 2e6 iterations, **convergence** to normal limit distributions for the conversion factors is achieved. The three parallel chains converge to the same values (Fig. 5.9C). The Gelman–Rubin criterion suggests convergence for all parameters ($\underset{d=1,\ldots,D}{\text{mean}}(\hat{R}_d) \approx 1.00071$). The residual plots indicate **mass balance** for all time points (Fig. 5.9A) and the numerical mean residuals are smaller than before (chain 1: -1.23×10^{-14}, chain 2: -1.16×10^{-14}, chain 3: -1.47×10^{-14}). The **posterior distributions** of the remaining parameters converge to nice normal distributions (Fig. 5.9B)

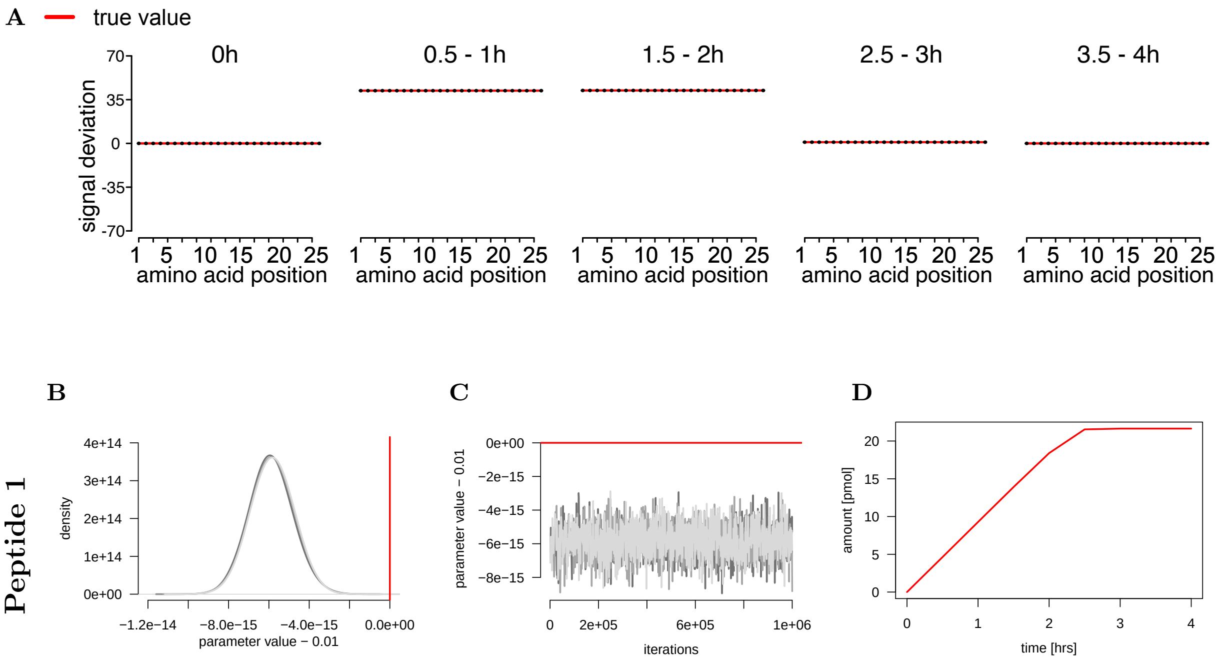


Fig. 5.9: Inference results of Ex. 2 when fixing one conversion factor to its true value. Only the inference results of parameter 1 are shown. The outcome for the others is similar. **(A) Residual plots (B) Posterior distributions of the conversion factors** and **(C) Trace plots of the Markov chains** After fixing parameter 6 to its true value, the chains for the remaining parameters converge and agree with their true values to high accuracy and precision. **(D) Estimated amount kinetics** Since the spread of the posterior distribution is so small, the uncertainty in the peptide amounts over time is invisible in the graphics.

Table 5.7: Statistics of Ex. 2 with a peptide titration. The sixth conversion factor was set to its true value $v_6 = 100$. Inferences were run for 2e6 iterations, burn-in was discarded and the three chains were combined into one. For the conversion factors, the true value was subtracted to visualise the accuracy. Values were rounded to two digits precision. **(A) Posterior statistics of one run.** Statistics of the posterior distributions of all remaining parameters in Ex. 2. **(B) Repeatability.** The inference on the dataset was repeated 100 times. This table reports the mean of the posterior means across runs and the corresponding standard deviations.

(A) Single run

Parameter	mean	sd
$v_1 - v_1^*$	-5.85×10^{-15}	6.17×10^{-13}
$v_2 - v_2^*$	-5.87×10^{-14}	7.10×10^{-13}
$v_3 - v_3^*$	2.10×10^{-13}	1.98×10^{-13}
$v_4 - v_4^*$	1.90×10^{-12}	1.98×10^{-12}
$v_5 - v_5^*$	2.01×10^{-13}	6.15×10^{-10}
σ	2.88×10^{-13}	1.53×10^{-14}

(B) Repeatability

Parameter	mean	sd
$v_1 - v_1^*$	-5.87×10^{-15}	8.42×10^{-18}
$v_2 - v_2^*$	-5.88×10^{-14}	1.53×10^{-16}
$v_3 - v_3^*$	2.10×10^{-13}	2.01×10^{-15}
$v_4 - v_4^*$	1.90×10^{-12}	8.93×10^{-15}
$v_5 - v_5^*$	2.01×10^{-13}	1.25×10^{-14}
σ	2.87×10^{-13}	6.64×10^{-17}

with small standard deviations (Tab. 5.7A). Like in Example 1, the curves do not center at the true values. Nevertheless, we say that the values are inferred accurately. All of the deviations lie below $5.88 \times 10^{-11}\%$ and the mean over all remaining peptides (excluding peptide 6) is $\delta_v = 3.15 \times 10^{-11}$. Again, the **high accuracy and high precision** transfer directly to small uncertainty in the **estimated peptide concentrations** (Fig. 5.9D) and the mean percent error is small: $\delta_c = 3.16 \times 10^{-11}$. Fixing one of the parameters results into reliable inference with every simulation. The mean and standard deviation of the means of all conversion factor estimates over 100 **repetitions** is shown in Tab.5.7B.

5.3.3.4 Generalisation

Providing the correct conversion factor of one peptide seems to help in the inference of the remaining conversion factors. Above, we selected the peptide with the largest contribution to the low-informative subspace, which in that case corresponds to the single least-informative eigendirection of the curvature matrix. Does it make a difference which peptide we choose for titration? We systematically investigated the inference results by fixing each conversion factor one by one to their correct values. Figure 5.10 shows the change in the lowest information content, as well as the resulting mean deviation in percent, the mean posterior standard deviations and the mean residuals. Fixing the value of peptides P_1, P_2, P_5 and P_6 yields a successful inference. The biggest improvement in the information content is visible for the titration of peptide P_1, however the inference results are qualitatively very similar. A titration of peptides P_3 or P_4 does not increase the information above the heuristic threshold, which is also mirrored in the simulations. The chains have not converged after 2e6 iterations and the deviation of the means from the true values is large in comparison to the other scenarios.

After confirming our observations on more datasets, we conclude, that setting the conversion factors of certain peptides to fixed (correct) values facilitates the inference, whereas fixing of certain other conversion factors might not have the desired effect in a reasonable iteration time. We propose a second rule of thumb:

Heuristic principle 5.2 (Titration candidate)**.** To increase the information content sufficiently for successful inference, the conversion factor of the peptide contributing most to the subspace of low-informative eigenvectors should be fixed.

However, we observed that the fixation of the recommended parameter does not always yield the largest increase in information. Nevertheless we found this rule of thumb reliable and useful in practice.

For this simple dataset, we only have one low-informative eigendirection, so a single additional peptide titration suffices to increase the amount of information to a reasonable amount that enables identifiability. A detailed analysis on a multitude of datasets revealed that for larger and more complicated datasets

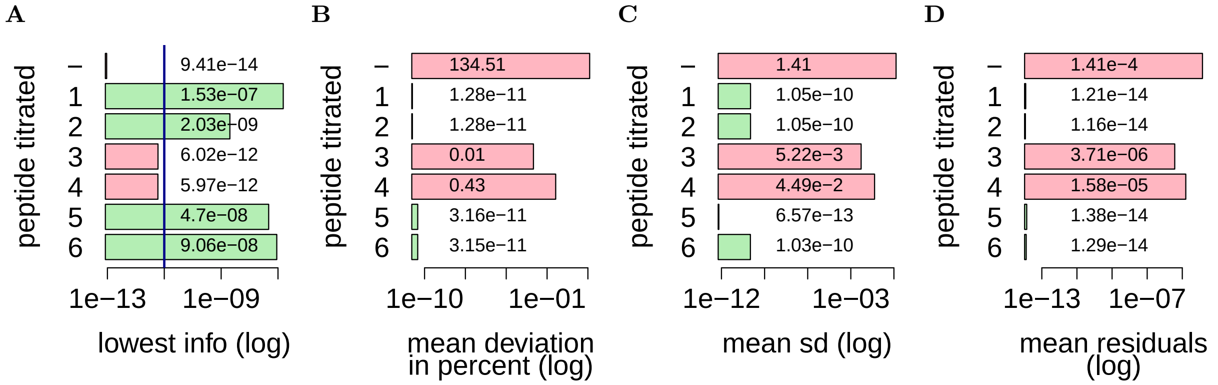


Fig. 5.10: Comparison of different titrations in Ex. 2. Identifiability analysis and inferences were performed for the dataset when fixing the values of the respective conversion factors to their true values. Simulations were run for 2e6 iterations. **(A) Lowest information.** The plot shows the information carried by the least-informative directions. The information is lowest for the original dataset discussed before, shown in the top bar. Titrations of peptides P_1, P_2, P_5, P_6 all increase the information to a level above the heuristic threshold (darkblue), peptides P_3, P_4 do not. **(B) Inference accuracy.** This plot depicts the means over the deviations of the parameter posterior medians from their true values in percent. **(C) Inference precision.** The means over the posterior standard deviations of all peptides are shown. **(D) Mass balance.** The residual means over all amino acids positions, time points and chains is summarised.

multiple low-informative eigendirections can arise and require multiple peptide titrations. We propose a summarising and generalising rule of thumb for multiple peptide titrations, that has been successfully applied to a multitude of datasets:

Heuristic principle 5.3 (Generalisation)**.** An information content of the order of e-12 is considered too low for identifiability. The number of low-informative directions indicates how many conversion factors should be provided. The parameters contributing most to the eigenspace of low-informative eigendirections are the recommended choice for additional peptide titrations.

5.3.3.5 Noisy peptide titrations

In the above analysis, we considered the ability to fix the parameter to the correct value. In reality, if the conversion factor is measured via titration, the value will be subject to measurement errors. In this paragraph, we will investigate the impact of titration measurement errors on the inference of the remaining conversion factors. Note that the signal intensities are still noise-free.

The heuristic 5.3 above was followed for choosing which parameter to fix. Instead of providing the true conversion factor value, it was perturbed by using a normally distributed titration error:

$$v_{\text{tit}} = v_{\text{true}} + \epsilon_{\text{tit}}, \qquad \text{where } \epsilon_{\text{tit}} \sim \mathcal{N}(0, \sigma_{\text{tit}}) \tag{5.7}$$

The standard deviations σ_{tit} were chosen systematically between 1e-7 and 20.

In the examples considered, we observed that the measurement error transfers directly to the estimations. Figure 5.11 shows the results for two datasets with a similar cleavage pattern to the one discussed in this section. The same substrate is digested with slightly different cleavage sites and different reaction rates, resulting in different but similar kinetics. Two datasets were simulated. A linear relationship between input titration error and output inference error was observed. The digestion product with the largest inference error is the cleavage sibling peptides of the titrated peptide. It is inferred with the same relative deviation from its true value as the titrated peptide. All other digestion products would have a lower error. This suggests that the QPuB inference is only as good as the conversion factor value provided.

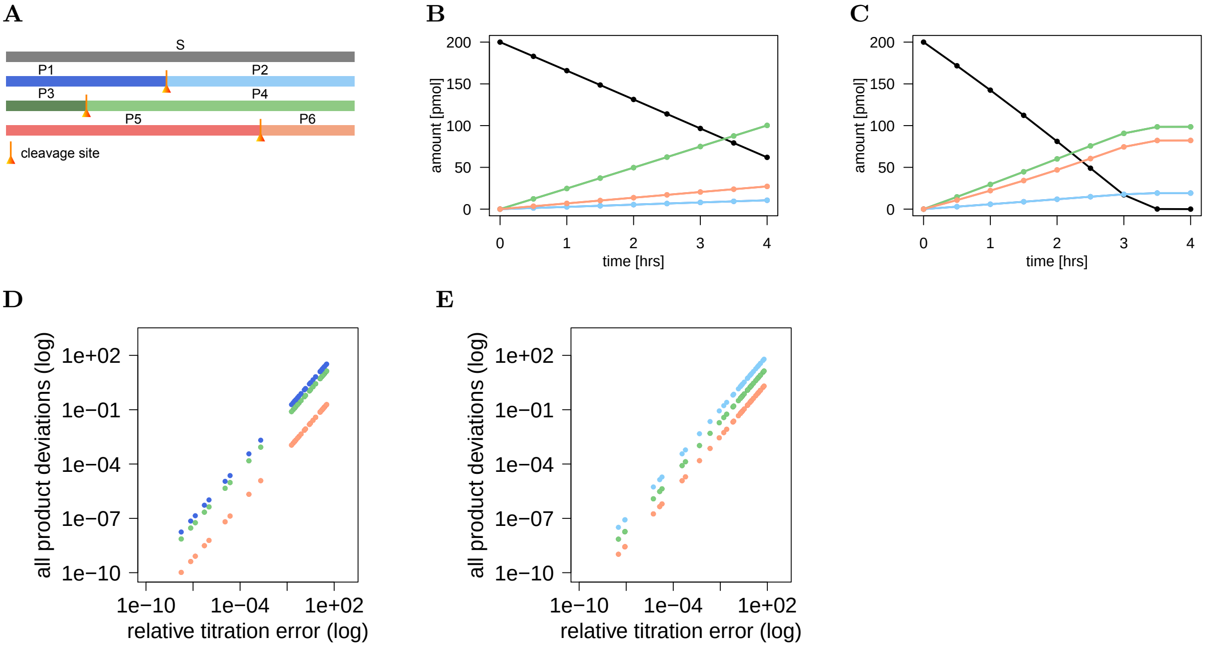


Fig. 5.11: Linear titration error in Ex. 2. To investigate the effect of erroneous measurements in the additional peptide titration, the conversion factor of the titrated peptide was fixed to normally perturbed values. The inference was run for 5e6 iterations achieving convergence. **(A) Cleavage pattern. (B-C) Concentration kinetics for the two replicates respectively. (D-E) Effect of the titration error.** The titrated peptide was chosen as recommended by the heuristic principle. In replicate 1, peptide 2 was fixed and in replicate 2, peptide 1 was fixed. The titration error is plotted on the x-axis as relative percent error. On the y-axis the mean percent error over the inferred conversion factors of the remaining peptides is drawn. Colors are as in (A). The relationship between the input titration error and the output inference error is linear for the two examples analysed. *[The simulations were run with a former version of the QPuB code with no major differences to the current version. The quality of the results is expected to be the same.]*

5.3.3.6 Summary

In datasets containing an insufficient amount of information impeding an effective inference, fixing one or more parameters to their correct values can cause improvement. We proposed a heuristic principle that recommends the number and identities of the peptides to titrate. By fixing their values, the others can be estimated successfully. This principle has proved useful in a variety of datasets. Our analysis suggests, that if the conversion factors are fixed to values perturbed from their true values, the resulting parameter estimations mirror the input error.

5.3.4 Strategy 2: Additional information through fitting several kinetics together

A second strategy to achieve identifiability and successful inference is presented in this section. Information on the parameters can be increased by measuring a second kinetic under different biological conditions. The conversion factor depends on the peptide's physico-chemical properties which determine its behaviour in the mass spectrometer. These properties should not be altered by the way the peptide was generated. Introduction of a protease inhibitor, digestion of the same substrate with two different protease isoforms or proteases from different cell lines or organisms, as well as changing the essay conditions or simply the ratio of substrate and enzyme in the digest only affect the kinetics of the digest. Combining two different kinetics should therefore add information on the conversion factors of those peptides which are observed in both datasets, facilitating inference. The idea is to provide a second set of synthetic data as QPuB input. In the calculation of the likelihood, QPuB will simply loop over the datasets, attempting to satisfy the mass balance condition for both of them simultaneously.

5.3.4.1 The additional data

The second dataset is created from the same cleavage pattern (Fig. 5.5A), but with different reaction rates, therefore resulting in a different concentration kinetics over time. The same conversion factors as before (Tab. 5.5B) are used to calculate the signal intensities. Numerical values are given in Tab. 5.8 and Fig. 5.12 juxtaposes the two reaction kinetics used in this example.

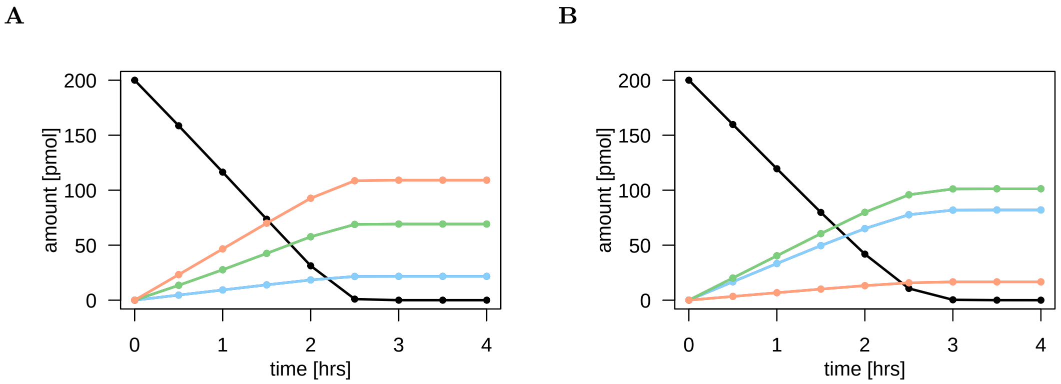


Fig. 5.12: Two concentration kinetics of the same cleavage pattern in Ex. 2. Different kinetics can arise due to different biological conditions. **(A) Kinetics under condition 1.** Peptide amounts over time of the unidentifiable dataset analysed before (see Fig. 5.5C). **(B) Kinetics under condition 2.** A second kinetics derived from the same cleavage pattern.

Table 5.8: Numerical data of the second kinetics in Ex. 2. (A) Concentration kinetics. Amount in [pmol] for the substrate and all peptide products over a digestion time of 4 hours. **(B) Conversion factors.** Conversion factors of all peptide products, systematically chosen. The conversion factor of the substrate is set to 1 by default. **(C) Kinetics of MS signal intensities.** MS signal intensities in [a.u.] for the substrate and all peptide products over a digestion time of 4 hours, as calculated from (A) and (B) using Eqn. (4.8). Shown are only the hourly time points, in the inference we also used the half-hourly measurements. Values in (A) and (C) are rounded to two digits precision for printing.

(A) Amounts [pmol]

Reactant	0h	1h	2h	3h	4h
S	200.00	119.51	41.86	0.35	5.21×10^{-5}
P_1	0.00	33.32	65.08	81.89	82.02
P_2	0.00	33.32	65.08	81.89	82.02
P_3	0.00	40.40	79.85	101.15	101.32
P_4	0.00	40.40	79.85	101.15	101.32
P_5	0.00	6.77	13.22	16.62	16.65
P_6	0.00	6.77	13.22	16.62	16.65

(B) Conversion factors

Parameter	Value
v_0	1
v_1	0.01
v_2	0.1
v_3	1
v_4	10
v_5	10.01
v_6	100

(C) MS signal intensities [a.u.]

Reactant	0h	1h	2h	3h	4h
S	2.00×10^{12}	1.20×10^{12}	4.19×10^{11}	3.45×10^{9}	521 376.06
P_1	0.00	3.33×10^{13}	6.51×10^{13}	8.19×10^{13}	8.20×10^{13}
P_2	0.00	3.33×10^{12}	6.51×10^{12}	8.19×10^{12}	8.20×10^{12}
P_3	0.00	4.04×10^{11}	7.99×10^{11}	1.01×10^{12}	1.01×10^{12}
P_4	0.00	4.04×10^{10}	7.99×10^{10}	1.01×10^{11}	1.01×10^{11}
P_5	0.00	6.76×10^{9}	1.32×10^{10}	1.66×10^{10}	1.66×10^{10}
P_6	0.00	6.77×10^{8}	1.32×10^{9}	1.66×10^{9}	1.67×10^{9}

Table 5.9: Improvement by fitting together two kinetics in Ex. 2. The table shows the information content and the performance of the inference when the datasets are run separately compared to in combination. The first column reports the information content of the least-informative direction. Note that it is below the heuristic threshold for both kinetics, but sufficiently large for the combination. The inference was stopped after 2e6 iterations. The accuracy is evaluated by the mean over the percent deviation from the true value (δ_v) as before. The precision is summarised as mean over the standard deviations for every conversion factor. Values were rounded to two digits precision for printing.

	info	accuracy	precision
cond 1	9.41×10^{-14}	134.51	1.65
cond 2	5.54×10^{-18}	117.85	0.58
together	3.98×10^{-10}	7.37×10^{-12}	1.03×10^{-10}

Table 5.10: Statistics of Ex. 2 with two kinetics. Inferences were run for 2e6 iterations, burn-in was discarded and the three chains were combined into one. For the conversion factors, the true value was subtracted to visualise the accuracy. Values were rounded to two digits precision. **(A) Posterior statistics of one run.** Statistics of the posterior distributions of all parameters in Ex. 2, when fitting two kinetics together. **(B) Repeatability.** The inference on the dataset was repeated 100 times. This table reports the mean of the posterior means across runs and the corresponding standard deviations.

(A) Single run

Parameter	mean	sd
$v_1 - v_1^*$	-9.70×10^{-16}	6.22×10^{-13}
$v_2 - v_2^*$	-9.01×10^{-15}	6.62×10^{-13}
$v_3 - v_3^*$	8.99×10^{-14}	1.00×10^{-13}
$v_4 - v_4^*$	9.01×10^{-13}	1.00×10^{-12}
$v_5 - v_5^*$	-9.95×10^{-14}	6.15×10^{-10}
$v_6 - v_6^*$	-1.41×10^{-12}	3.48×10^{-12}
σ	2.91×10^{-13}	1.00×10^{-14}

(B) Repeatability

Parameter	mean	sd
$v_1 - v_1^*$	-1.03×10^{-15}	1.18×10^{-15}
$v_2 - v_2^*$	-9.69×10^{-15}	1.15×10^{-14}
$v_3 - v_3^*$	9.02×10^{-14}	1.02×10^{-13}
$v_4 - v_4^*$	1.01×10^{-12}	9.93×10^{-13}
$v_5 - v_5^*$	-1.49×10^{-13}	4.02×10^{-13}
$v_6 - v_6^*$	-1.62×10^{-12}	4.02×10^{-12}
σ	3.18×10^{-13}	8.89×10^{-14}

5.3.4.2 Identifiability analysis

The combination of the two kinetics increases the information content. Table 5.9 shows the information contents of the two kinetics separately compared to the information content of the combination. Following the heuristic principle 5.1, it should now be possible for QPuB to infer the correct conversion factors for all peptide products.

5.3.4.3 Inference results

Indeed, the information content has been increased enough for a successful inference. After 2e6 iterations, the Markov chains have **converged**, visually (Fig. 5.13) as well as by the Gelman–Rubin diagnostics ($\text{mean}_i(\hat{R}_i) \approx 1.01$). The **posterior distributions** are now settled at a single peak with a narrow standard deviation (Fig. 5.13B). The means of the estimated conversion factors perfectly match the correct values with **low deviation**. All conversion factor deviations are below $1.001 \times 10^{-11}\%$ with the mean over all peptides being $\delta_v = 7.37 \times 10^{-12}\%$. The **calculated peptide amounts** over time have a very narrow uncertainty range for both datasets (Fig. 5.13D). The residual plots show perfect **mass balance** for both replicates (Fig. 5.13A). The means of the residuals in each chain are small (chain 1: -2.16×10^{-14}, chain 2: -2.48×10^{-14}, chain 3: -2.26×10^{-14}), now with a comparable order of magnitude as the results in Example 1. Also, after improving the identifiability of this example, the results are highly **repeatable** (Tab. 5.10B).

5.3.4.4 Summary

Using multiple kinetics under different biological conditions together improves the inference results substantially. The information content is increased so that all parameters are now identifiable.

For larger and more complicated datasets unfortunately this technique might only reduce the number of low-informative eigendirections, but not sufficiently. In these cases, fitting more then two biological replicates together can improve the identifiability, as presented in the next example.

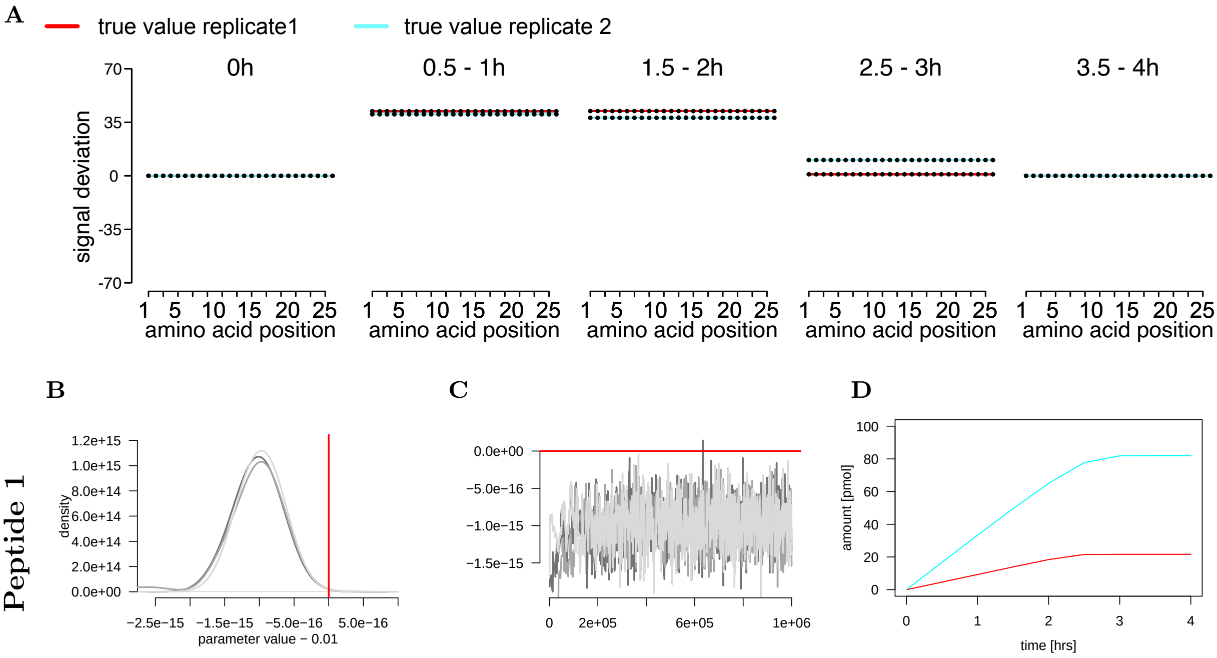


Fig. 5.13: Inference results of Ex. 2 when fitting two kinetics obtained under different biological conditions together. Only the inference results of parameter 1 are shown. The outcome for the others is similar. **(A) Residual plots** The red and cyan lines represent the amount of substrate degraded in the two kinetics respectively. The parameters are inferred such that mass balance is achieved for both conditions simultaneously. **(B) Posterior distributions of the conversion factors** and **(C) Trace plots of the Markov chains** Per chain, a single density is obtained of the values that best explain the data of both kinetics. **(D) Estimated amount kinetics** The inferred parameters yield peptide amounts that fit the true amounts perfectly. The uncertainty is so small that it is invisible in the plot.

5.4 Example 3

In this section, a more complex example with many peptide products is discussed. Application of the heuristic principles proposed above suggests non-identifiability and calls for additional measures. We will pursue the titration approach as well as the approach of considering multiple replicates in the inference, as well as combinations thereof.

5.4.1 The data

The same 26-mer substrate as in the previous examples is digested by a proteasome-like protease with a complex cleavage specificity shown in Figure 5.14A. Over a digestion time of four hours, 45 peptide products are produced. Canonical as well as spliced peptides are considered to increase the peptide diversity. Three kinetics under different biological conditions were simulated. Reaction rates were randomly sampled and kinetics generated using the julia code presented in Section 4.8.1. This time, also the conversion factors were randomly sampled; half of them on a range of [0,1], the other half in [1,100]. Values are shown in Tab. 5.11D. The kinetics of the peptide amounts and the resulting input MS signal intensities of one condition are shown in Fig. 5.14. Tables of the reaction rates and the kinetics under all conditions considered can be found in Tab. 5.11.

5.4.2 Results

5.4.2.1 Identifiability analysis

First, the information content of the individual replicates was calculated to investigate identifiability. According to the heuristic principle 5.3, each of them has a few low-informative eigendirections. Table 5.12 shows the number of vanishing eigenvalues of each dataset together with the parameters that contribute most to the low-informative eigenspace. Therefore, trying to run them individually in their original state will probably not result in successful inference in reasonable iteration time. For dataset A, the titration approach would require fixing six parameters, dataset B requires six and dataset D five titrations, respectively. The approach of combining several kinetics decreases the number drastically. Combining kinetics A and B reduces the number of required titrations to one. When fitting all three datasets together, no additional titrations are needed. In the following, we will focus on three strategies:

- Scenario 1: dataset A by itself with six peptide titrations,
- Scenario 2: datasets A and B together with titration of peptide P_{41},
- Scenario 3: all three kinetics together without additional peptide titration.

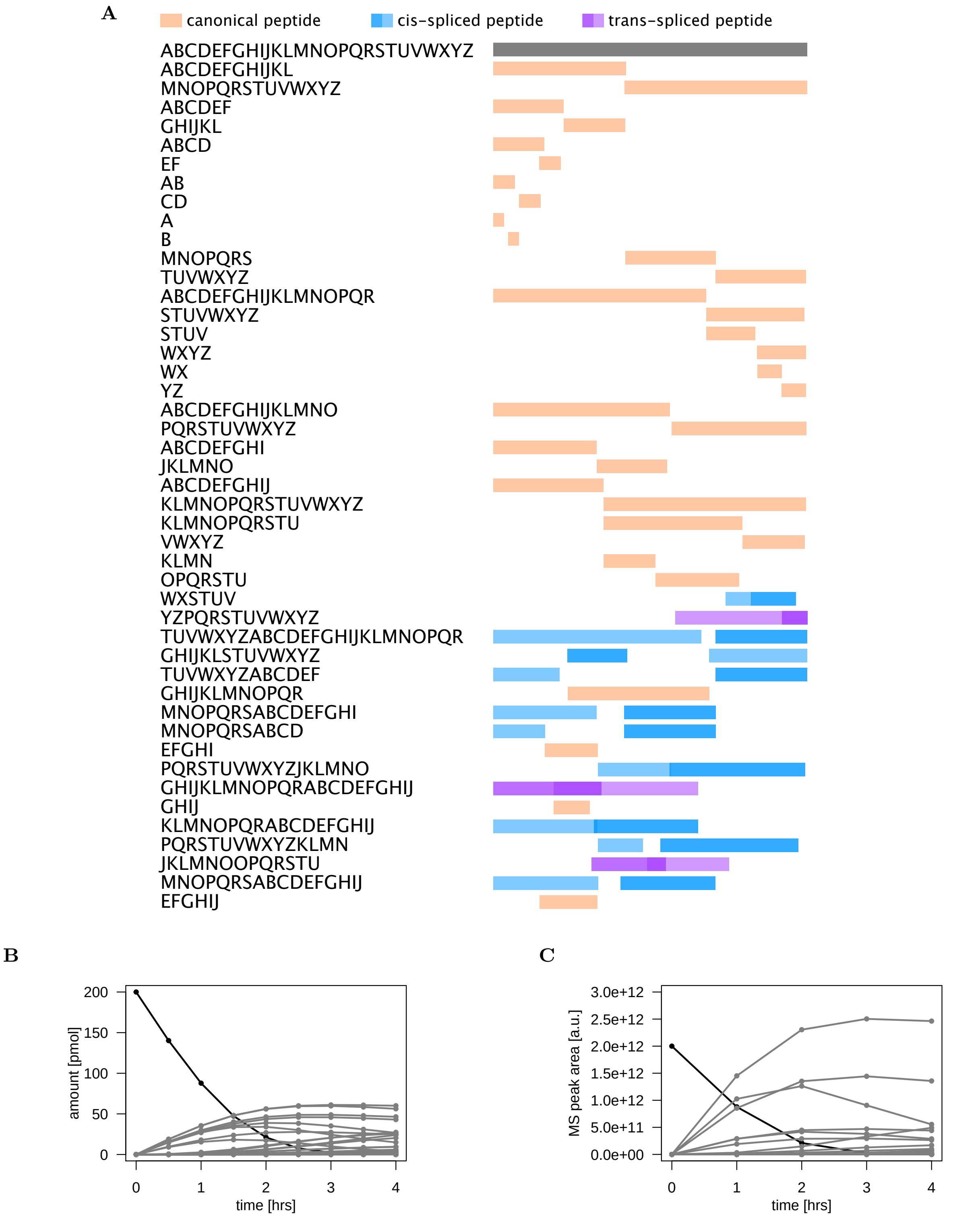


Fig. 5.14: Schematic of the cleavage pattern of Ex. 3 and resulting kinetics. (A) Digestion pattern. A 26-mer substrate is digested by a proteasome-style peptidase into 45 peptide products over the course of 4 hours. The enzyme hydrolyses the peptide bonds at four different positions in the substrate sequence and also digests multiple of the peptide products downstream. To further increase the peptide repertoire, also proteasome-catalysed spliced peptides are generated. **(B) Concentration kinetics of kinetic A.** Depicted are the amounts of the substrate and all peptide products over the digestion time, as simulated. For numerical values see Tab. 5.11A. **(C) Kinetics of MS signal intensities of kinetic A.** Depicted are the MS signal intensities of the substrate and all peptide products over the digestion time, as computed from the amounts in (B) using the conversion factors in Tab. 5.11D.

Table 5.11: Numerical data in Ex. 3. (A-C) Concentration kinetics. Amounts in [pmol] for the substrate and all peptide products over a digestion time of 4 hours for three different biological conditions. In this example, only hourly measurements are created. **(D) Conversion factors.** Conversion factors of all peptide products, randomly sampled. The conversion factor of the substrate is set to 1 by default. MS signal intensities for the substrate and all peptide products can be calculated from (A) and (B) using Eqn.(4.8). Values are rounded to two digits precision for printing. [Table continued on next page.]

(A) Amounts of kinetic A [pmol]

	0h	1h	2h	3h	4h
S	200.00	87.80	21.11	2.23	0.13
P1	0.00	15.48	17.40	9.98	3.98
P2	0.00	17.85	26.94	27.34	24.66
P3	0.00	2.71	11.07	20.68	25.73
P4	0.00	1.11	1.16	0.91	0.59
P5	0.00	0.02	0.20	0.85	1.99
P6	0.00	0.02	0.20	0.89	2.17
P7	0.00	0.00	0.00	0.04	0.14
P8	0.00	0.00	0.00	0.04	0.18
P9	0.00	3.17×10^{-6}	0.00	0.01	0.04
P10	0.00	3.17×10^{-6}	0.00	0.01	0.04
P11	0.00	0.07	0.07	0.07	0.07
P12	0.00	0.10	0.10	0.10	0.09
P13	0.00	29.88	46.31	48.85	46.41
P14	0.00	27.65	34.03	24.44	14.94
P15	0.00	0.87	3.74	7.31	9.56
P16	0.00	0.86	3.63	7.11	9.34
P17	0.00	0.01	0.12	0.20	0.21
P18	0.00	0.01	0.16	0.65	1.43
P19	0.00	35.38	56.17	61.03	60.04
P20	0.00	35.41	56.00	59.79	56.23
P21	0.00	0.17	0.82	2.05	3.60
P22	0.00	0.04	0.04	0.04	0.04
P23	0.00	28.03	43.61	45.83	42.98
P24	0.00	26.98	38.90	35.18	26.67
P25	0.00	1.32	6.10	13.61	20.37
P26	0.00	1.34	6.36	14.91	23.96
P27	0.00	0.01	0.06	0.14	0.22
P28	0.00	0.02	0.26	1.29	3.56
P29	0.00	0.00	0.05	0.57	1.74
P30	0.00	0.00	0.01	0.13	0.52
P31	0.00	0.26	1.56	3.71	5.92
P32	0.00	1.61	10.11	20.66	27.30
P33	0.00	0.00	0.07	0.41	1.20
P34	0.00	0.00	0.06	0.23	0.51
P35	0.00	0.00	0.01	0.06	0.16
P36	0.00	0.01	0.12	0.63	1.79
P37	0.00	1.51×10^{-5}	0.00	0.01	0.04
P38	0.00	0.13	0.80	2.07	3.72
P39	0.00	0.00	0.02	0.16	0.58
P40	0.00	2.14×10^{-6}	0.00	0.02	0.12
P41	0.00	2.14×10^{-6}	0.00	0.02	0.12
P42	0.00	0.01	0.20	1.16	3.37
P43	0.00	1.62×10^{-5}	0.00	0.01	0.04
P44	0.00	0.28	1.52	3.45	5.19
P45	0.00	0.01	0.12	0.63	1.76

(B) Amounts of kinetic B [pmol]

	0h	1h	2h	3h	4h
S	200.00	78.02	31.52	12.52	4.15
P1	0.00	19.30	24.41	24.47	22.23
P2	0.00	18.80	23.11	22.41	19.52
P3	0.00	1.38	3.55	5.47	7.01
P4	0.00	0.62	0.91	1.26	1.89
P5	0.00	0.14	0.77	1.89	3.61
P6	0.00	0.15	0.83	2.11	4.23
P7	0.00	0.01	0.06	0.22	0.62
P8	0.00	0.01	0.06	0.22	0.62
P9	0.00	9.20×10^{-6}	0.00	0.00	0.01
P10	0.00	9.20×10^{-6}	0.00	0.00	0.01
P11	0.00	1.21	1.12	0.88	0.76
P12	0.00	0.11	0.10	0.09	0.08
P13	0.00	60.38	80.37	86.07	86.00
P14	0.00	44.41	41.06	28.38	15.29
P15	0.00	16.26	32.71	37.93	34.53
P16	0.00	14.31	28.86	33.88	30.56
P17	0.00	1.95	3.84	4.05	3.97
P18	0.00	0.89	2.42	7.30	22.06
P19	0.00	11.47	10.73	7.66	4.47
P20	0.00	14.02	10.25	2.05	0.06
P21	0.00	3.85	6.89	8.25	8.14
P22	0.00	4.32	9.67	14.10	18.08
P23	0.00	22.73	31.51	35.05	36.49
P24	0.00	22.21	30.01	32.39	32.42
P25	0.00	0.54	1.56	2.74	4.12
P26	0.00	0.55	1.62	2.91	4.50
P27	0.00	0.01	0.04	0.12	0.32
P28	0.00	0.01	0.05	0.12	0.20
P29	0.00	0.71	8.72	22.98	40.70
P30	0.00	1.77	10.14	19.72	22.61
P31	0.00	1.91	5.55	9.43	13.64
P32	0.00	0.91	3.47	6.32	9.35
P33	0.00	0.01	0.04	0.11	0.23
P34	0.00	0.00	0.00	0.01	0.01
P35	0.00	0.70	3.24	4.89	5.49
P36	0.00	0.10	1.27	3.80	7.64
P37	0.00	0.09	1.24	3.72	7.50
P38	0.00	0.32	1.69	2.71	2.86
P39	0.00	0.00	0.03	0.09	0.18
P40	0.00	0.00	0.00	0.02	0.05
P41	0.00	0.00	0.00	0.02	0.05
P42	0.00	0.00	0.02	0.05	0.06
P43	0.00	0.00	0.01	0.05	0.18
P44	0.00	0.02	0.05	0.07	0.07
P45	0.00	0.00	0.03	0.08	0.14

Table 5.11: continued.

(C) Amounts of kinetic D [pmol]

	0h	1h	2h	3h	4h
S	200.00	81.68	19.53	3.56	0.57
P1	0.00	11.28	15.82	15.40	13.74
P2	0.00	11.10	15.06	14.01	11.85
P3	0.00	0.44	1.87	3.67	5.35
P4	0.00	0.30	0.55	0.60	0.63
P5	0.00	0.00	0.05	0.19	0.42
P6	0.00	0.00	0.06	0.21	0.47
P7	0.00	7.54×10^{-5}	0.00	0.02	0.05
P8	0.00	7.57×10^{-5}	0.00	0.02	0.05
P9	0.00	3.49×10^{-7}	2.93×10^{-5}	0.00	0.00
P10	0.00	3.49×10^{-7}	2.93×10^{-5}	0.00	0.00
P11	0.00	0.26	0.28	0.26	0.22
P12	0.00	0.47	0.97	0.97	0.86
P13	0.00	33.03	48.54	50.32	48.58
P14	0.00	31.69	43.09	39.96	33.60
P15	0.00	1.35	5.45	8.71	9.65
P16	0.00	1.19	4.11	6.14	6.67
P17	0.00	0.16	1.34	2.58	2.99
P18	0.00	0.05	0.18	0.42	1.21
P19	0.00	12.55	16.48	14.59	11.62
P20	0.00	12.99	15.47	9.57	2.71
P21	0.00	0.94	3.98	7.65	10.93
P22	0.00	0.55	0.79	0.80	0.76
P23	0.00	59.54	89.42	94.46	92.35
P24	0.00	57.62	81.80	79.56	70.48
P25	0.00	2.19	9.04	16.66	22.49
P26	0.00	2.29	10.19	20.74	31.39
P27	0.00	0.09	0.72	2.17	5.18
P28	0.00	0.10	1.00	3.15	6.08
P29	0.00	0.00	0.36	2.67	6.99
P30	0.00	0.11	1.52	4.83	8.77
P31	0.00	0.15	1.55	3.46	4.81
P32	0.00	0.14	1.38	3.28	5.19
P33	0.00	0.01	0.18	0.85	2.05
P34	0.00	0.00	0.01	0.02	0.02
P35	0.00	2.75×10^{-5}	0.00	0.00	0.00
P36	0.00	0.00	0.05	0.20	0.46
P37	0.00	2.78×10^{-7}	1.40×10^{-5}	9.05×10^{-5}	0.00
P38	0.00	0.39	3.05	5.92	7.36
P39	0.00	0.00	0.17	0.81	1.93
P40	0.00	1.93×10^{-5}	0.00	0.02	0.09
P41	0.00	1.93×10^{-5}	0.00	0.02	0.09
P42	0.00	0.01	0.43	1.91	3.72
P43	0.00	0.00	0.15	0.93	2.82
P44	0.00	0.37	2.36	4.82	7.03
P45	0.00	0.00	0.05	0.20	0.46

(D) Conversion factors

Parameter	Value
v_0	1.00
v_1	87.83
v_2	0.93
v_3	45.96
v_4	46.84
v_5	21.37
v_6	0.30
v_7	0.25
v_8	32.83
v_9	0.78
v_{10}	13.20
v_{11}	26.17
v_{12}	0.66
v_{13}	91.92
v_{14}	0.27
v_{15}	50.23
v_{16}	0.55
v_{17}	82.83
v_{18}	29.17
v_{19}	0.24
v_{20}	0.41
v_{21}	0.87
v_{22}	92.35
v_{23}	0.97
v_{24}	0.93
v_{25}	0.41
v_{26}	69.39
v_{27}	0.76
v_{28}	25.36
v_{29}	53.76
v_{30}	0.99
v_{31}	8.23
v_{32}	47.23
v_{33}	11.07
v_{34}	0.69
v_{35}	69.66
v_{36}	0.55
v_{37}	0.29
v_{38}	1.66
v_{39}	0.08
v_{40}	0.49
v_{41}	72.41
v_{42}	5.43
v_{43}	0.60
v_{44}	0.51
v_{45}	32.81

Table 5.12: Identifiability analysis in Ex. 3. The curvature matrix was calculated for the individual datasets as well as for possible combinations. The heuristic principle 5.3 was applied. The table shows the number of small eigenvalues and the parameters which contribute most to the space spanned by low-informative eigenvectors. The single datasets have very low information content and call for many additional peptide titrations. When combining two of the kinetics, the information content increases significantly. When combining all of the kinetics, no additional peptide titrations are required.

replicate	number of small eigenvalues	biggest contribution
A	6	35, 41, 10, 22, 8, 17
B	6	10, 9, 44, 34, 27, 7
D	5	35, 10, 37, 8, 41
A,B	1	41
A,B,D	0	-

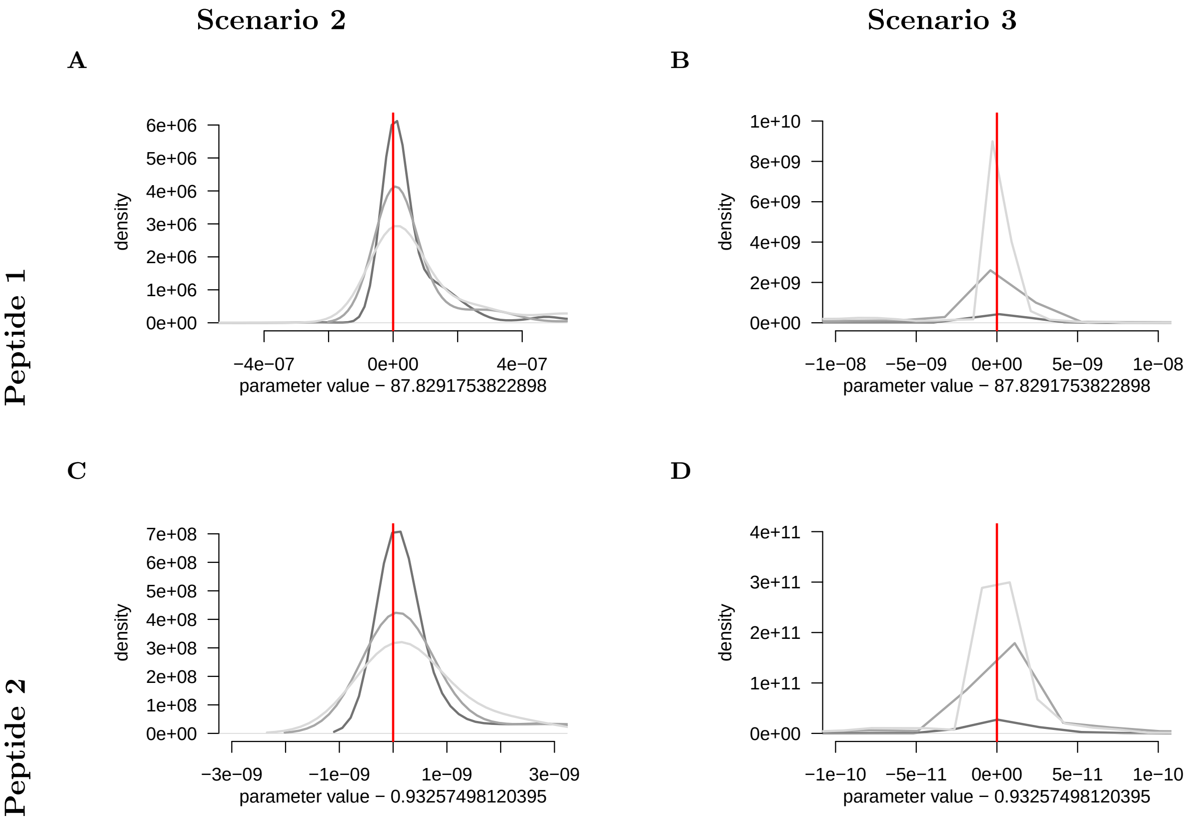


Fig. 5.15: Inference results of Ex. 3 for Scenario 2 and 3. Posterior distributions of the conversion factors of peptides 1 and 2 are shown. The outcome for the others is similar. To visualise the accuracy and precision of the estimate, the correct values were subtracted of the parameter values on the x-axes. **(A)** Density of parameter 1 when combining datasets A and B and fixing the value of v_{41}. **(B)** Density of parameter 1 when combining datasets A, B and D. **(C)** Density of parameter 2 when combining datasets A and B and fixing the value of v_{41}. **(D)** Density of parameter 2 when combining datasets A, B and D.

5.4.2.2 Inference results

All inferences were run for 2e7 iterations. As expected, in the inferences of the cases which the rule of thumb predicts to be problematic, we did not obtain **convergence** after a reasonable iteration time (data not shown). Surprisingly, Scenario 1 did not converge over iteration time (visually as well as Gelman–Rubin), although the rule of thumb did not indicate any difficulties. In Scenarios 2 and 3 the chains did converge with $\underset{d=1,\ldots,D}{\text{mean}}(\hat{R}_d) \approx 1.06$ and $\underset{d=1,\ldots,D}{\text{mean}}(\hat{R}_d) \approx 1.03$, respectively. We will omit the results of scenario 1 in the following, only reporting results of the two scenarios that converged. In both scenarios, the **posterior densities** of all (remaining) conversion factors are normally distributed with small standard deviations. Figure 5.15 shows the densities of peptides 1 and 2 as representatives reflecting the inference outcome. The densities of the other conversion factors are qualitatively similar. We report here the mean over the standard deviations of all inferred conversion factors to summarise the precision:

$$\underset{d=1,\ldots,D}{\text{mean}}(\text{sd}(v_d)) = 1.33 \times 10^{-6} \qquad \text{(Scenario 2)} \tag{5.8}$$

$$\underset{d=1,\ldots,D}{\text{mean}}(\text{sd}(v_d)) = 5.26 \times 10^{-7} \qquad \text{(Scenario 3).} \tag{5.9}$$

The estimated conversion factors of all (remaining) peptides are **highly accurate** and coincide with their true values. The mean of the deviations in percent are

$$\delta_v = 2.62 \times 10^{-8} \qquad \text{(Scenario 2)} \tag{5.10}$$

$$\delta_v = 1.14 \times 10^{-10} \qquad \text{(Scenario 3).} \tag{5.11}$$

Likewise, the distributions of the peptide concentrations are very narrow for every replicate. In both scenarios, the mass balance was satisfied to a high degree:

chain	Scenario 2	Scenario 3
chain 1	-1.10×10^{-10}	-6.80×10^{-12}
chain 2	-6.78×10^{-11}	-1.97×10^{-13}
chain 3	-3.56×10^{-10}	-2.80×10^{-15}

(5.12)

The simulations were **repeatable**, but we did not study it in large scale.

5.4.3 Summary

Some large datasets have multiple bad directions and the rule of thumb suggests multiple peptide titrations. Here, the inference can be successful when fixing the recommended conversion factors to their true values. Unfortunately, the information gain through many additional titrations sometimes might not be enough for a successful inference in reasonable iteration time. Fitting together multiple datasets simulated to represent measurements under different biological conditions sufficiently constrains the parameters to infer. With a certain number of kinetics, the need for additional peptide titrations can be eliminated.

6 Discussion

The aim of this project was to implement a computational pipeline to absolutely quantify all peptide products of an *in vitro* protein digestion analysed by mass spectrometry. This chapter will summarise the key findings and discuss the value of the results. We will review the limitations and recommend implementations for improvement.

6.1 Discussion of the results

We here presented QPuB, a tool for absolute Quantification of Peptide products $u$sing Bayesian inference. It is based on the principle of mass balance and the linear relationship between the peptide amount and its MS signal response. The underlying algorithm applies Bayesian inference in a Markov Chain Monte Carlo scheme to iteratively estimate the conversion factors enabling computational transition from measured signal intensities to desired peptide amounts. To develop and calibrate the algorithm and to evaluate its performance, it was tested on synthetic noise-free data where the correct solution is known. The results indicate that QPuB is able to successfully infer the correct solution with high precision and accuracy under the premise that sufficient information is conveyed by the data. If the data does not sufficiently constrain the parameters, then QPuB cannot give a reliable estimate. As a proof of concept, we demonstrated the QPuB pipeline on the simplest possible example of only two cleavage products. It would be interesting to investigate the maximal possible precision the sampler can achieve. However, for biological applications, this level of precision is usually not required. The same argument explains why we judge the estimates to be satisfactory although the true value lies several standard deviations away from the mean of the posterior (Fig. 5.2B, 5.4B, 5.9B, 5.13B). This small error in the conversion factor values translates directly into the precision of the concentrations, and a deviation of the order of e-15 pmol is usually not relevant in applications.

Investigation of the curvature of the likelihood surface seems to provide a way to anticipate whether the data contain sufficient information to constrain the parameters for successful inference. Since the likelihood function is Gaussian the corresponding curvature matrix is independent of the parameters. This allows us to assess the sensitivity of the likelihood around the optimum before running the inference and gives an indication whether it is worth starting a simulation. This is convenient, because it saves the user hours or days of computational runtime until they see that the system is not identifiable and no unique estimate can be made. If the Hessian matrix has eigenvalues which are zero or very small compared to the largest eigenvalues, this means that there is large variance along the direction of the corresponding eigenvector. A deviation of the parameter value in a direction of low curvature leads to no or small change in the likelihood value, making it difficult to find the optimal value. In case the curvature matrix has multiple vanishing eigenvalues, the problem expands to multiple low-informative directions which can be arbitrarily combined. With little information through the likelihood evaluation, the chain is not guided and diffuses through the space spanned by low-informative directions. This is the case in Example 2. Here, all products are of first order and none of them is further digested. Therefore sibling peptides have the same concentration kinetics and the kinetics of the different peptide pairs are very parallel to each other. Due to this linear dependence, the measurements over digestion time do not provide new information. In Example 1, this dependence is not so strong because one of the products is further digested, leading to a more diverse kinetics. Although the cleavage pattern of Example 2 might not be very realistic, the same behaviour can occur in different, more complex cleavage patterns like Example 3. The conversion factors of the peptides are not sufficiently constrained and an infinite number of combinations yields the same sum of mass for a particular amino acid position. The parameters are correlated; if the mass of one of the peptides is increased, another is decreased and infinitely many parameter combinations satisfy the mass balance condition. This leads to the problem of non-identifiability. The chains sample along the state space with nowhere to settle. Since the prior range is a bounded uniform distribution, the chains will converge to a uniform posterior after many iterations. Using the estimate of such an inference — a premature one as well as the converged — will lead to wrong calculated peptide amounts in the sample. Thus, the identifiability analysis on the dataset a priori is a valuable tool which can save time and prevent from ill-considered conclusions.

The analysis of the Hessian not only allows to appraise whether or not the parameters probably will be identifiable, but also provides a framework to remedy the problem of non-identifiability. In general, additional data can yield additional information which facilitates the inference. Identification of the low-informative parameter directions with large variance and determination of the parameters that contribute most to the space spanned by them allows to spot the parameters which are hardest to infer. One way to obtain more information about these parameters would be to measure the conversion factor value through titration of a synthetic equivalent of the corresponding peptide, as shown in Sec. 5.3.3. Fixing the parameter to a certain value reduces the variability between the parameters and therewith constrains the value of the others, resulting in successful inference.

The identifiability analysis reveals that the curvature matrices of all examples shown in this thesis have a large difference in magnitude between the largest and smallest eigenvalues. The machine precision of a double float in R is 2.220446e-16. Therefore, the smallest eigenvalue will appear negligible in comparison to the others and the likelihood in the corresponding direction appears very flat around its maximum. In Example 2, the difference in magnitude is of the order e13 (Fig. 5.7), which leads to numerical difficulties. The eigenvalues are so small, i.e. the curvature is so flat, that navigation in the respective directions is not possible anymore. Even in Example 1, the difference in magnitude is already very large with an order of e9. Nevertheless, the curvature is still large enough for the chain to find its way and the algorithm succeeds to make use of the little information there is to return a reliable estimate. This is a considerable strength of the algorithm.

Since the order of magnitude of eigenvalues of a matrix changes under scaling of the data, we use the normalised eigenvalues, the proportion of variance explained, to assess identifiability. The analysis of multiple datasets lead us to the observation, that the transition between identifiable and non-identifiable should lie somewhere in the order of e-12. However, specifying a clear cut threshold when the parameters are not constrained enough and QPuB will most likely fail to infer the correct values in reasonable iteration time is difficult. The current implementation uses a threshold of 1e-12. If at least one normalised eigenvalue is below this threshold, QPuB gives a warning. In Scenario 1 of Example 3, the threshold of 1e-12 identifies six low-informative directions. Yet, with six additional peptide titrations, the curvature matrix still has three normalised eigenvalues in the order of e-12. However, fixing nine parameters to their correct values also does not enable the chains to converge in the iteration time tested. On the other hand, Scenario 2 still has one value of order e-12 but displays successful inference. In the border area assessment of identifiability can become vague. Making definite statements in the order of machine precision is tricky. Nevertheless, the heuristic principle builds a guideline that is easily applicable and proved useful in practice.

Calculation of the conversion factor from the experimentally measured calibration curve will be subject to some measurement error. First results indicate that fixing the parameter to an erroneous value results in estimates whose posterior median deviates from the true value by the same relative error (Fig. 5.11). The conversion factor estimate of the sibling peptide exhibits the same error that was input. This makes sense because in the cleavage pattern studied they have the same concentration by construction. All other cleavage products can be inferred to a higher accuracy. Further investigation should include other cleavage patterns as well. This analysis was performed on noise-free data. How the error of the titration propagates with the error in the kinetics needs to be investigated. Also how the error accumulates when multiple additional peptide titrations are advised can be subject to further studies. In a similar fashion, the effects of an erroneous substrate titration on the inferred peptide concentrations could be studied. These findings suggest that QPuB can only be as accurate as the data provided. If the user has an estimation of the level of error of the titration performed they can size up the error of the inferred conversion factors.

In general, the titration approach requires the user to measure the digestion kinetics, consult QPuB, then purchase recommended peptide equivalents and perform the respective titrations. While kinetics and titration should better be performed back to back for comparability, this procedure is theoretically feasible in the application, if the synthesis of the inquired peptides does not take too long and no major changes were made to the MS setup in the meantime (cleaning or calibration). However, the digestion kinetics could also be remeasured. A practical limitation of the titration approach can be reached for large and complex datasets where many additional peptide titrations are recommended. However, for low numbers of low-informative eigendirections this is a relatively fast procedure to make the measured dataset identifiable.

Another way to introduce more information to the system is to measure a second kinetics under different biological conditions resulting in ideally the same set of peptide products. If the same peptide is produced in the different digests, then the amount of information on its conversion factor in the data is increased. Like the titration procedure, this provides more information about the unconstrained parameters, but instead of just fixing a single value, many new data points now contribute to the calculation of the likelihood. In addition, this strategy also provides more data on all the identifiable parameters as well, therewith facilitating the inference even more. This could explain why Scenario 1 in Example 3 does not converge in time even with nine additional peptide titrations, whereas it does with the additional kinetic. We consider this in general the superior approach. If the user can, they are advised to provide a second (or more) kinetics to the inference to decrease the number of additional peptide titrations needed. Luckily, in many use cases, the researcher is already investigating two or more biological situations in comparison, interested in the differences in peptide amounts under the two more conditions. Instead of a purely relative assessment of abundance between the same peptides, with the application of QPuB they would have the benefit of being able to compare abundances between different peptides as well. If, in addition, they provide the substrate titration, even absolute amounts for all peptides in the digestions are obtained in one go.

6.2 On the way to real data

Our investigation indicates that QPuB is able to successfully infer the correct conversion factors if the data is noise-free, satisfies mass balance and sufficiently constrains all parameters. However, in reality measurements will be far from perfect. Further *in silico* investigation is advised to analyse the performance on more realistic datasets.

The QPuB predecessor QME was developed to handle small conversion factors reflecting the small dynamic range of older mass spectrometers. Modern devices have a larger mass range and are potentially able to detect peptides with conversion factors ranging from very small ($\sim 10^{-4}$) to very large ($\sim 10^{4}$). The examples investigated in this thesis used conversion factors in the range $[0, 100]$, which already exceeds QME's capability. Attempts to increase the **conversion factor range** to $[0, 10\,000]$ have been made. Preliminary analyses indicate that the parameters are harder to infer if their values are dispersed over a larger range. It might also be interesting to analyse how easy it is for the algorithm to distinguish between two conversion factors with very similar values.

In this thesis, we made the assumption that all peptide products produced in the digestion are detected and identified. However, in reality, this is rarely the case. For a discussion of this assumption see Sec. 6.4. Studying the effect of **missing peptides** *in vitro* would be very important to draw conclusions about applicability of QPuB to real data. Depending on the sensitivity of the mass spectrometer used, low abundant peptides might escape the analysis. Due to poor ionisation, fragmentation or transport of the ions, peptides can induce low signal intensities which can be indistinguishable from background noise. Detectability of short amino acid sequences depends on the sensitivity and precision of the instrument as well as the identification software used. Usually, peptides shorter than three amino acid residues escape detection. Also randomly missing peptides should be studied [125]. Preliminary results indicate that the effect of a missing peptide depends on the total number of peptides in the digest, on the position of the peptide in the cleavage hierarchy, and its abundance. In a small digestion with few products, the mass is largely off balance even with a single peptide missing. The larger the peptide pool, the smaller the effect will be. However, in the setting of Example 3, removing even one out of 45 products can have a considerable effect.

The **substrate sequence** used in this thesis was the latin alphabet, which is not very realistic. In QPuB, the choice of characters used does not make a difference for the inference, as long as the sequences can be compared and aligned. However, what would be interesting to investigate is the effect of amino acid repetitions in the sequence. Some amino acids are more frequently observed in proteins than others and the probability, that a repetition occurs in a sequence of length 26 is quite high. Of particular interest would be the simulation of a tryptic digest, where cleavage occurs C-terminally of lysine and arginine residues. Preliminary results indicate that repetitions of single amino acids along the substrate sequence do not hinder successful inference. Peptide products can most likely be uniquely aligned to the substrate sequence because of the conventions used in the assignment step (if a peptide sequence is a direct subsequence of the parental sequence, it is identified as canonical hydrolysis product). However, repetitions of longer amino acid stretches inside the substrate sequence can lead to peptides of ambiguous origin, which causes uncertainty in the mass balance calculation. In this case, inference can be hampered, as preliminary analysis suggests.

Related to the study of trypsin is the study of other **types of proteases**. QPuB does not need the information about which protease was used for the digest, it solely compares the sequences and uses mass balance. In principle, the digestion of any kind of protease can be analysed. It only becomes problematic if some peptide products are failed to be identified due to poor ionisation or inconsistent kinetic behaviour and the argument comes back to the satisfaction of mass balance. A protease of interest is the endoplasmic reticulum aminopeptidase (ERAP I), which is one of the key players in the MHC I antigen processing and presentation pathway of the adaptive immune system. It ensures, that peptides have the optimal length of 9 to 15 amino acids to be loaded onto the major histocompatibility complex (MHC) class I for presentation to T cells. Aminopeptidases are exopeptidases. Exopeptidases cleave off single amino acid residues or short fragments at the N- or C-terminus of their substrate. This constitutes a challenge of mass loss that could be accounted for in QPuB. A former version of QPuB would take the information whether the protease was an endo- or exopeptidase as input, then figure out the longest common subsequence of the peptides produced and only consider the overlapping amino acid positions in the calculation of the mass balance. However, this implementation was removed with the rationale that endopeptidases can also produce short undetectable products as well as the aim to reduce required user input. It might be helpful for the inference to reimplement this feature to at least factor in this expected systematic mass loss where it is guaranteed to occur. Preliminary inference on simulated datasets mimicking exopeptidatic digestions with all fragments detected seem to be successful for small number of products. For more products, the curvature matrix reveals a number of low-informative eigendirections, that can be reduced by following the approach of fitting together multiple kinetics under different biological conditions as described in Sec. 5.3.4. Further investigation is needed. The impact of not detecting the single amino acid products has not yet been investigated.

The main factor to investigate *in silico* is the question of **data noise**. In reality, the measurements of the MS signal intensities over digestion time will be subject to random and systematic error. How the level of noise influences the identifiability of the parameters and the quality of the resulting concentrations is of major interest in regard of potential real-life applications. In the generation of synthetic data (Sec. 4.8.1), multiplicative noise can be added to the data as described in Eqn. (4.23). This would represent technical noise from the device. Another possibility would be to include biological noise that depends on the concentration and behaviour of the peptides in the mixture over time. Noisy data should also always be provided in biological and technical replicates to account for some uncertainty. The impact of data noise on the inference results should be investigated in practical terms — how will the level of noise effect the success or uncertainty of the inference? — as well as from an analytical point examining the information content change under different levels of noise. In the same context, the influence of the **number of replicates** can be examined. When adding noise to identifiable examples, we expect the precision of the estimates to decrease depending on the spread of the replicates, i.e. the more noise the wider the posterior distributions. The accuracy of the estimate will probably depend on the mean of the replicates and how much this represents the true values. In theory, the more replicates are measured, the more their mean represents the true underlying value [177]. Some investigations on noise have already been done. We generated noisy data using Eqn. (4.23) in two or more replicates for different levels of noise, i.e. different values of σ_{noise}. Preliminary analysis on the dataset of Example 1 shows, that the precision and accuracy of the estimate decrease as the level of noise increases. A systematic analysis can reveal the relationship between the noise level and the inference error. In non-identifiable datasets, it is possible that the noise can have the effect of making the problem identifiable. This is what was observed when adding noise to Example 2. Above a certain level of noise, the eigenvalues are raised above the heuristic threshold of e-12. The low-informative directions are then determined by the measurement error. This allows the chains to converge to a solution after the same number of iterations it did not converge in before. However, the inferred conversion factors are not very precise nor accurate. Increasing the number of replicates or time points measured in the data could increase the information in the system, allowing for an improved inference. This however is strongly dependent on the shape of the kinetics and requires further investigation. In the case of noisy data, single inference results are maybe not conclusive about the underlying systematic. An analytical examination of the likelihood and its curvature matrix under noise could be useful to gain a deeper understanding of the difficulties arising.

Once QPuB is calibrated to realistic synthetic data, the thorough *in silico* analysis can be followed by calibration on experimental data. As opposed to the controlled environment of simulated data, the correct conversion factors are not known in reality. To be able to benchmark the performance and validate the results, the concentrations of the digestion products need to be experimentally obtained via titration. The inferred concentrations can then be compared to the measured kinetics. The estimated concentration range should overlap with the “true” values. After satisfying accuracy as well as precision has been achieved for a sufficiently large number of datasets, QPuB can be applied in practice and serve as a convenient tool to computationally quantify digestion products.

6.3 Discussion of the implementation

A handy characteristic of the Bayesian approach is that it is built from modules which can easily be replaced [145]. A different prior can be defined, the likelihood function can be adjusted and the proposal algorithm can be substituted by a more efficient one. The current default **prior** of QPuB is a wide but bounded uniform prior for every parameter. If the user has more information about the conversion factors of certain peptides, they can replace the default. On the one hand, information could be gained experimentally. Especially in the case of additional peptide titrations, it would make sense to define a normal prior around the measured value instead of simply setting the value to a fixed number. On the other hand, the prior can be informed by a former posterior. It should not be informed by another inference on the same dataset. However, if sufficient timepoints or replicates have been measured, a subset could be used to run a preliminary inference whose posterior can be used as prior of a subsequent run. Also, since the conversion factors of the peptides should be the same under different biological conditions, the inference on the dataset of one kinetics could be used to inform the prior of a second. Different prior distributions can be tested to see how they affect the resulting posterior. Preliminary results show that if the chosen prior range of a parameter is too narrow such that the true value is not covered, then the estimates of all conversion factors are affected to make up for this error. A drawback of the uniform prior as defined is that small values below 1 are less probable to be sampled than larger values. A commonly chosen prior for scaling parameters like the conversion factors is a logarithmical uniform prior.

The **likelihood** function can be adjusted to incorporate new features. For example, future work could deal with the violation of the mass conservation. Peters et al. [37] punished mass gain by defining an unsymmetric distance metric that weights an unlikely mass gain more than a (systematic) mass loss. A former QPuB implementation accounted for this in the definition of the likelihood. With an additional scaling parameter, the standard deviation of the normally distributed likelihood would be narrowed for the case of mass gain. Initially, this punishment parameter was set to a fixed value, that was later replaced by a nuisance parameter to be inferred. For the inference on noise-free datasets, this parameter did not seem to make a difference in the output, so it was omitted again. For noisy data, it could be reimplemented. However, since this punishment also restrains the possibility of mass gain due to measurement error instead of true mass error, this concept might have to be reconsidered.

There is no one-size-fits-all inference algorithm. In the development of QPuB we went through the implementation of a range of different **MCMC algorithms**. We started with the basic Metropolis–Hastings algorithm [137, 138, 139], which required cumbersome tuning of the proposal and did not converge in reasonable iteration time even for low-dimensional problems. This issue was solved through adding an adaptive scheme in the form of adaptive Metropolis [160, 189]. We went through implementations of the

Rao-Blackwellised AM algorithm (Alg. 3 in [190]), introducing global adaptation (Alg. 4 in [190]), as well as a componentwise approach (Alg. 6 in [190]), but none performed to satisfaction. A colleague worked on a Hamiltonian Monte Carlo approach, which is promising for high-dimensional problems but ran into problems in our model.

Finally, we decided to take advantage of the learning capacity of population-based sampling. By taking into account the states of multiple chains in parallel, a more efficient proposal can be achieved. The Differential Evolution Markov Chain algorithm (DE-MC) by Ter Braak [41] looked promising for a successful quantification of up to 100 peptides. The need to run 50 to 200 chains in parallel, however, seemed impractical. In follow-up publications, the authors managed to reduce the number of chains required to a minimum of three by incorporating features like sampling from an archive of past states and increasing the variability in the proposal step [42, 43]. They successfully applied the DREAM algorithm to high-dimensional, multi-modal and nonlinear target distributions, respectively [43].

In the current implementation of QPuB, some adjustments can be done to increase the performance. The default in QPuB are **start values** of the three chains which are dispersed over the prior region, namely from the lower and upper bound and the middle point respectively. We also implemented random initialisation or user-defined starting positions. The repeatability analysis in the results section was performed 100 times starting from the same initialisation, because this is the default setting. The analysis might be more representative if started from random start values. However, in our tests on noise-free data it did not seem to make a difference where to start sampling, the true solution was reached every time. The **number of parallel chains** run by the sampler is set to three by default. This is recommended by the DREAM literature, since the time it takes to reach convergence increases drastically the more chains are run [42]. However, for a large number of peptides in the sample, it could be beneficial to increase the number of chains [45]. An advantage of multi-chain algorithms is their ability to parallelise [144]. Because the proposal works on past states instead of the current ones, it can be distributed to multiple processors [44]. This can substantially decrease the algorithm runtime when many chains are used to infer many parameters. A boost in convergence rate could be achieved by tuning the **proposal function**. In the parallel direction update, to increase the efficiency, the gamma jump rate could be adjusted. The factor β_0 can be tuned to improve the acceptance rate [44]. The jump rate γ_s in the snooker update is currently uniformly sampled like in [42]. However, they remark that this might be suboptimal for normal target distributions. Therefore, fixing $\gamma_s = 2.38/\sqrt{2}$ could be an option. The occasional unit gamma jump is performed every 10 iterations [41]. This could be increased to a frequency of every 5 iterations like in [43] and onwards. Surprisingly, although the authors recommend a unit jump distance of 0.98 [41], their implementations use a jump distance of 1 [44, 45]. Additionally, the noise factors e_1 and e_2 could be tuned. There are different default values used in the different DREAM publications. In the current implementation of QPuB, the noise factor e_2 was set to zero. This is not a good choice, since this constant ensures ergodicity of the chains, i.e. that the chains can potentially reach every part of the state space. The definition of dispersed starting values should make up for this mistake. Nevertheless, it should be corrected in the next generation implementation. The frequency of the occasional snooker update could be optimised. Preliminary results indicate that for simple examples the snooker update might even hamper the convergence speed. More recommendations on improving the snooker proposal are given in [42]. In higher dimensions, it can be beneficial not to update all parameter values at once, but rather only a subset. In what the authors call the "crossover" step, some parameters are chosen randomly to stay at their current position instead of being updated by their proposal candidate [41]. This procedure can further be optimised by a self-adaptive crossover step [43]. This step increases the variability in the proposal and should result in a more effective search of the state space resulting in faster convergence.

The **convergence** is currently accessed using the Gelman–Rubin criterion. However, in practice, we always set the iteration number to a large fixed value and judge convergence based on the graphical output. In our experience, the threshold of 1.2 recommended by Gelman and Rubin is too generous. We observed, that the posterior histograms are not converged to normal distributions yet and need to run longer. For automated convergence monitoring, a lower threshold would be advisable. However, an automated stopping criterion can be unreliable [150]. QPuB would benefit from consulting a second diagnostic. Vrugt et al. use a combination of different criteria to decide whether convergence has been achieved [44]. Apart from evaluating convergence, much research has also been done to accelerate convergence [150]. In general, the efficiency of QPuB should be improved. For simple examples it takes quite long until the chain is fully converged. The correct order of magnitude for every conversion factor is found relatively fast, but it takes many more iterations to refine the estimate. In [44], the authors list the number of samples needed for the DREAM algorithm to reliably infer the parameters of a "not too complicated" (quote [44]) posterior distribution, after burn-in has been subtracted. According to this list, for five to ten parameters only 5 000–10 000 iterations should be sufficient. This guideline is given for the DREAM algorithm, so for the more efficient DREAM$_{(\mathrm{ZS})}$ this should be even less. One reason for this under-performance is the large thinning parameter of QPuB, which was chosen to be 1 000 by mistake. The originators of the algorithm recommend a thinning rate of 10 [42, 45]. With a value too large, the dominance of the "fake" archive is overcome only after $1000 \cdot 10d$ iterations, which means 70 000 for Example 1. Most importantly, the thinning rate should only affect the convergence rate and not the final outcome, i.e. we expect the results of this thesis to be qualitatively the same. If the chains need too many iterations to converge, QPuB will run into memory issues. Since every iteration is saved to the chain object (4.15), it soon becomes a very large array, especially if many parameters are inferred

(Ex. 3 with 46 parameters and three chains: 60 MB). However, also the authors experience a rather low acceptance rate of the DREAM$_{(ZS)}$ algorithm for some examples [168].

If the above measures do not sufficiently improve the efficiency of the algorithm, a next-generation QPuB could employ a more advanced member of the DREAM family. The MT-DREAM$_{(ZS)}$ algorithm [168] is built on the former DREAM$_{(ZS)}$ with the additional feature of multi-try sampling [191]. The efficiency of the proposal step is increased drastically by proposing multiple candidates at once (e.g. five), of which the best is selected based on their posteriors and forwarded to the Metropolis acceptance step, which itself is modified in a complex fashion. The generation and evaluation of a high number of proposals is computationally demanding, but this is compensated by "spectacular performance" (quote [168]) due to distributed computing. The authors emphasise the algorithm's suitability for high-dimensional problems and apply MT-DREAM$_{(ZS)}$ to dimensions up to 241 parameters with fast convergence time as well as overall runtime.

6.4 Discussion of the general assumptions

A main assumption underlying the QPuB framework is the linearity of the calibration curve. Peptide amounts and the resulting measured signal response are related, but whether the relationship is a true linear one cannot be guaranteed. Physico-chemical properties of the peptides can influence their behaviour in the mass spectrometer and therewith the resulting signal intensities measured. Signal suppression by dominant peptides or saturation effects for large abundances can distort the linearity [37, 125]. The second assumption used is the principle of mass conservation. In the closed and controlled environment of *in vitro* analyses, theoretically the law of mass conservation holds. Every amino acid in the sample preparation must be preserved over the course of the experiment. With a complete peptide coverage, quantification of the digestion products should in theory be possible. However, in reality mass conservation cannot be guaranteed [37, 125, 192, 193]. There is a potential for systematic loss of mass at every step of the experimental pipeline. Even with well controlled experiments, careful sample handling and calibration of laboratory equipment, sample preparation can be error-prone due to environmental conditions or small variations. Traces of peptides will remain in the laboratory equipment like pipettes and tubes, as well as in the instrument, e.g. the chromatography column [personal communication]. With the choice of high-resolution instruments and optimisation of the MS process, uncertainty can be further limited but not excluded. The nature of a peptide influences its ionisation and transportability through the mass spectrometer. Fragmentation efficiency and resolution of the instrument play a role. Peptides with strong signals can lead to detector saturation that masks peptides of lower intensity. Short peptides of length smaller than three amino acids usually escape detection. The Fourier transform required in modern orbitrap devices can also be a source of error. Even after careful calibration, random noise of electrical origin and systematic effects in the measurements remain, like background noise and other interfering signals of contaminants [192, 194, 28]. After the measurements, the postprocessing steps can further introduce uncertainty. Data processing can lead to errors, especially when not performed in an automated fashion. The most important part in the MS workflow is the peak assignment. It is a complex procedure that depends on many factors. Filtering and smoothing steps can be performed to facilitate effective feature detection [192]. Reliable peptide identification depends on the method used, the quality and quantity of the database searched and the false discovery rates [192]. In this step, even large peptides can remain unidentified because of the mass range or modifications [39]. Subsequent data filtering steps can further decrease the number of peptides in a dataset. In summary, many factors in the MS-based peptide identification pipeline can lead to a violation of mass conservation [37, 125, 192, 193]. The impact of missing products on the peptide quantification using QPuB needs further investigation. However, with the advancement in mass spectrometry technology leading to increasing precision and sensitivity, both the assumptions of mass conservation and linearity become more and more justified.

6.5 Applicability of QPuB

QPuB is an easy to use software package. It is independent of the operating system and only requires the installation of the R software environment and relies on a few R packages. Currently, the QPuB pipeline is started using a single command line from the terminal. However, it can also be run from within R or RStudio. Multiple runs can be started using convenient bash scripts. The program comes with a lot of default settings for the layman user. The only mandatory input are CSV files with the signal intensities of the substrate and products in replicates and the peptide sequences. To calculate absolute peptide amounts, the substrate titration data and the amount of substrate loaded in the digestion must be specified. Otherwise only normalised signal intensities are returned, which allow for relative quantification between peptides. QPuB does not require any other specific settings, especially no instrument specific settings that have to be laboriously calibrated prior to application, as is the case for its predecessor QME by Mishto et al. [39]. To change the default settings, QPuB comes with an input text file, where output preferences and algorithm parameters can be specified. The advanced user has the opportunity to customise the code, as modules can easily be adjusted. Thanks to its flexible implementation, QPuB has the potential to become a universal tool which can be applied to any *in vitro* enzymatic digest. The maximum number of peptide products that could be analysed depends on the algorithms ability for high-dimensional parameter inference and the computational hardware available. With the advancement in mass spectrometry hardware, the number of products measured will increase steadily. For common laboratory techniques, absolute quantification of many peptides becomes expensive and laborious [117]. A well-designed computational approach could keep the experimental work to a minimum and indicate which experimental measurements would be most beneficial to obtain the information desired. The computational cost depends on the hardware and inferential method used. With the advancement in computer technology, algorithms can become more efficient, saving resources and runtime. QME in its current implementation struggles with high-dimensional data [personal communication]. For large numbers of peptides in a digest, label-free quantification can be applied. Unfortunately it only allows for relative quantification between the same peptides in different samples [21]. With QPuB it is possible to compare relative abundances of different peptides, without additional experimental effort. With minimal effort in the form of a substrate titration, QPuB brings the great advantage of easy absolute quantification. Another advantage compared to QME is that QPuB returns a full posterior distribution of possible parameter values instead of a single point estimate. The conversion factor distribution can be summarised into a single value and the spread of the distribution gives an estimate on the uncertainty in the data. This is particularly useful for noisy data or when the amount of data is limited. In addition, the current implementation of QME struggles to find global solutions and might get stuck in a local optimum [personal communication]. The DREAM$_{(ZS)}$ algorithm promises to be robust to that [43]. In summary, the QPuB framework offers an easy and reliable approach to relative and absolute quantification with minimal experimental effort.

6.6 The greater picture

Proteases are more than non-specific degradative enzymes. Through regulation of protein activity they control biological processes from the control of the cell cycle progression to the activation of the immune system [195, 196]. Analysing the dynamics of the degradome, i.e. the proteases, their substrates and inhibitors [197, 195], as well as the resulting peptidome is important to understand their role in the cell. Identification and quantification of the peptide pool generated by the proteases plays a major part in understanding the protease's cleavage preferences and how it is affected by various factors. Inside cells, proteases are usually not acting in isolation and much more information is contained in the greater system they are part of. From protein synthesis to degradation, subject to activation and inhibition, following spatial and temporal distributions, proteases are highly dynamic. Many proteases can digest a variety of substrates and a complex connection and interaction between different types of proteins results in a hierarchical synergy that asks to be understood [195]. Abundances of target proteins of proteases and their degradation products play a crucial part in this process. Quantitative information provided by QPuB can be useful in these studies.

Quantitative modelling provides a way to develop a deeper understanding of small scale processes like molecular mechanisms of individual processes up to their influence on large-scale cellular networks [198, 199]. A prominent example is the MHC I antigen processing and presentation pathway. A network of different proteases, transporters and assistant protein complexes acts together as part of the adaptive immune system. Antigens in the cell plasma are digested by the proteasome and trimmed by aminopeptidases into smaller fragments. These are transported to the endoplasmic reticulum, where they are further degraded by ERAP to fit the binding groove of the major histocompatibility complex I. This complex is finally shuttled to the cell surface via the common secretory pathway, where T cells recognise the load and potentially initiate an immune response [200, 201, 202, 203, 79, 204, 4]. A deeper understanding of this pathway can be gained through the development of models [205, 206, 38, 207, 208, 209, 210, 211, 212, 213, 214]. Quantitative modelling would not only allow to understand the specificity and turn-over of the proteases involved but also potentially predict which of the possible MHC I bound peptides – both non-spliced and spliced – will be produced by the proteasome and presented on the cell surface [215]. The absolute amount of peptides presented on MHC 1 is one of the crucial factors for T cell activation. QPuB could help to analyse the dynamics of the proteases *in vitro* [216].

As mentioned in Sec. 2.2.2, proteasomes are not only able to catalyse peptide hydrolysis but are also involved in transpeptidation that creates spliced peptides. This was an interesting discovery [73] and caused many groups to investigate the frequencies of spliced peptides and their biological relevance [20, 217, 218, 219, 220, 221, 222, 223, 224]. QPuB could provide a layer of quantitative information on spliced peptides that could advance this area of research. Another aspect in this context is the proteasome specificity for cleavage and splice sites along a substrate sequence. It was previously demonstrated, that the different catalytically active β subunits of the 20S proteasome and its isoforms exhibit differences in quantities of the generated peptide products. A quantitative investigation of the proteasome dynamics in the presence of inhibitors for the specific subunits could elucidate this [225, 209].

The small size and high specificity of peptides makes them ideal targets for various medical treatments against food allergies, diabetes and cancer, among others [5, 6]. In immunotherapy, mutation-derived peptide vaccines can be used to trigger a T cell response [226, 227, 228, 229]. T cell transfer therapy aims at multiplying T cells specific to tumor-associated antigens [230, 231]. These and other approaches require the identification of potential peptide candidates that are efficiently produced by the proteasome [232, 233]. Quantification of peptides in an *in vitro* digestion via QPuB could largely facilitate this endeavour [118].

7 | Conclusions

Quantification of peptide products is an important step on the way to understanding protease specificities and dynamics. However, measurements obtained by mass spectrometry of *in vitro* digests only provide relative quantification of same peptides in different samples. In order to obtain absolute quantities, laboratory techniques involving synthetic peptides have to be applied, which can become expensive and laborious for large peptide mixtures. In this thesis, we proposed a label-free quantification pipeline called QPuB, Quantification of Peptides using Bayesian inference, which computationally infers the peptide amounts from the MS signal intensities with minimal experimental effort. Based on the principle of mass balance and the linear relationship between the peptide amounts and the corresponding MS signal intensities, a conversion factor for every peptide is estimated, that allows to calculate normalised signal intensities which can be used for relative quantification between different peptides in a sample. If in addition the substrate titration is provided, then absolute quantities of all peptides generated can be obtained. For this, QPuB employs Bayesian inference with an underlying Differential Evolution Markov chain algorithm, which returns full posterior distributions for the conversion factors. These provide a best guess about the conversion factors and also represent the uncertainty of the estimate. With these distributions, the concentration kinetics for each individual peptide can be calculated. In its current implementation, QPuB is able to reliably infer the correct parameter values for a variety of noise-free synthetic datasets that satisfy the mass balance condition. If the measured data provided is not informative enough and does not sufficiently constrain the parameters, then more information can be provided in the form of additional peptide titrations or more signal kinetics measured under different biological conditions. Once QPuB has been validated on experimental data, we believe that it has the potential to add a valuable alternative to the pool of quantification techniques.

Bibliography

[Citing pages are listed after each reference.]

[1] Harold Jeffreys. Theory of probability. Vol. 2, 1961. [Cited on page 3]

[2] Iman Tavassoly, Joseph Goldfarb, and Ravi Iyengar. Systems biology primer: the basic methods and approaches. *Essays in Biochemistry*, 62(4):487–500, 2018. [Cited on page 1]

[3] Jeremy M. Berg, John L. Tymoczko, and Lubert Stryer. *Biochemie.* Spektrum Akademischer Verlag, 2009. [Cited on pages 1, 5, 6, and 9]

[4] Novalia Pishesha, Thibault J. Harmand, and Hidde L. Ploegh. A guide to antigen processing and presentation. *Nature Reviews Immunology*, 2022. [Cited on pages 1 and 102]

[5] Miloš Erak, Kathrin Bellmann-Sickert, Sylvia Els-Heindl, and Annette G. Beck-Sickinger. Peptide chemistry toolbox – transforming natural peptides into peptide therapeutics. *Bioorganic & Medicinal Chemistry*, 26:2759–2765, Jan 2018. [Cited on pages 1 and 102]

[6] Pablo Scodeller and Eliana K. Asciutto. Targeting tumors using peptides. *Molecules*, 25(4):808, Feb 2020. [Cited on pages 1 and 102]

[7] Chloe Bleuez, Wolfgang F. Koch, Carole Urbach, Florian Hollfelder, and Lutz Jermutus. Exploiting protease activation for therapy. *Drug Discovery Today*, 27(6):1743–1754, Jun 2022. [Cited on page 1]

[8] Jeffrey C. Silva, Marc V. Gorenstein, Guo-Zhong Li, Johannes P.C. Vissers, and Scott J. Geromanos. Absolute quantification of proteins by LCMSE. *Molecular & Cellular Proteomics*, 5(1):144–156, Jan 2006. [Cited on pages 1 and 2]

[9] C. Kreutz, M.M. Bartolome Rodriguez, T. Maiwald, M. Seidl, H.E. Blum, L. Mohr, and J. Timmer. An error model for protein quantification. *Bioinformatics*, 23(20):2747–2753, Sep 2007. [Cited on page 1]

[10] Ankit Sinha and Matthias Mann. A beginner's guide to mass spectrometry–based proteomics. *The Biochemist*, 42(5):64–69, 2020. [Cited on pages 2, 7, 11, 12, and 13]

[11] Liang Tang and Paul Kebarle. Dependence of ion intensity in electrospray mass spectrometry on the concentration of the analytes in the electrosprayed solution. *Analytical Chemistry*, 65:3654–3668, 1993. [Cited on pages 2 and 14]

[12] Randy W. Purves, Wojciech Gabryelski, and Liang Li. Investigation of the quantitative capabilities of an electrospray ionization ion trap/linear time-of-flight mass spectrometer. *Rapid communications in mass spectrometry*, 12:695–700, 1998. [Cited on pages 2 and 14]

[13] Robert D. Voyksner and Heewon Lee. Investigating the use of an octupole ion guide for ion storage and high-pass mass filtering to improve the quantitative performance of electrospray ion trap mass spectrometry. *Rapid communications in mass spectrometry*, 13:1427–1437, 1999. [Cited on pages 2 and 14]

[14] Dirk Chelius and Pavel V. Bondarenko. Quantitative profiling of proteins in complex mixtures using liquid chromatography and mass spectrometry. *Journal of Proteome Research*, 1(4):317–323, 2002. [Cited on pages 2 and 14]

[15] Pavel V. Bondarenko, Dirk Chelius, and Thomas A. Shaler. Identification and relative quantitation of protein mixtures by enzymatic digestion followed by capillary reversed-phase liquid chromatography- tandem mass spectrometry. *Analytical Chemistry*, 74(18):4741–4749, 2002. [Cited on pages 2 and 14]

[16] Stephen J. Callister, Richard C. Barry, Joshua N. Adkins, Ethan T. Johnson, Weijun Qian, Bobbie-Jo M. Webb-Robertson, Richard D. Smith, and Mary S. Lipton. Normalization approaches for removing systematic biases associated with mass spectrometry and label-free proteomics. *Journal of Proteome Research*, 5(2):277–286, Feb 2006. [Cited on pages 2 and 14]

[17] Jürgen Cox, Marco Y. Hein, Christian A. Luber, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Molecular & Cellular Proteomics*, 13(9):2513–2526, Sep 2014. [Cited on pages 2 and 15]

[18] Erik L. de Graaf, Piero Giansanti, A.F. Maarten Altelaar, and Albert J.R. Heck. Single-step enrichment by Ti4$^+$-IMAC and label-free quantitation enables in-depth monitoring of phosphorylation dynamics with high reproducibility and temporal resolution. *Molecular & Cellular Proteomics*, 13(9):2426–2434, Sep 2014. [Cited on page 2]

[19] Michal Bassani-Sternberg, Sune Pletscher-Frankild, Lars Juhl Jensen, and Matthias Mann. Mass spectrometry of human leukocyte antigen class i peptidomes reveals strong effects of protein abundance and turnover on antigen presentation. *Molecular & Cellular Proteomics*, 14(3):658–673, Mar 2015. [Cited on page 2]

[20] Juliane Liepe, Fabio Marino, John Sidney, Anita Jeko, Daniel E. Bunting, Alessandro Sette, Peter M. Kloetzel, Michael P. H. Stumpf, Albert J. R. Heck, and Michele Mishto. A large fraction of HLA class I ligands are proteasome-generated spliced peptides. *Science*, 354(6310):354–358, Oct 2016. [Cited on pages 2, 11, and 102]

[21] Marcus Bantscheff, Simone Lemeer, Mikhail M. Savitski, and Bernhard Kuster. Quantitative mass spectrometry in proteomics: critical review update from 2007 to the present. *Anal Bioanal Chem*, 404(4):939–65, Sep 2012. [Cited on pages 2, 14, 15, and 101]

[22] Michael J. MacCoss, Christine C. Wu, Hongbin Liu, Rovchan Sadygov, and John R. Yates III. A correlation algorithm for the automated quantitative analysis of shotgun proteomics data. *Analytical Chemistry*, 75:6912–6921, 2003. [Cited on page 2]

[23] Claire E. Eyers, Craig Lawless, David C. Wedge, King Wai Lau, Simon J. Gaskell, and Simon J. Hubbard. CONSeQuence: Prediction of reference peptides for absolute quantitative proteomics using consensus machine learning approaches. *Molecular & Cellular Proteomics*, 10(11):M110.003384, Nov 2011. [Cited on page 2]

[24] Hongbin Liu, Rovchan Sadygov, and John R. Yates III. A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical Chemistry*, 76:4193–4201, 2004. [Cited on pages 2 and 15]

[25] Seungmook Lee, Min-Seok Kwon, Hyoung-Joo Lee, Young-Ki Paik, Haixu Tang, Jae K Lee, and Taesung Park. Enhanced peptide quantification using spectral count clustering and cluster abundance. *BMC Bioinformatics*, 12(423), 2011. [Cited on pages 2 and 15]

[26] Christine Vogel and Edward M. Marcotte. Label-free protein quantitation using weighted spectral counting. *Methods Mol Biol*, 893:321–41, 2012. [Cited on page 2]

[27] Sarah Gerster, Taejoon Kwon, Christina Ludwig, Mariette Matondo, Christine Vogel, Edward M. Marcotte, Ruedi Aebersold, and Peter Bühlmann. Statistical approach to protein quantification. *Molecular & Cellular Proteomics*, 13(2):666–677, Feb 2014. [Cited on page 2]

[28] Cheng Chang, Jiyang Zhang, Changming Xu, Yan Zhao, Jie Ma, Tao Chen, Fuchu He, Hongwei Xie, and Yunping Zhu. Quantitative and in-depth survey of the isotopic abundance distribution errors in shotgun proteomics. *Analytical Chemistry*, 88(13):6844–6851, Jun 2016. [Cited on pages 2, 15, and 100]

[29] Cheng Chang, Zhiqiang Gao, Wantao Ying, Yan Fu, Yan Zhao, Songfeng Wu, Mengjie Li, Guibin Wang, Xiaohong Qian, Yunping Zhu, and Fuchu He. LFAQ: Toward unbiased label-free absolute protein quantification by predicting peptide quantitative factors. *Anal Chem*, 91(2):1335–1343, Jan 2019. [Cited on page 2]

[30] Kenneth R. Chalcraft, Richard Lee, Casandra Mills, and Philip Britz-McKibbin. Virtual quantification of metabolites by capillary electrophoresis-electrospray ionization-mass spectrometry: Predicting ionization efficiency without chemical standards. *Analytical Chemistry*, 81:2506–2515, 2009. [Cited on page 2]

[31] M. A. Raji, P. Fryčák, C. Temiyasathit, S. B. Kim, G. Mavromaras, J.-M. Ahn, and K. A. Schug. Using multivariate statistical methods to model the electrospray ionization response of GXG tripeptides based on multiple physicochemical parameters. *Rapid Communications in Mass Spectrometry*, 23(14):2221–2232, Jun 2009. [Cited on page 2]

[32] Varvara J. Mandra, Maria G. Kouskoura, and Catherine K. Markopoulou. Using the partial least squares method to model the electrospray ionization response produced by small pharmaceutical molecules in positive mode. *Rapid Communications in Mass Spectrometry*, 29(18):1661–1675, Aug 2015. [Cited on page 2]

[33] Christopher J. Cramer, Joshua L. Johnson, and Amin M. Kamel. Prediction of mass spectral response factors from predicted chemometric data for druglike molecules. *Journal of the American Society for Mass Spectrometry*, 28(2):278–285, Nov 2016. [Cited on page 2]

[34] Joseph H.A. Guillaume, John D. Jakeman, Stefano Marsili-Libelli, Michael Asher, Philip Brunner, Barry Croke, Mary C. Hill, Anthony J. Jakeman, Karel J. Keesman, Saman Razavi, and Johannes D. Stigter. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environmental Modelling & Software*, 119:418–432, 2019. [Cited on pages 2, 33, 34, 35, and 36]

[35] Andreas Raue, Clemens Kreutz, Fabian Joachim Theis, and Jens Timmer. Joining forces of bayesian and frequentist methodology: a study for inference in the presence of non-identifiability. *Phil. Trans. R. Soc. A.*, 371:20110544, 2013. [Cited on pages 2, 19, 20, 24, 35, and 36]

[36] Oana-Teodora Chis, Alejandro F. Villaverde, Julio R. Banga, and Eva Balsa-Canto. On the relationship between sloppiness and identifiability. *Mathematical Biosciences*, 282:147–161, 2016. [Cited on page 2]

[37] Björn Peters, Katharina Janek, Ulrike Kuckelkorn, and Hermann-Georg Holzhütter. Assessment of proteasomal cleavage probabilities from kinetic analysis of time-dependent product formation. *Journal of Molecular Biology*, 318(3):847–62, May 2002. [Cited on pages 2, 3, 41, 44, 96, 99, and 100]

[38] Michele Mishto, Fabio Luciani, Hermann-Georg Holzhütter, Elena Bellavista, Aurelia Santoro, Kathrin Textoris-Taube, Claudio Franceschi, Peter M. Kloetzel, and Alexey Zaikin. Modeling the in vitro 20S proteasome activity: The effect of PA28−$\alpha\beta$ and of the sequence and length of polypeptides on the degradation kinetics. *Journal of Molecular Biology*, 377(5):1607–1617, Apr 2008. [Cited on pages 3 and 102]

[39] Michele Mishto, Andrean Goede, Kathrin Textoris-Taube, Christin Keller, Katharina Janek, Petra Henklein, Agathe Niewienda, Alexander Kloss, Sabrina Gohlke, Burkhardt Dahlmann, Cordula Enenkel, and Peter M. Kloetzel. Driving forces of proteasome-catalyzed peptide splicing in yeast and humans. *Molecular & Cellular Proteomics*, 11(10):1008–1023, Oct 2012. [Cited on pages 3, 8, 16, 41, and 100]

[40] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. Bayesian data analysis (with errors fixed as of 15 february 2021), 2020. [Cited on pages 4, 17, 18, 20, 21, 27, and 28]

[41] Cajo J. F. ter Braak. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16(3):239–249, Sep 2006. [Cited on pages 4, 29, 30, 31, 32, 48, 97, and 98]

[42] Cajo J. F. ter Braak and Jasper A. Vrugt. Differential evolution Markov chain with snooker updater and fewer chains. *Statistics and Computing*, 18(4):435–446, Oct 2008. [Cited on pages 4, 29, 30, 32, 48, 97, and 98]

[43] Jasper A. Vrugt, Cajo J. F. ter Braak, Cees G. H. Diks, Bruce A. Robinson, James M. Hyman, and Dave Higdon. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences & Numerical Simulation*, 10(3):271–288, Mar 2009. [Cited on pages 4, 28, 29, 30, 31, 32, 48, 97, 98, and 101]

[44] Jasper A. Vrugt. Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software*, 75:273–316, Jan 2016. [Cited on pages 4, 26, 27, 28, 29, 30, 31, 32, 48, 97, and 98]

[45] Erin M. Shockley, Jasper A. Vrugt, and Carlos F. Lopez. PyDREAM: high-dimensional parameter inference for biological models in python. *Bioinformatics*, 34(4):695–697, Oct 2017. [Cited on pages 4, 25, 26, 28, 29, 30, 31, 32, 48, 97, and 98]

[46] Francis H.C. Crick. On protein synthesis. *Symposia of the Society for Experimental Biology*, 12:138–163, 1958. [Cited on page 5]

[47] Valerie C. Wasinger, Stuart J. Cordwell, Anne Cerpa-Poljak, Jun X. Yan, Andrew A. Gooley, Marc R. Wilkins, Mark W. Duncan, Ray Harris, Keith L. Williams, and Ian Humphery-Smith. Progress with gene-product mapping of the mollicutes: Mycoplasma genitalium. *Electrophoresis*, 16:1090–1094, 1995. [Cited on page 5]

[48] Marc R. Wilkins, Jean-Charles Sanchez, Andrew A. Gooley, Ron D. Appel, Ian Humphery-Smith, Denis F. Hochstrasser, and Keith L. Williams. Progress with proteome projects: Why all proteins expressed by a genome should be identified and how to do it. *Biotechnology and Genetic Engineering Reviews*, 13(1):19–50, 1995. [Cited on page 5]

[49] Marc R. Wilkins, Christian Pasquali, Ron D. Appel, Keli Ou, Olivier Golar, Jean-Charles Sanchez, Jun X. Yan, Andrew. A. Gooley, Graham Hughes, Ian Humphery-Smith, Keith L. Wllliams, and Denis F. Hochstrasser. From proteins to proteomes: Large scale protein identification by two-dimensional electrophoresis and amino acid analysis. *Nature Biotechnology*, 14:61–65, Jan 1996. [Cited on page 5]

[50] Steven L. Salzberg. Open questions: How many genes do we have? *BMC Biology*, 16(94), 2018. [Cited on page 5]

[51] Cassandra Willyard. Expanded human gene tally reignites debate. *Nature*, 558:354–355, Jun 2018. [Cited on page 5]

[52] Ole Nørregaard Jensen. Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry. *Curr Opin Chem Biol*, 8(1):33–41, Feb 2004. [Cited on page 5]

[53] Adrian P. Bird. Gene number, noise reduction and biological complexity. *Trends in Genetics*, 11(3):94–100, Mar 1995. [Cited on page 5]

[54] Albert J.R. Heck and Jeroen Krijgsveld. Mass spectrometry-based quantitative proteomics. *Expert Review of Proteomics*, 1(3):317–326, Oct 2004. [Cited on page 5]

[55] Martin Beck, Alexander Schmidt, Johan Malmstroem, Manfred Claassen, Alessandro Ori, Anna Szymborska, Franz Herzog, Oliver Rinner, Jan Ellenberg, and Ruedi Aebersold. The quantitative proteome of a human cell line. *Molecular Systems Biology*, 7(1):549, Jan 2011. [Cited on page 5]

[56] Jürgen Cox and Matthias Mann. Quantitative, high-resolution proteomics for data-driven systems biology. *Annual Review of Biochemistry*, 80:273–299, 2011. [Cited on page 5]

[57] Michael Kinter and Nicholas E. Sherman. *Protein sequencing and identification using tandem mass spectrometry.* Wiley-Interscience Series on Mass Spectrometry. John Wiley & Sons, Inc., 2000. [Cited on pages 5, 6, and 10]

[58] Ruedi Aebersold and Matthias Mann. Mass-spectrometric exploration of proteome structure and function. *Nature*, 537(7620):347–355, Sep 2016. [Cited on page 6]

[59] Anna Radzicka and Richard Wolfenden. Rates of uncatalyzed peptide bond hydrolysis in neutral solution and the transition state affinities of proteases. *Journal of the American Chemical Society*, 118(26):6105–6109, 1996. [Cited on page 6]

[60] Liana Tsiatsiani and Albert J.R. Heck. Proteomics beyond trypsin. *The FEBS journal*, 282(14):2612–2626, 2015. [Cited on page 7]

[61] W. Kühne. über das trypsin. *Verhandlungen des Heidelberger Naturhistorischen und Medizinischen Vereins*, 1877. [Cited on page 7]

[62] Michael Schrader, Peter Schulz-Knappe, and Lloyd D. Fricker. Historical perspective of peptidomics. *EuPA Open Proteomics*, 3:171–182, 2014. [Cited on page 7]

[63] Gerd Specht, Hanna P. Roetschke, Artem Mansurkhodzhaev, Petra Henklein, Kathrin Textoris-Taube, Henning Urlaub, Michele Mishto, and Juliane Liepe. Large database for the analysis and prediction of spliced and non-spliced peptide generation by proteasomes. *Scientific Data*, 7(1):1–12, 2020. [Cited on page 7]

[64] Joseph D. Etlinger and Alfred L. Goldberg. A soluble ATP-dependent proteolytic system responsible for the degradation of abnormal proteins in reticulocytes. *Proceedings of the National Academy of Sciences of the United States of America*, 74(1):54–58, Jan 1977. [Cited on page 7]

[65] Ahron Ciehanover, Yaacov Hod, and Avram Hershko. A heat-stable polypeptide component of an ATP-dependent proteolytic system from reticulocytes. *Biochemical and Biophysical Research Commununications*, 81(4):1100–1105, Apr 1978. [Cited on page 7]

[66] Lars Thelander (Member of the Nobel Committee for Chemistry). Advanced information on the Nobel Prize in chemistry 2004 - ubiquitin-mediated proteolysis, 2004. [Cited on pages 7 and 8]

[67] Jan Löwe, Daniela Stock, Bing Jap, Peter Zwickl, Wolfgang Baumeister, and Robert Huber. Crystal structure of the 20S proteasome from the archaeon T. acidophilum at 3.4å resolution. *Science*, 268(5210):533–539, Apr 1995. [Cited on page 7]

[68] D. Voges, P. Zwickl, and W. Baumeister. The 26S proteasome: A molecular machine designed for controlled proteolysis. *Annual Review of Biochemistry*, 68:1015–1068, 1999. [Cited on page 7]

[69] Indrajit Sahu and Michael H. Glickman. Proteasome in action: substrate degradation by the 26S proteasome. *Biochemical Society Transactions*, 49:629–644, 2021. [Cited on pages 7 and 8]

[70] Peter Zwickl, Dieter Voges, and Wolfgang Baumeister. The proteasome: a macromolecular assembly designed for controlled proteolysis. *Philosophical Transactions of the Royal Society of London*, 354:1501–1511, 1999. [Cited on page 7]

[71] Indrajit Sahu and Michael H. Glickman. Structural insights into substrate recognition and processing by the 20S proteasome. *Biomolecules*, 11(148), 2021. [Cited on page 8]

[72] Gili Ben-Nissan and Michal Sharon. Regulating the 20S proteasome ubiquitin-independent degradation pathway. *Biomolecules*, 4(3):862–84, Sep 2014. [Cited on page 8]

[73] Nathalie Vigneron, Vincent Stroobant, Jacques Chapiro, Annie Ooms, Gérard Degiovanni, Sandra Morel, Pierre van der Bruggen, Thierry Boon, and Benoît J. Van den Eynde. An antigenic peptide produced by peptide splicing in the proteasome. *Science*, 304(5670):587–590, 2004. [Cited on pages 8 and 102]

[74] Edus H. Warren, Nathalie J. Vigneron, Marc A. Gavin, Pierre G. Coulie, Vincent Stroobant, Alexandre Dalet, Scott S. Tykodi, Suzanne M. Xuereb, Jeffrey K. Mito, Stanley R. Riddell, and Benoît J. van den Eynde. An antigen produced by splicing of noncontiguous peptides in the reverse order. *Science*, 313(5792):1444–1447, 2006. [Cited on page 8]

[75] Alexandre Dalet, Nathalie Vigneron, Vincent Stroobant, Ken ichi Hanada, and Benoît J. van den Eynde. Splicing of distant peptide fragments occurs in the proteasome by transpeptidation and produces the spliced antigenic peptide derived from fibroblast growth factor-5. *The Journal of Immunology*, 184(6):3016–3024, Feb 2010. [Cited on page 8]

[76] Alexandre Michaux, Pierre Larrieu, Vincent Stroobant, Jean-François Fonteneau, Francine Jotereau, Benoît J. van den Eynde, Agnès Moreau-Aubry, and Nathalie Vigneron. A spliced antigenic peptide comprising a single spliced amino acid is produced in the proteasome by reverse splicing of a longer peptide fragment followed by trimming. *The Journal of Immunology*, 192(4):1962–1971, Jan 2014. [Cited on page 8]

[77] Michele Mishto, Muhammad L. Raza, Dario de Biase, Teresa Ravizza, Francesco Vasuri, Morena Martucci, Christin Keller, Elena Bellavista, Tonia J. Buchholz, Peter M. Kloetzel, Annalisa Pession, Annamaria Vezzani, and Uwe Heinemann. The immunoproteasome β5i subunit is a key contributor to ictogenesis in a rat model of chronic epilepsy. *Brain, Behavior, and Immunity*, 49:188–196, May 2015. [Cited on page 8]

[78] Ken ichi Hanada, Jonathan W. Yewdell, and James C. Yang. Immune recognition of a human renal cancer antigen through post-translational protein splicing. *Nature*, 427:252–256, 2004. [Cited on page 8]

[79] Michele Mishto and Juliane Liepe. Post-translational peptide splicing and T cell responses. *Trends in Immunology*, 38(12):904–915, Dec 2017. [Cited on pages 8 and 102]

[80] F. Ebstein, K. Textoris-Taube, C. Keller, R. Golnik, N. Vigneron, B. J. van den Eynde, B. Schuler-Thurner, D. Schadendorf, F. K. M. Lorenz, W. Uckert, S. Urban, A. Lehmann, N. Albrecht-Koepke, K. Janek, P. Henklein, A. Niewienda, P. M. Kloetzel, and M. Mishto. Proteasomes generate spliced epitopes by two different mechanisms and as efficiently as non-spliced epitopes. *Scientific Reports*, 6(1), Apr 2016. [Cited on page 8]

[81] Celia R. Berkers, Annemieke de Jong, Karianne G. Schuurman, Carsten Linnemann, Hugo D. Meiring, Lennert Janssen, Jacques J. Neefjes, Ton N. M. Schumacher, Boris Rodenko, and Huib Ovaa. Definition of proteasomal peptide splicing rules for high-efficiency spliced peptide presentation by MHC class I molecules. *The Journal of Immunology*, 195(9):4085–4095, Sep 2015. [Cited on page 8]

[82] William J. Lennarz and M. Daniel Lane. *Encyclopedia of Biological Chemistry*. Academic Press, 2013. [Cited on page 8]

[83] Victor Henri. *Lois générales de l'action des diastases.* Librairie Scientifique A. Hermann, 1903. [Cited on page 8]

[84] Leonor Michaelis and Maud L. Menten. Die kinetik der invertinwirkung. *Biochem. z*, 49(333-369):352, 1913. [Cited on page 8]

[85] George Edward Briggs and John Burdon Sanderson Haldane. A note on the kinetics of enzyme action. *Biochemical journal*, 19(2):338, 1925. [Cited on page 8]

[86] Athel Cornish-Bowden. One hundred years of Michaelis–Menten kinetics. *Perspectives in Science*, 4:3–9, 2015. [Cited on page 8]

[87] William W. Chen, Mario Niepel, and Peter K. Sorger. Classic and contemporary approaches to modeling biochemical reactions. *Genes Dev*, 24(17):1861–75, Sep 2010. [Cited on pages 9, 10, 34, 37, and 51]

[88] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003. [Cited on page 10]

[89] Rachel E. Foreman, Amy L. George, Frank Reimann, Fiona M. Gribble, and Richard G. Kay. Peptidomics: A review of clinical applications and methodologies. *Journal of Proteome Research*, 20(8):3782–3797, 2021. [Cited on page 10]

[90] Pehr Edman, Erik Högfeldt, Lars Gunnar Sillén, and Per-Olof Kinell. Method for determination of the amino acid sequence in peptides. *Acta Chemica Scandinavica*, 4(7):283–293, 1950. [Cited on page 10]

[91] Pehr Edman and Geoffrey Begg. A protein sequenator. In *European Journal of Biochemistry*, pages 80–91. Springer, 1967. [Cited on page 10]

[92] Emmalyn J. Dupree, Madhuri Jayathirtha, Hannah Yorkey, Marius Mihasan, Brindusa Alina Petre, and Costel C. Darie. A critical review of bottom-up proteomics: The good, the bad, and the future of this field. *Proteomes*, 8(3):14, 2020. [Cited on pages 11 and 12]

[93] Joseph J. Thomson. On the appearance of helium and neon in vacuum tubes. *Science*, 37(1):360–364, Jan–Jun 1913. [Cited on page 11]

[94] Joseph J. Thomson. Bakerian lecture: Rays of positive electricity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 89(607):1–20, 1913. [Cited on page 11]

[95] Francis W. Aston. A positive ray spectrograph. *Philos. Mag.*, 38(228):707–714, Dec 1919. [Cited on page 11]

[96] Francis W. Aston. Neon. *Nature*, 104(2613):334–334, 1919. [Cited on page 11]

[97] Francis W. Aston. The constitution of the elements. *Nature*, 104(2616):393–393, 1919. [Cited on page 11]

[98] Francis W. Aston. Isotopes and atomic weights. *Nature*, 105(2646):617–619, 1920. [Cited on page 11]

[99] Francis W. Aston. Some problems of the mass-spectrograph. *Philosophical Magazine*, 43(255):514–528, March 1922. [Cited on page 11]

[100] Francis W. Aston. Bakerian lecture - a new mass-spectrograph and the whole number rule. *Proc. R. soc. Lond. Ser. A-Contain. Pap. Math. Phys. Character*, 115(772):487–U8, Aug 1927. [Cited on page 11]

[101] Matthias Mann, Ronald C. Hendrickson, and Akhilesh Pandey. Analysis of proteins and proteomes by mass spectrometry. *Annual Review of Biochemistry*, 70(1):437–473, 2001. [Cited on page 11]

[102] Michael A. Baldwin and Fred W. McLafferty. Liquid Chromatography-Mass Spectrometry interface – I: The direct introduction of liquid solutions into a chemical ionization mass spectrometer. *Organic Mass Spectrometry*, 7:1111–1112, May 1973. [Cited on page 12]

[103] Malcolm Dole, L. L. Mack, R. L. Hines, R. C. Mobley, L. D. Ferguson, and M. B. Alice. Molecular beams of macroions. *Journal of Chemical Physics*, 49(5):2240–2249, Sep 1968. [Cited on page 12]

[104] Masamichi Yamashita and John B. Fenn. Electrospray ion source. another variation on the free-jet theme. *Journal Physical Chemistry*, 88(20):4451–4459, 1984. [Cited on page 12]

[105] John B. Fenn, Matthias Mann, Chin Kai Meng, Shek Fu Wong, and Craig M. Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246:64–71, 1989. [Cited on page 12]

[106] Karin Markides and Astrid Gräslund. Mass spectrometry (MS) and nuclear magnetic resonance (NMR) applied to biological macromolecules. Advanced information on the Nobel Prize in Chemistry 2002, Oct 2002. [Cited on page 12]

[107] Alexander Makarov. Electrostatic axially harmonic orbital trapping: A high-performance technique of mass analysis. *Analytical Chemistry*, 72(6):1156–1162, Mar 2000. [Cited on page 12]

[108] Qizhi Hu, Robert J. Noll, Hongyan Li, Alexander Makarov, Mark Hardman, and R. Graham Cooks. The orbitrap: a new mass spectrometer. *Journal of Mass Spectrometry*, 40(4):430–443, 2005. [Cited on page 12]

[109] Jesper V. Olsen, Boris Macek, Oliver Lange, Alexander Makarov, Stevan Horning, and Matthias Mann. Higher-energy C-trap dissociation for peptide modification analysis. *Nat Methods*, 4(9):709–12, Sep 2007. [Cited on page 12]

[110] Darryl J. C. Pappin, Peter Hojrup, and Alan J. Bleasby. Rapid identification of proteins by peptide-mass fingerprinting. *Current Biology*, 3(6):327–332, 1993. [Cited on page 13]

[111] David N. Perkins, Darryl J. C. Pappin, David M. Creasy, and John S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551–3567, 1999. [Cited on page 13]

[112] Jimmy K. Eng, Ashley L. McCormack, and John R. Yates, III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5:976–989, Jun 1994. [Cited on page 13]

[113] Jürgen Cox and Matthias Mann. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26(12):1367–72, Dec 2008. [Cited on page 13]

[114] C. Bartels. Fast algorithm for peptide sequencing by mass spectrometry. *Biomed. Environ. Mass. Spectrom*, 19(363-368):36, 1990. [Cited on page 13]

[115] Jing Zhang, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng Zhang, Gilles A Lajoie, and Bin Ma. PEAKS DB: de novo sequencing assisted database search for sensitive and accurate peptide identification. *Molecular & cellular proteomics*, 11(4), 2012. [Cited on page 13]

[116] Michele Mishto, Yehor Horokhovskyi, John A. Cormican, Xiaoping Yang, Steven Lynham, Henning Urlaub, and Juliane Liepe. Database search engines and target database features impinge upon the identification of post-translationally cis-spliced peptides in hla class i immunopeptidomes. *Proteomics*, 22(10):2100226, 2022. [Cited on page 13]

[117] Miroslav Nikolov, Carla Schmidt, and Henning Urlaub. *Quantitative Methods in Proteomics*, volume 893 of *Methods in Molecular Biology*, chapter 7 Quantitative Mass Spectrometry-Based Proteomics: An Overview. Springer Science and Business Media LLC, 2012. [Cited on pages 13, 14, 15, and 101]

[118] Michele Mishto, Artem Mansurkhodzhaev, Ge Ying, Aruna Bitra, Robert A. Cordfunke, Sarah Henze, Debdas Paul, John Sidney, Henning Urlaub, Jacques Neefjes, Alessandro Sette, Dirk M. Zajonc, and Juliane Liepe. An in silico–in vitro pipeline identifying an HLA-A*02:01$^+$ KRAS G12V$^+$ spliced epitope candidate for a broad tumor-immune response in cancer patients. *Frontiers in Immunology*, 10(2572), Nov 2019. [Cited on pages 14, 16, and 102]

[119] Shao-En Ong, Blagoy Blagoev, Irina Kratchmarova, Dan Bach Kristensen, Hanno Steen, Akhilesh Pandey, and Matthias Mann. Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. *Molecular & Cellular Proteomics*, 1(5):376–386, May 2002. [Cited on page 14]

[120] Steven P. Gygi, Beate Rist, Scott A. Gerber, Frantisek Turecek, Michael H. Gelb, and Ruedi Aebersold. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature Biotechnology*, 17:994–999, Oct 1999. [Cited on page 14]

[121] Andrew Thompson, Jürgen Schäfer, Karsten Kuhn, Stefan Kienle, Josef Schwarz, Günter Schmidt, Thomas Neumann, and Christian Hamon. Tandem mass tags: A novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Analytical Chemistry*, 75(8):1895–1904, Apr 2003. [Cited on page 14]

[122] Scott A. Gerber, John Rush, Olaf Stemman, Marc W. Kirschner, and Steven P. Gygi. Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *PNAS*, 100(12):6940–6945, Jun 2003. [Cited on page 15]

[123] Weixun Wang, Haihong Zhou, Hua Lin, Sushmita Roy, Thomas A. Shaler, Lander R. Hill, Scott Norton, Praveen Kumar, Markus Anderle, and Christopher H. Becker. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Analytical Chemistry*, 75(18):4818–4826, 2003. [Cited on page 15]

[124] Karlie A. Neilson, Naveid A. Ali, Sridevi Muralidharan, Mehdi Mirzaei, Michael Mariani, Gariné Assadourian, Albert Lee, Steven C. van Sluyter, and Paul A. Haynes. Less label, more free: approaches in label-free quantitative mass spectrometry. *Proteomics*, 11(4):535–53, Feb 2011. [Cited on page 15]

[125] Christine Vogel and Edward M. Marcotte. Absolute abundance for the masses. *Nature Biotechnology*, 27(9):825–826, 2009. [Cited on pages 15, 93, 99, and 100]

[126] Haixu Tang, Randy J. Arnold, Pedro Alves, Zhiyin Xun, David E. Clemmer, Milos V. Novotny, James P. Reilly, and Predrag Radivojac. A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics*, 22(14):e481–8, Jul 2006. [Cited on page 15]

[127] Andrew F. Jarnuczak, Dave C. H. Lee, Craig Lawless, Stephen W. Holman, Claire E. Eyers, and Simon J. Hubbard. Analysis of intrinsic peptide detectability via integrated label-free and SRM-based absolute quantitative proteomics. *J Proteome Res*, 15(9):2945–59, Sep 2016. [Cited on page 15]

[128] Sven Nahnsen, Chris Bielow, Knut Reinert, and Oliver Kohlbacher. Tools for label-free peptide quantification. *Mol Cell Proteomics*, 12(3):549–56, Mar 2013. [Cited on page 15]

[129] Chongle Pan, Guruprasad Kora, David L. Tabb, Dale A. Pelletier, W. Hayes McDonald, Gregory B. Hurst, Robert L. Hettich, and Nagiza F. Samatova. Robust estimation of peptide abundance ratios and rigorous scoring of their variability and bias in quantitative shotgun proteomics. *Analytical Chemistry*, 78(20):7110–7120, 2006. [Cited on page 15]

[130] Maurice George Kendall. On the reconciliation of theories of probability. *Biometrika*, 36:101–116, 1949. [Cited on page 17]

[131] James O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer Science and Business Media LLC, 2 edition, 1985. [Cited on pages 17, 18, 19, 20, and 21]

[132] Rens van de Schoot, Sarah Depaoli, Ruth King, Bianca Kramer, Kaspar Märtens, Mahlet G Tadesse, Marina Vannucci, Andrew Gelman, Duco Veen, Joukje Willemsen, and Christopher Yau. Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1):1–26, 2021. [Cited on pages 18, 19, and 21]

[133] Matthew Richey. The evolution of Markov chain Monte Carlo methods. *The American Mathematical Monthly*, 117(5):383–413, May 2010. [Cited on pages 22 and 26]

[134] Charles J. Geyer. *Handbook of Markov Chain Monte Carlo*, chapter 1 Introduction to Markov Chain Monte Carlo. 2011. [Cited on pages 22 and 23]

[135] Sean P Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability.* Springer Science and Business Media LLC, 1 edition, 1993. [Cited on page 23]

[136] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer Science and Business Media LLC, 2 edition, 2004. [Cited on pages 23, 24, and 26]

[137] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, Jun 1953. [Cited on pages 23 and 96]

[138] Nicholas Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science*, Special Issue, 1987. [Cited on pages 23 and 96]

[139] Wilfred K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. [Cited on pages 23 and 96]

[140] Ming-Hui Chen, Qi-Man Shao, and Joseph G. Ibrahim. *Monte Carlo methods in Bayesian computation.* Springer Science & Business Media, 2012. [Cited on page 23]

[141] Jaewook Lee, Woosuk Sung, and Joo-Ho Choi. Metamodel for efficient estimation of capacity-fade uncertainty in li-ion batteries for electric vehicles. *Energies*, 8:5538–5554, 06 2015. [Cited on page 24]

[142] Charles J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–483, Nov 1992. [Cited on pages 25 and 26]

[143] Adrian E. Raftery and Steven M. Lewis. [practical markov chain monte carlo]: Comment: One long run with diagnostics: Implementation strategies for Markov Chain Monte Carlo. *Statistical Science*, 7(4):493–497, Nov 1992. [Cited on page 25]

[144] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–511, 1992. [Cited on pages 25, 26, 27, and 97]

[145] Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Paul-Christian Bürkner, Lauren Kennedy, Jonah Gabry, and Martin Modrák. Bayesian workflow. *ArXiv*, 2020. [Cited on pages 25, 26, 36, 37, and 96]

[146] Vivekananda Roy. Convergence diagnostics for Markov chain Monte Carlo. *Annual Review of Statistics and Its Application*, 7:387–412, 2020. [Cited on pages 25, 26, and 27]

[147] William A. Link and Mitchell J. Eaton. On thinning of chains in mcmc. *Methods in Ecology and Evolution*, 3(1):112–115, 2012. [Cited on page 26]

[148] Art B. Owen. Statistically efficient thinning of a markov chain sampler. *Journal of Computational and Graphical Statistics*, 26(3):738–744, 2017. [Cited on page 26]

[149] Marina Riabiz, Wilson Chen, Jon Cockayne, Pawel Swietach, Steven A. Niederer, Lester Mackey, and Chris Oates. Optimal thinning of mcmc output. *arXiv preprint arXiv:2005.03952*, 2020. [Cited on page 26]

[150] Mary K. Cowles and Bradley P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, Jun 1996. [Cited on pages 26, 27, 28, and 98]

[151] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer Series in Statistics. Springer Science and Business Media New York, 2004. [Cited on pages 26 and 28]

[152] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in bayesian workflow. *Journal of the Royal Statistical Society, Series A Statistics in Society*, 182(2):389–402, 2019. [Cited on page 26]

[153] Kerrie Mengersen, Sonia Knight, and Christian Robert. MCMC: how do we know when to stop? *Bulletin of the International Statistical Institute*, 58, 1999. [Cited on pages 27 and 28]

[154] Stephen P. Brooks and Gareth O. Roberts. Assessing convergence of Markov chain Monte Carlo algorithms. *Statistics and Computing*, 8(4):319–335, 1998. [Cited on page 27]

[155] Stephen P. Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998. [Cited on page 27]

[156] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2):667–718, 2021. [Cited on page 28]

[157] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Draft version 1.1 edition, Feb 1998. [Cited on page 28]

[158] Walter R. Gilks, Gareth O. Roberts, and Edward I. George. Adaptive direction sampling. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(1, Special Issue: Conference on Practical Bayesian Statistics):179–189, 1994. [Cited on page 28]

[159] Heikki Haario, Eero Saksman, and Johanna Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Computational statistics*, 14(3):375–395, 1999. [Cited on page 28]

[160] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, pages 223–242, 2001. [Cited on pages 28 and 96]

[161] Heikki Haario, Eero Saksman, and Johanna Tamminen. Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2):265–273, 2005. [Cited on page 28]

[162] Faming Liang and Wing Hung Wong. Evolutionary monte carlo: Applications to Cp model sampling and change point problem. *Statistica Sinica*, 10:317–342, 2000. [Cited on page 28]

[163] Faming Liang and Wing Hung Wong. Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association*, 96(454):653–666, Jun 2001. [Cited on page 28]

[164] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997. [Cited on page 29]

[165] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution - A Practical Approach to Global Optimization.* Natural Computing Series. Springer-Verlag Berlin Heidelberg, 2005. [Cited on page 29]

[166] Malcolm J.A. Strens. Evolutionary MCMC sampling and optimization in discrete spaces. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC*, 2003. [Cited on page 29]

[167] Malcolm J.A. Strens, Mark Bernhardt, and Nicholas Everett. Markov chain Monte Carlo sampling using direct search optimization. *Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia*, 2002. [Cited on page 29]

[168] Eric Laloy and Jasper A. Vrugt. High-dimensional posterior exploration of hydrologic models using multiple-try DREAM(ZS) and high-performance computing. *Water Resources Research*, 48, 2012. [Cited on pages 29, 31, 32, 48, and 99]

[169] Lennart Ljung and Torkel Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, 1994. [Cited on pages 33 and 36]

[170] Guillaume Basse and Iavor Bojinov. A general theory of identification. 2020. [Cited on page 33]

[171] Ernesto San Martín and Jorge González. Bayesian identifiability: Contributions to an inconclusive debate. [Cited on pages 33 and 34]

[172] Jean-Pierre Florens and Anna Simoni. Revisiting identification concepts in bayesian analysis. *Annals of Economics and Statistics*, (144):1–38, 2021. [Cited on page 33]

[173] Joseph B. Kadane. Testing a subset of the overidentifying restrictions. *Econometrica: Journal of the Econometric Society*, pages 853–867, 1974. [Cited on page 33]

[174] Oana-Teodora Chis, Julio R. Banga, and Eva Balsa-Canto. Structural identifiability of systems biology models: a critical comparison of methods. *PLoS One*, 6(11):e27755, 2011. [Cited on page 34]

[175] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. [Cited on page 35]

[176] David C. Lay, Steven R. Lay, and Judi J. McDonald. *Linear algebra and its applications.* Pearson, 2016. [Cited on page 36]

[177] John R. Taylor. *An Introduction to Error Analysis - The Study of Uncertainties in Physical Measurements.* University Science Books, 2 edition, 1997. [Cited on pages 37, 38, 39, and 95]

[178] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. [Cited on page 41]

[179] RStudio Team. *RStudio: Integrated Development Environment for R.* RStudio, PBC., Boston, MA. [Cited on page 41]

[180] Henrik Bengtsson. *R.utils: Various Programming Utilities*, 2020. R package version 2.10.1. [Cited on page 41]

[181] Henrik Bengtsson. A unifying framework for parallel and distributed processing in r using Futures, aug 2020. [Cited on page 41]

[182] Jeroen Ooms. *sys: Powerful and Reliable Tools for Running System Commands in R*, 2020. R package version 3.4. [Cited on page 41]

[183] Henrik Bengtsson. *matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors)*, 2021. R package version 0.58.0. [Cited on page 41]

[184] Florian Hartig, Francesco Minunno, and Stefan Paul. *BayesianTools: General-Purpose MCMC and SMC Samplers and Tools for Bayesian Statistics*, 2019. R package version 0.1.7. [Cited on page 48]

[185] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. [Cited on page 53]

[186] T. Loman, Y. Ma, V. Ilin, S. Gowda, N. Korsbo, N. Yewale, C. V. Rackauckas, and S. A. Isaacson. Catalyst: Fast biochemical modeling with julia. *bioRxiv*, 2022. [Cited on page 53]

[187] Christopher Rackauckas and Qing Nie. Differentialequations.jl – a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017. [Cited on page 53]

[188] Niklas Korsbo et al. 2017. `https://github.com/korsbo/Latexify.jl`. [Cited on page 53]

[189] Christian Lorenz Mueller. Exploring the common concepts of adaptive mcmc and covariance matrix adaptation schemes. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2010. [Cited on page 96]

[190] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, dec 2008. [Cited on page 97]

[191] Jun S. Liu and Faming Liang. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, Mar 2000. [Cited on page 99]

[192] Salvatore Cappadona, Peter R. Baker, Pedro R. Cutillas, Albert J. R. Heck, and Bas van Breukelen. Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino Acids*, 43(3):1087–1108, Jul 2012. [Cited on pages 99 and 100]

[193] Mike Sargent. Guide to achieving reliable quantitative LC-MS measurements. *RSC analytical methods committee*, 68, 2013. [Cited on pages 99 and 100]

[194] Chris Bielow. *Quantification and Simulation of Liquid Chromatography-Mass Spectrometry Data.* PhD thesis, Freie Universität Berlin, 2012. [Cited on page 100]

[195] Carlos López-Otín and Christopher M. Overall. Protease degradomics: A new challenge for proteomics. *Nature Reviews Molecular Cell Biology*, 3(7):509–519, Jul 2002. [Cited on page 101]

[196] Irina Lyapina, Vadim Ivanov, and Igor Fesenko. Peptidome: Chaos or inevitability. *International Journal of Molecular Sciences*, 22(23):13128, 2021. [Cited on page 101]

[197] G. Angus McQuibban, Jiang-Hong Gong, Eric M. Tam, Christopher A. G. McCulloch, Ian Clark-Lewis, and Christopher M. Overall. Inflammation dampened by gelatinase A cleavage of monocyte chemoattractant protein-3. *Science*, 289(5482):1202–1206, 2000. [Cited on page 101]

[198] Alex Mogilner, Roy Wollman, and Wallace F. Marshall. Quantitative modeling in cell biology: What is it good for? *Developmental Cell*, 11(3):279–287, Sep 2006. [Cited on page 102]

[199] Jonathon Howard. Quantitative cell biology: the essential role of theory. *Molecular Biology of the Cell*, 25(22):3438–3440, Nov 2014. [Cited on page 102]

[200] Jacques J. Neefjes and Frank Momburg. Cell biology of antigen presentation. *Current Opinion in Immunology*, 5:27–34, 1993. [Cited on page 102]

[201] A. Williams, C. Au Peh, and T. Elliott. The cell biology of MHC class I antigen presentation. *Tissue Antigens*, 59:3–17, Jan 2002. [Cited on page 102]

[202] Tim Elliott and Jacques Neefjes. The complex route to MHC class I-peptide complexes. *Cell*, 127(2):249–251, Oct 2006. [Cited on page 102]

[203] Jacques Neefjes, Marlieke L. M. Jongsma, Petra Paul, and Oddmund Bakke. Towards a systems understanding of MHC class I and MHC class II antigen presentation. *Nature Reviews Immunology*, 11(12):823–836, Nov 2011. [Cited on page 102]

[204] Darija Muharemagic, William Scott, and Michele Mishto. Antigen presentation to lymphocytes. pages 1–8, Oct 2019. [Cited on page 102]

[205] Björn Peters. *Modeling the MHC-I pathway.* PhD thesis, Humboldt-Universität zu Berlin, 2003. [Cited on page 102]

[206] S. Tenzer, B. Peters, S. Bulik, O. Schoor, C. Lemmel, M. M. Schatz, P.-M. Kloetzel, H.-G. Rammensee, H. Schild, and H.-G. Holzhütter. Modeling the MHC class I pathway by combining predictions of proteasomal cleavage,TAP transport and MHC class I binding. *CMLS Cellular and Molecular Life Sciences*, 62(9):1025–1037, May 2005. [Cited on page 102]

[207] Neil Dalchau, Andrew Phillips, Leonard D. Goldstein, Mark Howarth, Luca Cardelli, Stephen Emmott, Tim Elliott, and Joern M. Werner. A peptide filtering relation quantifies MHC class I peptide optimization. *PLoS Computational Biology*, 7(10):e1002144, Oct 2011. [Cited on page 102]

[208] Juliane Liepe, Herman-Georg Holzhütter, Peter Kloetzel, Michael Stumpf, and Michele Mishto. Modelling proteasome and proteasome regulator activities. *Biomolecules*, 4(2):585–599, Jun 2014. [Cited on page 102]

[209] Juliane Liepe, Hermann-Georg Holzhütter, Elena Bellavista, Peter M. Kloetzel, Michael P.H. Stumpf, and Michele Mishto. Quantitative time-resolved analysis reveals intricate, differential regulation of standard- and immuno-proteasomes. *eLife*, 4(e07545), 2015. [Cited on page 102]

[210] R. Charlotte Eccleston, Peter V. Coveney, and Neil Dalchau. Host genotype and time dependent antigen presentation of viral peptides: predictions from theory. *Scientific Reports*, 7(1), Oct 2017. [Cited on page 102]

[211] R. Charlotte Eccleston, Shunzhou Wan, Neil Dalchau, and Peter V. Coveney. The role of multiscale protein dynamics in antigen presentation and T lymphocyte recognition. *Frontiers in Immunology*, 8, Jul 2017. [Cited on page 102]

[212] Laura Parshotam. *Dynamic modelling of the processing of peptides for presentation on major histocompatability complex class I proteins.* PhD thesis, University College London, 2017. [Cited on page 102]

[213] Denise S. M. Boulanger, Ruth C. Eccleston, Andrew Phillips, Peter V. Coveney, Tim Elliott, and Neil Dalchau. A mechanistic model for predicting cell surface presentation of competing peptides by MHC class I molecules. *Frontiers in Immunology*, 9, Jul 2018. [Cited on page 102]

[214] Alan J. Hayes, Sanket Rane, Hannah E. Scales, Gavin R. Meehan, Robert A. Benson, Asher Maroof, Juliane Schroeder, Michio Tomura, Neil Gozzard, Andrew J. Yates, Paul Garside, and James M. Brewer. Spatiotemporal modeling of the key migratory events during the initiation of adaptive immunity. *Frontiers in Immunology*, 10, Apr 2019. [Cited on page 102]

[215] Jonathan W. Yewdell, Eric Reits, and Jacques Neefjes. Making sense of mass destruction: quantitating MHC class I antigen presentation. *Nature Reviews Immunology*, 3(12):952–961, Dec 2003. [Cited on page 102]

[216] Ting Wu, Jing Guan, Andreas Handel, David C. Tscharke, John Sidney, Alessandro Sette, Linda M. Wakim, Xavier Y.X. Sng, Paul G. Thomas, Nathan P. Croft, Anthony W. Purcell, and Nicole L. La Gruta. Quantification of epitope abundance reveals the effect of direct and cross-presentation on influenza ctl responses. *Nature communications*, 10(1):1–14, 2019. [Cited on page 102]

[217] Pouya Faridi, Chen Li, Sri H. Ramarathinam, Julian P. Vivian, Patricia T. Illing, Nicole A. Mifsud, Rochelle Ayala, Jiangning Song, Linden J. Gearing, Paul J. Hertzog, Nicola Ternette, Jamie Rossjohn, Nathan P. Croft, and Anthony W. Purcell. A subset of hla-i peptides are not genomically templated: Evidence for cis- and trans-spliced peptide ligands. *Science Immunology*, 3(28):eaar3947, 2018. [Cited on page 102]

[218] Roman Mylonas, Ilan Beer, Christian Iseli, Chloe Chong, Hui-Song Pak, David Gfeller, George Coukos, Ioannis Xenarios, Markus Müller, and Michal Bassani-Sternberg. Estimating the contribution of proteasomal spliced peptides to the hla-i ligandome. *Molecular & Cellular Proteomics*, 17(12):2347–2357, 2018. [Cited on page 102]

[219] Zach Rolfs, Stefan K Solntsev, Michael R. Shortreed, Brian L. Frey, and Lloyd M. Smith. Global identification of post-translationally spliced peptides with neo-fusion. *Journal of Proteome Research*, 18(1):349–358, 2018. [Cited on page 102]

[220] Juliane Liepe, John Sidney, Felix K.M. Lorenz, Alessandro Sette, and Michele Mishto. Mapping the MHC class I–spliced immunopeptidome of cancer cells. *Cancer Immunology Research*, 7(1):62–76, Nov 2018. [Cited on page 102]

[221] Katherine Woods, Pouya Faridi, Simone Ostrouska, Cyril Deceneux, Stephen Q. Wong, Weisan Chen, Ritchlynn Aranha, Nathan P. Croft, Divya Duscharla, Chen Li, Rochelle Ayala, Jonathan Cebon, Anthony W. Purcell, Ralf B. Schittenhelm, and Andreas Behren. The diversity of the immunogenic components of the melanoma immunopeptidome. *BioRxiv*, page 623223, 2019. [Cited on page 102]

[222] Wayne Paes, German Leonov, Thomas Partridge, Annalisa Nicastri, Nicola Ternette, and Persephone Borrow. Elucidation of the signatures of proteasome-catalyzed peptide splicing. *Frontiers in Immunology*, page 2355, 2020. [Cited on page 102]

[223] Ilan Beer. Commentary: An in silico – in vitro pipeline identifying an HLA-A*02:01+ KRAS G12V+ spliced epitope candidate for a broad tumor-immune response in cancer patients. *Frontiers in Immunology*, 12, 2021. [Cited on page 102]

[224] Michele Mishto, Guillermo Rodriguez-Hernandez, Jacques Neefjes, Henning Urlaub, and Juliane Liepe. Response: Commentary: An in silico–in vitro pipeline identifying an HLA-A*02:01+ KRAS G12V+ spliced epitope candidate for a broad tumor-immune response in cancer patients. *Frontiers in Immunology*, 12, 2021. [Cited on page 102]

[225] Michele Mishto, Juliane Liepe, Kathrin Textoris-Taube, Christin Keller, Petra Henklein, Marion Weberruß, Burkhardt Dahlmann, Cordula Enenkel, Antje Voigt, Ulrike Kuckelkorn, Michael P. H. Stumpf, and Peter M. Kloetzel. Proteasome isoforms exhibit only quantitative differences in cleavage and epitope generation. *European Journal of Immunology*, 44(12):3508–3521, Nov 2014. [Cited on page 102]

[226] Annika Nelde, Juliane Sarah Walz, Daniel Johannes Kowalewski, Heiko Schuster, Olaf-Oliver Wolz, Janet Kerstin Peper, Yamel Cardona Gloria, Anton W. Langerak, Alice F. Muggen, Rainer Claus, Irina Bonzheim, Falko Fend, Helmut Rainer Salih, Lothar Kanz, Hans-Georg Rammensee, Stefan Stevanović, and Alexander N. R. Weber. HLA class I-restricted MYD88 L265P-derived peptides as specific targets for lymphoma immunotherapy. *OncoImmunology*, 6(3):e1219825, Dec 2016. [Cited on page 102]

[227] Patrick A. Ott, Zhuting Hu, Derin B. Keskin, Sachet A. Shukla, Jing Sun, David J. Bozym, Wandi Zhang, Adrienne Luoma, Anita Giobbie-Hurder, Lauren Peter, Christina Chen, Oriol Olive,

Todd A. Carter, Shuqiang Li, David J. Lieb, Thomas Eisenhaure, Evisa Gjini, Jonathan Stevens, William J. Lane, Indu Javeri, Kaliappanadar Nellaiappan, Andres M. Salazar, Heather Daley, Michael Seaman, Elizabeth I. Buchbinder, Charles H. Yoon, Maegan Harden, Niall Lennon, Stacey Gabriel, Scott J. Rodig, Dan H. Barouch, Jon C. Aster, Gad Getz, Kai Wucherpfennig, Donna Neuberg, Jerome Ritz, Eric S. Lander, Edward F. Fritsch, Nir Hacohen, and Catherine J. Wu. An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, 547(7662):217–221, Jul 2017. [Cited on page 102]

[228] Matthias Kloor, Miriam Reuschenbach, Claudia Pauligk, Julia Karbach, Mohammad-Reza Rafiyan, Salah-Eddin Al-Batran, Mirjam Tariverdian, Elke Jäger, and Magnus von Knebel Doeberitz. A frameshift peptide neoantigen-based vaccine for mismatch repair-deficient cancers: A phase I/IIa clinical trial. *Clinical Cancer Research*, 26(17):4503–4510, Jun 2020. [Cited on page 102]

[229] Annika Nelde, Hans-Georg Rammensee, and Juliane S. Walz. The peptide vaccine of the future. *Molecular & Cellular Proteomics*, 20, 2021. [Cited on page 102]

[230] Paul F. Robbins, Mona El-Gamil, Yutaka Kawakami, and Steven A. Rosenberg. Recognition of tyrosinase by tumor-infiltrating lymphocytes from a patient responding to immunotherapy. *Cancer Research*, 54(12):3124–3126, 1994. [Cited on page 102]

[231] Alexandre Dalet, Paul F. Robbins, Vincent Stroobant, Nathalie Vigneron, Yong F. Li, Mona El-Gamil, Ken-ichi Hanada, James C. Yang, Steven A. Rosenberg, and Benoît J. van den Eynde. An antigenic peptide produced by reverse splicing and double asparagine deamidation. *Proceedings of the National Academy of Sciences*, 108(29):E323–E331, 2011. [Cited on page 102]

[232] Anouk C.M. Platteel, Juliane Liepe, Kathrin Textoris-Taube, Christin Keller, Petra Henklein, Hanna H. Schalkwijk, Rebeca Cardoso, Peter M. Kloetzel, Michele Mishto, and Alice J.A.M. Sijts. Multi-level strategy for identifying proteasome- catalyzed spliced epitopes targeted by CD8+ T cells during bacterial infection. *Cell Reports*, 20:1242–1253, Aug 2017. [Cited on page 102]

[233] Sergio Gonzalez-Duque, Marie Eliane Azoury, Maikel L. Colli, Georgia Afonso, Jean-Valery Turatsinze, Laura Nigi, Ana Ines Lalanne, Guido Sebastiani, Alexia Carré, Sheena Pinto, et al. Conventional and neo-antigenic peptides presented by β cells are targeted by circulating naïve CD8+ t cells in type 1 diabetic and healthy donors. *Cell metabolism*, 28(6):946–960, 2018. [Cited on page 102]