University of Wrocław
Faculty of Mathematics and Computer Science
Mathematical Institute

Hasselt University
Centre for Statistics

Mateusz Staniak

Statistical Methods for Mass Spectrometry Proteomics Data with Multiple Membership Structure

Doctoral thesis
written under the supervision of
prof. dr hab. Małgorzata Bogdan, University of Wrocław
prof. dr Tomasz Burzykowski, Hasselt University

Wrocław 2025

Contents

Chapter 1

Overview of the dissertation

Proteins carry out most functions of cells. Proteomics, the study of all proteins in a biological system such as a cell or a tissue, is thus essential to all biomedical research (Guo, J. A. Steen, and Matthias Mann, 2025). A core technology capable of determining both identities and quantities of proteins in a sample is mass spectrometry (MS). It has found use in various areas of academic and applied research involving proteins, including drug discovery, clinical proteomics, structural studies, the investigation of protein-protein interactions, and the study of post-translational modifications.

Mass spectrometry experiments generate large and complex data sets that require advanced computational approaches at every stage of analysis — from signal extraction from raw spectra, through normalization and filtering, to peptide identification and protein inference, and ultimately downstream statistical modeling.

This dissertation addresses practical challenges in the statistical analysis of mass spectrometry–based data, requiring both the development and benchmarking of new models, as well as the design and implementation of supporting software. A central focus is on problems that fall into the class of *multiple membership* models. Unlike standard analysis of variance, where each observation belongs to exactly one group, multiple membership arises when observations are simultaneously associated with several groups, contributing information to the estimation of multiple effects at once.

Two motivating examples of this structure are considered in detail.

- In **protein quantification**, a related problem arises in the inclusion of *shared peptides*, that is, peptides that map to more than one protein. Standard approaches often discard these peptides, thereby reducing statistical power and introducing biases, particularly when proteins are homologous. Modeling shared peptides as belonging to multiple proteins creates a multiple membership problem: a single peptide's observed abundance informs the abundance estimates of each protein to which it maps.

- In **hydrogen–deuterium exchange mass spectrometry (HDX-MS)**, the experimental output is peptide-level spectra, whereas the scientific goal is to infer exchange probabilities at the level of protein segments, often individual residues or short contiguous stretches. Each observed peptide spans multiple residues, and many residues are covered by several overlapping peptides. As a result, a single residue contributes to the uptake of multiple peptides, and each peptide provides information about several residues. This overlap creates a multiple membership structure: peptide-level data must be modeled as a composite of the exchange behavior of all residues it contains, while each residue's parameters are informed by multiple peptides.

For both of these problems, we propose new statistical models and effective computational algorithms that explicitly account for the multiple membership structure, allowing more accurate inference while efficiently pooling information across overlapping observations.

Finally, this dissertation also reports on the redesign and refactoring of `MSstats`, a widely used statistical software suite for the analysis of differential protein abundance from mass spectrometry

data, originally developed by Prof. Olga Vitek's group at Northeastern University (US). The statistical method for incorporating shared peptides introduced in this thesis has been integrated into the `MSstats` framework.

The dissertation is organized as follows.

- Chapter 2 first introduces all essential mathematical concepts that will be useful in explaining proposed statistical models. Then, we explain the technical background of MS-based proteomics studies and describe them in statistical terms. We provide an overview of the state-of-the-art methods for analyzing various types of proteomics data. Finally, we briefly describe the biological data sets that will be used for evaluating the proposed methods.

- Chapter 3 presents a new statistical model and computational algorithm, extending the `MSstats` framework to incorporate shared peptides into the quantification of protein abundance. We evaluated the method on both simulated and real data, focusing on two key aspects: the precision of protein abundance estimates across biological conditions (e.g., healthy vs. diseased patients), and the statistical properties of the resulting significance tests.

- Chapter 4 introduces a new statistical method for inferring the probabilities of hydrogen–deuterium exchange in protein fragments that are smaller than the directly observable peptide units. We evaluated the proposed approach on both simulated and real data, focusing on the precision of probability estimates and the quantification of their uncertainty.

- Chapter 5 describes the contribution to existing MSstats software tools for mass spectrometry proteomics. We describe proposed changes and improvements to the design of this family of statistical packages, and evaluate them in terms of complexity of their structure, extensibility, and performance. Finally, we introduce practical, free, and open-source implementations of the statistical methods developed in Chapters 3, 4, including an integration of the method for incorporating shared peptides into the MSstats workflow.

Funding acknowledgements and research output

This dissertation was prepared as a part of joint PhD studies in cooperation between University of Wrocław and Hasselt University. Hasselt University financially supported this cooperation with a BOF 2020 BILA grant (BOF20BL12-R11125), which made my research visits to UHasselt possible.

My work on the MSstats software suite in Olga Vitek's Lab (Northeastern University, US) was financially supported by the 2019 Chan Zuckerberg Essential Open Source Software Award, awarded to Prof. Vitek. The initial phase of this collaboration was supported by a travel grant from the NAWA programme at the University of Wrocław (2020). The work described in Chapter 3 was done as a part of a collaboration between Olga Vitek's Lab and Genentech, Inc.

Statistical work presented in this thesis was financially supported by the National Science Centre (Poland) grant PRELUDIUM 2020/37/N/ST6/04070 *Protein inference and quantification: a regularization approach* awarded in 2020.

Chapter 3 is based on a publication:

- Mateusz Staniak, Ting Huang, Amanda M Figueroa-Navedo, Devon Kohler, Meena Choi, Trent Hinkle, Tracy Kleinheinz, Robert Blake, Christopher M Rose, Yingrong Xu, Pierre M Jean Beltran, Liang Xue, Małgorzata Bogdan, and Olga Vitek, *Relative quantification of proteins and post-translational modifications in proteomic experiments with shared peptides: a weight-based approach*, Bioinformatics, Volume 41, Issue 3, March 2025, btaf046, DOI: 10.1093/bioinformatics/btaf046.

Chapter 4 is based on a publication in preparation:

- Mateusz Staniak, Jürgen Claesen, Tomasz Burzykowski, *Estimation of peptide-segment-level kinetic exchange rates based on the isotope distributions of overlapping peptides*, 2025.

Chapter 5 is based on publications:

- Devon Kohler, Mateusz Staniak, Tsung-Heng Tsai, Ting Huang, Nicholas Shulman, Oliver M. Bernhardt, Brendan X. MacLean, Alexey I. Nesvizhskii, Lukas Reiter, Eduard Sabido, Meena Choi, and Olga Vitek. *MSstats Version 4.0: Statistical Analyses of Quantitative Mass Spectrometry-Based Proteomic Experiments with Chromatography-Based Quantification at Scale*, Journal of Proteome Research 2023 22 (5), 1466-1482, DOI: 10.1021/acs.jproteome.2c00834.

- Devon Kohler, Mateusz Staniak, Fengchao Yu, Alexey I. Nesvizhskii, and Olga Vitek, *An MSstats workflow for detecting differentially abundant proteins in large-scale data-independent acquisition mass spectrometry experiments with FragPipe processing*, Nature Protocols 19, 2915–2938 (2024), DOI: 10.1038/s41596-024-01000-3.

Other relevant works include:

- Ting Huang, Mateusz Staniak, Felipe da Veiga Leprevost, Amanda M. Figueroa-Navedo, Alexander R. Ivanov, Alexey I. Nesvizhskii, Meena Choi, and Olga Vitek, *Statistical Detection of Differentially Abundant Proteins in Experiments with Repeated Measures Designs and Isobaric Labeling*, Journal of Proteome Research 2023 22 (8), 2641-2659 DOI: 10.1021/acs.jproteome.3c00155.

- Devon Kohler, Maanasa Kaza, Cristina Pasi, Ting Huang, Mateusz Staniak, Dhaval Mohandas, Eduard Sabido, Meena Choi, and Olga Vitek, *MSstatsShiny: A GUI for Versatile, Scalable, and Reproducible Statistical Analyses of Quantitative Proteomic Experiments*, Journal of Proteome Research 2023 22 (2), 551-556, DOI: 10.1021/acs.jproteome.2c00603.

Acknowledgements

I want to express my gratitude to my Advisors, Prof. Małgorzata Bogdan and Prof. Tomasz Burzykowski, for their guidance, patience, and all the learning opportunities.

I am grateful to Prof. Olga Vitek for including me in the MSstats team and providing me with numerous opportunities to engage with the proteomics community. I want to thank the Lab members: Meena Choi, Ting Huang, Devon Kohler, Tony Wu, and others, for our collaboration.

I want to thank Jürgen Claesen, who first introduced me to the world of mass spectrometry data. I would also like to thank him, Katarzyna Górczak and Piotr Prostko for their hospitality during my visits to Hasselt. I also thank my other colleagues from Wrocław, Hasselt, Warsaw and Boston, in particular Krystyna Grzesiak, for the work we have done and the time we have spent together.

I want to thank my Family - my wife, my parents, and my parents-in-law for being by my side. And finally, I thank God for all the good I received during my PhD studies.

Chapter 2

Background

In this chapter, we first introduce the mathematical background required to present our contributions. Next, we describe the technological and biological background of mass spectrometry-based proteomics studies. Finally, we connect the two topics by characterizing mass spectrometry data in statistical terms and summarizing state-of-the-art approaches to the analysis of such data.

2.1 Optimization

In this section, we introduce basic terminology and methods of optimization that will be relevant to fitting the proposed statistical methods.

2.1.1 Mathematical basics

Multivariate real-valued functions $f : \mathbb{R}^p \to \mathbb{R}$ will be our main tool for statistical modeling. In particular, we will use them to measure how closely the outputs of a statistical model match the observed data. To do this, we first need to define a notion of distance between vectors.

Definition 1 (Norm and distance) *Let us consider a function $L : \mathbb{R}^p \mapsto \mathbb{R}, p \geq 1$. If L satisfies the following conditions:*

1. $\forall_{x,y \in \mathbb{R}^p} L(x+y) \leq L(x) + L(y)$ *(triangle inequality),*
2. $\forall_{x \in \mathbb{R}^p} \forall_{c \in \mathbb{R}} L(cx) = |c| L(x)$*,*
3. $\forall_{x \in \mathbb{R}^p} L(x) = 0$ *if and only if* $x = 0$*,*

then we call it a norm. Moreover, such a function L can be used to define a real non-negative function $d(x,y), x, y \in \mathbb{R}^p, p \geq 1$ which measures the distance between its arguments by setting $d(x,y) = L(x-y)$.

A commonly used example of a distance measure in a p-dimensional space is the *Euclidean distance*, defined as

$$L_2(x,y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2},$$

for vectors $x = (x_1, \ldots, x_p)$ and $y = (y_1, \ldots, y_p)$. The corresponding Euclidean norm will be denoted by $\|\cdot\|_2$.

Definition 2 (Limit of a function) *Let us consider a function $f : X \mapsto \mathbb{R}, X \subseteq \mathbb{R}^p$, and a distance function $d : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$. We say that y_0 is a limit of function f as arguments approach (tend to) a given point x_0, denoted as $\lim_{x \to x_0} f(x)$, if*

$$\lim_{x \to x_0} f(x) = y_0 \iff \forall_{\varepsilon > 0} \ \exists_{\delta > 0} : \ \ d(x, x_0) < \delta \implies d(f(x), y_0) < \varepsilon.$$

Definition 3 (Continuity of a function) *Let f be real function $f : X \mapsto \mathbb{R}, X \subseteq \mathbb{R}^p$. We say that f is continuous at x_0 if $\lim_{x \to x_0} f(x) = f(x_0)$. If a function is continuous for all points in its domain, we call it a continuous function.*

Definition 4 (Derivative of a function) *First, let us consider a real function $f : \mathbb{R} \mapsto \mathbb{R}$. A derivative f' of f at point x_0 is defined by a limit*

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

If this limits exists and is finite, we say that f is differentiable at x_0. Alternatively, a derivative of f with respect to x can be denoted as $\frac{d}{dx} f$.

A multivariate counterpart of this definition involves norms of relevant vectors. However, from a practical point of view, a **partial derivative** is a more useful concept. A partial derivative of a real function of p variables with respect to i-th variable is a derivative computed while treating all other variables as constants:

$$\frac{d}{dx_i} f(x_1, \ldots, x_p) = \lim_{h \to 0} \frac{f(x_0, \ldots, x_i + h, \ldots, x_p) - f(x_0, \ldots, x_i, \ldots, x_p)}{h}. \tag{2.1}$$

A vector of all partial derivatives of f, $(\frac{d}{dx_1} f, \ldots, \frac{d}{dx_p} f)$, is referred to as gradient of f and denoted as $gradf$ or ∇f.

Points x such that $\nabla f = 0$ ($f'(x) = 0$ in univariate case), along with points where the gradient is undefined, are known as stationary points of f.

Moreover, a second partial derivative $\frac{d}{dx_i dx_j} f$ can be defined by applying this definition again to the partial derivative:

$$\frac{d}{dx_i dx_j} f = \frac{d}{dx_i} \left[\frac{d}{dx_j} f \right]. \tag{2.2}$$

This leads to the following notation.

Definition 5 (Hessian matrix) *Let f be a real function $f : X \to \mathbb{R}$, with $X \subseteq \mathbb{R}^p$, $p \geq 1$. The* Hessian matrix *of f at a fixed point $x_0 = (x_1^0, \ldots, x_p^0) \in X$ is the $p \times p$ matrix $H(x_0) = [H_{i,j}]$ whose entries are given by*

$$H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x_0), \quad i, j = 1, \ldots, p.$$

The Hessian matrix is often denoted by $\nabla^2 f(x_0)$ and describes the local curvature of f around x_0.

Basic definitions concerning real functions enable us to define minima of a function and introduce formally the notion of an optimization problem. These definitions are crucial, as algorithms of solving optimization problems enable application of statistical methods to data.

2.1.2 Core concepts of optimization

Definition 6 (Local and global minimum of a function) *Let $f : X \to \mathbb{R}$ be a real-valued function defined on a set $X \subseteq \mathbb{R}^p$, and let $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a distance function. A point $x_0 \in X$ is called a* local minimum *of f if there exists $\delta > 0$ such that*

$$d(x, x_0) < \delta \quad \Longrightarrow \quad f(x_0) \leq f(x) \quad \textit{for all } x \in X.$$

If the inequality $f(x_0) \leq f(x)$ holds for all $x \in X$, then x_0 is called a global minimum *of f.*

Definition 7 (Minimization) *Let $f : X \to \mathbb{R}$ be a real-valued function defined on a set $X \subseteq \mathbb{R}^p$. A **minimization problem** consists of finding a point $x_0 \in X$ such that $f(x_0)$ is minimal. In this context, f is called the* objective function*, x is the* optimization variable*, and a minimizer x_0 is referred to as an* optimal solution *of the problem.*

When multiple local minima exist, a *global minimum* is typically of primary interest. However, in practice, finding a global minimum may be less feasible than identifying a local one.

Definition 8 (Constrained Minimization Problem) *A **constrained minimization problem** consists of finding a point*

$$x_0 \in X$$

that minimizes a real-valued function $f : X \to \mathbb{R}$ while satisfying a set of restrictions. These restrictions, called constraints*, define a feasible set*

$$S \subseteq X$$

and are typically expressed as equalities or inequalities:

$$g_i(x) \leq 0, \quad h_j(x) = 0, \qquad i = 1, \ldots, m,\ j = 1, \ldots, k,$$

where g_i and h_j are real-valued functions on X. The goal is to find $x_0 \in S$ such that

$$f(x_0) \leq f(x) \quad \textit{for all } x \in S.$$

Let us note that, in many cases, a constrained optimization problem can be transformed into an unconstrained problem by appropriately changing the optimization variable. For example, consider an optimization problem in which the variable x is subject to the constraint $x > 0$. By introducing a new variable y such that

$$x = \exp(y),$$

the constraint is automatically satisfied, and the problem can be reformulated as an unconstrained optimization in terms of y.

There is a particular class of functions which guarantees that a local minimum is also a global minimum: convex functions. We introduce related terminology.

Definition 9 (Convex set) *Let consider a set $X \subseteq \mathbb{R}^p$. We say that X is convex if*

$$\forall_{x,y \in X} \forall_{\alpha \in [0,1]} \ \alpha x + (1 - \alpha) y \in X.$$

Definition 10 (Convex function) *Let $f : X \mapsto \mathbb{R}, X \subseteq \mathbb{R}^p$, be a real function. If X is convex and*

$$\forall_{x,y \in X} \ \forall_{\theta \in [0,1]} \ f\left(\theta x + (1 - \theta) y\right) \leq \theta f(x) + (1 - \theta) f(y), \tag{2.3}$$

then we say that f is convex. If a strict inequality holds for all $x \neq y$, we call such a function strictly convex.

Condition (2.3) can, in principle, be checked for any function. However, if the function f has additional properties, convexity can be verified more easily using the following criteria:

- **First-order condition:** If f is differentiable, it is convex on a convex set X if, for all $x, y \in X$,

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

- **Second-order condition:** If f is twice differentiable, it is convex on a convex set X if its Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in X$.

A less standard, but very useful class of optimization problem involves functions that are not convex, but become convex when restricted to a subset of their arguments. Optimization of such functions will be a major focus of Chapter 3.

Definition 11 (Bi-convex Sets) *Let $X \subseteq \mathbb{R}^p$ and $Y \subseteq \mathbb{R}^q$, and let $B \subseteq X \times Y$. For each $y \in Y$ and $x \in X$, define the* sections *of B by*

$$B_{\cdot,y} = \{x \in X : (x, y) \in B\},$$
$$B_{x,\cdot} = \{y \in Y : (x, y) \in B\}.$$

*The set B is called **bi-convex** if, for every $y \in Y$, the section $B_{\cdot,y}$ is convex in X, and for every $x \in X$, the section $B_{x,\cdot}$ is convex in Y.*

Definition 12 (Bi-convex Functions) *Let $X \subseteq \mathbb{R}^p$ and $Y \subseteq \mathbb{R}^q$, and let $Z \subseteq X \times Y$ be a bi-convex set. Consider a real-valued function $f : Z \to \mathbb{R}$. For each $(x, y) \in Z$, define the* sections *of f by*

$$f_{x,\cdot} : B_{x,\cdot} \to \mathbb{R}, \qquad f_{x,\cdot}(y) = f(x, y),$$
$$f_{\cdot,y} : B_{\cdot,y} \to \mathbb{R}, \qquad f_{\cdot,y}(x) = f(x, y),$$

where $B_{x,\cdot}$ and $B_{\cdot,y}$ are the sections of Z defined as in Definition 11.

*The function f is called **bi-convex** if Z is bi-convex, and if, for each $x \in X$, the function $f_{x,\cdot}$ is convex in y, and for each $y \in Y$, the function $f_{\cdot,y}$ is convex in x.*

Definition 12 can be naturally extended to *multi-convex functions* by partitioning the function's arguments into more than two sub-vectors and requiring convexity of f with respect to each sub-vector while keeping the others fixed.

A *bi-convex minimization problem* is an optimization problem of the form

$$\min_{(x,y) \in X \times Y} f(x, y) \quad \text{subject to } (x, y) \in B,$$

where B is a bi-convex set and $f(x, y)$ is a bi-convex function.

Since bi-convex functions are not convex in general, they can admit multiple local minima. As a result, finding the global minimum is often difficult due to their complex, nonlinear structure. In practice, popular algorithms therefore focus on identifying partial optima of the bi-convex function instead.

Definition 13 (Partial optimum) *Let $f : Z \mapsto \mathbb{R}, B = X \times Y \subseteq \mathbb{R}^{p+q}$ be a real function and let $(x_0, y_0) \in Z$. Point (x_0, y_0) is a partial optimum of f if*

$$\forall_{x \in B_{\cdot,y_0}} f(x_0, y_0) \leq f(x, y_0)$$
$$\forall_{y \in B_{x_0,\cdot}} f(x_0, y_0) \leq f(x_0, y).$$

For real, bi-convex, differentiable functions f, partial optima coincide with stationary points of f. Thus, solving a bi-convex optimization problem in search of partial optima may in fact lead to minima. Recall, however, that a stationary point of a function may correspond to a minimum, a maximum, or a saddle point.

Bi-convex optimization methods take advantage of the underlying bi-convex structure of the objective function. A commonly used approach for such problems is the *Alternate Convex Search* (ACS) method Gorski, Pfeuffer, and Klamroth, 2007 which can be seen as a special case of the more general *Block-Relaxation* De Leeuw, 1994 and *Block-Coordinate Descent* methods (see, for example, Bertsekas, 1997).

2.1.3 Optimization methods

We begin by introducing a standard approach for solving nonlinear equations, known as the Newton, or Newton–Raphson method (Nesterov et al., 2018). Given a differentiable function $g : \mathbb{R} \to \mathbb{R}$, the goal is to find a root x_0 such that $g(x_0) = 0$. Starting from an initial guess x, we use a first-order Taylor expansion of g around x:

$$g(x + \Delta x) = g(x) + g'(x)\,\Delta x + o(|\Delta x|).$$

Neglecting the remainder term, solving $g(x) + g'(x)\,\Delta x = 0$ yields

$$\Delta x = -\frac{g(x)}{g'(x)}.$$

Iterating this update gives the Newton–Raphson method:

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}.$$

To apply this idea to optimization, we set $g(x) = \nabla f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice-differentiable objective function. Finding a stationary point of f then reduces to solving the system of equations $\nabla f(x) = 0$. Applying Newton's method to this system leads to the multivariate Newton iteration

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k). \tag{2.4}$$

A stationary point x^* obtained in this way corresponds to a local minimum of f provided that the Hessian $\nabla^2 f(x^*)$ is positive definite. Thus, Newton's method serves as a fundamental building block for many optimization algorithms.

For bi-convex optimization problems, we consider the *Alternate Convex Search* (ACS) algorithm as defined in Gorski, Pfeuffer, and Klamroth, 2007. Let

$$\text{minimize } f(x, y) \quad \text{subject to } (x, y) \in B$$

be a bi-convex optimization problem. The ACS algorithm is an iterative procedure that generates a sequence of points (x_i, y_i), $i = 0, \ldots, M$, where M is a predefined maximum number of iterations. At each iteration, the function is optimized with respect to one block of variables while keeping the other block fixed. Specifically, given (x_i, y_i), the next iterate (x_{i+1}, y_{i+1}) is computed as

$$x_{i+1} = \arg\min_{x \in B_{x,\cdot}} f(x, y_i), \quad y_{i+1} = \arg\min_{y \in B_{\cdot,y}} f(x_{i+1}, y),$$

where $B_{x,\cdot}$ and $B_{\cdot,y}$ denote the feasible sets for x and y, respectively.

To terminate the procedure before reaching the maximum number of iterations, a stopping criterion is employed. Common choices include monitoring the absolute or relative change between consecutive iterates:

$$\|(x_{i+1}, y_{i+1}) - (x_i, y_i)\| < \delta, \quad \frac{\|(x_{i+1}, y_{i+1}) - (x_i, y_i)\|}{\|(x_i, y_i)\|} < \delta,$$

where $\delta > 0$ is a user-defined tolerance parameter.

For simplicity, we adopt the following notation:

$$f(x \mid y) := f_{\cdot,y}(x),$$

which indicates that the elements of the vector y are held fixed, while the elements of the vector x are treated as the variable arguments of the function $f(x \mid y)$.

Algorithm 1: Bi-convex optimization using the Alternate Convex Search algorithm.

1 Start with an arbitrary point $(x_0, y_0) \in B$, set $i = 0$.
2 **while** $i \leq M$ *and the stopping criterion is not satisfied* **do**
3 $\quad x_{i+1} = \arg\min_{x \in B_{\cdot,y_i}} f(x|y_i)$
4 $\quad y_{i+1} = \arg\min_{y \in B_{x_{i+1},\cdot}} f(y|x_{i+1})$
5 $\quad$ Set $i \leftarrow i + 1$.
6 **end**
7 **Output:** (x_i, y_i).

In both steps (3) and (4), we assume that solutions to the respective optimization problems exist. The order of optimization in these steps may be interchanged. If either of the problems in steps (3) or (4) has no solution, the procedure terminates immediately without producing one. The sequence $(f(x_i, y_i))_i$ is always nonincreasing. If, in addition, f is bounded from below, then this sequence is convergent (Gorski, Pfeuffer, and Klamroth, 2007). Moreover, if the objective function is biconvex and continuous with compact level sets, and the partial minima are unique, then ACS with exact block-wise minimization converges to a stationary point (Gorski, Pfeuffer, and Klamroth, 2007).

2.2 Mathematical and statistical terminology

In this section, we review statistical concepts that form the basis for modeling mass spectrometry-based proteomics data. We describe two major classes of statistical approaches: linear and non-linear regression models. For both approaches we present their statistical framework, properties and associated optimization techniques which enable fitting these models to data.

We use the basic concepts of probability theory and statistics such as random variables and estimators without formal introduction, which can be found in standard textbook such as (Feller, 1991). However, we will begin with a brief recap of hypothesis testing problem, as the related concepts will be used in the evaluation of proposed methods and in the description of state-of-the-art models.

2.2.1 Hypothesis testing

Statistical hypothesis testing in the frequentist sense involves a choice between two statements about true parameters of an underlying probability distribution or a function of these parameters (see standard statistical textbooks such as Hogg, McKean, and Craig, 2005). The two choices are referred to as null hypothesis (H_0) and the alternative (H_1 or H_A). Hence, two errors can be made when we compare the

outcome of a statistical test to the truth: rejecting H_0 (accepting H_A) when H_0 is true and accepting H_0 when it is false. The former is referred to as type I error, while the latter - the type II error.

Procedures that generate a decision regarding H_0 and H_A based on a statistic are called statistical tests. Each statistical test is characterized by two measures that describe its error rate: probabilities of type I and type II errors. Usually, statistical power is used in place of the type II error probability in characterization of statistical tests. When H_1 states that the unknown parameter θ belongs to a set Θ_1, power function γ is defined by $\gamma(\theta_1) = \mathbb{P}(H_0 \text{ is rejected } |\theta = \theta_1)$, $\theta_1 \in \Theta_1$. On the other hand, when H_0 states that $\theta \in \Theta_0$, maximum type I error probability over H_0, $max_{\theta_0 \in \Theta_0} \mathbb{P}(H_0 \text{ is rejected } |\theta = \theta_0)$ is known as the size or significance level of the test. A decision whether the null hypothesis should be rejected is based on a test statistic that can be calculated based on the data. Typically, the test statistics is compared to a threshold that is selected in order to control the type I error at a selected level, while minimizing type II error (equivalently: maximizing power). Often, the test statistic T is converted to a p-value via a transformation involving cumulative distribution function of T. For example, when null hypothesis is rejected when $T > c$ for c such that $\mathbb{P}(T > c|H_0) \leq \alpha$, where α is a selected significance level (maximum accepted probability of type I error), the p-value is defined as $p = \mathbb{P}(T > t^*|H_0)$ where t^* is the value of T computed based on the sample. After this transformation the test rejects H_0 when $p < \alpha$.

Often, both in the context of medical or biological applications, and in statistical learning, rejection of a null hypothesis is referred to as a positive result or a discovery. Similarly, a lack of rejection (accepting H_0) is referred to as a negative result. Then, type I error is referred to as a false positive results, and type II error as a false negative result.

Analysis of large datasets, such as proteomic data collected for many proteins, often requires testing multiple hypotheses simultaneously. A relevant example is measuring the abundances of many (say n) proteins in two groups: healthy and diseased. Performing all n tests at the same unadjusted significance level α results, on average, in $n\alpha$ false discoveries purely due to random chance. To address this, various generalizations of type I error measures have been proposed in the context of multiple testing to control overall error rates.

Of particular importance is the *False Discovery Rate* (FDR) and the method for controlling it introduced by Benjamini and Hochberg (1995). Let V denote the number of false discoveries among the n tests, and S the number of true discoveries. The *false discovery proportion* (FDP) is defined as

$$\text{FDP} = \frac{V}{\max(V + S, 1)}.$$

The FDR is then defined as the expected value of the FDP over independent replications of the experiment:

$$\text{FDR} = \mathbb{E}[\text{FDP}].$$

The Benjamini–Hochberg procedure controls the FDR at a given level q when testing hypotheses $H_1, \ldots, H_n$. Let $p_{(i)}$ denote the i-th smallest p-value in the ordered sequence of p-values. The procedure controls the FDR at level q by finding

$$k^* = \max \left\{ k : p_{(k)} \leq \frac{k}{n} q \right\}$$

and rejecting all hypotheses corresponding to p-values $p_{(i)}$ for $i \leq k^*$.

2.2.2 Convolutions of discrete probability distributions

For two independent random variables X and Y with probability mass functions f and g, respectively, a convolution $f * g$ of f and g is the probability mass function of their sum $X + Y$. In this section, we briefly introduce a method of finding the convolution $f * g$ for given functions f and g based on a polynomial representation of distributions f and g.

Definition 14 (Generating function) *First, let us consider a finite discrete random variable X such that $\mathbb{P}(X = i) = p_i$ for $i = 0, \ldots, K$. In general, a probability generating function (PGF) ψ_X of X is defined for $t > 0$ as*

$$\psi_X(t) = \mathbb{E}t^X$$

which in the finite discrete case can be expressed as

$$\psi_X(t) = \sum_{i=0}^{K} p_i t^i. \tag{2.5}$$

Finding convolutions of discrete probability distributions using generating functions The usefulness PGFs can be seen when considering the distribution of a sum of independent random variables. Let X and Y be independent random variables such that $P(X = k) = p_k, k = 0, \ldots, K$ and $P(Y = m) = q_m, m = 0, \ldots, M$. PGFs of these random variables are $\psi_X(t) = \sum_{i=0}^{K} p_i t^i$ and $\psi_Y(t) = \sum_{i=0}^{M} q_i t^i$, respectively. A well known result states that the PGF of a random variable $X + Y$ is given by

$$\psi_{X+Y} = \psi_X(t)\psi_Y(t). \tag{2.6}$$

2.2.3 Regression modeling

In this section, we introduce basic concepts related to regression modeling, which decomposes observed data into a function of various fixed data characteristics (variables) and noise terms modeled using random variables. The function that connects observations to variables may be defined by a set of parameters (parametric regression) or by a more general description such as a requirement that the function is smooth (non-parametric regression). Depending on assumption about this function and a distribution of random variables that model noise in data, fitting regression models may require different optimization techniques. Hence, we also describe relevant optimization techniques.

2.2.3.1 Linear models

The crux of linear models is a decomposition of a designated random variable (explained or independent variable) into a linear combination of unknown parameters and observed (dependent) variables, and a random noise terms. For a given observation with index i, such a model can be written as

$$Y_i = \sum_{k=0}^{p} \beta_k X_{i,k} + \varepsilon_i, \tag{2.7}$$

where Y_i denotes the value of an independent variable, $X_{i,k}$ is the value of k-th dependent variable (also called a predictor or a regressor) and terms $\beta_0, \beta_1, \ldots, \beta_p$ are the model parameters. Typically, the β_0 term is associated with a constant variable $X_0 \equiv 1$ and referred to as an intercept. A useful matrix notation treats both the independent variable and predictors as vectors such that i-th element of such a vector is the observed value for i-th observation. Using this notation we can re-write the linear model as

$$\mathbb{Y} = \mathbb{X} \times \beta + \varepsilon$$

where $\mathbb{Y} = (Y_1, \ldots, Y_n)^T$, $\mathbb{X}$ is a matrix such that its value in i-th row and j-th column is $X_{i,k+1}$ and $X_{i,1} = 1$ which corresponds to the intercept term, $\beta = (\beta_0, \ldots, \beta_p)^T$, $\varepsilon \sim MVN(\underline{0}, \sigma^2\mathbb{I})$, where $MVN(\mu, \Sigma)$ denotes a multivariate normal distribution with a mean vector μ and a covariance matrix Σ. Here, $\mathbb{I}$ denotes the identity matrix. This standard assumption about the random term corresponds to symmetric errors with a constant variation that do not exhibit high deviations (outliers) and lead to analytically traceable likelihood function with known asymptotic properties. However, in the context of mass-spectrometry data analysis this assumption often does not hold due to the presence of outlying

feature-level intensities, and alternative error term distributions must be considered, which will be discussed along with optimization techniques for regression models.

Explanatory (independent) variables in linear models can potentially be treated as random, but they are usually considered fixed. Likewise, the model effects may be considered fixed or treated as random variables from a specified distribution (random effects). The latter choice changes their interpretation. We will briefly discuss this case, as resulting models are useful in modeling various sources of variation in MS-based proteomics data.

In some models, it is beneficial to treat some effects corresponding to grouping (categorical) variables as random. Typically, such effects correspond to variables for which not all possible levels are observed. In the simplest case of a single grouping variable such a model can be written as (Gałecki and Burzykowski, 2012)

$$Y_i = X\beta + Z_i b_i + \varepsilon_i \tag{2.8}$$

where Y_i is a vector of n_i responses for observation in the i-th group, $i = 1, \ldots, N$, X is the design matrix for fixed effects and β is a vector of corresponding fixed parameters. Z_i is a design matrix for q random effects, $q \geq 1$, while $b_i = (b_{i,q}, \ldots, b_{i,q})^T$ is a vector of unobservable random effects. Classical normal assumptions state that $b_i \sim \mathcal{N}(\mathbf{0}, \mathbb{G})$, $\varepsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and b_i, ε_i are independent. Maximum likelihood estimation of such models is often based on the marginal distribution of y_i, as effects b_i are unobservable.

2.2.3.2 Non-linear models

Let us consider a more general model which includes the linear model as a special case. In this model specification, instead of the parametric formula of Equation 2.7, only the conditional mean and variance of the independent variable given observations are specified. A general linear model (GLS) enables modeling alternative variance structures compared to the simple linear model with independent errors, which facilitates modeling the correlation between observations. In this approach, the connection between the observed values of the independent variable $Y_j, j = 1, \ldots, n$ and related values of predictors x_j is established via the relation

$$\mathbb{E}\left[Y_j | x_j\right] = f(x_j, \beta), Var\left[Y_j | x_J\right] = \sigma^2 g^2\left(\beta, \theta, x_j\right) \tag{2.9}$$

where f is an known function of unknown parameters β and predictors x_j, and g is a function that specifies the relationship between the variance of observations and model parameters β and predictors x_j, which also depend on an additional parameter θ. If f is a linear function, and g is an identity function, this model simplifies to the standard linear model. This model is more general than a model such as given by Equation 2.7, but can be be formulated similarly by assuming that the conditional distribution of $Y_j | x_j$ is normal with mean and variance given by Equation 2.9.

2.2.3.3 Estimation

A standard statistical approach to fitting regression models to data is based on the likelihood principle. Let observations $\mathbb{Y} = (Y_1, \ldots, Y_n), n \geq 1$ have a joint distribution $f(\mathbb{Y}, \theta)$. We will also denote this joint distribution by $f(\theta)$ for simplicity. Parameter θ may be uni- or multi-dimensional. If random variables Y_i are independent and identically distributed such that $Y_i \sim \tilde{f}$, then $f = \prod_{i=1}^{n} \tilde{f}(\theta)$. As a function of θ, f is referred to as the likelihood function, as it describes the probability of observing this set of observations given the parameter θ. For convenience, maximization is often replaced by minimizing negative logarithm of the likelihood function.

In case of regression models, likelihood-based optimization typically leads to minimizing a function of differences between observed and predicted values of the independent variables, known as residuals. Let us consider two basic examples.

Example 1 (Likelihood estimation for a linear model with Gaussian errors) *Let us consider a linear regression model given by*

$$Y_i = \sum_{j=0}^{p} X_{i,j}\beta_j + \varepsilon_i, \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

under the assumption that σ^2 is known. This implies that $Y_i \overset{iid}{\sim} \mathcal{N}\left(\sum_{i=0}^{p} X_i\beta_i, \sigma^2\right)$. Since ε_i are independent, the likelihood function is given by

$$L(\beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(y_i - \sum_{j=0}^{p} X_{i,j}\beta_j)^2}{2\sigma^2}\right)$$

for $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$, and y_i, x_i denoting observed values of random variables Y_i and X_i, respectively. Then, the $-\log L(\beta) = \ell(\beta)$ is given by

$$\frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{p} X_{i,j}\beta_j\right)^2.$$

With a known σ^2, minimizing this function can be reduced to minimizing the sum of squared residuals $\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{p} X_{i,j}\beta_j\right)^2$. Hence, the linear model with iid standard normal errors leads to the so-called ordinary least squares loss function. In this case, the solution can be found analytically. Let us note that the simplicity of the least squares criterion, in particular due to its differentiability, leads to its use in many practical applications irrespective of statistical assumptions.

Example 2 (Likelihood estimation for a linear model with Laplace errors) *Let us consider a two parameter Laplace distribution $Laplace(\mu, \sigma)$ with a density function*

$$f(x, \mu, \sigma) = \frac{1}{2\sigma}\exp\left(\frac{-|x-\mu|}{\sigma}\right).$$

A linear regression model with iid errors from this distribution lead to the following likelihood function:

$$L(\beta) = \prod_{i=1}^{n} \frac{1}{2\sigma}\exp\left(\frac{-|y_i - \sum_{j=0}^{p} X_{i,j}\beta_j|}{\sigma}\right),$$

as $\varepsilon \overset{iid}{\sim} Laplace(0, \sigma$ implies $Y_i \overset{idd}{\sim} Laplace(\sum_{j=0}^{p} X_{i,j}\beta_j, \sigma)$. When σ is known, this leads to negative log-likelihood function of the form

$$\ell(\beta) = n\log(2\sigma) + \frac{1}{\sigma}\sum_{i=1}^{n}|y_i - \sum_{j=0}^{p} X_{i,j}\beta_j|.$$

Similarly to the Gaussian case, vector of parameters β can be estimated by minimizing the sum $\sum_{i=1}^{n}|y_i - \sum_{j=0}^{p} X_{i,j}\beta_j|$. Such estimator is more robust towards outliers than the least squares estimator.

These and analogous examples lead to the observation that fitting regression models is typically based on a comparison between observed and predicted values of the independent variable. For a given set of parameters $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$, let us define a predicted value for an i-th observation as

$$\hat{Y}_i = \sum_{j=0}^{p} X_{i,j}\beta_j.$$

Then, as a generalization of the provided examples, parameters can be fitted to a given set of observations $(X_i, Y_i), i = 1, \ldots, n$ by minimizing the total discrepancy between observed and predicted values

$$\ell(\hat{\beta}) = \sum_{i=1}^{n} L(Y_i - \hat{Y}_i),$$

where ℓ is the objective function and we will refer to L as a loss function. This approach is known as the M-estimation. Various assumptions about the error terms ε_i lead to different loss functions. We denote the loss functions originating from Gaussian and Laplace errors as L_2 and L_1, respectively. It is important to note that not every loss functions corresponds to a distribution of the random term. A popular choice of L which leads to robust estimation akin to the $Laplace$ noise case, but without the non-differentiability issue is the smooth robust Huber loss (P. J. Huber, 1992) given by

$$L_H(x, M) = \begin{cases} 2M|x| - M^2, & |x| \geq M, \\ |x|^2, & |x| < M, \end{cases}$$

where x is a scalar input and M is a positive hyperparameter. Lower values of this parameter increase the robustness of this procedure. Figure 2.1 presents examples of described norms for a range of values of residuals around 0.

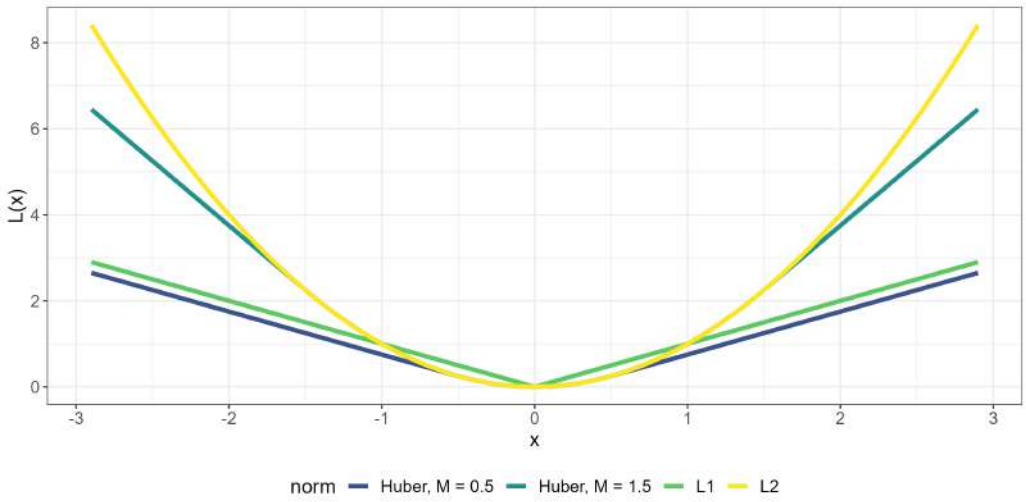


Figure 2.1: Comparison of L_2, L_1, and Huber norms for $M \in \{1.5, 0.5\}$, for a range of possible values of residuals. The Huber norm is quadratic in a neighbourhood of 0 that depends on the robustness parameter M, while resembling L_1 norm otherwise.

Maximum likelihood is less straightforward with nuisance parameters. Let us turn our attention to the general linear model given by Equation 2.9 with a Gaussian distribution assumption. Independence of error terms ε_i can still be assumed, but they no longer have identical distributions. Moreover, neither the scale parameter σ nor the parameter θ are assumed to be known. Then, the log-likelihood estimation leads to solving the equation (Davidian and Giltinan, 1995)

$$\sum_{i=1}^{n} g^{-2}(\beta, \theta, x_i) \left(y_i - f(x_i, \beta)\right) \frac{\partial}{\partial \beta} f(x_i, \beta) = 0$$

which requires a joint optimization over β, θ and σ, which complicates the estimation. As an alternative, pseudolikelihood approach replaces nuisance parameters θ and σ by their consistent estimators. Davidian and Giltinan, 1995 describes the following iterative procedure (PL-GLS) to estimate the GLS model using pseudolikelihood approach. Let M be the maximum number of iterations. A stopping criterion is required, which can be based on a comparison of β estimators from consecutive iterations, for example.

Algorithm 2: An iterative approach to solving PL-GLS problems.

1 Start with an arbitrary point estimator $\widehat{\beta}^{(0)}$, set $k = 0$.
2 **while** $k \leq M$ *and the stopping criterion is not satisfied* **do**
3 Estimate θ by $\theta^{(k)}$. Compute weights $\widehat{w}_i = g^{-2}\left(\widehat{\beta}^{(k)}, \widehat{\theta}^{(k)}, x_i\right)$. Re-estimate β by solving $\sum_{i=1}^{n} g^{-2}(\beta, \theta, x_i)\,(y_i - f(x_i, \beta))\,\frac{\partial}{\partial\beta} f(x_i, \beta) = 0$. Set $k \leftarrow k + 1$.
4 **end**
5 **Output:** $\widehat{\beta}$ after the final iteration.

2.2.3.4 Multiple membership data

Regression models can be used to compare means between groups. Standard statistical test such as Student's t test and analysis of variance (ANOVA) can be represented by a linear model with binary explanatory variables that indicate group membership of observations. For example, a comparison between two groups A and B is equivalent to fitting a linear model $Y_i = \mu + \beta_1 X_1$, where $X_1 = \mathbb{1}(\text{observation } i \text{ belongs to group A})$ and testing the hypothesis $\beta_1 = 0$, as the mean of group A under this model is equal to $\mu + \beta_1$, and the mean of group B is equal to μ. This logic can be extended to multiple groups, and using additional variables to represent various sources of variation. Examples of such models in the context of the analysis of mass spectrometry data will be given in Section 2.4

A different way to extend such models is to consider observations that may belong to multiple groups at the same time. This leads to the so-called multiple (or mixed) membership modeling (Blei, Ng, and Jordan, 2003; Erosheva, Fienberg, and Lafferty, 2004, recently Marco et al., 2024).

Under multiple membership, membership of the i-th observation in group g is represented by a variable $X_{i,g}$, in analogy to the example of a Student's test given earlier. However, it is not required that $X_{i,g} \in \{0, 1\}$. Instead, $X_{i,g}$ is a continuous value such that $X_{i,g} \in [0, 1]$. The value of $X_{i,g}$ is interpreted as a degree to which observation i belongs to group g. Often, it is considered a probability, or a ratio of, say, time associated with group g. Examples include the probability of a topic occurring in a document (natural language processing (Blei, Ng, and Jordan, 2003)), or a time spent by a child in a given school. In these cases, it is reasonable that for all groups $g = 1, \ldots, G$ we have $\sum_{g=1}^{G} X_{i,g} = 1$ for each observation i. The values $X_{i,g}$ describing the multiple membership structure may be known, assumed to be proportional to some observable variables and treated as constants, or considered unknown.

2.2.4 Graph theory

We conclude this chapter with several basic definition regarding graphs based on Wilson, 1972. These concepts will be useful for a rigorous description of modeling sets of proteins identified in mass spectrometry experiments.

Definition 15 (Graph) *Let V be a non-empty finite set and let E be a finite set of (unordered) pairs of distinct elements of V. A simple graph G consists of two elements: a set of vertices V and a set of edges E. We denote this by $G = (V, E)$.*

We say that an edge $v, w \in E$ joins vertices v and w. Allowing pairs of non-distinct elements of V in the edge set E leads to so-called loops. Additionally, it is possible to generalize this definition by allowin E to be a collection rather than a set, meaning that an identical pair may occur multiple times. Such a generalized definition describe a general graph, often referred to simply as a graph. Finally, let us note that edges of graph may be directed.

If graphs X and Y are defined by vertex sets V and W, respectively, and by edge sets E and F, respectively. We say that Y is a **subgraph** of X if $W \subset V$ and $F \subset E$. It is also possible to create

a new graph based on X and Y. If E and F are disjoint and a graph Z consists of a set of vertices $U = W \cup V$ and a set of edges $G = E \cup F$, then we say that Z is a union of X and Y. This allows us to introduce an important property of graphs. If a graph cannot be represented as a union of two graphs, it is referred to as a **connected graph**. Conversely, if a graph is a union of at least two graphs, then it is disconnected.

Definition 16 (Bipartite graph) *Let G be a graph defined by a vertex set V and an edge set E. If V can be decomposed into a union of two disjoint set $V = W \cup U$, and every edge $w, u \in E$ is such that $w \in W$ and $u \in U$, then G is bipartite graph.*

In general, every graph can be represented by a matrix which either provides information about the number of edges between given vertices, or encodes each edge by indicating its constituting vertices. In case of simple bipartite graphs which are of particular importance for this thesis, the former representation is both useful and simple. Let us consider a simple bipartite graph $G = (V, E)$ such that $V = W \cup U$, and W, U are disjoint. Let $W = w_1, \ldots, w_n$ and $U = u_1, \ldots, u_m$. Then, G can be represented by an **adjacency matrix** M such that

$$M_{i,j} = \begin{cases} 1, w_i, u_j \in E, \\ 0, w_i, u_j, \notin E. \end{cases}$$

A transposition of this matrix also defines an adjacency matrix for this graph.

2.3 Mass spectrometry experiments

In this section, we introduce biological motivations and principles behind mass spectrometry-based proteomics. Then, we describe technical limitations and characteristics of mass spectrometry (MS) as a measurement technique that are relevant to the development of statistical methods. We describe a conceptual framework for interpreting MS data.

2.3.1 Biological context

Proteins are chemical molecules responsible for executing cellular functions in living organisms. The instructions for their synthesis are encoded in DNA; however, it is not possible to fully predict the protein content of a cell, tissue or an organism, i.e. the proteome, based on that information alone due to the fact that proteins can, for instance, undergo post-translational modifications. Consequently, while studying proteins is crucial for understanding biological processes such as disease progression, it cannot be reduced to genetic analysis. The difference in complexity between genome and proteome is staggering: for instance, the human genome contains roughly 20,000 genes, yet it can give rise to up to one million proteoforms (Munoz and Heck, 2014).

Amino acids are the building blocks of proteins. There are 21 amino acids. Each amino acid has two ends, referred to as N-terminus and C-terminus, derived from amine ($-NH_2$) and carboxyl ($-COOH$) groups in their chemical structure, respectively. In addition to those groups, they include a hydrogen atom and a variable side chain referred to as a **residue**. When several amino acids react, they form peptide (amide) bonds. Hence, such chains of amino acids are referred to as peptides. Figure 2.2 presents a schematic representation of how amino acids form peptide bonds to create oligo- and polypeptides. Amino acids are represented by two different notations: three-letter codes (for example alanine - ala) and one-letter codes (alanine - A). We give an example of the three-letter codes in Figure 2.3, but use the short, one-letter notation throughout the dissertation. In this case, sequences of amino acids are represented simply by sequences of letters from the 21 letter alphabet.

Sometimes, a distinction is made between shorter chains of up to 20 amino acids (oligopeptides) and longer chains (polypeptides). As such, **proteins** can be defined as polypeptides with biological

functions. However, functions of proteins cannot be determined solely based on their amino acid composition. They also depend on higher order characteristic of proteins, referred to as structures. Amino acid composition is merely the primary structure, while their folding is the secondary protein structure. Even protein-protein interactions affect their functions (V. S. Rao et al., 2014).

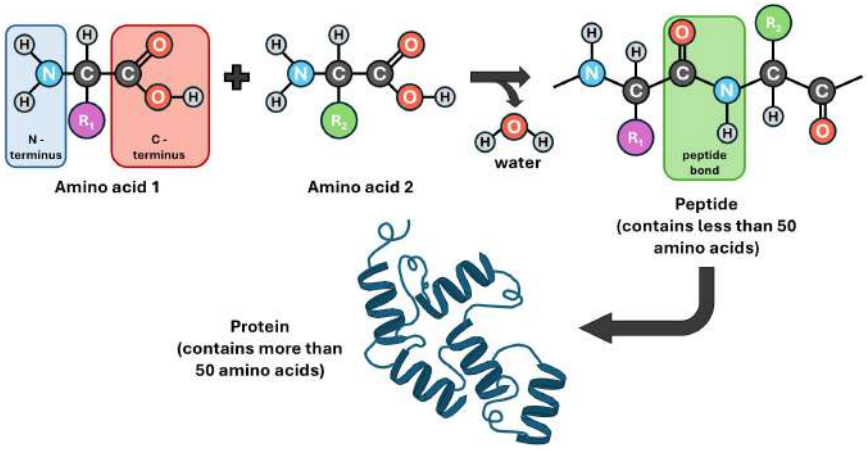


Figure 2.2: A schematic representation of how amino acids form peptides and proteins.

A core technology that is used to identify protein content of a biological sample and to study various characteristics of proteins is mass spectrometry. It addition to determining identities (amino acid sequences) of proteins in a sample, it is capable of measuring their abundance. It is widely used in various fields of research related to proteins. The following section provides basic information about measurements in mass spectrometry-based proteomics.

2.3.2 Technical background

In this section, we give a high-level view of mass spectrometry (MS)-based proteomics studies. Structure and characteristics of MS data are largely driven by the technical limitations of mass spectrometry. Thus, we begin by describing the measurement process while emphasizing challenges relevant to the proposed statistical methods.

2.3.2.1 Bottom-up and top-down proteomics

There are two major approaches to identification and quantification of proteins via mass spectrometry. The top-down approach quantifies intact proteins. However, due to technical limitations, its applicability is limited to proteins with high abundance and low molecular weight (Miller and L. M. Smith, 2023). The alternative is to perform the measurements on peptides. This enables higher coverage of protein sequences and identification of lower abundance proteins compared to the top-down approach. Peptides are created in the process of **digestion**: before being subjected to measurements in a mass spectrometer, proteins are broken down into smaller pieces by hydrolization of peptide bonds via proteolysis (Shuken, 2023). From the primary protein structure perspective, amino acid sequences of peptides are subsequences of the original protein sequence. Digestion can be specific or non-specific. In the former case, peptide bonds are cleaved after specific amino acids patterns, while in the latter case, they are cleaved randomly. For example, a commonly used protease trypsin cleaves proteins after amino acids arginine and lysine, unless they are followed by proline. In practice, digestion is not complete and some cleavage sites remain intact.

2.3.2.2 Mass spectrometers

A mass spectrometer is a device used for the identification of chemical compounds based on their mass and mass of their fragments. It consists of three main components: an ion source which vaporizes the input analytes, giving them charge in the ionization process; a mass analyzer which separates the resulting ions based on their mass to charge ratio (denoted m/z); and a detector which records a signal that is amplified and stored as pairs of m/z and intensity values (Matthiesen and Bunkenborg, 2020). These intensity values are generated based on electric current proportional to the abundance of ions. Thus, these values are used to estimate the abundance of proteins. However, as peptides are identified based on mass information, it is important to also understand the m/z dimension of spectra. Section 2.3.2.4 provides details of spectra interpretation, including important concepts and distinctions regarding mass of peptides. Figure 2.3 presents the the basic working principle of mass spectrometry. Protein molecules are digested into peptides. Mass spectrometers are capable of separating resulting smaller molecules of peptides based on their mass, which enables their identification and quantifiation.

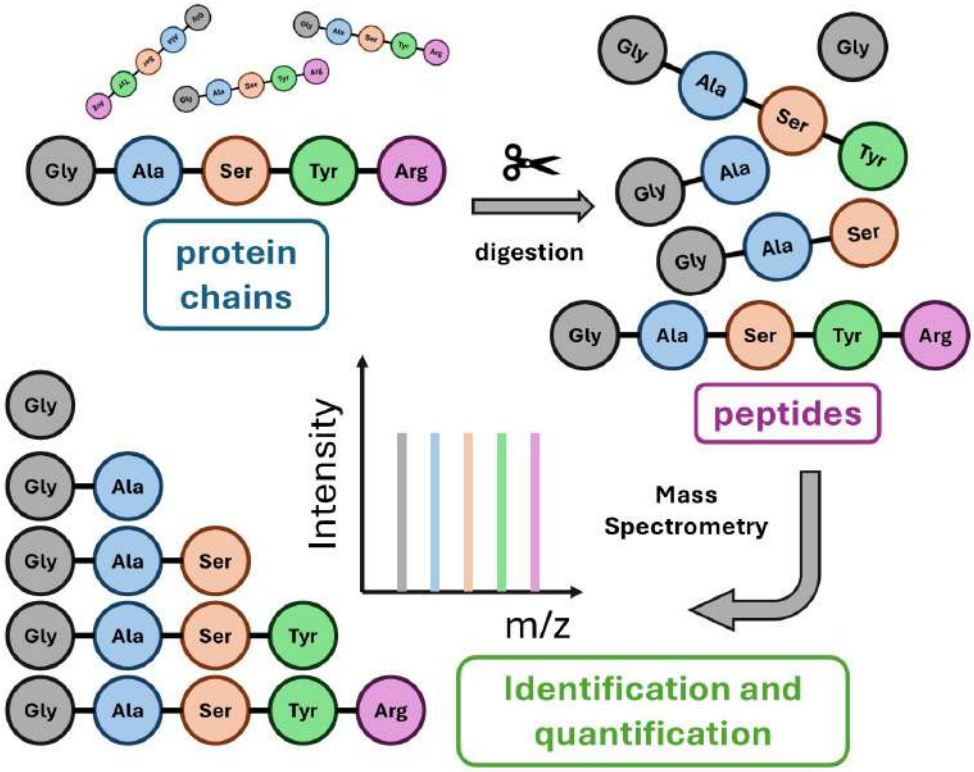


Figure 2.3: Schematic representation of a mass spectrometry measurement.

Quality of mass spectrometry measurements depends heavily on complexity and purity of the sample. The latter issue pertains to the composition of the sample. Presence of contaminants (proteins that are not of interest, but come from outside sources such as clothing of lab personnel) may lead to both spurious identifications of peptides similar to those originating from contaminants, and to the suppression of signal originating from target proteins of the study. On the other hand, high complexity of a sample leads to issues with peptides competing for ionization (Matthiesen and Bunkenborg, 2020), which leads to fewer identified peptides and lower coverage. To alleviate this issue, separation techniques are used to group peptides with similar physicochemical properties. One of the most popular methods is liquid chromatography (LC). LS-MS enables collecting a series of MS spectra taken as the molecules elute from the chromatography column (Matthiesen and Bunkenborg, 2020).

2.3.2.3 MS1 and MS2 spectra

The sequence of MS spectra generated by LC-MS is often referred to as survey scan, as they enable further investigation of selected analytes found in such spectra. This is particularly useful, because analytes are only described by m/z values and corresponding intensities after MS analysis. Peptide identity (amino acid sequence) needs to be recovered from the sequence of m/z values: a problem known as peptide sequencing (H. Steen and Mann, 2004). There is a commonly used procedure designed to aid in this challenge. In MS/MS approach, a subset of peptide ions recorded in each mass spectrum is subjected to additional fragmentation and another set of spectra is taken for the fragments of peptides. Initial spectra are referred to as **peptide MS scans** (MS1 for short), while the the other spectra are called **fragment MS scans** (MS2 for short). Predictable fragmentation patterns give rise to specific types of peptide fragments which improves sequencing. The next section introduces core concepts related to mass which are necessary to understand how mass spectra are used to identify and quantify peptides, and how it affects peptide sequencing.

2.3.2.4 Interpretation of MS1 spectra

Mass of a peptide is a sum of masses of its amino acids reduced by mass of H_2O created by peptide bonds. However, the mass of each amino acid can vary due to differences in masses of atoms. Variants of elements that differ by a number of neutrons in the atom's nucleus (and hence mass) are referred to as **isotopes**. The number of protons in the nucleus on the other hand is constant for a given element. Standard notation that describes isotopes puts the sum of the number of protons and neutrons, called the mass number, in a superscript before the chemical symbol of an element. For example, ^{12}C denotes carbon-12: carbon isotope with a mass number equal to 12. Per convention, this particular isotope serves as a base for mass measurements in mass spectrometry. The unit of measurement is **unified atomic mass unit** (u), defined as $\frac{1}{12}$ of the mass of a ^{12}C atom (Volmer and Leslie, 2007). This unit is also referred to as Dalton (Da) and this terminology will be used throughout this thesis.

Each isotopic variant of an element has a certain probability of occurrence. Hence, each sequence of amino acids is associated with a probability distribution of occurrence over possible masses. In probability theory terms, this probability distribution is a convolution of probability distributions over isotopic variants of elements that constitute the peptide. As a consequence, variants of a peptide with different masses are observed during measurement in a mass spectrometer. This is reflected in MS1 spectra: each peptide is represented by multiple peaks found at m/z values corresponding to masses of these variants. Moreover, their intensities are proportional to isotopic probabilities.

Theoretical mass of an ion whose charge state, amino acid and isotopic compositions are known is referred to as the **exact mass** (Brenton and Godfrey, 2010). There are two ways in which isotopic masses are aggregated into a single mass value that characterizes a molecule. **Average mass** is calculated as a mean of isotopic masses weighted by probabilities of occurrence. **Monoisotopic mass** is the sum of exact masses of most abundant isotopes of constituent atoms (Volmer and Leslie, 2007).

While exact mass refers to a theoretical value, accurate mass is defined as *the experimentally determined mass of an ion measured to an appropriate degree of accuracy and precision used to determine, or limit the possibilities for, the elemental formula of the ion* Brenton and Godfrey, 2010. To make the terminology clearer, terms *measured accurate mass* and *calculated exact mass* have been proposed (Brenton and Godfrey, 2010).

Comparisons between theoretical and observed masses are typically done on a relative scale. For a molecule with calculated exact mass m_e and measured accurate mass value m_a, mass measurement error δ_m is defined (Volmer and Leslie, 2007) as

$$\delta_m = \frac{m_a - m_e}{m_e} \times 10^6 \tag{2.10}$$

and the unit of such a measurement is referred to as **parts per million** (ppm). Measurement error

describes the accuracy of a mass spectrometer. As the mass is expressed in Da, and the unit of charge is coulomb, the unit of m/z values is defined as a ratio of these, known as the Thomson unit.

Two additional terms describe the ability of an instrument to distinguish peaks at close m/z values (Volmer and Leslie, 2007). The theoretical capability is termed resolving power. A corresponding value estimated from mass spectra is called resolution. Thus, a mass spectrometer capable of separating peaks at close m/z values is said to have a high resolving power, while the resulting spectra are said to be of high resolution. High resolution is important both for extracting peptide identities from spectra (by ensuring accuracy in the mass dimension) and for quantifying their abundance (by reducing interferences from co-eluting species).

In the intensity dimension, **dynamic range** describes the ability of an instrument to detect and quantify analytes with abundances in different orders of magnitude.

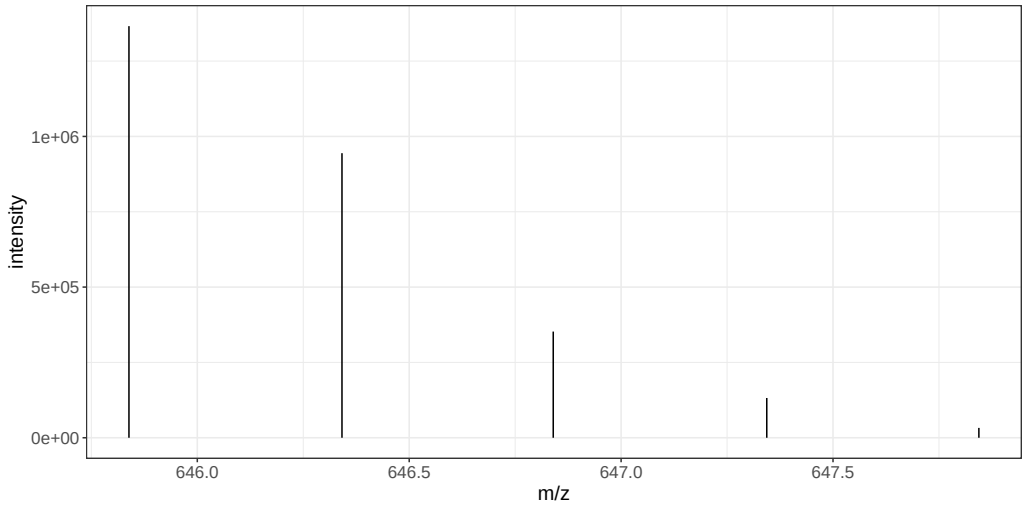


Figure 2.4: Example sequence of isotopic peaks for a doubly charged peptide AADFIDQALAQK. Taken from a dataset published in The et al., 2018.

Example 3 (Isotopic peaks in MS1 spectrum) *Figure 2.4 shows an example of an isotopic peak cluster in an MS1 spectrum for the doubly charged peptide* `AADFIDQALAQK`*. The leftmost peak corresponds to the monoisotopic ion. Table 2.1 provides additional details.*

The monoisotopic mass of this peptide is 1289.662 Da. For a charge state of $+2$*, the monoisotopic peak is therefore expected at 645.84 m/z. Subsequent peaks appear at intervals of approximately* 0.5 *Th, because isotopic variants differ by roughly* 1 *Da in mass, and the spacing in m/z is equal to* 1 *divided by the charge.*

The column absolute mass error reports the absolute values of the mass measurement error, δ_m*, for each peak in parts per million. All observed peaks were detected very close to their theoretical m/z values.*

In addition, the observed intensities closely match the expected isotopic distribution when standardized by the total intensity of the cluster.

2.3.2.5 Acquisition modes

Selection of peptide ions for fragmentation based on MS1 spectra leads to another distinction between two major types of mass spectrometry experiments referred to as data acquisition mode. In first approach, a fixed number of ions corresponding to most intense peaks is selected for further fragmentation. Since abundance information is used in this approach, it is referred to as data-dependent acquisition (DDA). As an alternative, all ions within a specified, wide m/z range are fragmented, which leads to more complex MS2 spectra with a large number of present peptide ions. This strategy

Obs. m/z	Theor. m/z	Intensity	Rel. intensity	Isot. prob.	Abs. Mass Err. [ppm]
645.84	645.84	1365985.25	0.48	0.48	1.05
646.34	646.34	944392.75	0.33	0.33	2.88
646.84	646.84	352392.53	0.12	0.13	1.52
647.34	647.34	131988.05	0.05	0.04	2.40
647.84	647.84	32716.76	0.01	0.01	1.18

Table 2.1: Characteristics of the sequence of isotopic peaks displayed in Figure 2.4.

is called data-independent acquisition (DIA). In principle, DDA approach generates data with a significant proportion of missing values due to differences in peptide identifications between runs, and makes quantification of lower abundance proteins difficult, while DIA alleviates these issues.

2.3.2.6 Isobaric labeling

With the standard approach to MS experiments, called label-free, each biological sample is measured in a separate run. Isobaric labeling is an approach which enables simultaneous processing of multiple samples. Such ability is referred to as multiplexing. There are multiple implementation of this quantification strategy, including stable isotope labeling by amino acids in cell culture (SILAC), isobaric tag for absolute and relative quantification (iTRAQ) and tandem mass tags (TMT). Reagents used to process N samples jointly are called N-plexes. One of the most widely used variants of isobaric labeling is 11-plex TMT (Sivanich et al., 2022). Figure 2.5 presents a schematic representation of the TMT labeling approaches. After digestion of protein samples, indicated by color, each sample is labeled with a mass tag. These tags have identical mass, but can be differentiated after fragmentation. Hence, all of these samples are processed together in a single MS run, and abundances of peptides labeled with each tag can be calculated.

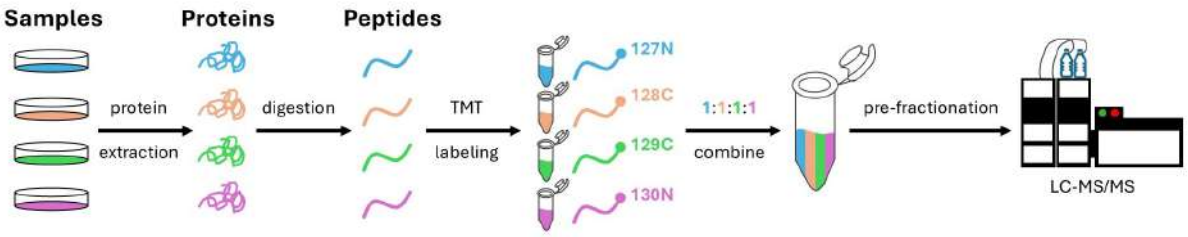


Figure 2.5: A schematic representation of an MS experiment with tandem mass tags labeling.

2.3.3 Peptide identification

In bottom-up proteomics, protein digestion produces peptides that must be identified based on MS2 spectra. Once peptides are identified, the next step is to determine which proteins are present in the sample using peptide-level evidence. This second task is known as the protein inference problem.

In this section, we introduce the terminology needed for a rigorous treatment of both protein inference and protein quantification. Since the statistical methods proposed in this thesis operate on peptide-level data to draw conclusions at higher levels, we do not delve into the details of peptide identification or the processing of raw spectra required for peptide ion quantification. For comprehensive overviews, we refer the reader to existing reviews such as Noor et al., 2021 or Huang, J. Wang, et al., 2012. Throughout the thesis, we assume that peptide identification or quantification has already been performed. Whenever additional spectral processing is needed, we explicitly describe all required analytical steps. For example, analyses involving HDX-MS require extraction of isotopic patterns from MS1 spectra.

We briefly outline the principles of peptide identification via database search, and then focus on protein inference in quantitative MS studies and modeling of peptide sequence overlap in HDX-MS applications.

Peptide identification via database search generally involves three steps: generating a list of candidate peptides, creating their theoretical spectra, and comparing those theoretical spectra to the observed ones.

Mass measurement errors, chemical or spectral interferences, sequence similarities, and post-translational or experimental modifications can lead to incorrect identifications. As a result, both false positives (peptides not present in the sample but incorrectly matched) and false negatives (true peptides that remain unidentified) can occur. This uncertainty in peptide identification propagates to uncertainty in the inferred set of proteins.

Search engines provide users with extensive control over search parameters, such as peptide length constraints, allowed charge states, and possible modifications. A variety of methods have also been developed to validate identifications and control error rates.

However, from our perspective, the most crucial aspect of the process is the selection of the database used for the search.

2.3.3.1 Protein databases

When peptides are generated using a specific protease, with predefined modification settings and peptide length constraints, the set of candidate peptides produced during database search is determined entirely by the proteins present in the chosen reference database.

In standard discovery studies, which aim to identify and quantify as many proteins as possible in a biological sample, large protein databases are typically used. These databases include all proteins with known sequences for the organism expected in the sample.

The most widely used repository of protein sequences is the UniProt Knowledgebase (UniProtKB), which consists of two sections:

- **Swiss-Prot** – manually curated and reviewed sequences, accompanied by rich annotations on protein function, properties, and relevant literature.
- **TrEMBL** – automatically annotated and unreviewed sequences that extend the coverage of Swiss-Prot.

Most protein sequences included in UniProtKB are derived from gene sequences submitted to related nucleotide sequence databases.

Highly similar proteins, including different forms of the same proteins (isoforms) may arise from alternative splicing, polymorphism and post-translational modifications (PTM) (Stastna and Van Eyk, 2012). Such proteins may originate from separate genes or from the same gene. Various isoforms of the same protein may play different biological roles (Stastna and Van Eyk, 2012; Bludau et al., 2021), so the ability to differentiate and quantify them is important for MS-based research. Accounting for presence of isoforms in the sample increases the computational cost of database search and amplifies statistical issues associated with multiple comparisons both at the PSM level and in downstream statistical analysis. For example, the addition of isoforms to human proteins database increases the number of proteins from 20,420 to 42,504 and the number of theoretical tryptic peptides from 665,736 to 702,368. The relatively low increase in number of tryptic peptides compared to increase in number of proteins is due to high sequence similarity between isoform proteins. Issues caused by this sequence overlap are described in the next section.

2.3.4 Protein inference: concepts and terminology

Let us consider the set of all peptides $\{R_m : m \leq M\}$ identified in a given MS experiment. These peptides match to proteins $\{P_k : k \leq K\}$. Matching is only based on amino acid sequences, so each peptide is a sub-sequence of at least one protein. The relationship between proteins and peptides can be represented with a bipartite graph in a following way. Proteins $\{P_k\}$ and peptides $\{R_m\}$ define the two classes of vertices. There is an edge between a protein P_k and a peptide R_m if the sequence of R_m is a subsequence of the sequence of P_k. We will refer to this graph as a peptide-protein graph. It can be represented by an adjacency $M \times K$ matrix $\mathcal{V}$ such that

$$\mathcal{V}_{mk} = \begin{cases} 1, \text{peptide m matches to protein k}, \\ 0, \text{peptide m does not match to protein k}. \end{cases}$$

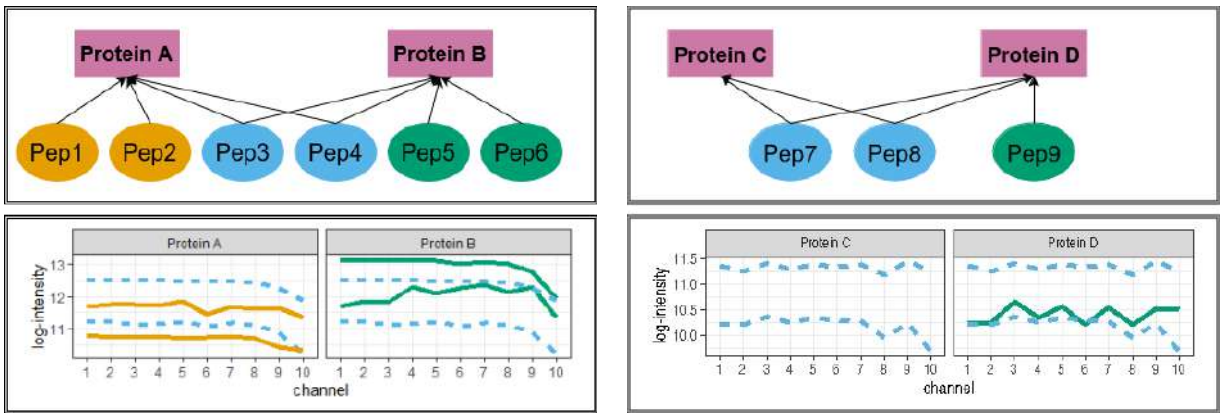
We will refer to $\mathcal{V}$ as the peptide-protein matrix. Usually, the peptide-protein matrix for an entire experiment is sparse as most peptides match to one or few proteins. Respective graph can be decomposed into a set of connected components (subgraphs). Proteins from different components are identified by disjoint sets of features. Peptides m^* such that $\sum_{k=1}^{K} V_{m^*k} = 1$ will be called **unique peptides**, as they only match to a single protein. Conversely, peptides m^* such that $\sum_{k=1}^{K} V_{m^*k} > 1$ will be referred to as **shared peptides**, as they match to multiple proteins. For a given peptide (or a spectral feature) f let us introduce the following notation to describe the set of all matching proteins among the K protein in a given cluster:

$$V(f) = \{k \in 1, \ldots, K : \text{feature } f \text{ matches Protein } k\}. \tag{2.11}$$

Due to the probabilistic nature of peptide identification, there is inherent uncertainty in determining the presence of peptides in a sample, which in turn introduces uncertainty in inferring protein presence. This uncertainty is particularly pronounced for proteins identified by a single peptide or by few unique peptides (Huang, J. Wang, et al., 2012). Shared peptides further increase uncertainty at the protein level, as any combination of matching proteins could theoretically be present in the sample (Huang, J. Wang, et al., 2012). Consequently, translating peptide-level evidence into protein-level conclusions is more complex than in cases where a protein is identified solely by unique peptides. Moreover, some peptide-level measurements are disproportionately affected by random noise due to interferences. Therefore, in practice, it is generally not advisable to use all identified peptides for protein-level quantification or to attempt to quantify every possible protein.

Figure 2.6 illustrates the protein inference problem on a minimal example. Proteins A and B both have unique peptides, and share two peptides. Hence, their presence in the sample is supported by unique evidence, but using information from shared peptides to quantify their abundance is a separate challenge. Together, proteins A and B form a cluster of proteins that can be analyzed separately from other proteins. Likewise, proteins C and D constitute another cluster. In this case, protein D has a unique peptide, but protein C is only identified by shared peptides. Such a protein is sometimes referred to as a subset protein (Nesvizhskii and Aebersold, 2005).

Based on presented considerations, we provide the following working definitions of the two steps in MS data analysis: protein inference and protein quantification. **Protein inference** is the task of selecting a subset of vertices and edges of the peptide-protein graph which corresponds to high-confidence peptide identifications and proteins whose presence in the sample can be inferred from this set of peptides. A protein inference method is any procedure which produces such reduced graph. Protein inference may be accompanied by **protein grouping**: creation of new protein label by concatenation of several *raw* protein labels (protein names that correspond to unique identifiers in the database). **Protein quantification** is the task of estimating either relative or absolute protein abundances. **Protein summarization** is the task of aggregating peptide-level information to protein-level



(a) (b)

Figure 2.6: **Simple examples of protein inference**. Rectangles are proteins, ovals are peptides. Arrows indicate protein membership of a peptide based on the amino acid sequence information. Green and orange: unique peptides. Blue: shared peptides. Quantitative profiles of each peptide across samples are shown below the graphs. Solid lines indicate unique peptides, dashed lines indicate shared peptides. Taken from Staniak et al., 2025.

estimates. A protein summarization method is any procedure which takes as input peptide-level quantitative data and returns estimates of protein abundances. Protein-level estimate for a given protein may consist of a single value per experiment, per replicate or per replicate and channel, depending on the context.

In principle, peptide identification (including any post-processing or validation) determines which peptides are present in a sample, while protein inference determines which proteins are present. Some protein inference methods, however, include a post-processing step for peptide-level data or allow users to decide whether to include or exclude shared peptides. For this reason, we consider the selection of peptides as part of protein inference, so that the output of a protein inference algorithm can serve directly as input for protein quantification.

Having provided the definitions, let us delve deeper into the inference and quantification problems. There are two well known sources of the protein inference problems: proteins identified by a single peptide (one-hit wonders) and peptides that match to multiple proteins (Huang, J. Wang, et al., 2012; Nesvizhskii and Aebersold, 2005). The former are characterized by high uncertainty regarding their presence in the sample due to possible identification errors. The latter are also referred to as degenerate or ambiguous peptides, and they pose a challenge not just at the inference step, but also for quantification. Currently, there is no widely accepted strategy for using shared peptides in abundance estimation (Bludau et al., 2021). Most methods quantify proteins or protein groups based on unique peptides (Kohler, Staniak, Tsai, et al., 2023; Goeminne, Gevaert, and Clement, 2018; Dermit et al., 2020). Conceptual issues with protein inference and quantification are so strong that some authors advocate forgoing protein summarization completely, and argue for peptide-level statistical analysis instead (Plubell, Käll, et al., 2022).

2.3.4.1 General protein inference approaches

Ambiguity introduced by shared peptides and one-hit wonders cannot be resolved based on sequence information in the peptide-protein graph only. While Huang, J. Wang, et al., 2012 thoroughly reviewed protein inference algorithms, The et al., 2018 introduced a useful categorization of inference algorithms. According to that classification, there are three major approaches to protein inference: **exclusion** which disregards evidence from shared peptides, **parsimony** which aims to find a minimal set of proteins that explain all identified peptides, and **inclusion** which attributes shared peptides to all their corresponding proteins. This last approach is capable of including additional information in the

inference process. External sources of information include: gene models (a priori information about protein clusters, (Gerster, Qeli, et al., 2010)), peptide detectability (sequence-dependent probability of detection by MS, (Huang, Gong, et al., 2013)), MS1 spectrum information (Spivak et al., 2012), PSM quality scores (for example probabilities of correct identification), and quantitative profiles of peptides (Price et al., 2007).

Moreover, The et al., 2018 characterized the performance of these three approaches on three controlled mixtures. One of these mixtures consisted of proteins that only shared peptides with other proteins in the mixture. In the other two mixtures, each protein shared peptides with another protein that was not present in the sample. They compared reported false discovery ratio (FDR) of protein inference algorithms to the actual FDR. All inferences principles performed similarly well when all proteins that corresponded to shared peptides were present in the sample. On the contrary, when some of the proteins assigned to shared peptides were not present in the sample, parsimony and inclusion approaches failed to control FDR. However, in principle, the exclusion of shared peptides results in loss of quantitative information about proteins: the number of peptides used to estimate protein abundances decreases, and proteins that are identified only by shared peptides cannot be quantified. Thus, there is a trade-off between protein discovery error rates and amount of quantitative information retained for downstream statistical analysis that needs to be taken into account while choosing protein inference approach.

Example 4 (Protein inference) *Applying the three general protein inference approaches to the example presented in Figure 2.6 would lead to the following outcomes. Exclusion approach would remove shared peptides from the analysis, resulting in a loss of protein C and reduction of number of available peptides for protein summarization by half for each remaining protein. The exclusion criterion is sometimes used even more stringently. A principle known as a two-peptide rule (criticized by (Serang et al., 2012)) proposes retaining only proteins with at least two unique peptides. Inclusion approach would assign every peptide to all matching proteins. Here, protein A would be characterized by peptides 1, 2, 3 and 4, and protein B - by peptides 3, 4, 5 and 6. In the other cluster, protein C would be identified by peptides 7 and 8, while protein D - by peptides 7, 8, 9, 10. With this approach no protein would be lost, but the abundances of shared peptides would be attributed in equal measure to all corresponding proteins. In case of divergence in quantitative patterns between shared and unique peptides for some proteins, this would result in quantitative bias. Parsimony approach would affect the cluster of proteins C and D, either by excluding protein C, or creating a new protein {C,D}. Similarly to inclusion, parsimony can group into a same protein label peptides with different quantitative patterns.*

2.3.4.2 Notion of uniqueness

It is important to note that the problem with shared peptides cannot be solved by technological advancement, as it is caused by inherent similarity of sequences between proteins (sequence homology). It is related to protein families (groups of evolutionarily-related proteins), protein variants (similar proteins originating from one gene or gene family), or, sometimes, redundant entries in the protein database. The shared peptides can constitute over 50% of all the possible peptides in the experiment, when all such events are considered (Schork et al., 2022; P. Wilmarth, 2020; Madhira, 2016). We will discuss the single non-biological and non-technological factor in the shared peptides problem: choice of a protein database.

Let us recall the example of two databases describing human proteome: canonical and isoforms. We stated that the larger database consists of over 50% more proteins and generates 5.5% more tryptic peptides, so from the peptide identity perspective, the difference between the two protein sets is small. However, the difference is significant from the perspective of peptide membership. 92.% of tryptic peptides generated from the canonical database are unique to their respective proteins. Meanwhile, only 44.4% of tryptic peptides based on the database which includes protein isoforms match to a single

protein. Hence, in this case, over half of all candidate peptides are shared by multiple proteins, if the larger database is used. These proportions remain similar if we restrict our attention to peptides that consist of at least 6 amino acids, which is typical in MS data analysis (96.4% and 46.1% of unique peptides, respectively). Thus, firstly, uniqueness of a given peptide depends on the database choice, and, secondly, this choice determines the severity of the protein inference issue. Whenever sample processing and MS instruments enable identifying and distinguishing protein isoforms present in the sample, appropriate databases should be used.

2.3.5 Modeling post-translational modifications sites

Post-translational modifications produce isoforms of native proteins. They cause small changes in mass of proteins. Hence, with enough resolving power and high mass accuracy, modified variants of proteins can be identified using mass spectrometry. In practice, this is achieved by time-of-flight and ion trap MS platforms. As a consequence, mass spectrometry is commonly applied to discover novel PTMs without strict assumptions about the kind or degree of modifications (Hermann, Schurgers, and Jankowski, 2022; Mnatsakanyan et al., 2018). While all MS strategies can be applied to PTM studies, we will focus on bottom-up experiments. One of the advantages of bottom-up approach in this problem is the smaller molecular mass of peptides compared to alternative approaches, which makes the relative molecular mass change of PTMs higher, reducing the technical requirements, including mass accuracy (Hermann, Schurgers, and Jankowski, 2022).

The general goal of mass spectrometry-based relative PTM quantification is to assess changes in occupancy of a PTM site across biological conditions. One of the major challenges to quantification is the confounding between changes in abundance of PTMs and changes in overall protein abundance (Kohler, Tsai, et al., 2023; Demeulemeester et al., 2024).

Kohler, Tsai, et al., 2023 provided a clear framework for designing PTM quantification studies, which we adopt as a conceptual guide for modeling such experiments. A brief overview of this approach is given in Section 2.4.

The main goal is to test whether the abundance of a PTM site differs between conditions. For example, in a simple case with two groups, i and j, let μ_i and μ_j represent the average log-abundance of the PTM site in each condition. The hypothesis test can be written as:

$$H_0 : \mu_i - \mu_j = 0 \quad \text{vs} \quad H_A : \mu_i - \mu_j \neq 0,$$

where H_0 states that there is no difference between the groups.

However, changes in PTM abundance can be influenced by changes in the overall protein abundance. To account for this, Kohler, Tsai, et al., 2023 proposed adjusting the hypothesis by removing the protein-level effect. Let μ_k^P denote the log-abundance of the protein corresponding to the PTM site in condition k. The adjusted test then becomes:

$$H_0 : (\mu_i - \mu_i^P) - (\mu_j - \mu_j^P) = 0 \quad \text{vs} \quad H_A : (\mu_i - \mu_i^P) - (\mu_j - \mu_j^P) \neq 0.$$

This adjusted formulation tests whether the PTM site changes independently of changes in the overall protein abundance.

Similarly to proteome profiling, observations in bottom-up studies are made at the peptide-level. Thus, estimation of PTM site-level quantities requires aggregation of quantitative information over multiple peptides. Moreover, not only multiple peptides carry the same modification site, but a single peptide may carry multiple modifications. Hence, the relationship between PTM sites and peptides is analoguous to the relationship between proteins and peptides in proteome profiling studies. Peptide that only carry a single modification can be considered unique to that modification's site. Peptides with multiple modifications can be treated as shared between corresponding PTM sites. Such peptides have long been recognized as a challenge in estimation of change in site occupancy (Mayya and

Han, 2009). Hence, it is common to only use unique peptides, similarly to global proteome profiling experiments. However, shared peptides (multi-site peptides) are treated differently. While there is again no generally accepted strategy for PTM site quantification with shared peptides, it is common to concatenate multiple sites found on the same peptide into a single artificial site. Then, such a site can be quantified based on all peptides that carry this particular set of modifications. Such peptides are treated as unique to this new PTM site.

2.3.6 Hydrogen-deuterium exchange studies

Hydrogen-deuterium exchange mass spectrometry (HDX-MS) takes advantage of the mass difference between hydrogen and its heavier isotope, deuterium (D). In proteins, amino acids are connected by peptide bonds, each of which contains a hydrogen atom attached to a nitrogen (the amide hydrogen). These hydrogens can exchange with deuterium when the protein is exposed to heavy water (D_2O). Since deuterium is approximately 1 Da heavier than hydrogen, this exchange increases the mass of the molecule. Importantly, every amino acid except proline has an amide hydrogen that can participate in this exchange.

In continuous labeling experiments, proteins are incubated in D_2O for an extended period of time and deuteration level is recorded as a function of labeling time. Hence, it is possible to study the hydrogen-deuterium exchange by measuring differences in masses of intact proteins or peptides. Mass spectrometry can be applied to this end, and information acquired with it may be used to describe the protein structure.

The HDX-MS process can be modeled using the Linderstrøm-Lang framework (James et al., 2021; Stofella, Grimaldi, et al., 2024; Linderstrom-Lang, 1955). Each amide hydrogen can be either accessible for exchange (open state) or protected (closed state). Accessibility is influenced by global unfolding and local conformational fluctuations. Exchange occurs only in the open state at a *chemical exchange rate* k_{ch}, which provides structural information about the protein.

$$H_{cl} \underset{k_{cl}}{\overset{k_{op}}{\rightleftarrows}} H_{op} \xrightarrow[D_2O]{k_{ch}} D_{op} \underset{k_{op}}{\overset{k_{cl}}{\rightleftarrows}} D_{cl} \tag{2.12}$$

The total H/D exchange rate k_{HDX} can be derived by considering the fraction of time an amide spends in the open state. Assuming the amide is mostly closed, the steady-state fraction of open hydrogens is approximately $\frac{k_{op}}{k_{cl}+k_{ch}}$. Multiplying by the chemical exchange rate k_{ch} gives the classical formula:

$$k_{HDX} = \frac{k_{op}k_{ch}}{k_{cl} + k_{ch}}. \tag{2.13}$$

Intuitively, the overall exchange rate depends on both how frequently the site opens (k_{op}) and how fast exchange occurs once it is open (k_{ch}), with the closing rate k_{cl} limiting the available time for exchange.

Further simplifications lead to two commonly discussed regimes:

- **EX1 regime**: $k_{ch} \gg k_{cl}$, so $k_{HDX} \approx k_{op}$ (exchange is fast once the site opens).
- **EX2 regime**: $k_{cl} \gg k_{ch}$, so $k_{HDX} \approx K_{op}k_{ch}$, with $K_{op} = k_{op}/k_{cl}$ (exchange is slow relative to site closing).

These regimes can be distinguished experimentally based on isotopic distributions in MS1 spectra: EX1 leads to bimodal patterns, while EX2 produces unimodal distributions.

HDX-MS experiments often use non-specific digestion or proteases that cut proteins at many different sites. This results in peptides with overlapping sequences. Since hydrogen-deuterium exchange

occurs at the level of individual amide hydrogens, many computational methods aim to infer exchange rates not just for whole peptides, but for individual residues as well. We review some of these methods in Section 2.4.4, after introducing additional terminology in Section 2.3.7.3.

2.3.7 Statistical design of mass spectrometry experiments

Having described the limitations of measurement methods and relevant types of experimental studies, we can describe the structure of experimental data and research objectives in statistical terms.

2.3.7.1 General experimental design

Mass spectrometry studies can address a wide range of questions about proteins, including their presence in biological samples, abundance levels, three-dimensional structures, post-translational modifications, and protein-protein interactions. In general, these studies characterize protein behavior under different conditions (e.g., healthy vs. diseased) across multiple biological samples obtained from various sources such as patients, tissues, or cells.

We will refer to biological replicates as samples derived from different subjects or sources, and technical replicates as multiple measurements or preparations of the same biological sample. A single mass spectrometry measurement is referred to as a run, so each experiment consists of a number of MS runs.

Both repeated measures and group comparison designs are common in proteomics. In a repeated measures design, samples from the same subject are collected under different conditions or at different time points. In a group comparison design, conditions are compared using samples from different, non-overlapping sets of subjects.

2.3.7.2 Data structure in a run

In bottom-up proteomics, each run measures a set of peptide ions. A single peptide can appear in multiple charge states, which gives it different chemical properties. Each charged form of a peptide is usually treated separately. These charged peptide ions are often referred to as spectral features—signals detected in MS1 spectra with a specific mass-to-charge ratio (m/z) and retention time, representing one observable form of a peptide. Multiple spectral features can correspond to the same peptide sequence, reflecting different charge states, isotopic variants, or modifications. For simplicity, we will generally refer to them simply as peptides, since all charged variants of a peptide correspond to the same sequence and match the same set of proteins.

Each peptide ion is observed across multiple MS1 spectra, often identified with the help of MS2 spectra, corresponding to different retention times. The information from these spectra can be aggregated, filtered, or analyzed together to characterize the peptide ion.

Figure 2.7 summarizes this data structure. Protein abundances that are typically of interest are not observable, but need to be recovered from peptide-level information. Intensities of peptides are observed in spectra recorded over retention time. In each MS1 spectra, multiple peptides are found, possibly with various charge states. Each peptide ion is characterized by several isotopic peaks.

The same peptide in different charge states is usually treated as a separate entity, since differences in ionization can affect its chemical properties and quantitative behavior. Information from multiple spectra corresponding to the same peptide ion can be combined in various ways. Typically, the intensity of a given m/z range (corresponding to a single peptide ion) is plotted over retention time to create an extracted ion chromatogram (XIC). This plot is then smoothed and aggregated into a single value that represents the intensity of the spectral feature in a particular run (R. Smith and Tostengard, 2020).

A single XIC may include multiple peptide-spectrum matches (PSMs)—instances where individual MS/MS spectra have been matched to the same peptide sequence. Usually, each peptide ion is

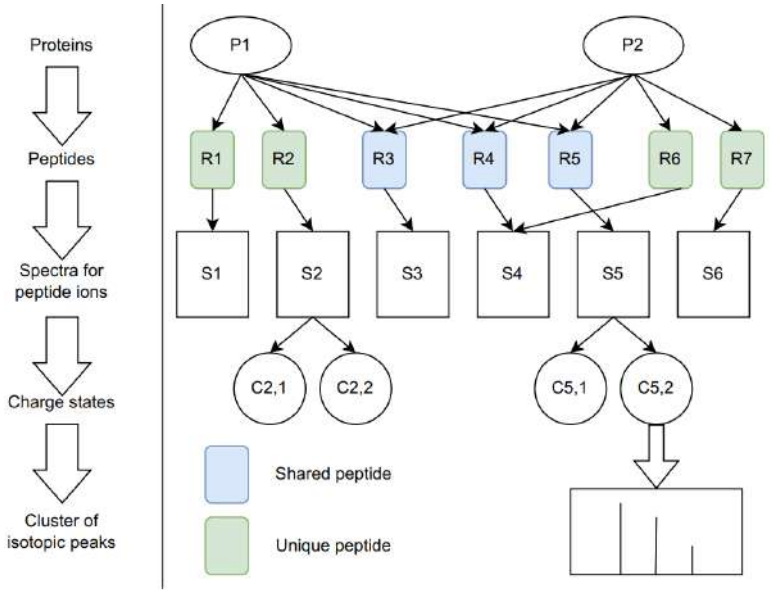


Figure 2.7: Structure of MS data in a single run.

summarized by a single numerical value per run, or by a few values corresponding to its PSMs. More complex data structures that retain less aggregation are also possible, depending on the analysis strategy.

2.3.7.3 Estimands: parameters and comparisons of interest

Relative protein quantification is possible due to the proportionality of observed intensities of peptide ions to their abundances. However, due to differences in ionization properties, intensities acquired for different peptides are not directly comparable (Ong and Matthias Mann, 2005). With consistent sample preparation, intensities acquired for the same peptide in different runs can be used to compare abundance of peptides in various biological conditions. Moreover, appropriate data processing, including normalization, enables aggregation of feature-level information into peptide-level summaries that describe abundance of each protein in a given sample (Goeminne, Argentini, et al., 2015). Thus, with appropriate sample and data processing, intensities can be compared, both at the protein- and peptide-level, depending on goals of a particular study. In this thesis, we focus on studies designed for protein-level conclusions. Let us also note that in practice, raw intensities are typically transformed with a logarithmic function to better conform to assumptions of standard statistical tools.

There are two main approaches for estimating differences in protein abundances across conditions (Goeminne, Argentini, et al., 2015). The first approach uses feature-level data for each protein to directly estimate treatment effects. Typically, these methods model the **log-intensities of features** as a linear function of a set of parameters, which includes a **treatment effect**. The treatment effect represents the change in protein abundance attributable to the experimental condition, after accounting for other sources of biological and technical variation. The exact set of parameters depends on the specific model used.

Let us denote the treatment effect for the i-th protein under the k-th condition as Treat_{ik}. The primary comparison of interest is usually the difference between conditions or groups of conditions. For example, when comparing two conditions, A and B, the **log-fold change** for the i-th protein is defined as:

$$\text{logFC}_i = \text{Treat}_{iA} - \text{Treat}_{iB}.$$

This logFC_i represents the expected difference in log-abundance of the i-th protein between the two conditions.

The alternative approach separates the analysis into two stages: first, feature-level intensities are aggregated into protein-level summary values—a step known as **protein summarization**—and second, treatment effects are estimated based on these protein-level summaries. Summarization-based approaches therefore involve two models: one for the feature-level data used to perform the aggregation, and another for the protein-level data used to estimate treatment effects.

Feature-level models for summarization can range from simple sum- or mean-based aggregation to more robust statistical methods. The protein-level model is then analogous to the feature-based models described earlier, with the treatment effect explicitly included. Comparisons between conditions have the same interpretation as in feature-level methods. Examples of both feature- and summarization-based approaches are provided in Section 2.4.

It is also possible to compare arbitrary linear combinations of treatment effects using a **contrast**. Formally, a contrast is a vector C of coefficients that specifies which conditions, or combination of conditions, are being compared. For the i-th protein, the corresponding log-fold change for this contrast is defined as

$$\log_2 FC(C) = C \cdot \text{Treat}_i,$$

where Treat_i is the vector of estimated treatment effects for the i-th protein based on protein-level summaries. Here, the symbol $\cdot$ denotes the **dot product** of two vectors, which is computed as

$$C \cdot \text{Treat}_i = \sum_j C_j \left(\text{Treat}_i\right)_j.$$

This operation produces a single number representing the log-fold change for the specific linear combination of treatment effects defined by C.

With appropriate replication, it is possible to estimate the variability of estimated log-fold changes. Hence, it is possible to construct statistical tests for comparisons of interest. Testing changes the scientific question from effect size estimation to a binary decision problem. This binary problem is often called differential expression (DE) analysis, and corresponds to a statistical hypothesis problem with a null hypothesis that states no difference between average abundances of a given protein in different condition, and an alternative hypothesis of a difference. Proteins that exhibit a significant difference are commonly referred to as regulated. Particular form of such a test depends on assumed model and some examples will be given in Section 2.4.2.

Model 2.12 is fundamental to the **interpretation of HDX-MS data**, but they can also be interpreted without a direct reference to it. The goals of HDX-MS studies vary, and include, among other possible objectives, comparisons of H/D exchange levels between conditions such as a free protein and a bound protein (complex), and analysis of protein folding based on degrees of protection of various parts of the amino acid sequence (James et al., 2021). Hence, both pure estimation and hypothesis testing problems appear in the analysis of HDX-MS data.

Bottom-up HDX-MS data describe peptides, which are smaller than proteins characterized by a given study, but are not the smallest possible units for which exchange can be theoretically estimated. Each residue with an at least one exchangeable hydrogen can be characterized by its own exchange rate. Therefore, various descriptors of H/D exchange may be readily estimated from data for peptides, but more granular estimation is of interest. Often, the ability to quantify exchange for smaller segments of protein sequence is referred to as higher spatial resolution. From technical point of view, resolution can be increased with protein digestion and fragmentation technique, with the latter solution introducing another issue known as *scrambling* (James et al., 2021). From computational point of view, estimability of individual exchange rates depends on the overlap between sequences of peptides, number of experimental time points, and noise level.

A general solution to equations derived from model 2.12 involves a convex combination of two exponential functions (Hvidt and Nielsen, 1966). However, depending on exchange regime, two simplified expressions can be derived for the deuteration level of a given residue. In the EX1 regime,

deuteration at time t, denoted by $d(t)$ can be approximated by $d(t) = 1 - \exp(-k_{op}t)$. Analogous formula for the EX2 regime is given by $d(t) = 1 - \exp\left(-\frac{k_{int}}{P}t\right)$, where the constant P defined by $P = \frac{k_{cl}}{k_{op}}$ is known as the protection factor. Estimating exchange rates that appear in these formulas for deuteration is the primary goal of the analysis of HDX-MS data. This is commonly done by fitting various models to experimental data based on calculated deuterium uptakes.

Information about H/D exchange in HDX-MS is typically summarized into a single value per peptide ion and time point, describing deuterium uptake in either absolute or relative terms (Stofella, Grimaldi, et al., 2024). This summarization is usually based on the isotopic distribution observed in the MS1 spectrum.

One common approach is to convert the isotopic peaks into a single value by calculating the intensity-weighted average of their m/z values. Alternatively, some analyses use only the monoisotopic peak and take its m/z value as the summary. Once a single m/z value m_z is assigned to a peptide ion at a given time point, its neutral mass m can be computed as

$$m = m_z \cdot z - z,$$

where z is the charge state (time index omitted for clarity), as ionization adds z protons to the neutral peptide. Let $m_{0\%}$ denote the mass of the undeuterated peptide. Then the *absolute uptake* is

$$m - m_{0\%}.$$

To express uptake in relative terms, one may use either the theoretical maximum number of exchangeable sites or an empirical estimate obtained after very long deuteration (e.g., 24 hours). Let $m_{100\%}$ denote this fully deuterated mass. The *relative uptake* is then calculated as

$$\frac{m - m_{0\%}}{m_{100\%} - m_{0\%}}.$$

Alternatively, the full isotopic distribution can be used to characterize H/D exchange, rather than reducing it to a single uptake value. Existing approaches to modeling exchange dynamics are discussed in Section 2.4.4. In general, HDX-MS analyses aim to estimate either exchange probabilities or protection factors, which describe how quickly individual amide hydrogens undergo exchange. Estimating these quantities requires either (i) first extracting peptide-level uptake values and modeling them, or (ii) directly modeling the full isotopic distributions.

2.4 State-of-the-art proteomics data analysis methods

This section describes a selection of relevant existing methods for protein quantification, including quantitative information in the protein inference process, estimating rates of hydrogen-deuterium exchange based on MS data, and quantification of post-translational modifications. We summarize common elements and limitations of described methods.

2.4.1 Protein inference assisted by quantitative data

While most protein inference algorithms rely on the amino acid information (Huang, J. Wang, et al., 2012), several recent approaches advocated for using quantitative information.

Quantifere (Lukasse and America, 2014) aims to increase protein coverage compared to exclusion or parsimony. This approach takes as input a result of parsimony-based inference (i.e. minimal set of proteins required based on data appropriately filtered to remove low-quality observations and proteins with low amount of available information. Proteins inferred by parsimony are classified as *primary*. *Secondary* proteins are selected based on correlation clustering of quantitative profiles of

peptides according to the following algorithm. Here, a quantitative profile is defined as a sequence $(X_{ir})_{r=1}^{R}, r = 1, \ldots, R$, where X_{ir} denotes a standardized intensity of i-th peptide in r-th MS run.

Peptides that share at least one matching protein are referred to as *sibling* peptides. Moreover, a distinction can be made between peptides that match to the same set of proteins (*full siblings*) and peptides that originated from different sets of proteins (*half siblings* or *not siblings*). The underlying assumption of Quantifere is that full sibling peptides would be characterized by high correlation, while the others by lower correlation. Additionally, it is said that a peptide is *foreign* to a protein if it is a sibling to one of its peptides but it does not match that protein.

Groups of sibling peptides are clustered based on their quantitative profiles using agglomerative hierarchical clustering with average-based linkage and $1-$Pearson's correlation as a distance measure. Clustering result depends on a user-selected correlation threshold which is used as a stopping criterion: a cluster grows only as long as the average correlation between its members is above this threshold.

Output of clustering then helps adjust the original inference output. Firstly, secondary proteins with at least one peptide which does not cluster with a foreign peptide are labeled as *inferred*. Corresponding peptides are in turn labeled as *inference peptides*. During an additional verification step, a peptide may lose this status if its presence can be explained with a smaller set of secondary proteins. Likewise, secondary proteins that do not have matching *inference* peptides after this additional filtering, lose their *inferred* status.

Authors proposed an additional scoring method for secondary proteins to increase the method's robustness to falsely identified and noisy peptides. In the proposed workflow, primary proteins are summarized by a sum of uniquely matching peptides, but no method for using shared peptides to quantify secondary proteins is implemented.

PeCorA (Peptide Correlation Analysis, (Dermit et al., 2020)) aims to improve on feature filtering methods by using discordant quantitative patterns to describe proteoforms rather than simply remove corresponding peptides.

Compared to Quantifere, PeCorA requires additional annotation of biological conditions for MS runs. Intensities can be pre-processed by filtering low values. Then, different charge states of each peptide are sum-aggregated and resulting intensities are log-transformed. log-intensities are median-normalized across MS runs. Moreover, peptide intensities are centered by subtracting the mean of control group's intensities.

Aggregated and normalized intensities are used to compare quantitative profile of each peptide to all other peptides matching to a protein that it originated from. The quantitative profiles are in fact summarized by their slopes across conditions. For a fixed peptide i, this is achieved by fitting a model

$$y = \beta_0^i + \beta_1^i X_1^i + \beta_2^i X_2 + \beta_3^i X_1^i X_2 + \varepsilon$$

where y denotes a vector of peptide-level log-intensities for a given protein, ε is random Gaussian noise under standard assumptions, β_0^i is an intercept, β_2^i is a condition parameter (X_2^i describes treatment group), and $X_1^i = 1$ for rows corresponding to peptide i, $X_1^i = 0$ otherwise. Consequently, parameter β_3^i describes the interaction between a biological condition and a peptide. Thus, it can be used to compare the consistency of this peptide's quantitative profile to other quantitative profiles matching to this protein. P-values for testing the significance of this parameter for each peptide are recorded and corrected with the Benjamini-Hochberg procedure within the set of peptides matching to the same protein.

While this procedure was originally applied only to unique peptides, the same principle can be applied to shared peptides, for example as post-processing of inclusion inference results. In this case, outlying quantitative patterns may be interpreted as separate protein isoforms.

COPF (COrrelation-based functional ProteoForm assessment, (Bludau et al., 2021)) defines a functional proteoform group, i.e. a group of peptides that are both derived from a same gene and have co-varying quantitative profiles. Such a group may or may not correspond to a particular proteoform. COPF approach assumes that if only one protein isoform is expressed or multiple expressed isoforms

share similar quantitative patterns, this pattern should be observed consistently at the peptide-level. However, if multiple protein isoforms are expressed in different ways, there corresponding peptides can be decomposed into groups with highly correlated quantitative profiles, each describing a different proteoform group.

This idea is implemented using hierarchical clustering. Quantitative profile of i-th peptide, $i = 1, \ldots, I$ is again defined as a sequence $(X_{ir})_{r=1}^{R}$ of log-intensities across MS runs $r = 1, \ldots, R$. For each protein, all pairwise Pearson's correlations cor_{ij} between profiles of peptides i and j are calculated, $i, j \in 1, \ldots, I$ Then, $1 - \text{cor}_{ij}$ defines a distance metric which is used in hierarchical clustering of peptides matching to a given protein. The resulting tree is cut into two clusters with at least two peptides per cluster. Based on such clustering, a proteoform score PS_k for k-th protein is calculated by comparing within-cluster correlation r_w to across-cluster correlation $r_{a,k}$ via the difference $PS_k = r_w - r_{a,k}$. Within-cluster correlation is estimated by a minimum of peptide correlations averaged for each cluster. Simple average of correlations over protein's matching peptides serves as an estimate of across-cluster correlation. The underlying assumptions states that proteoforms with different quantitative patterns will have higher scores than proteins that consists of a single proteoform or without discordant patterns across proteoforms. Authors proposed a test statistic and a method of p-value computation which, in tandem with a multiple testing correction, can be used to asses statistical significance of candidate proteoforms.

VIQoR (Tsiamis and Schwämmle, 2022) is an approach that facilitates both protein inference and differential analysis of protein abundance. It aims to improve on a Diffacto (B. Zhang et al., 2017) algorithm and possesses two notable features. Firstly, soft-parsimony approach to protein inference is capable of inferring some subset proteins by concatenating them with their superset proteins (i.e. protein grouping). More importantly, VIQoR uses quantitative profiles of peptides to estimate weight which are then used to aggregate peptide-level data using the following workflow.

Peptide-level intensities are log-transformed, centered and normalized across MS runs. Then, factor analysis using fast-FARMS algorithm (B. Zhang et al., 2017) is applied to assign protein-specific weights to shared peptides. Then, peptides are filtered based on user-specified minimum weight threshold. Finally, peptide-level intensities are aggregated to peptide-level estimates, which can be used to calculate log-fold changes. Summarization can be done either with regular or weighted summation. Visualization tools implemented in VIQoR support comparisons between two conditions in balanced designs.

Unlike VIQoR, the original fast-FARMS-based method Diffacto used weights to aggregate log-fold changes calculated based on peptide-level profiles rather than summarizing peptide intensities to protein-level abundances.

All presented approaches assume that peptide originating from the same protein isoform share consistent profiles and typically aim to discover sets of peptides with similar quantitative patterns and attribute them to either a single protein or a concatenation of several isoforms.

In our terminology, Quantifere and COPF work with subsets of protein clusters: COPF focuses on a single protein label, while Quantifere peptides that share protein matches. The former strategy silently assumes that a small database was used for peptide identification and there are possibly multiple protein isoforms under the same label. Similarly, PeCorA searches for discordant quantitative profiles for each protein separately. VIQoR reduces the set of proteins in a cluster by grouping proteins. Hence, each method simplifies the full peptide-protein graph, with VIQoR using the most information out of the described methods. Each method provides a way to enrich protein inference results of parsimony or inclusion approaches.

PeCorA method is capable of removing discordant features from the analysis. Alternative approaches only perform peptide filtering based on pre-defined metrics that describe the quality of quantitative information such as signal intensity or number of non-missing values.

As a protein inference post-processing method, Quantifere does not provide a protein quantification method nor a way to include shared peptides in such a process. Similarly, PeCorA is foremost

a peptide filtering method and in principle it can be used with any differential analysis tool. Authors of the original manuscript used MSstats (Kohler, Staniak, Tsai, et al., 2023) for this purpose. COPF strategy was designed for a certain type of MS experiments, but is broadly applicable. For differential analysis, authors used the analysis of variance (ANOVA) model with an interaction of condition and proteoform as groups. VIQoR assumes that data come from a balanced experiments and facilitates comparisons between two conditions, but does not provide a way to determine statistical significance of the results.

Additionally, some of the methods require additional aggregation (such as summation over different charge state) or force certain kinds of normalization. Moreover, they require specific input formats and provide types of access that may not be optimal for programmatic or large-scale use.

2.4.2 Protein quantification

Protein inference methods provide researchers with a set of proteins or proteins groups present in the measured samples, together with their peptides. For quantitative studies, the next step in data interpretation is estimation of protein-level changes in abundance between conditions of interest. This can be done based on peptide-level data or after aggregation of peptide abundances into a single quantity per protein label and sample (referred to as summarization). In this section, we describe statistical methods proposed for the quantification task.

2.4.2.1 Peptide-based models

Goeminne, Argentini, et al., 2015 evaluated three peptide-based models that use unique peptides to directly estimate treatment effects. Most general model is defined for each protein i separately by

$$y_{ijklm} = \text{pep}_{ij} + \text{treat}_{ik} + \text{sample}_{ikl} + \text{exp}_{ikl} + \varepsilon_{ijklm} \tag{2.14}$$

where y_{ijklm} denotes log-transformed intensities of j-th peptide in m-th spectrum of l-th experiment and k-th treatment (condition). Error term ε_{ijklm} has a normal distribution with mean 0 and variance σ^2. Terms pep_{ij}, sample_{ikl} and exp_{ikl} model peptide-, MS run- and experiment-specific variation. The former effects describes technical variation due to laboratory, instrument and replicate. Effects treat_{ik} are used for comparisons between conditions. Mixed effects version of the model assumes that sample_{ikl} is a random effect with a normal distribution $\mathcal{N}(0, \sigma^2_{\text{sample},i})$. Alternative, reduced fixed-effects model drops this term altogether. This model is used to directly estimate condition effects.

Peptide-based models have been used to facilitate inclusion of shared peptides in protein quantification. Jin et al., 2008 proposed a method of relative protein quantification which estimates a common abundance factor for all proteins from the same cluster. In this work, protein clusters were referred to as Peptide-Sharing Closure Groups (PSCG). The method is suitable for designs with two groups, as it estimates the common abundance factor of a treatment group relative to a control group as

$$\begin{aligned} \widehat{C}_t &= \sum_{i=1}^{k} c_{it}\mu_{it}, \\ c_{it} &= \frac{1}{\sigma_{it}^2 \sum_{i=1}^{k} \frac{1}{\sigma_{it}^2}}, \end{aligned}$$

where μ_{it} and σ_{it} denote, the mean and standard deviation, respectively, of the log-abundances of an i-th peptide in a t-th PSCG. By the definition of a PSCG, this model is incapable of differentiating between differently expressed proteins within a cluster.

Bukhman et al., 2008 also proposed a method based on reducing the set of quantified proteins. In this approach, a single *representative* protein (so-called *anchor* protein) from each cluster is chosen.

Quantification is only done for this subset of proteins via the following model

$$y_{ij} = \gamma_i + \phi_i \sum_k \delta_{ik}\theta_{jk} + \varepsilon_{ij},$$

where y_{ij} denotes the signal intensity of peptide i in sample j, while γ_i and ϕ_i are the background signal and *sensitivity* of peptide i, respectively. The parameter ϕ_i models differences in physicochemical and detection properties. It allows the proportionality between the MS signal and abundance to vary for different peptides. Coefficient δ_{ik} is the (i,k) entry of a peptide-protein matrix, thus $\delta_{ik} = 1$ if and only if peptide i matches to protein k. Parameter θ_{jk} denotes the unknown abundance of protein k in sample j, and ε_{ij} is the residual random error. Again, such model cannot differentiate the abundances of proteins from the same cluster.

Blein-Nicolas et al., 2012 proposed an extension of this model, named the *all-proteins* model:

$$\log(y_{itr}) = \log\left(\sum_k \delta_{ik}\exp(\theta_{kt})\right) + D_i + B_r + C_{tr} + \varepsilon_{itr},$$

where y_{itr} denotes the measured intensity, $\exp(\theta_{kt})$ is the abundance of protein k, δ_{ik} is an element (i,k) of the peptide-protein matrix, D_i is a random peptide effect, B_r is a random effect of the biological variation in replicate r, C_{tr} is a random effect of the technical variation in treatment t, replicate r, and ε_{itr} is the residual random error. Estimation of this model is computationally challenging because of the nonlinear term $\exp(\theta_{kt})$) and thus Bayesian methods were applied. The complex random effects structure requires a substantial sample size for estimation. The logarithmic transformation imposes a different error structure than the other models: multiplicative rather than additive on the original intensity scale.

More recently, Jacob, Combes, and Burger, 2018 proposed a procedure for testing differential abundance based on a linear approximation of a similar model:

$$\log(y_i) \sim \mathcal{N}\left(\sum_{k=1}^{p} \delta_{i,k}\theta_k + \alpha_i, \sigma^2\right)$$

where $\delta_{i,k}$ again denotes the indicator of peptide i matching protein k, $\log(y_i)$ is a log-intensity of i-th peptide, θ_k denotes the abundance of protein k, and α_i is a peptide-specific effect.

Yet another model was introduced in Gerster, T. Kwon, et al., 2014:

$$y_i = \alpha + \beta \sum_k \delta_{ik}\theta_k + \varepsilon_i,$$

where y_i is a peptide abundance, $\delta_{i,k}$ denotes the indicator of peptide i matching protein k, θ_k denotes the abundance of protein k that is assumed to be normally distributed with mean μ and variance 1, and ε is a residual random error. The common distribution assumption implies that a priori, abundances of all proteins oscillate around a single value μ.

2.4.2.2 Summarization-based approach: MSstats

Multiple methods for protein summarization exist, ranging from simple means or medians of peptide abundances in each sample, to model-based approaches such as MSqRob (Sticker et al., 2020), Triqler (Truong, The, and Käll, 2023) and MSstats (Kohler, Staniak, Tsai, et al., 2023). All these methods assume that proteins or protein groups are characterized by uniquely matched peptides. Below we describe the statistical framework for protein-level summarization of experiments with TMT labeling in the open-source software MSstatsTMT (Huang, Choi, et al., 2020). As a comparison between different summarization or differential abundance detection methods is outside the scope of this work,

we do no describe the alternative methods. Such comparisons were done both in the original papers such as Kohler, Staniak, Tsai, et al., 2023; Sticker et al., 2020, and in review articles such as Bai et al., 2023; Lin et al., 2022.

Consider a protein (or a protein group) characterized by $f = 1, \ldots, F$ spectral features, i.e. peptide ions matched to that protein or group. The experiment profiles $b = 1, \ldots, B$ biological samples from each of $g = 1, \ldots, G$ groups (also called conditions), in $c = 1, \ldots, C$ channels for each of $m = 1, \ldots, M$ mixtures. For simplicity, we assume that the experiment has no technical replicates. Experiments with $M > 1$ mixtures typically dedicate one channel per mixture for reference material used for normalization (Plubell, P. A. Wilmarth, et al., 2017). For the purposes of protein summarization, MSstatsTMT models each protein and each mixture separately with a linear model

$$X_{fc} = \mu + Feature_f + Channel_c + \varepsilon_{fc}, \tag{2.15}$$
$$\sum_{f=1}^{F} Feature_f = 0, \sum_{c=1}^{C} Channel_c = 0,\ \varepsilon_{fc} \overset{iid}{\sim} \cdot(0, \sigma^2_\varepsilon)$$

where X_{fc} denotes the observed $\log_2$-intensity of feature f in channel c, μ denotes the overall mean protein abundance, $Feature_f$ and $Channel_c$ denote the additive main effects of feature f and channel c, and ε_{fc} denotes independent, identically distributed and non-systematic noise. The model is linear in parameters μ, $Feature_f$ and $Channel_c$. MSstatsTMT estimates the parameters using robust Tukey Median Polish (TMP) (Kohler, Staniak, Tsai, et al., 2023). Finally, the estimate of protein abundance in channel c is

$$Y_c = \widehat{\mu} + \widehat{Channel}_c,\ c = 1, \ldots, C \tag{2.16}$$

Y_c serves as input to the downstream differential analysis. The indices of proteins and mixtures in Equations (2.15) and (2.16) are omitted for simplicity.

2.4.2.3 Differential analysis based on summarized data

Once protein abundances are summarized, the next step specifies a statistical model for the protein-level summaries. The model characterizes the available sources of variation, and serves as the basis for tests for differential abundance. Many statistical models have been proposed, e.g. DeqMS (Y. Zhu et al., 2020), MSqRob (Sticker et al., 2020) or MSstats (Kohler, Staniak, Tsai, et al., 2023). They were reviewed in detail in Bai et al., 2023. Below we describe MSstatsTMT, which flexibly accommodates diverse experimental designs in experiments with TMT labels (Huang, Choi, et al., 2020; Huang, Staniak, et al., 2023). MSstatsTMT fits a separate linear model to each protein summary. In its most general form for group comparison designs it decomposes the variation in protein-level abundances in the following way:

$$Y_{gbm} = \mu + Mixture_m + TechRep(Mixture)_{t(m)} + Condition_g + Subject_{mgb} + \varepsilon_{mtgb} \tag{2.17}$$

where

$$\sum_{g=1}^{G} Condition_g = 0, Mixture_m \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_M), Subject_{mgb} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_S),$$
$$TechRep(Mixture)_{t(m)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_T), \text{and } \varepsilon_{gbm} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

As another example, consider a more complex design that profiles biological replicates across multiple groups, collects repeated measurements on the biological replicates in time, and allocates measurements from each biological replicate to its own mixture. MSstatsTMT fits the model

$$Y_{ctm} = \mu + ConditionTime_{gt} + Mixture_m + \varepsilon_{mct} \tag{2.18}$$

where $\sum_{ct} ConditionTime_{gt} = 0$,

$$Mixture_m \overset{iid}{\sim} \mathcal{N}(0, \sigma_M^2),\ \varepsilon_{gtm} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

In this notation, $ConditionTime_{gt}$ represents all the combinations of conditions and times, and $Mixture$ is confounded with $Subject$. For each model and each protein, tests of differential abundance specify a null hypothesis, e.g. $H_0 : Condition_g = Condition_{g'}$ or $H_0 : ConditionTime_{gt} = ConditionTime_{gt'}$. All the model parameters are estimated using restricted maximum likelihood. The parameter estimates and their standard errors are combined into t-statistics to derive p-values, which in turn are adjusted to control False Discovery Rate using the Benjamini-Hochberg correction Benjamini and Hochberg, 1995.

2.4.3 Relative PTM quantification

In addition to general modeling strategies based on linear mixed models, two tailored methods have been recently proposed for PTM summarization: msqrob2PTM (Demeulemeester et al., 2024) and MSstatsPTM (Kohler, Tsai, et al., 2023). Msqrob2PTM repeatedly uses peptides with multiple modifications to quantify each PTM site, effectively implementing the inclusion approach from protein inference. In contrast, MSstatsPTM combines the two modification sites into an artificial site (called concatenation), and estimates a separate log-fold change for this combination, effectively implementing protein grouping.

Similarly to proteome profiling, the next step is statistical modeling of the summarized abundances. To distinguish changes in PTM site occupancy from overall changes in protein abundance, msqrob2PTM normalizes feature-level $\log_2$-intensities by subtracting the estimated abundance of the unmodified protein in the sample. The normalized feature intensities are then modeled with a robust linear model that accounts for differences between biological conditions. In contrast, MSstatsPTM separately summarizes the modified and the unmodified features corresponding to a PTM site with the MSstats workflow (Equation (2.15)). It then fits a separate protein-level model (e.g., Equations (2.17-2.18)) to each summary to reflect the experimental design. Finally, the null hypothesis compares changes in the expected abundance of the PTM site between conditions to the changes of the unmodified protein.

2.4.4 Estimation of hydrogen-deuterium exchange rates

Several methods, both based on modeling isotopic distributions of observed peptides and on centroids, have been proposed to estimate segment- or residue-level H/D exchange parameters. In this section, we give a quick overview of a selection of these models. Stofella, Grimaldi, et al., 2024 provided a thorough review of revelant methods.

Saltzberg et al., 2017 proposed a Bayesian approach to estimation of residue-level deuterium uptakes. Deuterium uptake $D_{f,t}$ for peptide f in time t was expressed as

$$D_{f,t} = D_0(N_f - \sum_i e^{k_i t})$$

where D_0 denotes deuterium fraction of the exchange buffer, N_f is the number of exchangeable hydrogens for peptide f and the summation index i goes over all residues with an exchangeable hydrogen. Noise is modeled via a truncated Gaussian distribution. Likelihood function enables using data from multiple replicates. The approach was implemented in an open source tool and is also available as a part of HDX Workbench software tool.

Babić, Kazazić, and D. M. Smith, 2019 proposed an algorithm to estimate residue-level exchange probabilities from peptide-level isotopic distributions. Probability of the H/D exchange at time t for residue r_j is modeled as $1 - \exp(-k_j t)$. Then, the expected isotopic distribution of a given peptide at

a fixed time point is modeled according to Equation 4.1 (assuming total intensity equal to 1). Then, the expected peak intensities are compared to normalized observed intensities with the following loss function:

$$L((p)_{a,i}) = \sum_t \left[\sum_a (q(P_{a,t}) - q^{obs})(P_{a,t}) W_{a,t} (q(P_{a,t} - q^{obs}(P_{a,t}))^T \right]$$

where t denotes time points, P_a denotes peptides, W_a is a matrix of weights and $q(P_a), q^{obs}(P_a)$ denotes a vector of expected and observed isotopic probabilities $p_{a,i}$, respectively. Vector q may also denote a vector of exchange probabilities. Exchange probabilities are recovered from isotopic distributions via numerical deconvolution procedure that uses Fast Fourier Transform. The approach was implemented in publicly available Matlab scripts.

Z. Zhang, 2020 proposed modeling average deuterium content d_q of an amide at a particular exchange time via the equation

$$d_q = D_0 \left[1 - \exp\left(-\frac{k_{int}}{P} t_{ex} - k_{ch} t_{op} \right) \right]$$

where t_{ex} denotes time of the exchange, t_{op} is a time spent by the molecule in an open state (when exchange is possible), D_0 denotes the deuterium concentration in the labeling solution, k_{ch} is a chemical exchange rate and P is an amide-specific protection factor. Hence, $\frac{k_{int}}{P}$ is a H/D exchange rate constant. Segment-level probability distribution is determined via a convolution of residue-level probabilities. Then, Equation 4.1 is used to model peptide-level isotopic distributions. Expected isotopic peaks are compared to peaks predicted based on model parameters (protection factors) via the following loss function.

$$\chi^2 = \frac{1}{\nu} \sum_p \sum_m \frac{(I^{pred}_{p,m} - I^{obs}_{p,m})^2}{\sigma^2_{m,p}}$$

where for a given peptide p, $I^{pred}_{p,m}$ and $I^{exp}_{p,m}$ denote m-th predicted and observed isotopic peak, respectively. Parameters $\sigma^2_{m,p}$ denote variances of peaks, which are estimated based on replicate measurements at each time-point. The approach was implemented in the commercial MassAnalyzer software.

Earlier, Abzalimov and Kaltashov, 2006 applied the deconvolution approach to a special case of estimating deuterium exchange probabilities for a segment based on measured distribution for a second segment and concatenation of the two segments. The exchange distribution of the unobserved segment B was modeled as a convolution of the observed distributions of neighbouring segment A and the concatenation AB of both segments. The deconvolution of the unknown probability distribution for segment B was done using maximum entropy method.

Alternative methods based on centroided data have been proposed. Recently, Stofella, Skinner, et al., 2022 modeled fractional deuterium uptake defined defined for a peptide that consist of N residues by

$$D^{pred}(t, P) = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \exp\left(-\frac{k_{int,i}}{P_i} t \right) \right)$$

where t denotes time, $k_{int,i}$ is the intrinsic exchange rate and P_i denotes the protection factor. Corresponding observed uptake are defined by

$$D^{obs}(t) = \frac{D_t - D_0}{D_{FD} - D_0}$$

where t denotes time, and D_0, D_t and D_{FD} are intensity-weighted centroids of isotopic distribution of an undeuterated peptide, measured at time t, and at full deuteration (determined experimentally), respectively. Protection factors were determined by minimizing the following loss function:

$$C(\lambda, \{P_i\}) = \sum_j \sum_k w_{j,k} \left[D^{pred}(t_k, \{P_i\}) - D^{obs}(t_k) \right]^2 + \lambda \sum_i \left[\ln(P_{i-1}) - 2\ln(P_i) + \ln(P_{i+1}) \right]^2$$

where t_k denotes k-th time point and $w_{j,k}$ are weights. The second term of this loss function penalizes large differences between protection factors of adjacent residues, and λ denotes a penalty parameter. No publicly available implementation of this approach was provided. Similar deuterium uptake curve-based solutions were proposed by Seetaloo, Kish, and Phillips, 2022a, Skinner et al., 2019.

With the exception of Abzalimov and Kaltashov, 2006, all described method start with a residue-level model and aim to estimate the exchange rates for each amide. Babić, Kazazić, and D. M. Smith, 2019, Z. Zhang, 2020 and Stofella, Skinner, et al., 2022 recognized the underdetermination of this problem in case when no disjoint segment of observed peptides corresponds to a particular single residue. Babić, Kazazić, and D. M. Smith, 2019 provided conditions for estimability of a given set of residues based on a system of linear equations that describes the peptide-segment structure of data. These equations exploit the fact that the deuterium uptakes are additive. Stofella, Skinner, et al., 2022 used a clustering approach to determine sectors of the amino acid sequence for which common protection factors are estimated. Z. Zhang, 2020 used 20 solutions to the underdetermined system to evaluate their variability. Babić, Kazazić, and D. M. Smith, 2019 provided confidence intervals for the exchange probabilities based on a likelihood ratio test. Statistical assumption of the model followed a classical linear model. Similarly, Skinner et al., 2019 modeled observed uptake values as a sum of expected values and random noise under standard assumptions. Non-trivial noise structure was used by Saltzberg et al., 2017 in a Bayesian context.

2.5 Data

In this section, we describe biological data sets that we used to evaluate proposed modeling approaches. The first case study, protein degrader data, were first published in Staniak et al., 2025 which this thesis is largely based on. The remaining data sets were previously used in the literature and we restrict their description to most important facts. However, in all cases, we refer to the original articles for details of sample preparation and mass spectrometry measurement. Instead, we focus on the experimental design each study, peptide identification results, and relevant data processing steps. As simulation studies used the proposed models in data generation process, we describe them in relevant sections later of the Contributions chapter.

2.5.1 Protein degrader

2.5.1.1 Experimental design

This study evaluated BET bromodomain degradation by GNE-0011 BET binder in EOL-1 cells. Two groups were considered: control (DMSO) and treatment (labeled GNE-001). For each group, measurements were made after 0, 30, 60, 120, and 480 minutes to estimate changes in protein abundances over time. Different samples were used for each treatment time, resulting in a group comparison design. For each time and group only one biological replicate was used. The samples were labeled with TMT-10plex in a single TMT mixture.

2.5.1.2 Data acquisition and processing

MS/MS spectra collected on an Orbitrap Fusion Lumos Mass Spectrometer (ThermoFisher Scientific) coupled to an RSLCnano U3000 liquid chromatography system (ThermoFisher Scientific) were searched using the Comet search algorithm version 2017.01 (Eng, Jahan, and Hoopmann, 2013) against a concatenated target/decoy database comprised of the Swissprot human protein sequences (version 2017.08) and contaminants. Search parameters were as follows: a 50 ppm precursor ion mass tolerance; 1.0005 fragment bin tolerance; tryptic digestion with up to two missed cleavages; fixed modifications: carbamidomethyl on cysteine residues, TMT 10-plex on Lysine and the peptide

N-term; variable modifications: methionine oxidation, TMT 10-plex on tyrosine. PSMs were filtered to a 1% peptide FDR at the search level using linear discriminant analysis (Kirkpatrick et al., 2013). Next, PSM data across fractions were aggregated and subsequently filtered to a 2% protein FDR. Reporter ion intensities (MS3) were determined using the Mojave algorithm (Zhuang et al., 2013) with an isolation width of 0.5.

The original processing matched shared peptides to an arbitrarily selected single protein.

2.5.2 Thermal proteome profiling

2.5.2.1 Experimental design

The original study (Xu et al., 2021) had a two-fold goal. From a biological perspective, the study evaluated the response of protein targets in K562 cell lysate to Staurosporine (kinase inhibitor) treatment compared to a control group treated with DMSO. From a technical perspective, the study compared two approaches to measurements: Thermal Proteome Profiling (TPP) and OnePot.

The TPP experiment utilized a hybrid design: each biological sample was heated at 11 increasingly high temperatures. Unlike the repeated measures design, the comparisons were made between samples measured separately rather than between conditions evaluated for the same sample. Samples were treated with Staurosporine at 25$\times$ the concentration of DMSO, with two samples per condition. Each sample was labeled with a TMT-10plex. The lowest temperature was used as between-mixture normalization channel, while the highest temperature was not used in the analysis. The two experimental conditions were measured separately in two TMT mixtures. In this version of the experiment, all proteins were expected to decrease in abundance across increasing temperatures, but at at different rate depending on treatment.

The OnePot experiment utilized a group comparison design. In this case, all temperate-subjected aliquots of a same biological replicate were pooled prior toTMT labeling. The experimental conditions were defined by four concentrations of Staurosporine ($1\times, 5\times, 10\times, 25\times$) and one DMSO (control) group. Three biological replicates per condition were measured.

Pooled samples were labeled with a TMTPro-16plex in a single TMT mixture. One remaining TMT channel was used for normalization. Since the OnePot experiment used a larger sample size in terms of biological replicates, and evaluated a higher number of concentrations, it was expected to produce more accurate results.

2.5.2.2 Data acquisition and processing

In both variants of the study, mass spectra were acquired with an Orbitrap Fusion Lumos Tribrid mass spectrometer (Thermo Fisher Scientific), searched against Homo Sapiens Swissprot database (v2017-10-25) and processed using Proteome Discoverer 2.4 (Orsburn, 2021). Additional details of data acquisition and processing can be found in the original article.

Proteome Discoverer provides a protein inference option which produces several columns describing protein labels. Depending on the choice of a column, shared peptides can be attributed to a leading protein or to a concatenation of labels of matching proteins. The latter option was used originally, with a restriction to proteins that had at least one unique peptide.

2.5.3 Multi-site PTM

2.5.3.1 Experimental design

The original study Maculins et al., 2021 compared samples of primary murine macrophages uninfected and infected with *S. flexneri* at three time points: uninfected, early infection and late infection. The

goal was to quantify abundance of total protein and phosphorylation in wildtype (WT) and ATG16L1-deficient (cKO) samples. The study made 9 comparisons in a group comparison design: KO Early-WT Early, KO Late-WT Late, KO Uninfected-WT Uninfected, KO Early-KO Uninfected, KO Late-KO Uninfected, WT Early-WT Uninfected, WT Late-WT Uninfected, Infected-Uninfected, and KO-WT. The 22 biological samples were split between two 11-plex TMT mixtures. Mixture 1 consisted of one replicate of uninfected WT and two replicates of uninfected cKO, while Mixture 2 consisted of one replicate of uninfected cKO and two of uninfected WT. There was no normalization channel in this study. Hence, all the analyses proceeded without normalization. Quantification of phosphorylation required adjusting changes of modified peptides for changes in global protein abundance.

2.5.3.2 Data acquisition and processing

Spectra were acquired on an Orbitrap Fusion Lumos mass spectrometer coupled to EASY nanoLC-1000 or nanoLC-1200 (ThermoFisher) liquid chromatography systems, then searched against a UniProt mouse and Shigella flexneri protein sequences database and processed with the Mojave algorithm (Zhuang et al., 2013). Modifications of interest included phosphorylation on serine, threonine, and tyrosinem, and the localization was done using a modification of AScore algorithm (Beausoleil et al., 2006). Additional details of data acquisition and processing can be found in the original article.

In the original processing of modification sites, multiple modifications on a single peptide were concatenated into a new modification.

2.5.4 HVEM case study

2.5.4.1 Experimental design

The original study (Kuncewicz et al., 2019) investigated the interaction between a T cell molecule CD160 and the herpes virus entry mediator (HVEM). This interaction provides an inhibitory signal to T cells. One of the goals of the study was to identify the binding sites of CD160 responsible for interaction with HVEM. It was achieved by comparing the levels of H/D exchange of CD16-HVEM complex and a CD160 free protein. Higher protection from exchange in a given region upon protein binding indicated participation in the interaction with HVEM. Deuteration levels for these two groups (CD160-HVEM complex vs CD160 free protein) were measured at 5 time points: 10s, 1 min, 5 min, 25, 2 h. At each time point, four technical replicates were measured. Additionally, Additionally, measurements were done to estimate minimum and maximum levels of exchange, with the latter corresponding to 24 h deuteration.

2.5.4.2 Data acquisition and processing

List of CD160 peptides was created using a nondeuterated protein sample. Sample was digested using two proteases: protease type XIII and pepsin. Peptides loaded onto a LC column before measurements with a SYNAPT GS HDMS mass spectrometer (Waters, Milford, MA). Spectra were acquired in MSE mode over the m/z range of 50-2000. Specific setting of the mass spectrometer and additional details of sample processing can be found in the original paper. Peptides were identified by using the ProteinLynx Global Server software (PLGS, Waters, Milford, MA) against a randomized database. Deuteration leveles of identified peptides were calculated by using the DynamX 3.0 HDX-MS data analysis software (Waters, Milford, MA) after additional filtering based on intensity, mass errors and other criteria. Spectra were manully curated.

2.5.5 Milisecond-resolution HDX data

2.5.5.1 Experimental design

The original study (Kish et al., 2023a) aimed to showcase and validate an online flow rapid mixing and quenching HDX system capable of attaining a time resolution of 1 ms. The system was design for automated bottom-up studies characterized by reproducibility and repeatability. To valide this experimental approach, a large protein glycogen phosphorylase b (GlyPb) was analyzed.

The goal was to measures exchange kinetics of GlyPb. High sequences coverage (96.5%) was achieved, which enabled calculation of protection factors for the majority of protein sequence. H/D exchange was measured at 9 time points: 0.05, 0.15, 0.25, 0.35, 0.5, 1, 5, 30, and 300 s. Three replicates were used at teach time point.

2.5.5.2 Data acquisition and processing

GlyPb was dissolved, diluted and subjected to digestion by using pepsin. Peptides were separated on a LC column (Waters). Measurements were done on a mass spectrometer with a quadrupole time of flight mass analyzer (Synapt G2-Si HDMS QTOF, Waters). Mass spectra were obtained in the Waters HDMSE mode in an m/z range from 50 to 2000. Details of sample prepartion, in particular the labeling process, can be found in the original article.

Peptides were identified based on the HDMSE fragment data with ProteinLynx Global Server 3.02 (PLGS) (Waters). Deuterium uptakes were calculated with DynamX 3.0 (Waters) after filtering based on intensity, mass error, and other criteria. Spectra were manually curated. Protection factors were determined based on fitting various types of exponential curves to experimentally determined uptake curves.

Contributions

Chapter 3

Relative protein abundance estimation with shared peptides based on isobaric labeling data

3.1 Introduction

In this chapter, we restrict our attention to modeling multiple membership data acquired in TMT-based mass spectrometry studies. We propose to extend protein summarization in MSstatsTMT (see Equation (2.15)) for experiments with TMT labels to simultaneously estimate the abundances of proteins with shared peptides. Similarly to Quantifere (Lukasse and America, 2014), PeCorA (Dermit et al., 2020) and COPF (Bludau et al., 2021), we consider similarities between the feature-level profiles, however we do not attempt to cluster the profiles or assign them to an isoform. Similarly to VIQoR (Tsiamis and Schwämmle, 2022), we directly quantify the contribution of a peptide to protein-level summaries in form of weights, however we output not log-fold changes but full protein-level summaries compatible with statistical modeling of various experimental designs

3.2 Proposed model

Let us recall that following the MSstatsTMT approach, we refer to peptide ions as features, and to samples labeled with TMT reagents as channels. Often, they will correspond to different experimental conditions. Let us recall that $V(f)$ denotes the set of proteins matching to peptide f, as defined by Equation 2.11. Moreover, let us denote by X_{cf} the $\log_2$-intensity of feature f in channel c. We extend the MSstatsTMT summarization method by using the following model for each cluster of proteins in each TMT run:

$$\begin{aligned} X_{cf} = \mu \;\; &+ \sum_{k \in V(f)} Weight_{fk} \left(Protein_k + Channel_{kc} \right) \\ &+ \; Feature_f + \varepsilon_{cf}, \end{aligned} \tag{3.1}$$

under the typical linear model constraints

$$\sum_{k=1}^{K} Protein_k = 0, \; \forall_k \; \sum_{c=1}^{C} Channel_{kc} = 0,$$
$$\sum_{f=1}^{F} Feature_f = 0$$

and two new additional constraints

$$\forall_f \; \sum_{k \in V(f)} Weight_{fk} = 1, \; \forall_{f,k} \, Weight_{fk} \geq 0, \tag{3.2}$$

We assume that $\mathbb{E}[\varepsilon_{fc}] = 0$ and that the noise term has a finite variance σ^2_ε. No further distributional assumptions on ε_{fc} are imposed. This makes the approach flexible and robust to the types of noise and outliers commonly encountered in mass spectrometry data.

Let us now analyze the differences between the proposed model and unique peptides-based model of MSstatsTMT. The term $Channel_{kc}$ differs from the additive term $Channel_c$ from Equation 2.15. Considering several protein profiles jointly requires a way of estimating varying expression patterns. In Equation 2.15 the $Channel_c$ effect is implicitly nested in a protein, as the model is fitted separately for each protein. In the proposed model, as multiple proteins are considered jointly, the $Channel_c$ effect needs to be replaced by analogous $Channel_{kc}$ effect which describes abundace of protein k in chanell c, and enables modeling protein-level quantitative patterns that differ in trend, and not just a shift of one pattern along the Y-axis.

Let us define the **quantitative pattern**, or **expression profile**, of protein k as the vector

$$Abundance_k = (Protein_k + Channel_{kc})_{c=1}^{C},$$

where c indexes the channels.

The parameters $Weight_{fk}$ are unknown auxiliary coefficients that link each feature (peptide) profile to the expression profiles of its matching proteins. For a given feature f, the weights $Weight_{f1}, \ldots, Weight_{fK}$ form a **convex combination** of the protein profiles $Abundance_k$ for all $k \in V(f)$, where $V(f)$ is the set of proteins matched by feature f.

For a unique peptide, the weight of the corresponding protein is 1, and 0 for all other proteins. Conversely, setting equal weights

$$W_{fk} := \frac{1}{|V(f)|}$$

for all related proteins is equivalent to assigning the feature f to all matching proteins, which corresponds to the "all-proteins" approach.

The magnitude of these weights reflects the similarity between protein-level and peptide-level expression patterns: higher weights are expected for proteins whose profiles are more similar to the peptide's pattern, as seen in Example 5. As such, these weights can be used to improve inclusion-based protein inference by incorporating quantitative information to guide the assignment of features to proteins.

The term $Feature_f$ denotes a feature-specific intercept that accounts for differences in average $\log_2$ intensities across features, while μ represents the overall mean. These parameters help with model specification and defining protein abundance, but they are not part of the bi-convex multiplicative structure of the model and therefore do not complicate model fitting. In contrast, the parameters $Weight_{fk}$ must be estimated simultaneously with $Protein_k$ and $Channel_{kc}$ because they enter the model in a multiplicative way, which complicates estimation.

Finally, let $\widehat{\theta}$ denote an estimator of parameter θ. In analogy to the MSstatsTMT summarization, unknown parameters μ, $Protein_k$, $Channel_{kc}$ describe the relative abundance Y_{kc} of protein k in channel c. Hence, we define the following estimator:

$$\widehat{Y}_{kc} = \hat{\mu} + \widehat{Protein}_k + \widehat{Channel}_{kc}. \tag{3.3}$$

Example 5 (Weights for shared peptides) *Figure 3.1 compares two profiles of peptides shared by proteins BRD2 and BRD4. The rate of decrease in (standardized)* $\log_2$*-intensity of the first peptide does not follow closely any of the patterns exhibited by summaries, instead showing a sort of a compromise pattern. Estimated weights reflect that with* $Weight_{1,BRD2} = 0.56$ *for BRD2, and* $Weight_{1,BRD4} = 0.44$ *for BRD4. On the other hand, the second peptide shows a similar decrease as the pattern estimated for BRD4, and hence the estimated weights were equal to* $Weight_{2,BRD2} = 0.12$ *for BRD and* $Weight2, BRD4 = 0.88$ *for BRD4. Under this model, both features contribute to peptide level summaries of both proteins, but with different strengths of the contribution, as measured by*

the weights. A classification model, as opposed to this multiple (mixed) membership model, would assign each peptide to exactly one protein, despite them matching to both proteins, and the compromise quantitative pattern of peptide 1.

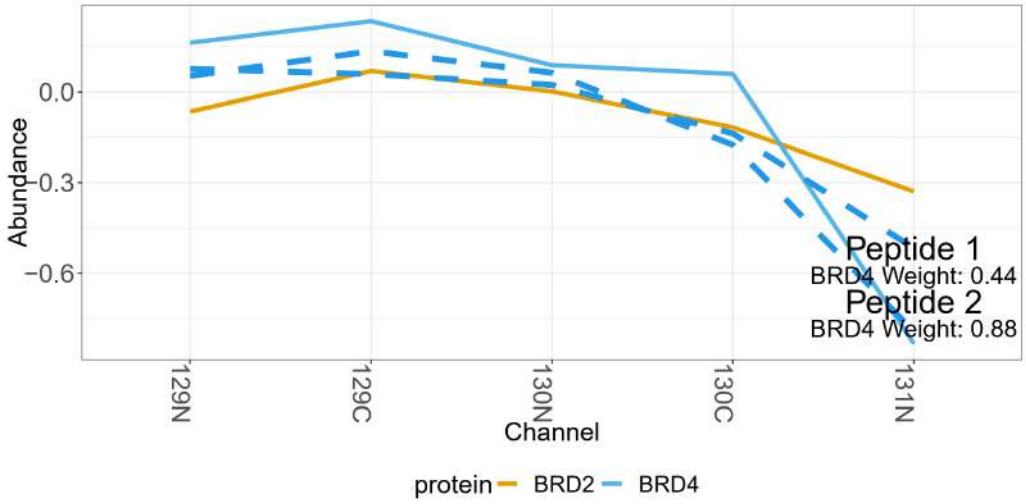


Figure 3.1: **Protein degrader study: motivation for using weights**. Solid lines denote protein-level summaries based on unique peptides, while dashed lines denote two individual shared features.

3.2.1 Data processing

Including shared peptides in protein summarization affects various steps of feature-level data processing. Firstly, it requires assigning shared peptides to all relevant proteins and finding a set of proteins that can be reliably quantified (quantifiable proteins). Depending on expectations and constraints set by the researchers, quantifiable proteins may be characterized by a minimum number of unique peptides, or by other criteria.- Secondly, it affects the way run-level normalization is performed. Lastly, it influences other operations such as missing values imputation. In this section, we discuss the two former aspects in detail.

1. **(List of candidate proteins)** Including shared peptides in protein-level summarization requires matching peptide to all possible proteins that include their sequences. Protein inference algorithms employed by popular signal processing tools such as Proteome Discoverer (Orsburn, 2021) or MaxQuant (Tyanova, Temu, and Juergen Cox, 2016) often perform protein inference using error rates for spurious protein identifications that may depend on the set and number of candidate proteins related to identified peptides. Hence, it is important that all relevant proteins are present in the database at the steps of peptide identification and protein inference. Finally, full sets of matching proteins rather than results of proteing grouping or filtering should be used.

2. **Peptide uniqueness and protein labels** A decision whether a protein is identified by at least one unique peptide should be done at the experiment-level rather than run-level, despite the fact that protein summarization is done for each run separately. Typically, we quantified only proteins with at least one unique peptide. In a given run of an experiment, multiple proteins may be identified by exactly the same set of peptides. In such a case, it is impossible to assign different summaries to each, at least without prior knowledge. Then, it would be reasonable to merge such proteins into a single label. However, it may be possible to differentiate between them in a different MS run of the same experiment. In such a case, merging them only in some runs would add spurious protein labels and make it impossible to match their summaries between runs, reducing the number of replicates at the protein level.

3. **(Filtering proteins identified by a single shared peptide)** Some proteins may be identified by a single shared peptide ion. The only possible summary for such a protein is a possibly rescaled profile of its feature. Thus, it is impossible to eliminate such a proteins from the analysis based on the similarity of its quantitative pattern to profiles originating in other proteins. Moreover, such a summary is characterized by high variance due to lowest possible sample size. Hence, we propose that without any prior knowledge such proteins should be removed from the analysis.

The problem of quantifying proteins without shared peptides is particularly challenging due to many possible structures of peptide-protein graphs for such clusters of proteins. The simplest scenario of two proteins where one of them has a unique peptide can be handled by the proposed approach, but the decision whether to quantify such a subset protein at all is up to researchers. Even in such a simple scenario, the protein could be merged into a single label, quantified separately, or the subset protein can be removed from the analysis.

However, in more complex scenarios, it may be impossible to clearly differentiate subset and superset proteins, particularly when no peptides in a cluster are unique. Figure 3.2 present the smallest possible example of such a cluster of proteins. In this example, all proteins are identified solely by shared peptides, all have the same number of matching peptides, and no proper subset of them can be selected to explain the presence of all identified peptides. Thus, in this case, there is no way to reduce the number of proteins without prior knowledge or additional assumptions.

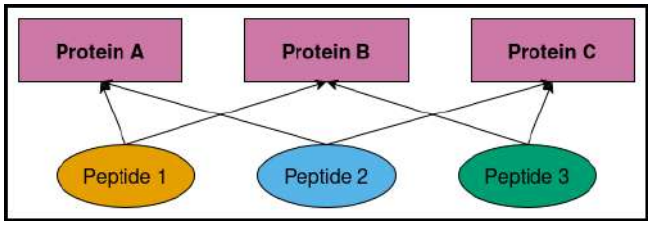


Figure 3.2: A toy example of a cluster of proteins identified by shared peptides in such a way that no superset or a leading protein exists.

From a statistical perspective, such clusters may suffer from identifiability issues for various reasons discussed for example in Marco et al., 2024. One of the possible conditions that guarantee identifiability of such mixed membership models is to require that, in our terms, each protein is characterized by at least one unique peptide. Hence, in our proposed processing, we only considered proteins with at least one matching unique peptides, except in a simulation study dedicated to the problem of subset proteins.

Huang, Choi, et al., 2020 discussed normalization of MS data at various steps of the analysis. From our perspective, spectrum-level (global) normalization is important, as it is done before protein-level summarization. It is required due to sample preparation or MS measurement artifacts and ensures comparability of intensities measured for different peptide ions in different runs of an experiment. Following the MSstatsTMT approach Huang, Choi, et al., 2020 we perform feature-level normalization by replacing raw $\log_2$-intensities with normalized $\log_2$-intensities X'_{cf} using the formula:

$$X'_{cf} = X_{cf} - \text{median}_{r,c} X_{cf} + \text{median}\{\text{median}_{r,c} X_{cf}\}.$$

3.2.2 Objective function for parameter estimation

Let L denote a loss function. We fit the model 3.1 separately for each cluster of proteins and each run of the experiment by minimizing the sum of differences between observed and fitted feature-level $\log_2$-intensities

$$\sum_{c=1}^{C}\sum_{f=1}^{F} L\Big(X_{cf} - \mu - Feature_f - \sum_{k\in V(f)} Weight_{fk}\,(Protein_k + Channel_{kc})\Big). \tag{3.4}$$

over parameters μ, $Feature_f$, $Protein_k$, $Channel_{kc}$, $Weight_{fk}$. Hence, in each cluster and run observed intensities are modeled by 1 global intercept, $F-1$ feature intercepts, and $KC-1$ parameters describing abundances of proteins, and $\sum_{f=1}^{F}(|V(f)|-1)$ weights. The number of observed $\log_2$-intensities is equal to FC assuming no missing values, and typically the sample sizes enable estimation of all relevant parameters.

The choice of loss function depends on assumptions about the error term ε_{fc}. In particular, $L = L_2(x) := x^2$ leads to ordinary least squares estimation which corresponds to the assumption that $\varepsilon_{fc} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. However, this loss function is not robust to outliers. Hence, other alternatives can be useful. The MSstats approach typically uses Tukey Median Polish for summarization which is related to the L_1 norm (Fink, 1988) associated with Laplace distribution of ε_{fc}. However, in practice, we observed that the proposed model encountered convergence issues with the non-differentiable L_1 loss. Hence, we use the smooth Huber loss (P. J. Huber, 1992) defined by Equation 2.2.3.3. Hyperparameter M needs to be selected in a way that it is small enough to ensure robustness but large enough to avoid the convergence issues of L_1 loss. For example, in the presented Case studies we used value $M = 0.001$ or $M = 10^{-6}$. The following Section provides details on optimization of this objective function.

3.2.3 Optimization of the objective function

We introduce a simplified notation that will be useful in presenting this model fitting approach. Let us define vectors of parameters: $\mathbf{P} = (Protein_k)_{k=1,\ldots,K}$, $\mathbf{D} = (Channel_{kc})_{k=1,\ldots,K,c=1,\ldots,C}$, $\mathbf{G} = (Feature_f)_{f=1,\ldots,F}$, $\mathbf{W} = (Weight_{fk})_{f=1,\ldots,F,k=1,\ldots,K}$, and let us denote the sum given by Equation 3.4 as a function of model parameters $\theta = (\mu, \mathbf{P}, \mathbf{D}, \mathbf{G}, \mathbf{W})$ by $\ell(\mu, \mathbf{P}, \mathbf{D}, \mathbf{G}, \mathbf{W})$. Moreover, let us denote by $\ell_{f,c}$ the component of ℓ that involves $\log_2$-intensity observed for feature f in channel c, so that $\ell = \sum_{f,c} \ell_{f,c}$.

The complete optimization problem 3.4 consists in finding a minimum of ℓ over the vector θ. However, as this problem is complicated due to the products of parameters $Weight_{fk}(Protein_k + Channel_{kc})$, we instead solve its biconvex counterpart by searching for a partial optimum.

Firstly, let us fix parameters $\mathbf{P}$ and $\mathbf{D}$. Then,

$$\ell_{f,c}(\mu, \mathbb{G}, \mathbb{W}|\mathbb{P}, \mathbb{D}) = L(X_{fc} - \mu - Feature_f - \sum_{k \in V(f)} Weight_{fk} Protein_k - \sum_{k \in V(f)} Weight_{fk} Channel_{kc})$$

$X_{fc} - Feature_f - \sum_{k \in V(f)} Weight_{fk} Protein_k - \sum_{k \in V(f)} Weight_{fk} Channel_{kc}$ is a linear function of parameters μ, $\mathbb{G}$, $\mathbb{W}$. Hence, the section $\ell_{f,c}(\mu, \mathbb{G}, \mathbb{W}|\mathbb{P}, \mathbb{D})$ is a convex function as a composition of a linear function and a convex function L. Analogously, function $X_{fc} - Feature_f - \sum_{k \in V(f)} Weight_{fk} Protein_k - \sum_{k \in V(f)} Weight_{fk} Channel_{kc}$ is linear in variables $\mu, \mathbb{G}, \mathbb{P}, \mathbb{D}$ when weights $Weight_{fk}$ are fixed, and hence the corresponding section $\ell(\mu, \mathbb{G}, \mathbb{P}, \mathbb{D}|\mathbb{W})$ is again a composition of a linear function and a convex function L. Hence, the optimization problem involving ℓ is bi-convex in terms of parameters $\mathbb{W}$ and $(\mathbb{P}, \mathbb{D})$. Parameters μ and $\mathbb{G}$, while important in modeling, do not affect the structure of the optimization problem.

Algorithm 3 describes iterative estimation procedure for the proposed model which adapts the Alternate Convex Search (De Leeuw, 1994) algorithm described in Section 2.1 to the proposed model.

Let $\mathbb{A}^i$ denote i-th update to parameter $\mathbb{A}$. Input data consist of peptide membership sets $V(f)$, $f = 1, \ldots, F$ and observed normalized $\log_2$-intensities $X_{fc}, c = 1, \ldots, C$ for each peptide. As the summaries are uniquely determined by weights and observed $\log_2$-intensities, the stopping criterion compares consecutive weights from iterations i and $i+1$. The procedure is terminated when $||\mathbb{W}^{i+1} - \mathbb{W}^i||^2 <$ tol. Additionally, there is a maximum number of iterations M.

Algorithm 3: Iterative parameter estimation for the weighted summarization model

Initialize $i = 0$, $\widehat{\mathbf{P}}^{(0)}, \widehat{\mathbf{D}}^{(0)}$ [e.g. by averaging the intensities of the unique features]
$i \leftarrow 0$ **while** $||\mathbb{W}^{i+1} - \mathbb{W}^{i}||^2 < tol$, $i <= M$ **do**
  $\left(\widehat{\mathbf{W}}, \widehat{\mu}, \widehat{\mathbf{G}}\right)^{(i)} \leftarrow \arg\min_{\mathbf{W},\mu,\mathbf{G}} \ell\left(\mathbf{W}, \mu, \mathbf{G}|\widehat{\mathbf{P}}^{i}, \widehat{\mathbf{D}}^{i}\right)$
  $\left(\hat{\mathbf{P}}, \hat{\mathbf{D}}, \widehat{\mu}, \widehat{\mathbf{G}}\right)^{i+1} \leftarrow \arg\min_{\mathbf{P},\mathbf{D},\mu,\mathbf{G}} \ell\left(\mathbf{P}, \mathbf{D}, \mu, \mathbf{G}|\widehat{\mathbf{W}}^{i+1i}\right)$
  $i \leftarrow i + 1$
end
Output: $\hat{Y}_{kc} = \hat{\mu} + \widehat{\mathbf{P}}_k + \widehat{\mathbf{D}}_{kc}, k = 1, \ldots, K, c = 1, \ldots, C$

Step (1) initializes the iterative procedure with equal weights for each protein $W_{fk} = \frac{1}{|V(f)|}, f = 1, \ldots, F$ and an arbitrary protein-level summaries $(\mathbb{P}, \mathbb{D})$. An easy way to obtain reasonable starting values is to use summaries based solely on unique peptides. At this step, $i = 0$.

Step (2) estimates weights $\mathbb{W}$ based on summaries obtained in Step (1). This requires solving an optimization problem $\min \ell(\mathbb{W}, \mu, \mathbb{G}|\mathbb{P}, \mathbb{D})$ under constraints given by Equation 3.2. This is equivalent to fitting a linear model

$$X_{f,c} \sim \mu + Feature_f + Weight_{f,1}(Protein_1^i + Channel_{1,c}i) + \ldots + Weight_{f,|V(f)|}(Protein_{|V(f)|}^i + Channel_{|V(f)|,c}^i) + \varepsilon_{fc}$$

where μ, $Feature_f, f = 1, \ldots, F$, and $Weight_{f,k}, f = 1, \ldots, F, k = 1, \ldots, V(f)$ are model parameters, and values $Protein_k^i + Channel_{k,c}^i$ are predictors derived in previous step of the algorithm. Random variables ε_{fc} denote random noise under the same assumptions as proposed model. This linear model is fitted using loss function L. Analogously, Step (3) estimates parameters $\mu, \mathbb{P}$ and $\mathbb{D}$ by solving an optimization problem $\min \ell(\mathbb{P}, \mathbb{D}, \mu, \mathbb{G}|\mathbb{W})$. Again, this is equaivalent to fitting a linear model. In this case, it is given by

$$X_{f,c} \sim \mu + Feature_f + Weight_{f,1}^{i+1} Protein_1 + Weight_{f,1}^{i+1} Channel_{1,c} + \ldots + Weight_{f,|V(f)|}^{i+1} Protein_{|V(f)|} + Weight_{f,|V(f)|}^{i+1} Channel_{|V(f)|,c} + \varepsilon'_{f,c}$$

where μ, $Feature_f, f = 1, \ldots, F$, $Protein_k, k = 1, \ldots, V(f)$, and $Channel_{k,c}, c = 1, \ldots, C$ are unknown parameters, and $Weight_{f,k}^{i+1}$ are fixed predictors found in Step (2).

Parameters μ and $Feature_f, f = 1, \ldots, F$ do not influence the biconvex structure of proposed model, so they can be estimated at both steps (Shen et al., 2017). We use the value of μ estimated at Step (3) to produce the final summaries.

As both problems are convex, we use standard tools for solving constrained convex optimization problem implemented in the CVXR package (Fu, Narasimhan, and Boyd, 2020) in R. The default solver used by this package is ECOS Domahidi, E. Chu, and Boyd, 2013 (embedded conic solver).

3.3 Evaluation

First, we introduce the data-generating process used for our simulation study. Then, we describe how case studies were used to evaluate the proposed approach. Finally, we define metrics used in the evaluation.

3.3.1 Simulated labeled mass spectrometry data

We considered two types of experimental designs that mimicked designs found in the biological case studies. Firstly, we considered a group comparison design similar to the protein degrader study 2.5.1,

but with a varying number of biological replicates in the same MS run. Secondly, we generated data from a hybrid design found in the TPP study, where comparisons were done between samples measured in different MS runs. Both designs, while using different biological replicates for each condition, produced data similar to the repeated measures designs due to a natural order between samples measured at different exposure times or temperatures. Hence, each simulated data set included natural protein-level quantitative profiles.

We considered two instances of the protein degrader-type group comparison design: low- and high-dimensional. In the low-dimensional setting, we generated a cluster of 5 proteins such that 3 proteins varied between conditions at different rates, and 2 other proteins did not change between conditions. In the high-dimensional setting, we generated a cluster of 20 proteins with 5 proteins that exhibited differential abundance and 15 background proteins with no changes. Both scenarios mimicked the behavior of proteins of interest from the protein degrader study 2.5.1 where only a subset of proteins showed changes in abundances between conditions.

In summary, this design of a simulation study enabled us to evaluate the proposed approach with data of different sizes and experiments with varying designs and amounts of available feature-level information.

3.3.1.1 Protein-level data

Protein-level data generated for the simulation studies were simulated from the model assumed by MSstatsTMT and given by Equation 2.17 which required choosing an experimental design first. As the proposed approach estimates the summaries separately in each run, we first focused on single run design similar to the protein degrader case study 2.5.1. Then, we chose five conditions with a number of biological replicates per condition varying from 1 to 3. The number of conditions was the same as in the protein degrader case study, and it ensured both realistic numbers of TMT channels (from 5 to 15) and a wide range of different $\log_2$-fold changes that could be considered simultaneously. The comparisons of interest compared each of four conditions to the first conditions.

With such experimental designs, simulating protein-level data based on Equation 2.17 required a choice of the global mean, sizes of group effects via the $Condition_g$ parameters, standard deviation of the subject-specific effect and standard deviation of the error term. We chose $\mu = 15$, $\sigma_S = 0.01$, $\sigma = 0.01$ and $Condition_g$ effects selected to obtain a wide range of $\log_2$-fold changes. Table 3.1 summarizes the resulting $\log_2 - foldchanges$ for all structures of simulated peptide-protein graph and experimental designs. The overall noise at the protein-level was kept low, as the proposed approach is focused on summarization rather than protein-level estimation.

Example protein-level profile plots are shown in Figure 3.3.

3.3.1.2 Feature-level data

Generating feature-level data requires choosing a structure of the peptide-protein network first. With 5 proteins in a cluster, there are 10 combinations of 2 proteins, 10 combinations of 3 proteins, and 5 combinations of 4 proteins. Hence, the peptide-protein graphs can be very complex. This complexity grows extremely fast with the number of proteins in a cluster. With 20 proteins, the second scenario used in our simulation study, the number of possible groups of, say, 7 proteins that share peptides may be well over 100,000, so the number of possible peptide-protein networks can be excessively high. To focus attention, we used each pair of proteins in the low-dimensional case, and each pair of proteins with 30 randomly selected triples in the high-dimensional case. We discuss the implications of this choice for input data sizes at the end of this section. In the biological case studies, cluster of 2 to 5 proteins constituted a vast majority of non-trivial clusters. In the TPP study, more than 70% of clusters included this number of proteins, while in other studies over 90% of cluster fell into this category. Moreover, typically over 50% non-trivial clusters consisted of just 2 proteins. We decided to evaluate the proposed model at the upper limit of these standard value (low-dimensional case) and

Cont.	1	2	3
1	.10	0.50	0.20
2	.20	0.75	0.30
3	.50	1.25	0.70
4	.75	1.50	1.00

(a)

Cont.	1	2	3	4	5
1	.10	0.50	0.20	.25	.30
2	.20	0.75	0.30	.30	.40
3	.50	1.25	0.70	.70	.50
4	.75	1.50	1.25	.80	.70

(b)

Cont.	1	2	3
1	.10	0.50	0.20
2	.20	0.75	0.30
3	.50	1.25	0.70
4	.75	1.50	1.00
5	.80	1.70	1.10
6	.85	1.95	1.20
7	.90	2.00	1.30

(c)

Table 3.1: Values of $\log_2$-fold changes in three simulation studies: (a) low-dimensional single run, (b) high-dimensional single run, (c) low-dimensional multi-run. For all remaining proteins in each study the $\log_2$-fold changes were equal to 0. Column *Cont.* denotes the index of a contrast (comparison), while numbered columns denote indices of proteins that showed changes in abundance between conditions, e.g. a value 0.8 in column 1 and row (contrast) 5 means that the true $\log_2$-fold change for first protein in the cluster and 5th comparison was equal to 0.8.

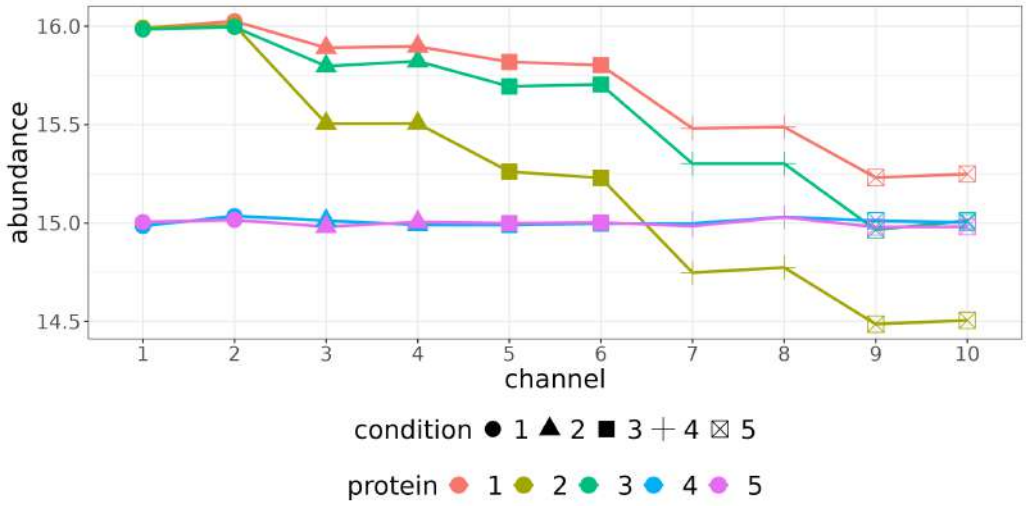


Figure 3.3: **Simulated data**: protein-level profiles.

with a much more complex cluster (high-dimensional case). This was motivated by expected increase in complexity of quantitative MS data with growing ability to differentiate protein isoforms.

Profiles of peptides were generated from the proposed model given by Equation 3.3. For unique peptides this reduced to the MSstatsTMT model. For shared peptides, we chose a fixed set of weights for features originating from a given set of proteins. For example, all peptides shared by protein 1 and protein 5 were given the same weights. In the low-dimensional case, since for two proteins $W_{f2} = 1 - W_{f1}$ for each feature f, it is enough to list the first weight for each pair. The weights were set to 0.7, 0.8, 0.5, 0.6, 0.9, 0.2, 0.3, 0.5, 0.4, 0.1. In high dimensional setup, the weights were sampled according to model constraints, as their number was too large to select them manually. $Feature$ effects were sampled uniformly from an interval $[-1, 1]$.

Overall, in the low-dimensional simulation studies, we varied the following parameters:

- standard deviation of feature-level random error: $\sigma_\varepsilon = 0.1, 0.2$,
- number of biological replicates: 1, 2, 3,
- number of unique peptides per protein: 1, 2, 3, 5, 10,

- number of shared peptides per pair of proteins: 3, 5, 10.

In the high dimensional case, we used 2 shared peptides per group of proteins sharing peptides, 1 or 3 unique peptides, and 2 to 3 biological replicates. 50 repetitions were done for each parameter configuration.

Table 3.2 present the sizes of peptide-protein networks and samples sizes for all types of data sets simulated in the low-dimensional case. In this study, we evaluated the model across cluster with 35 to 125 features, 65 to 225 edges in the peptide-protein graph and 324 to 3375 total observations.

no. unique	no. shared	no. bio. rep.	no. edges	no. features	sample size
1	3	1	65	35	325
1	3	2	65	35	650
1	3	3	65	35	975
1	5	1	105	55	525
1	5	2	105	55	1050
1	5	3	105	55	1575
1	10	1	205	105	1025
1	10	2	205	105	2050
1	10	3	205	105	3075
2	3	1	70	40	350
2	3	2	70	40	700
2	3	3	70	40	1050
2	5	1	110	60	550
2	5	2	110	60	1100
2	5	3	110	60	1650
2	10	1	210	110	1050
2	10	2	210	110	2100
2	10	3	210	110	3150
3	3	1	75	45	375
3	3	2	75	45	750
3	3	3	75	45	1125
3	5	1	115	65	575
3	5	2	115	65	1150
3	5	3	115	65	1725
3	10	1	215	115	1075
3	10	2	215	115	2150
3	10	3	215	115	3225
5	3	1	85	55	425
5	3	2	85	55	850
5	3	3	85	55	1275
5	5	1	125	75	625
5	5	2	125	75	1250
5	5	3	125	75	1875
5	10	1	225	125	1125
5	10	2	225	125	2250
5	10	3	225	125	3375

Table 3.2: Characteristics of feature-level data sets used in the simulation study with 5 proteins.

Table 3.3 presents the sizes of peptide-protein networks and samples sizes for the four cases used in a high-dimensional case. With around 500 features, 100 edges in the network and 9,600 to 15,000

total observations, these examples enabled evaluation of the proposed method for very large protein clusters at the practical limit of its current implementation.

no. unique	no. shared	no. bio. rep.	no. edges	no. features	sample size
1	2	2	960	518	9600
1	2	3	960	518	14400
3	2	2	1000	500	10000
3	2	3	1000	500	15000

Table 3.3: Characteristics of feature-level data sets used in the simulation study with 20 proteins.

3.3.2 Evaluation strategy

To evaluate the proposed weighted protein summarization method, we used three biological data sets and simulated data. Case studies represented three types of experiments: protein degradation study, thermal proteome profiling, and relative PTM quantification. Hence, presented results are based on diverse types of studies with different goals, workflows and experimental designs. Protein degrader, PTM and OnePot studies represent the group comparison design, while the TPP study represents the repeated measures design. Moreover, these data sets varied in complexity of the protein-peptide graphs. Protein degrader and PTM quantification studies were characterized by a simpler structure, with an average number of proteins in a cluster equal to 1.09 and 1.11, respectively. Both thermal profiling studies generated more complex structures, with averages of 1.94 and 3.81 proteins per cluster in OnePot and TPP data sets, respectively. Additionally, the proportion between unique and shared peptides varied between clusters, which was also reflected in the design of the simulation study.

All described studies were biological investigations with no available ground truth information regarding identified proteins or changes between conditions. However, we selected subsets of proteins of interest for which external information was available and used them to evaluate the proposed approach and to guide the design of simulation study.

In the **protein degrader** study, we chose a cluster of protein BRDT, BRD2, BRD3, and BRD4. The similarit of amino acid sequences of the proteins was around 60%, as calculated using Madeira et al., 2022. An external western blot assay study confirmed that BRD2 and BRD4 exhibited significant BET bromodomain degradation, but with different rates (Blake, 2019). In the full cluster, BRD2, BRD3 and BRD4 had 14, 12, and 22 unique peptides, respectively, while BRDT had none. Hence, BRDT protein was removed from the analysis. Moreover, the cluster included 5 shared peptides: 3 of them matched to BRD2, 4 matched to BRD3, and 5 matched to BRD4. Exact structure of sharing these peptides is presented in Figure 3.4, along with a profile plot of quantitative patterns of all peptides. The largest differences in abundances between proteins were observed at later time points. Moreover, there the quantitative patterns of shared peptides do not match the patterns of unique peptides.

BRD cluster included a large number of unique peptides. We used it as a basis for a resampling-based simulation study by using random subsets of unique features for each protein, along with all available shared peptides, to mimick the case of a cluster with few unique peptides. The number of unique peptides varied and in different scenarios we used 1-5 or 10 unique peptides per protein. Protein-level summaries based on such subsets were then compared to the results obtained based on all unique peptides.

Moreover, we used this case study to evaluate the quality of protein-level estimates in the presence of outliers in feature-level profiles. Outliers can understood in two ways: as observations characterized by much higher variation than others or as observations that exhibit quantitative patterns inconsistent with other observations. We used the latter definition. In our framework, profiles of shared peptides are combinations of various protein-level expression patterns, and do not simply match to one particular protein-level pattern in a binary sense. Hence, we focused on outliers in unique peptides. To

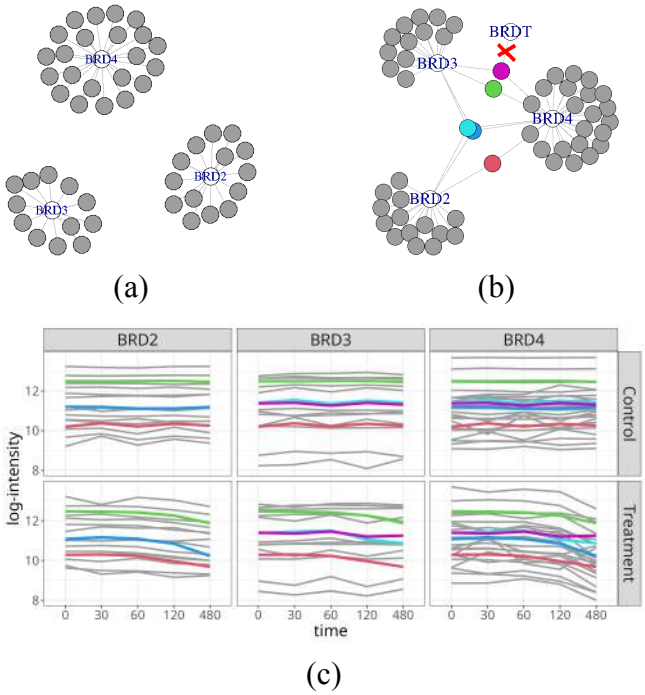


Figure 3.4: **Protein degrader: modeling the contribution of shared peptides transformed the disjoint subgraphs into a connected graph with heterogeneous peptide patterns.** (a) Proteins characterized by unique peptides. Grey nodes: unique peptides. Edges: matches between peptide and protein sequence. (b) As in (a), but with shared peptides (colored nodes). (c) Quantitative profiles of the peptides. Line colors match node colors in (a) and (b). Figure first published in Staniak et al., 2025.

simulate the presence of outliers, we calculated pairwise correlation coefficients between profiles of all unique peptides separately for each BRD protein. Then, we calculated average absolute value of correlation with other profiles for each feature and ranked them by this metric. The smallest values corresponded to features with patterns most dissimilar to other features of the same protein. Hence, we treated them as a proxy of noisy quantitative profiles. An example is given in Figure 3.5. Orange lines denote profiles of unique features that exhibited patterns inconsistent with other unique features of their respective proteins.

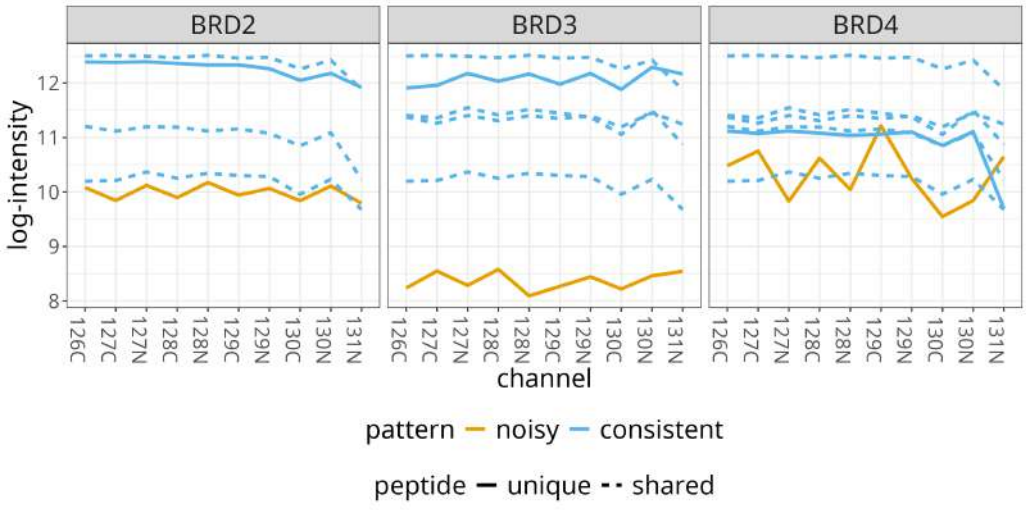


Figure 3.5: **Protein degrader study: example of noisy profiles of features.** Taken from Staniak et al., 2025.

In the **thermal profiling** study, we used a set of known interactors (Figueroa-Navedo, 2023) identified using the KinHub database (Davis et al., 2011; Eid et al., 2017) as a proxy of ground truth knowledge about differential abundance. We also compared the detected changes in abundances to the results based on a more sensitive OnePot portion of the study. For this purpose, we selected non-trivial clusters of proteins that included at least one known interactor. Additionally, we restricted our attention to proteins with at least one identified unique peptide. The TPP portion of the study had 75 such proteins in 27 clusters, while the OnePot portion had 93 proteins in 34 clusters.

Additionally, we illustrated the influence of including shared peptides in summarization based on a cluster of three proteins P16591, P16591-2, and P16591-3. The former two proteins were inferred from identical sets of proteins, and hence had to be merged into a new, common protein identifier. Sequence similarity between these two proteins and a known interactor protein P16591-3 was close to 90% (Madeira et al., 2022). Proteoform P16591-3 was identified by two unique peptides.

In the **PTM quantification** study we selected two modification sites: S236 and S240 on a single protein E9Q6J5. Again, there was no ground truth information available for this study, so we compared the proposed processing and approach to the original MSstatsPTM analysis (Kohler, Tsai, et al., 2023) in terms of the identity of characterized modification sites and resulting summarized quantitative profiles.

3.3.3 Evaluation metrics

In this section we motivate and define the exact metrics used to compare the proposed approach to two alternative strategies: using only unique peptides (*unique-only*) and using all available peptides as if they were unique (*all-peptides*).

3.3.3.1 Precision of estimation of differences between experimental conditions

In the **protein degrader** study, we used the summaries based on all available unique peptides for each protein as a reference for a restricted analysis based on a subset of unique peptides and all shared peptides. Based on such gold standard summaries, we calculated $\log_2$-fold changes $G_{i,k}$ for comparisons between control and treatment samples at each time point $T_k, k = 1, \ldots, 5$ for i-th protein among BRD2, BRD3, and BRD4. For each considered number of unique peptides per protein, we sampled subsets $j, j = 1, \ldots, 100$ and computed protein-level summaries using the proposed approach and two alternative strategies. Then, we calculated $\log_2$-fold changes $LF_{i,k,j}$ for k-th time point and i-th protein in j-th repetition of the study. In each repetition, we defined the mean-squared error (MSE) of $\log_2$-fold changes estimation as $\frac{1}{5}\sum_{k=1}(LF_{i,k,j} - G_{i,k})^2$ for each protein $i, i = 1, 2, 3$. Finally, we compared the MSE between the three strategies.

In the **simulation study**, we generated a range of known $\log_2$-fold changes $G_{k,c}$ for a comparison $c, c = 1, \ldots, 4$ and protein $k, k = 1, \ldots, K$. We considered $K = 5$ and $K = 20$. In both scenarios, abundances of most proteins did not change between conditions. For each fixed set of parameters of the simulation (given in Section 3.3.1) and j-th repetition of the study, we estimated summaries based on the three evaluated strategies (proposed, unique-only and all-peptides) and calculated $\log_2$-fold changes $LF_{k,c,j}$ based on these summaries for each protein k and comparison c. We defined the MSE for the simulated cluster of proteins in j-th repetition as $\frac{1}{4K}\sum_{k=1}^{K}\sum_{c=1}^{4}(LF_{k,c,j} - G_{c,k})^2$ and compared MSE values between the three strategies.

3.3.3.2 Error rates of testing significance of differential abundance

Joint summarization of protein isoforms influences the outcomes of differential expression testing in two major ways. Firstly, it enforces a certain way of processing protein inference results. This way, it promotes inference results that include more proteins than standard grouping or parsimony approaches

but reduce the overall number of protein groups. Secondly, it modifies the protein-level summaries compared to other approaches such as the unique peptides-only analysis which results in different estimated $\log_2$-fold changes. Thus, when comparing the results of proposed analysis method to other approaches, we expect both different sets of testable proteins and different quantification results, which potentially translates to different labels of proteins, numbers of comparisons, and different abundance estimates. Hence, we evaluated the two effects separately. Using biological data sets, we compared the assignments of shared peptides between the proposed processing approach and outputs of standard processing tools which perform protein inference and grouping. Then, using the simulated data sets, we compared the error rates of hypothesis testing between the three approaches.

Consider a fixed sequence database with M proteins. For a given experimental study, the application of peptide identification and protein inference algorithms provides a set of identified peptides and M_{obs} inferred proteins. Some of these proteins may be grouped under a single label and subjected to filtering rules (such as requiring at least two unique peptides per protein), further reducing the number of estimable items (proteins and protein groups) to M_g. Based on outputs of peptide identification by respective signal processing tools, we compared the number of estimable items M_g between the original approaches and the proposed processing. Moreover, we used the example cluster from the **thermal processing** and **PTM quantification** studies to illustrate how the proposed methods affects the number of estimable items M_g based on a fixed set of inferred proteins for non-trivial clusters of proteins.

Based on the **simulated data**, we used the MSstatsTMT protein-level model to compare the significance of observed differences in abundances between conditions. For each comparison c and protein k, we considered the null hypothesis $H_{0,c,k} : Condition^k_{c+1} - Condition^k_1 = 0$ versus the alternative hypothesis $H_{1,c,k} : Condition^k_{c+1} - Condition^k_1 \neq 0$ where $Condition^k_g$ denotes $Condition$ effect estimated for group g of protein k in the MSstatsTMT protein-level model equation 2.17. In practice, the decisions of statistical tests for biological studies depend on the chosen multiple testing correction. Thus, to make our results independent of the number of total proteins considered and comparable between different evaluations, we used raw (non-adjusted) p-values. To compare the properties of hypothesis testing in this context, we compared the power and FDR of tests based on the three strategies.

Additionally, for the **thermal proteome profiling** study we used the set of 237 known interactors as a proxy of true differential abundance. Hence, we compared the labels of proteins identified as significantly differentially abundant by all three approaches and presented their intersection with the known interactors and the results of a more sensitive onePot approach.

3.4 Results

In this section, we evaluate the proposed approach in comparison to the standard analysis based solely on unique peptides and a naive *inclusion*-type approach that treats all peptides matching a given protein as unique. As previously, we will refer to these alternative strategies as unique-only and all-peptides, respectively. We organized the results into two groups: first, related to the convergence and variation of estimates, and second, focused on the implications of shared peptides-based analyses for biological conclusions. Code and data required to reproduce these results can be found in a reproduction repository https://github.com/mstania/TMT_reproduction.

3.4.1 Fitting the model

As discussed earlier, block-coordinate optimization approaches converge in the bi-convex sense. We illustrate practical aspects of fitting the proposed model using this strategy using two resampling-based simulation studies based on the **protein degrader** study. Let us consider the proposed approach using Huber loss with a penalty parameter $M = 10^{-6}$, a 1% tolerance, a starting point based on unique

peptides, and a maximum of 100 iterations. Table 3.4 summarizes the fitted models.

	no. fitted		no. converged		mean no. iter.		mean final diff.	
no. unique	all	noisy	all	noisy	all	noisy	all	noisy
1	100	100	99	97	6.79	11.40	0.0036	0.0033
2	100	100	100	100	4.59	5.91	0.0034	0.0039
3	100	100	100	100	7.16	8.98	0.0042	0.0045
4	100	100	100	100	5.13	6.01	0.0034	0.0038
5	100	100	100	100	3.61	4.22	0.0036	0.0040
10	100	100	100	100	3.50	3.44	0.0040	0.0048

Table 3.4: **Protein degrader** study: summary of model convergence information across 100 repetitions of the resampling-based simulation study. Column *no. unique* indicates the number of sampled unique peptides. *All* refers to sampling from all possible unique peptides, while *noisy* refers to the sampling scheme that ensured the presence of one noisy peptide per protein. *No. fitted* describes the number of obtained fits (a number smaller than 100 would indicate solver error or other issues), *no. converged* refers to the number of models that reached the convergence criterion in 100 iterations or fewer, *mean no. iter.* refers to the number of iterations until convergence (based on *no. converged* fits), and *mean final diff.* describes the average value of a relative difference between consecutive sequences of estimated weights at convergence.

The number of steps required for convergence was not fully monotonic, but roughly decreased with the number of available unique peptides per protein. The largest number of average required steps was achieved with a single noisy unique peptide with around 11 iterations. In other cases, when either the quality or quantity of unique information was higher, the algorithm converged in 3-8 iterations. With a tolerance of 1%, the final average difference between consecutive sequences of estimated weights was between 0.003 and 0.004. Hence, the model converged in a reasonable number of iterations. We discuss the quality of protein-level summaries in the following Section.

The three models that did not converge in the noisy case simply required a higher number of iterations to converge. The model that did not converge with one unique peptide per protein when sampling from all possible peptides was an interesting case. Technically, the procedure did not converge, as the final average relative difference between weights was equal to 7. However, this was because all shared peptides matching the BRD4 protein were assigned a weight of 0 for that protein because of an outlying, unique peptide. Hence, all of them were subsequently treated as unique, which constituted the final summary. Using an alternative flat starting point resulted in convergence from both formal and practical perspectives. This rare example illustrates one of the possible edge cases for the method.

3.4.2 Importance of the robust loss

Protein degrader study Let us recall that noisy features can be understood either as features with larger variation in quantitative profiles or as features with patterns significantly different from other features that match the same protein. The latter understanding is more useful in the context of MS data analysis. Hence, we illustrate the effect of varying loss functions used to fit the proposed model using a resampling-based simulation study in a scenario where each subset of unique peptides includes an outlying pattern.

Figure 3.6 presents the MSE of $\log_2$-fold change estimation calculated across 100 repetitions of the simulation as a function of the number of unique peptides. As each subset included one noisy feature, the case with a single unique feature resulted in the largest error. Huber loss was compared to estimation using the ordinary least squares approach - a popular, but non-robust, choice of a loss function.

Using the Huber loss reduced both the average MSE and the variability of estimated $\log_2$-fold changes. The difference between the two approaches decreased as the number of available unique peptides increased, which, in turn, increased the total sample size.

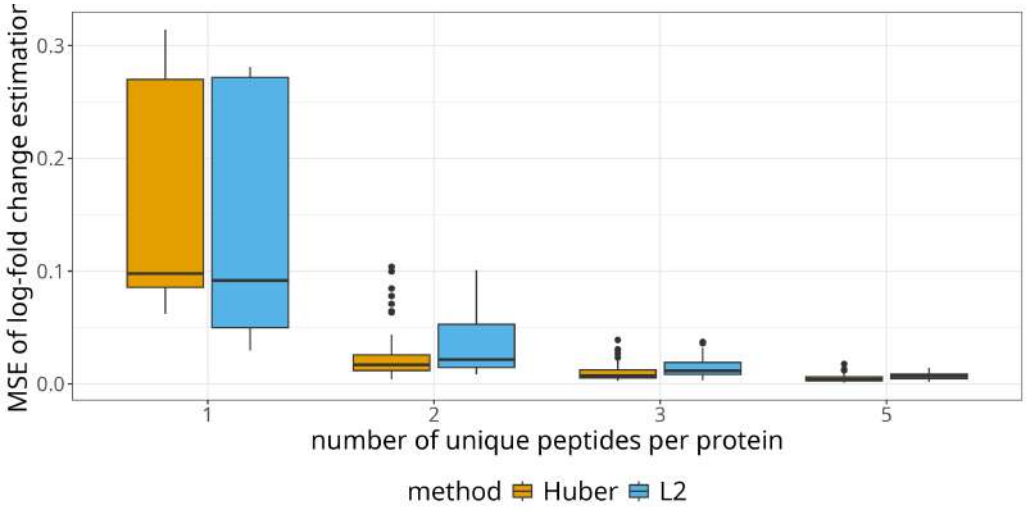
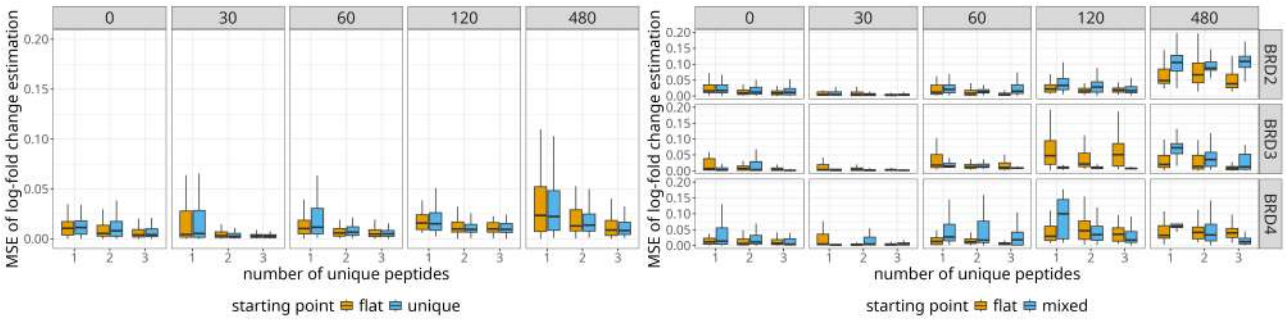


Figure 3.6: **Protein degrader study: the robust Huber loss function reduced the MSE of $\log_2$-fold changes estimation compared to quadratic norm**. Taken from Staniak et al., 2025.

3.4.3 Starting point selection

Protein degrader study With unique peptides available for each protein, various starting points for optimizing the loss function of the proposed approach should not lead to significantly different results. This was confirmed by a resampling-based simulation study, which confirmed two natural choices for an initial protein-level summary from the perspective of $\log_2$-fold change estimation quality. The two starting points were based on either all available unique peptides (denoted *unique*) or constantly equal to the median of $\log_2$-intensities of features (denoted *flat*). Figure 3.7(a) summarizes these results. With unique peptides, the differences in $\log_2$-fold changes were small, with slightly lower error rates achieved by the *flat* approach. However, the *unique* approach was not much different, and we used it in other simulations as a natural choice.



(a) Each protein had unique peptides.

(b) One protein (indicated in the row) did not have unique peptides.

Figure 3.7: **Protein degrader study: when unique peptides were available, different starting points did not lead to significantly different conclusions at the protein-level**. The boxplots summarize 50 repetitions of a resampling-based simulation study with a noisy unique peptide selected for each protein.

Figure 3.7(b) presents analogous results in a case where one of the proteins (indicated in the rows of the Figure) was only identified by shared peptides. In this case, the selection of a starting point had a much larger influence on quantification results. Here, the two approaches were labeled *flat* (starting summary constantly equal to the median of $\log_2$-intensities of features) and *mixed* (starting summary based on unique peptides for proteins that included them, with a flat summary otherwise). In this case, the flat summary typically produced lower average $\log_2$-fold changes.

3.4.4 Simplified set of quantifiable proteins

3.4.4.1 Data set-level complexity

Table 3.5 compares the complexity of peptide-protein networks in biological case studies between original processing methods that used standard protein inference methods to reduce the ambiguities in peptide assignments and the proposed approach, which matched shared peptides to all candidate proteins.

		Case study			
		1	2a	2b	3
Number of protein labels	original	7,482	7,043	8,447	26,004
	proposed	6,323	11,084	25,043	24,809
Number of peptide ions	original	81,851	89,423	164,863	43,585
	proposed	73,881	90,223	165,906	43,585
Number of protein clusters	proposed	5,818	5,699	6,559	22,285
Mean number of proteins per cluster	proposed	1.09	1.94	3.81	1.11
Mean number of shared peptides per cluster	proposed	7.06	15.2	24.8	1.71

Table 3.5: **Acknowledging shared peptides modified the number of quantifiable proteins and available feature-level information per protein in each case study**. Labels 1, 2a, 2b, and 3 refer to protein degrader, Onepot, TPP, and PTM case studies, respectively. Taken from Staniak et al., 2025.

Protein inference and grouping algorithms are prone to reporting concatenated labels that include many protein isoforms. This resulted in a much larger discrepancy in the total number of protein identifiers between the original approach (which uses the Master Protein Accessions output from Proteome Discoverer) and the proposed approach for the TPP case study. A similar, but much smaller effect was observed for its OnePot counterpart. The opposite effect was observed for PTM and protein degrader studies, as the former merged existing site identifiers into new ones. At the same time, the latter ignored some isoforms by randomly assigning shared peptides.

It is essential to reiterate that more complex peptide-protein graphs in OnePot and TPP studies encompassed many subset proteins that could not be easily resolved into a smaller set of quantifiable proteins. Hence, we made an exception in these cases and retained the subset proteins, as removing them would result in the loss of one-third of all peptide ions. Moreover, we included those proteins to highlight the disparity between the actual complexity of the network and the results of protein inference algorithms that attempt to reduce ambiguities in peptide assignments rather than modeling them explicitly.

The differences in peptide ion counts were due to MSstats processing, which, by default, removes shared peptides (usually shared between protein groups defined by protein grouping algorithms) and removes proteins identified by a single feature. Moreover, the proposed processing removed proteins identified only by shared peptides from the protein degrader study and proteins identified by a single shared peptide from all studies. All these characteristics were altered by including shared peptides in the analysis.

The counts of proteins per cluster reveal a disparity between BRD and PTM studies, on the one hand, and OnePot and TPP studies, on the other hand. The average number of proteins per cluster was close to 1 in the former studies, indicating a simpler peptide-protein structure dominated by proteins identified only by unique peptides. Average counts of proteins close to 2 and 3 indicate a more complex structure with many non-trivial clusters of proteins. Even in the latter cases, a smaller cluster of 2-5 proteins was typical (with 2 proteins being the most common), but larger clusters of tens of proteins were present, too.

3.4.4.2 Cluster-level complexity

The impact of recognizing shared peptides for downstream analysis is best understood in examples. Table 3.6 describes a cluster of proteins from the **OnePot** study. The protein inference algorithm used by Proteome Discoverer assigned peptides shared by a cluster of three proteins, Q7Z5L9, Q7Z5L9-2, and Q9H1B7, into 5 protein groups. In addition to individual protein labels, it reported groups (Q7Z5L9; Q7Z5L9-2) and (Q7Z5L9; Q7Z5L9-2; Q9H1B7) for peptides shared by a pair and a triple of proteins, respectively. On the other hand, the proposed approach models such peptides directly as matching to these proteins, reducing the number of labels required to describe the cluster to 3. In particular, Proteome Discover allocated 15 shared peptides to a new protein group (Q7Z5L9;Q7Z5L9-2). The proposed approach instead distributed quantitative information from these peptides between both matching proteins, with the strength of the contribution measured by weights based on their profiles.

Proposed protein group	Proteome Discoverer protein groups				
	Q7Z5L9	Q7Z5L9-2	Q9H1B7	Q7Z5L9; Q7Z5L9-2	Q7Z5L9; Q7Z5L9-2; Q9H1B7
Q7Z5L9	1	0	0	15	3
Q7Z5L9-2	0	2	0		
Q9H1B7	0	0	13	0	

Table 3.6: **Thermal profiling, OnePot case study: inclusion of shared peptides removed the need for concatenated protein groups, which enabled estimation at the level of individual proteins**. Taken from Staniak et al., 2025.

Such a simplified set of proteins has two important advantages. From the perspective of downstream statistical analysis, it simplifies the issue of multiple testing correction by reducing the number of testable proteins. From the perspective of biological conclusions, it increases the reproducibility of results, as various studies may identify slightly different sets of proteins, resulting in unnecessarily large sets of labels of protein groups that share some individual proteins, but can be differentiated by additional ones. For example, if another study identified peptides shared only by proteins Q7Z5L9-2 and Q9H1B7, the protein grouping approach would assign them to a label (Q7Z5L9-2 and Q9H1B7), which does not occur in this study, making comparisons more difficult. Modeling shared peptides explicitly in quantification does not require such labels, uses more information per protein, and results in a more parsimonious set of testable proteins.

Another example of this issue can be found in the analysis of PTMs. Some peptides may carry multiple modification sites, while each individual site can be quantified based on other peptides. Standard approaches, such as Kohler, Tsai, et al., 2023, concatenate such sites into a new label, which creates new modifications that correspond to a common occurrence of their components. Table 3.7 illustrates such an example from the multi-site PTM case study.

The cluster of modification sites of protein E9Q6J included sites S236 and S240. For each site, 2 uniquely matching peptides were identified, and there were 2 peptides covering both sites. The

Proposed PTM sites of protein E9Q6J	Original PTM sites of protein E9Q6J		
	S236	S240	S236_S240
S236	2	0	2
S240	0	2	

Table 3.7: **Multi-site PTM: modeling peptides that carried multiple modifications as peptides shared between individual sites of interest simplified the total set of quantifiable sites**. Taken from Staniak et al., 2025.

original processing approach created a multi-site S236_S240. Meanwhile, the proposed approach did not require any additional site labels. The table compares the number of available peptides per site label in both approaches. Instead of a new site with 2 peptides, the proposed approach distributed the quantitative information from shared peptides between the existing sites S236 and S240, and measured their contribution to each site-level summary using estimated weights. Figure 3.8 illustrates the quantitative impact of the proposed approach. Mixture 2 contained the shared peptides, while Mixture 1 only contained unique peptides. The quantitative profiles of shared peptides were very similar to the profiles of unique peptides carrying the S236 alone. The proposed approach estimated weights equal to 1 for this site and both features, fully allocating them to one site. This reduced the total number of quantified sites compared to the concatenation approach. At the same time, it increased the sample size per site and increased the similarity of summaries between mixtures for both sites.

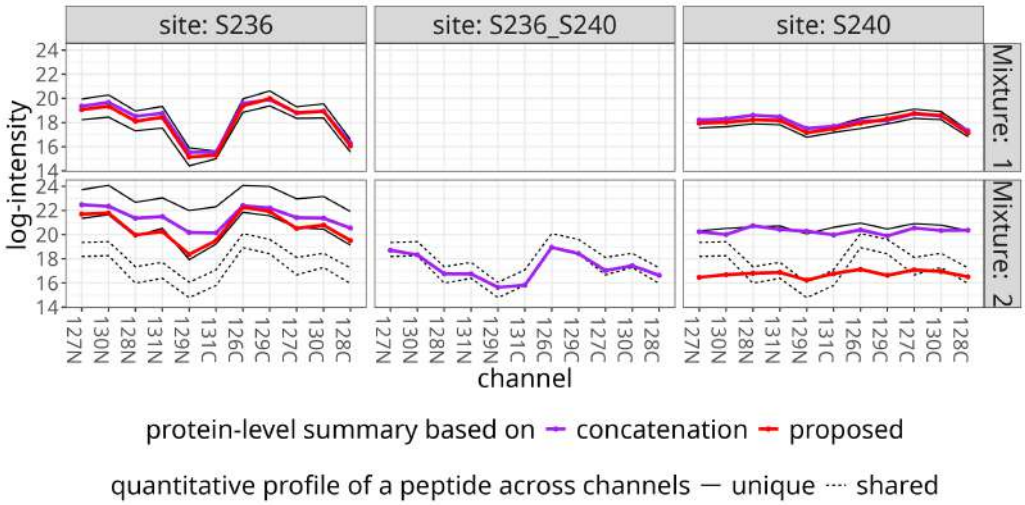


Figure 3.8: **Multi-site PTM: explicitly modeling the shared peptides simplified the set of quantifiable sites without distorting the overall quantitative patterns.** Taken from Staniak et al., 2025.

3.4.5 Improved estimation of differences in abundance

Protein degrader case study First, we investigated the impact of the number of unique peptides available for each protein on the change in precision of $\log_2$-fold change estimation after the addition of shared peptides. Figure 3.9 summarizes the results of a resampling-based simulation study for each protein in the BRD cluster. The proposed approach was contrasted with strategies that use only unique peptides or match each peptide to all possible proteins.

The estimation error decreased with the increasing number of shared peptides for all methods, both in terms of variance and bias. Summaries based on the all-peptides approach appear to overfit to the quantitative pattern of the BRD2 protein, resulting in small MSE values for this particular protein and relatively high errors for other proteins, even with a moderate number of unique peptides in the case of the BRD4 protein.

The proposed approach produced the lowest average MSE and reduced the variance of estimated $\log_2$-fold changes compared to the unique-only analysis. This pattern was consistent across the three proteins, with the largest differences between methods observed for the BRD4 protein, which exhibited the highest changes in relative abundance. The proposed approach provided the largest benefit with a small number of available unique peptides.

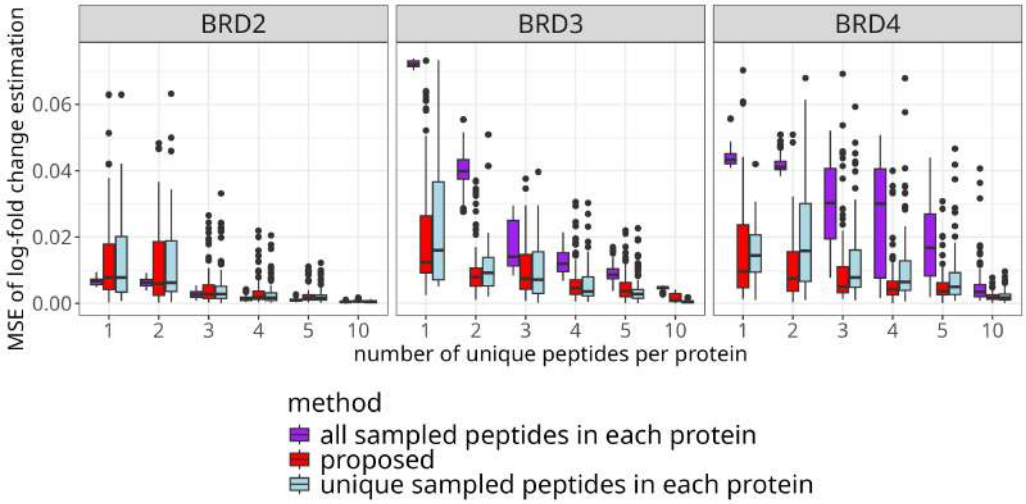


Figure 3.9: **Protein degrader study: weighted summarization with shared peptides improved the mean-squared error of $\log_2$-fold change estimation for BRD proteins**. The boxplots summarize 100 repetitions of the resampling-based simulation study. MSE of $\log_2$-fold change estimation is presented as a function of the number of unique peptides available per protein.

Figure 3.10 details the results of this simulation study for the case of two unique peptides per protein by presenting estimated $\log_2$-fold changes across all repetitions of the experiment. The most important comparison from a practical perspective is the comparison at the final time point. There, the all-peptides approach exhibited very low variance at the cost of high bias. Protein-level summaries produced by this strategy were clearly biased towards the quantitative pattern of the BRD2 protein. The proposed approach reduced the variance of unique-only estimation by increasing the sample size without increasing the bias.

We investigated the effect of modeling the contributions of shared peptides to protein-level abundance estimates on the precision of $\log_2$-fold change estimation in a series of model-based simulation studies.

Simulated data: single-run case. Let us begin with a specific simulated scenario that extends the design of the protein degrader study. Figure 3.11 summarizes 50 repetitions of the experiment involving 5 proteins with 2 unique peptides per protein, 2 shared peptides for each pair of proteins, and 2 biological replicates per condition by plotting estimated $\log_2$-fold changes against the true effects. These results lead to the same conclusions as the biological case study. The bias in all-peptides estimation was apparent, particularly for large effects associated with the fastest-changing protein, as well as for the lack of change between conditions. The difference in bias between weighted summarization and the unique-only approach was small, but there was an improvement in the variance of the estimates.

Figure 3.12 presents complete results of the simulation study in terms of MSE of $\log_2$-fold change estimation as a function of the proportion of unique peptides in a cluster. Rows of the plot indicate the sizes of the standard deviation of the error term at the feature-level, while columns indicate the number of biological replicates per condition. The MSE of the all-peptides strategy was typically much higher than that of the alternative approaches. As the proportion of unique peptides in a cluster approached 50%, the difference between the proposed approach and the unique-only analysis decreased. However, with a smaller amount of available information (a lower ratio of unique peptides or fewer biological

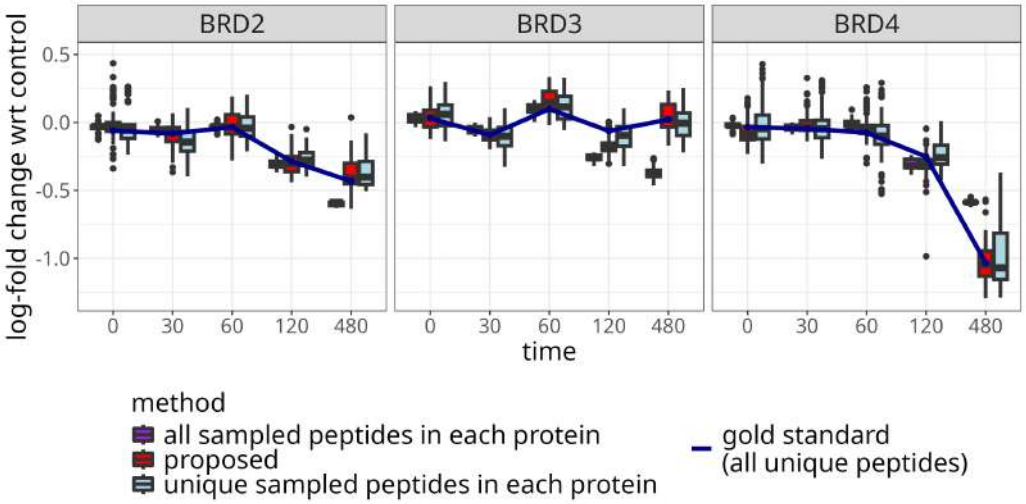


Figure 3.10: **Protein degrader study: weighted summarization with shared peptides improved the $\log_2$-fold change estimation with a small number of unique peptides per protein**. The boxplots summarize 100 repetitions of the resampling-based simulation study with 2 unique peptides per protein. Dark blue line denotes the $\log_2$-fold changes estimated based on all available unique peptides, which were treated as the ground truth values. Changes in relative abundances for BRD2 and BRD4 proteins were confirmed experimentally by Western blot.

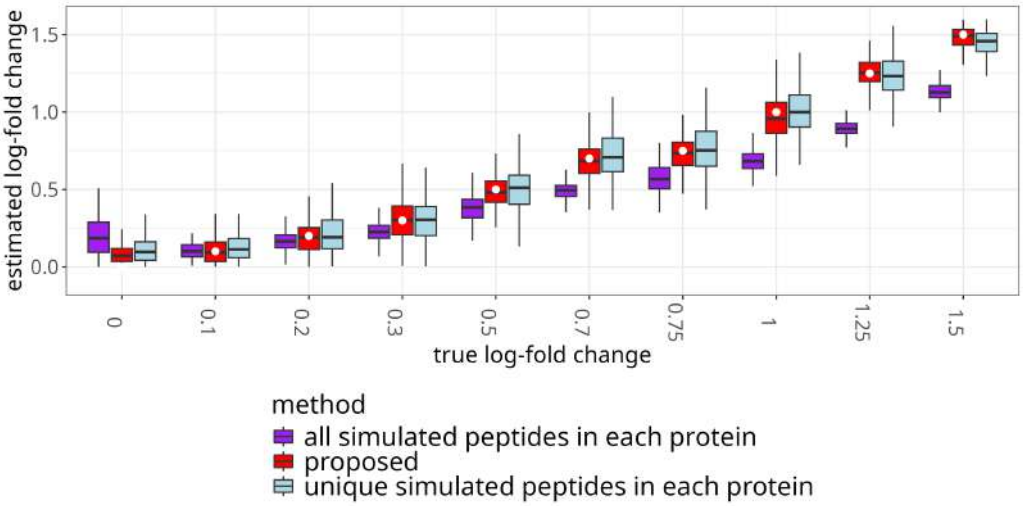


Figure 3.11: **Simulated data: modeling the contribution of shared peptides improved the estimation of a range of $\log_2$-fold changes**. The boxplots summarize 50 repetitions of the experiment with high noise ($\sigma = 0.2$) plotted against the true sizes of effects. White dots also indicate the true values to simplify the evaluation of the bias.

replicates), the benefit of including shared peptides in summarization increased. Such conditions correspond to biological investigations in which reliable identification of protein isoforms is possible.

Simulated data: multi-run case Figure 3.13 presents analogous results for the experimental design with multiple MS runs. Here, protein-level summaries are estimated separately for each run; hence, the quantitative profiles are simpler and more natural than in the previous setting, as replicates describing the same condition appear in different profiles. In this case, we observe the same trends as in the single-run case: using peptides with varying quantitative patterns produces biased estimates. At the same time, the proposed approach improves on the unique-only analysis at low amounts of available unique information.

Simulated data: high-dimensional case Due to the total size of data, we limited the number of simulated settings in the high-dimensional case. Figure 3.14 presents the MSE of $\log_2$-fold change

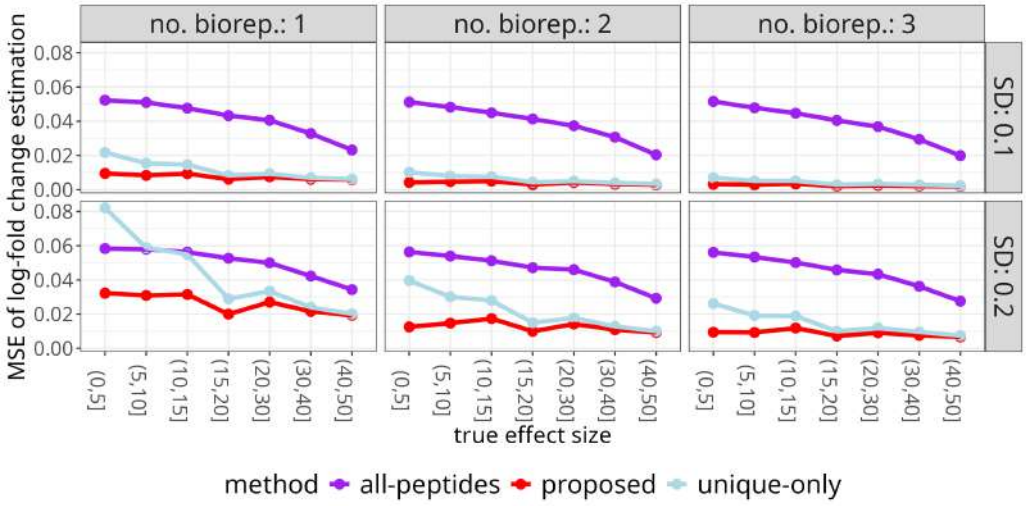


Figure 3.12: **Simulated data: weighted summarization with shared peptides reduced the error of $\log_2$-fold change estimation, particularly with a low amount of available unique information**. Each range of proportions of unique peptides in a cluster corresponds to multiple compositions of the peptide-protein clusters involving different numbers of unique and shared peptides.

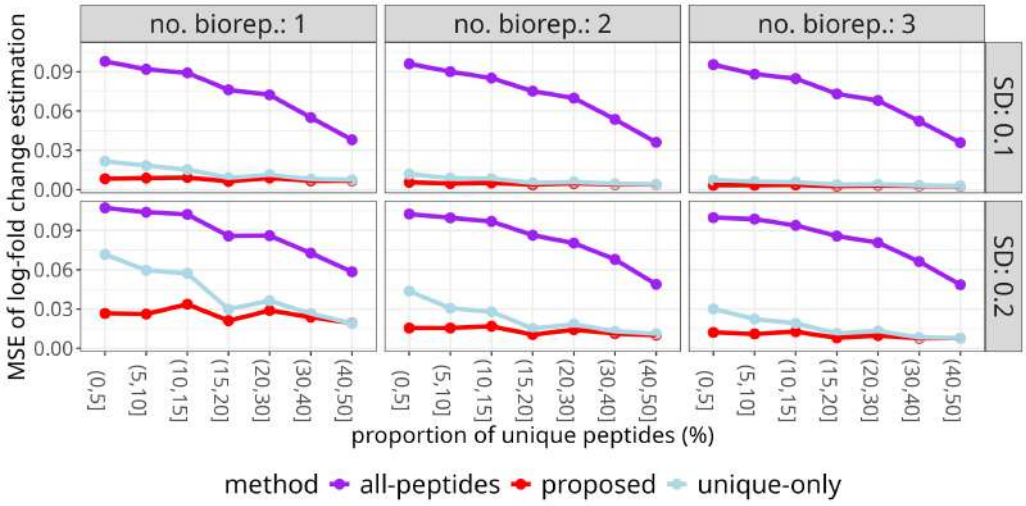


Figure 3.13: **Simulated data: weighted summarization with shared peptides reduced the error of $\log_2$-fold change estimation in a design with multiple MS runs**. Again, the benefit of modeling shared peptides was most pronounced when a low amount of unique information was available.

estimation as a function of the number of unique peptides per protein. Columns of the plot indicate the number of biological replicates per condition, while rows indicate the standard deviation of the random noise term at the feature-level. Again, we observe significantly higher errors based on the all-peptides approach, and the proposed approach improves on using only unique peptides. Such a large cluster corresponds to studies where both the identification of protein isoforms was possible and an appropriately large database was used.

3.4.6 Robustness to noise in unique peptides

Protein degrader study Outliers, understood as peptide features with quantitative patterns that differ strongly from the overall trend in features observed for the same protein, are pretty common in MS-based studies. We used the resampling-based simulations based on the protein degrader study to evaluate the impact of unique but noisy information on $\log_2$-fold change estimation with shared

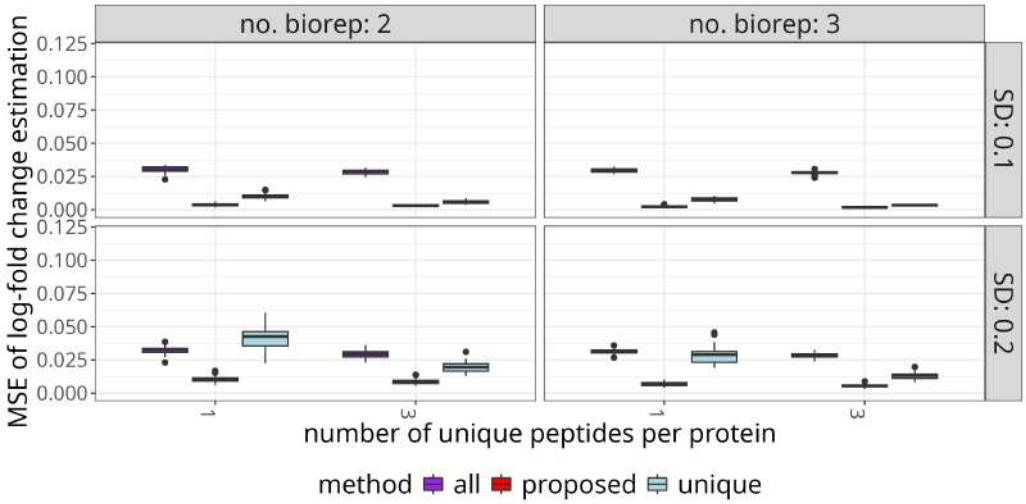


Figure 3.14: **Simulated data: weighted summarization with shared peptides reduced the error of $\log_2$-fold change estimation in a large cluster of proteins**. The results describe a complex peptide-protein network with a small proportion of unique information and a large number of observed proteins.

peptides.

In each repetition of this experiment, one unique peptide for each protein was selected from the pool of four features least correlated with other features. Hence, the case of a single unique peptide corresponds to the extreme case where all unique information is noisy and weights must be estimated from such noisy data.

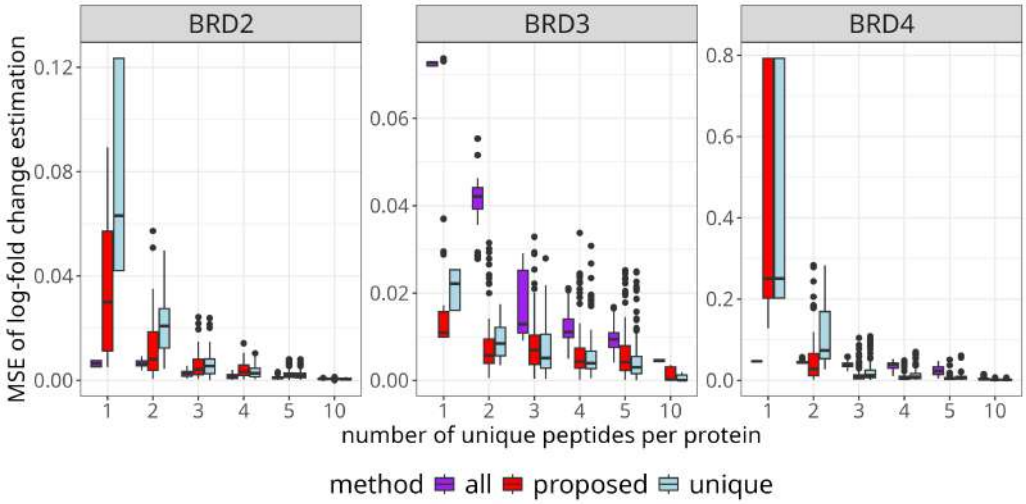


Figure 3.15: **Protein degrader study: weighted summarization reduced the bias and variance of $\log_2$-fold change estimation in the presence of noisy quantitative patterns in unique peptides, particularly with a low amount of unique information**. The boxplots summarize 100 repetitions of the resampling-based simulation study.

Figure 3.15 characterizes the MSE of $\log_2$-fold change estimation as a function of the number of unique peptides available for each protein. In this scenario, the all-peptides approach failed to capture the changes in abundance for the BRD3 protein. The unique-only approach produced estimates with a larger bias than the proposed approach. As expected due to the noise, the difference was most pronounced with a minimal amount of unique information available. Modeling the contribution of shared peptides alleviated the issues with outliers in unique peptides.

We give a more detailed analysis of these results in the case of two unique peptides per protein in

Figure 3.16. The all-peptides approach resulted in the highest bias at the final time point. Weighted summarization yielded unbiased estimates of $\log_2$-fold changes with variance smaller than that of the unique-only approach. The bias in estimation based on unique peptides was manifestly higher than previously observed in a random sampling scenario.

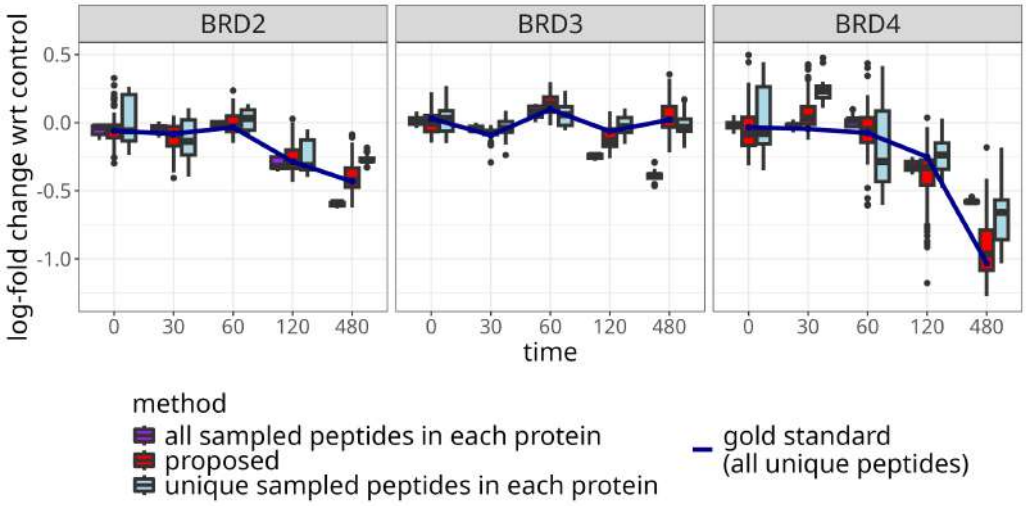


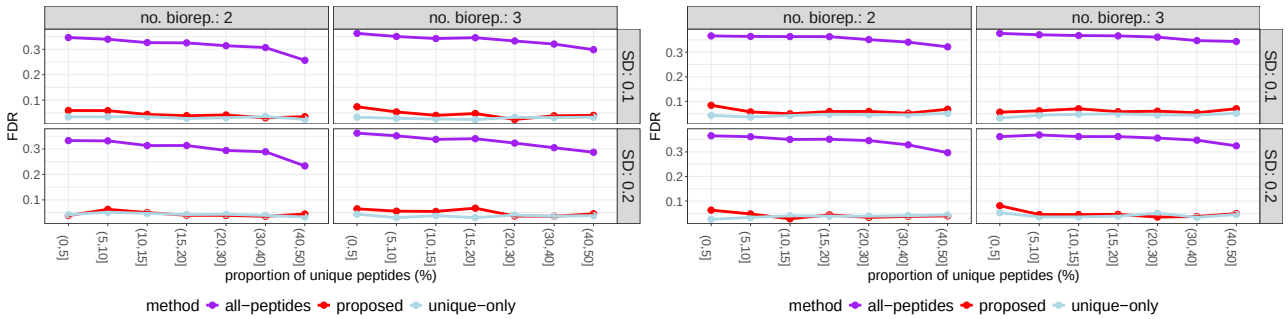
Figure 3.16: **Protein degrader study: weighted summarization reduced the bias and variance of $\log_2$-fold change estimation in the presence of noisy quantitative patterns in unique peptides**. The boxplots summarize 100 repetitions of the resampling-based simulation study in the case of 2 unique peptides per protein. Dark blue line indicates $\log_2$-fold changes calculated based on all available shared peptides, which were treated as the ground truth. The differential abundance of BRD2 and BRD4 proteins was experimentally determined using Western blot.

3.4.7 Reduced FDR of detecting differential abundance

Biological studies and controlled mixtures related to the protein inference problem lack ground truth regarding the differential abundance of proteins. Hence, we compared the proposed approach to alternative strategies from the perspective of statistical inference in a series of simulation studies. In this section, we present results regarding the type I error. P-values were calculated based on the models used by MSstatsTMT, as described in Section 2.4.2. In each simulated scenario and each repetition of the respective experiment, the number of comparisons was the same for all methods. Hence, we used unadjusted p-values as a basis for a fair comparison of the three modeling strategies to avoid the dependence on the total number of proteins in a study. As a result, absolute values of FDR are much higher than if they were adjusted, and we focus on relative comparisons between methods rather than a discussion of FDR control at a given level.

Simulated data: single-run case In an experimental design resembling the protein degrader study, we included additional biological replicates to enable statistical inference. Figure 3.17(a) summarizes the results of this simulation study. The FDR slightly decreased as the proportion of unique peptides in a cluster increased, with the most significant difference observed for the all-peptides approach. An additional biological replicate or increased noise level did not make as much of a difference. The proposed approach produced a slightly higher FDR level compared to the unique-only approach, particularly at low proportions of unique peptides. Overall, the FDR of both approaches was significantly better than that of using the all-peptides approach, which highlights the issue with aggregating peptides exhibiting different quantitative patterns under a single protein label. **Simulated data: multi-run case** We observed analogous behavior of all considered methods in an experimental design resembling the thermal profiling study, as seen in Figure 3.17(b). Again, the difference between the unique-only

and proposed approaches was much smaller than compared to all-peptides analysis. The overall FDR level was relatively consistent across different proportions of unique peptides in a cluster, with slightly higher values for weighted summarization at low amounts of unique information.



(a) Single-run case

(b) Multi-run case

Figure 3.17: **Simulated data: weighted summarization reduced the FDR compared to the all-peptides approach**. Each range of proportions of unique peptides in a cluster corresponds to multiple combinations of the number of unique and shared peptides per protein or pair of proteins, respectively. Each combination was evaluated in 50 repetitions of the experiment.

Simulated data: high-dimensional single-run case Additionally, we evaluated the proposed approach in a high-dimensional setting. As 20 proteins were simulated, the total number of comparisons was significantly higher than in previous studies, resulting in larger absolute values of FDR levels. Again, using all matching peptides for each protein resulted in high FDR values. FDR values for the proposed approach were very close to the ones obtained using unique peptides only.

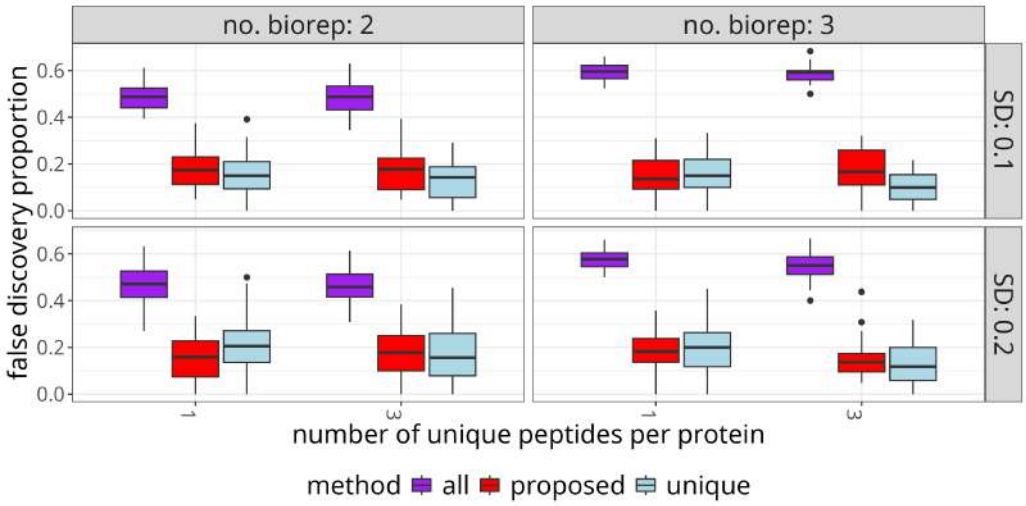


Figure 3.18: **Simulated data (high-dimensional setting): weighted summarization reduced the FDR compared to all-peptides approach**.

3.4.8 Improved power of detecting differential abundance

As previously discussed, biological studies in general lack a ground truth for differential abundance. However, we recall that in the case of the thermal proteome profiling and OnePot studies, known protein interactors can be treated as a proxy of known differential abundance. Hence, we begin the discussion of the statistical power of detecting changes between conditions based on different protein-level summaries with this example. Next, we present the results of simulation studies.

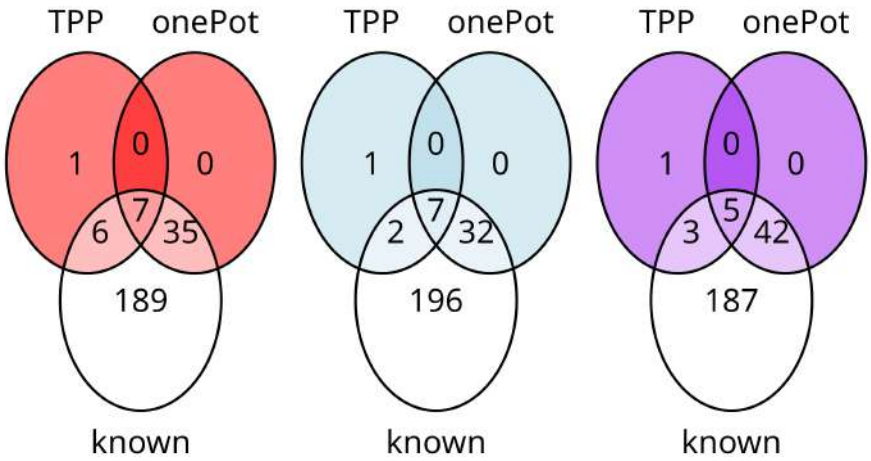


Figure 3.19: **Thermal proteome profiling: weighted summarization improved power of differential abundance testing compared to the unique-only approach.** Colors indicate methods: red - proposed approach, blue - unique-only analysis, purple - all-peptides.

Thermal proteome profiling study. Overall, the OnePot quantification strategy proved to be more sensitive than the TPP portion of the study. Figure 3.19 summarizes the results of statistical inference based on the two studies for each considered analysis method. Among the selected clusters of proteins that included at least one known interactor, the proposed approach was the most sensitive in the TPP study and the second most sensitive in the OnePot portion of this dataset. In the former case, it identified 4 and 5 proteins more than the alternative approaches. In the latter case, it correctly identified 3 additional proteins compared to the unique-only approach, and 7 proteins fewer than the all-peptides approach. In this example, each method produced only a single false discovery; hence, we did not discuss these results in the previous section on FDR. However, as seen in simulation results, the additional statistical power of the all-peptides strategy is usually associated with much higher FDR rates.

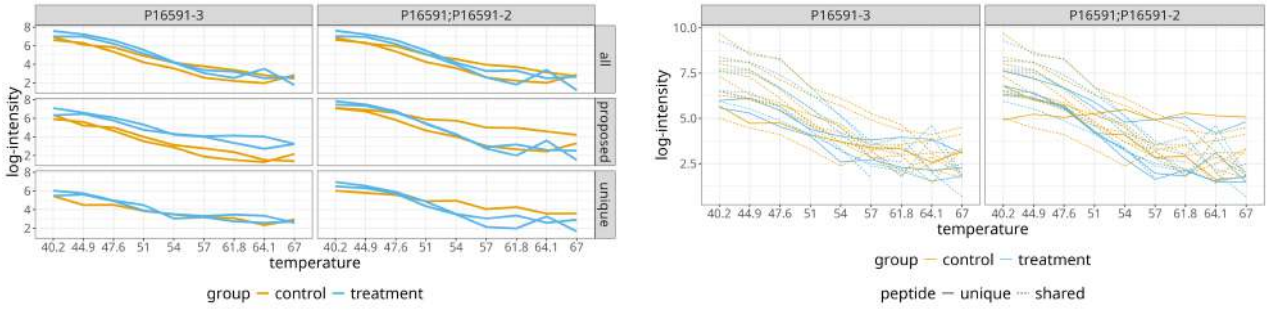
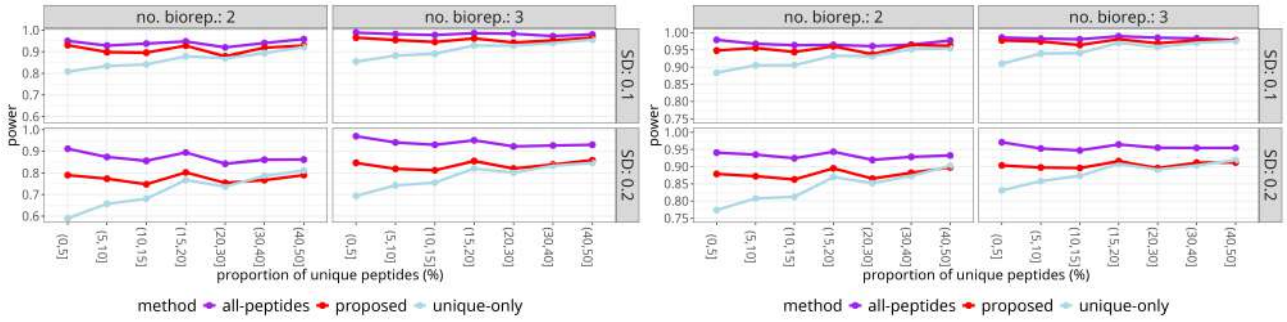


Figure 3.20: **Thermal proteome profiling study: The proposed approach detected the differential abundance of a known interactor P16591-3**. (a) Protein-level summaries for each protein in each run based on the three summarization approaches indicated by the rows. (b) Feature-level data, including both shared and unique peptides. For readability, Y-axes start at 0.6 and 0.75, respectively.

Figure 3.20 presents additional details about an example cluster of proteins P16591, P16591-2, and P16591-3, where the latter is a known interactor. Some unique peptides matching to P16591-3 exhibited smaller changes than shared peptides, similarly to a single outlier matching to proteins P16591 and P16591-2. Shared peptides exhibited a consistent pattern, and a weighted summary that included all available quantitative information enabled a confirmation of differential abundance. Quantification based on weighted summary resulted in a $\log_2$-fold change of 1.65 for this protein, compared to -0.03 estimated based on unique peptides only. Similarly, the all-peptides strategy yielded a smaller $\log_2$-

fold change of 0.17, and neither alternative approach identified this protein as differentially abundant.



(a) Single-run case

(b) Multi-run case

Figure 3.21: **Simulated data: weighted summarization improved the statistical power of detecting differential abundance compared to the unique-only strategy**. Each range of proportions of unique peptides in a cluster corresponds to multiple combinations of the number of unique and shared peptides per protein or pair of proteins, respectively. Each combination was evaluated in 50 repetitions of the experiment. Let us point out that the y-axis, describing power, does not start at 0.

Simulated data Figure 3.21(a) presents the statistical power of each method as a function of the proportion of unique peptides in a cluster. The power of the all-peptides approach was consistently higher than alternative approaches. However, as shown in the previous section, this high power was associated with an incomparably higher FDR. At high concentrations of unique peptides (approximately 50%), all methods performed similarly. At lower proportions of unique peptides, the proposed approach outperformed unique-only analysis by 5-10pp. Overall, the power increased with both additional replicates and lower noise. We observed the same trends in the case of the multi-run design. Figure 3.21 summarizes these results. On the other hand, Figure 3.22 presents the results of this simulation study as a function of the true effect size. The differences between methods were typically seen for small $\log_2$-fold changes.

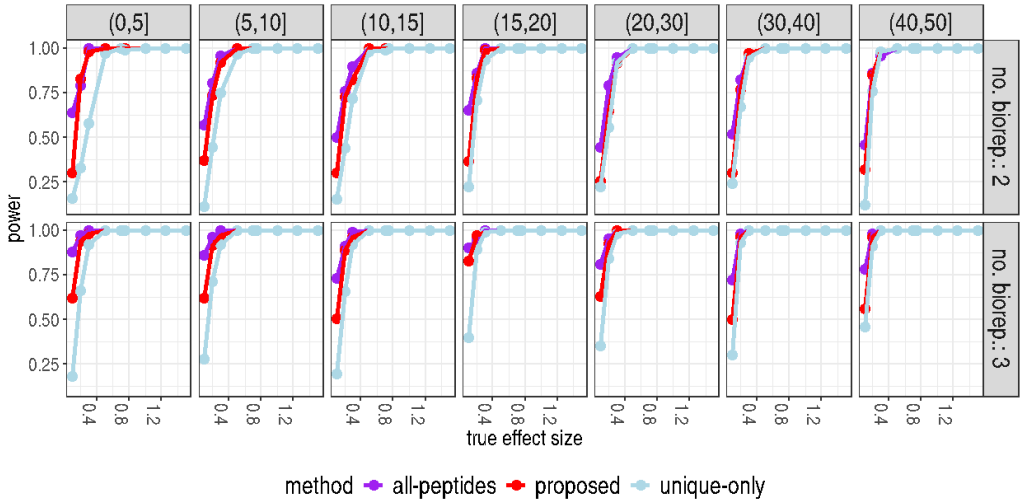
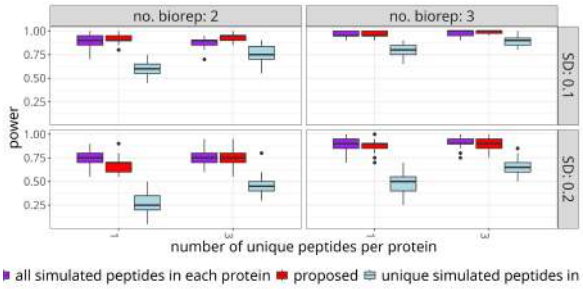


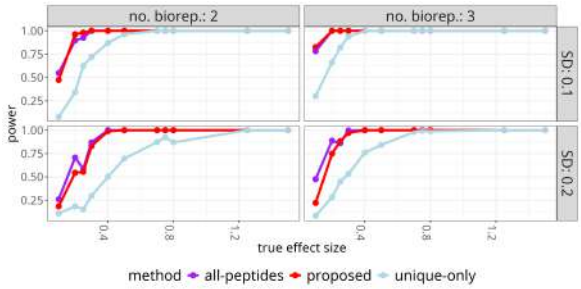
Figure 3.22: **Simulated data**.

Simulated data: high-dimensional case Figure 3.23 characterizes statistical power of weighted summarization compared to its alternatives as a function of 3.23(a) number of unique peptides per protein, and 3.23(b) true effect size. Despite the same experimental design as in the previously described single-run setup, the difference between the proposed and unique-only approaches is much larger. Again, the differences were mostly seen for smaller $\log_2$-fold changes. However, in this case, the

power of all methods was significantly smaller than previously seen. The proposed approach outperformed the unique-only approach by approximately 20 percentage points, while achieving power very close to that of the all-peptides strategy. Overall, the power of all methods improved with a smaller noise level and an additional replicate.



(a) Statistical power of the proposed approach and alternative strategies as a function of the number of unique peptides per protein.



(b) Statistical power of the proposed approach and alternative strategies as a function of the true effect size.

Figure 3.23: **Simulated data (high-dimensional case): weighted summarization improved the statistical power of detecting differential abundance compared to the unique-only strategy.**. The boxplots summarize 50 repetitions of the experiment.

3.5 Discussion

Based on a review of state-of-the-art protein quantification approaches, we concluded that existing methods typically lack a way to utilize information from shared peptides, particularly without additional assumptions such as the ability to assign each shared peptide to a single protein fully. Moreover, the availability of such tools capable of working with general data formats and experimental designs was limited. We proposed a novel protein summarizaion method that addressed these limitations. Firstly, by building on the MSstats framework, it can easily accommodate various experimental designs. This is achieved by performing summarization separately for each MS run and each cluster of proteins, and applying the MSstatsTMT protein-level model to the resulting summaries. Secondly, it does not enforce any particular peptide-protein structure. Instead, we propose quantifying the similarity or association between peptide-level and feature-level quantitative patterns by using weights. In principle, all weights describing the connection between a given peptide and related proteins may be non-zero. In such a case, its quantitative profile contributes to the estimated summary of each relevant protein. Thirdly, it was implemented in a free and open-source package that extends the data formats and a general workflow of the MSstatsTMT method, and produces results compatible with its protein-level data analysis tools.

Using the proposed approach, we evaluated the influence of using shared peptides for protein quantification based on biological case studies and simulations. We demonstrated that acknowledging shared peptides rather than discarding them or treating them as unique to either the original proteins or re-labeled, artificial groups of proteins, increases the accuracy of estimating the difference in protein abundances between experimental conditions and provides better statistical inference results.

The practical relevance of the approach depends on the MS measurement's capabilities in a given experiment. Simple mixtures or low ability to identify peptides that could differentiate proteins that they originated from may not provide enough feature-level information to benefit from the proposed approach. However, modern experiments are capable of measuring enough peptides to analyze protein isoforms. As seen in the results, using shared peptides is most advantageous when the amount of unique information is low. Again, this is common in some of the modern studies. Hence, the proposed quantification model is a valuable tool for protein quantification.

Several key areas require further research to enhance protein isoform quantification. Firstly, the set of quantifiable proteins is entirely determined by the protein inference algorithm used. Grouping peptides to limit the number of peptides shared under the new labels has become a standard, and most signal processing tools return protein groups as output. Such approaches influence both protein-level and peptide-level estimation of identification error rates. At the same time, simply reporting all matching proteins is not a satisfactory solution to this issue. As discussed in Section 3.2.1, even peptide-protein networks that acknowledge all shared peptides and candidate proteins can be complex and provide no obvious way of resolving ambiguities that appear there. In particular, sets of proteins identified by shared peptides with no superset or leading protein are challenging. Hence, alternative approaches to protein inference are needed that aim to return all reasonably quantifiable proteins. In particular, it would be beneficial to consider all three possible source of information: theoretical quantifiability (maximum set of theoretical tryptic peptides generated by a given isoform), prior knowledge from other studies (whether the protein has been identified in a related problem), and information provided by a particular study (identified peptides and their ability to differentiate various related isoforms).

As peptide-protein networks can be complex, effective visualization tools can help researchers understand their data. In this work, we used plots of peptide-protein graphs and standard profile plots with panels indicating proteins and colors or line types denoting the uniqueness of features. However, improved visualizations that combine network and quantitative information, possibly with prior information, would help guide the analyses.

Secondly, using shared peptides for quantification affects feature-level data processing. The presented biological case studies utilized a sample labeling strategy to improve protein coverage, increase sample throughput, and reduce the data missingness. Hence, some of the standard issues in MS data analysis were not relevant to our considerations. In the most common label-free studies, missing values are a significant issue for two reasons: differences in peptides identified in each MS run and peaks missing due to low abundance.

Missing values treatment in MS-based proteomics remains an active field of research. Whenever possible, the MSstats approach categorizes missing values into two categories: missing-at-random and missing-due-to-low-abundance. It imputes the latter using an accelerated failure time (AFT) model while ignoring the former. Alternative models were proposed, too. However, with shared peptides, the assumption of the AFT method that quantitative patterns of peptides differ only by a shift is violated. Hence, missing values cannot be imputed separately for each protein as shared peptides require *consensus* imputed values, and cannot be imputed using AFT, which would not take into account the differences in quantitative patterns between features observed in the same cluster. Hence, a new approach to missing data treatment is required to make the proposed approach applicable to label-free experiments.

Thirdly, modern experiments often employ repeated measures designs. The thermal proteome profiling experiments presented here exemplify this trend. In this case, it is possible to leverage the functional nature of observations and model continuous data, rather than treating each effect as discrete and independent of others. Such a model was recently developed in the Bayesian context by Marco et al., 2024. This would enable the proper modelling of the correlation structure of such data and possibly reduce model complexity. Fitting smooth functions to data with feature-specific weights will be a subject of our future research. However, unlike the MSstats quantification workflow, which is in principle applicable to all experimental designs, such analysis is restricted to one type of design, and incorporating it requires additional conceptual work to efficiently and seamlessly integrate it into existing workflows. In fact, the same is true for the proposed approach in general: the current implementation works with both non-trivial clusters of proteins and proteins identified only by unique peptides, and it is implemented in a way that enables the easy integration of unique-only-based results of MSstatsTMT analysis and the results of the proposed summarization. However, users may be interested in performing weighted summarization only for selected clusters of proteins of interest.

Moreover, the proposed summarization is more time-consuming than the MSstatsTMT summarization for proteins identified only by unique peptides. Hence, it would be beneficial to enhance both the interface and implementation of the analysis workflow for large experiments and to incorporate prior knowledge or hypotheses into the process. On the other hand, as the cost of computation is generally low, the use of high-performance computing tools can make the wide-scale application of the proposed approach viable.

Moreover, following the MSstats approach, we summarize feature-level information separately in each MS run. This is limiting in two ways. Firstly, it would be beneficial to share information about weights or protein expression patterns between runs to improve the precision of the estimation. Such sharing of information has already proved helpful when we used information about the availability of unique information in some runs to adjust protein labels in other runs. Similar benefits might be observed from a quantitative perspective. Secondly, this approach imposes a specific understanding of a profile: it is defined by all samples measured in a given run (in the case of labelled data) or across an entire study (in the case of label-free data). The second issue will be addressed in the future by enabling a more flexible definition of a quantitative profile, but additional research is required to evaluate the influence of such a change on parameter uncertainty estimation and protein-level summary definition and modeling.

Next, we observed that estimation of weights affected the variance of protein abundance effects. Similarly, in some examples, the presence of outliers made the weights susceptible to overfitting. Both of these issues can be addressed by introducing an additional parameter to penalize deviations of estimated weights from constant weights equal to $W_{fk} = \frac{1}{|V(f)|}$ for all k for a given feature f. With this addition, the model would require higher similarity between profiles to assign non-constant weights. We implemented a proof-of-concept version of this approach and observed a decrease in the number of false discoveries based on weighted summaries. However, a too large penalty parameter reduces the proposed approach to the all-peptides summarization, negating the positive effect of penalization. Hence, this modification of the proposed approach would require a cross-validation-type method of selecting the penalty separately for each protein cluster and each run of the experiment. This would significantly increase the required computation time and memory usage. Hence, we considered weights penalization too costly for its benefit at this stage of method development, but we consider it an important research direction for the future.

The current iteration of the proposed model uses weights solely to produce protein-level summaries, which are then used in protein-level modeling. It is possible to improve on this in two ways. Firstly, estimated weights can be an object of interest in their own right. As higher weights indicate stronger similarity between feature-level patterns and protein-level expression profiles, weights could be used as a measure of evidence for the presence of proteins in the sample. This way, they could help post-process protein inference results based on the weights. The proposed method partially does that by estimating some weights as equal to 0, which simplifies the peptide-protein graph. However, this behavior could be further studied and used both in a quantitative context and for the design of visualizations of complex peptide-protein networks. One of the current limitations is the fact that sum-to-one constraint imposed on weights makes them incomparable across features that match to different numbers of proteins. Hence, a method of converting weights from relative *similarity* or *membership* scores into absolute *evidence* scores is needed. Secondly, currently, the proposed approach requires that each spectral feature is assigned to at least one protein. Removing this constraint would provide two benefits: redundant proteins with not enough quantitative evidence could be removed from the analysis, and outlying or noisy features that do not match any particular quantitative pattern at the protein-level could be discarded at the summarization step. Both of these goals can be achieved by penalizing or thresholding weights, and we plan to investigate these possibilities in a future iteration of the project.

Due to the complex, non-convex form of the objective function required to fit the proposed model, its theoretical analysis was challenging. However, it would be beneficial to provide users with additional descriptors of the variation of the estimates. The uncertainty of estimation of both weights

and protein effects could be reported alongside point estimates. The latter problem is still open for MSstats-type summarization-based approaches. Both the model form and the general distributional assumptions make this challenging, but it is an important task for future studies to address.

Lastly, we restricted the proposed approach to the summarization step of the MSstats workflow. However, it was a major simplification from a statistical significance perspective. Whenever multiple experimental conditions are measured on the same quantitative profile, the difference between conditions is implicitly used to both assign weights and estimate the protein-level profile, which later serves as input to the model which quantifies differential abundance between the same conditions. Hence, an inquiry into this issue and its influence on the calculation of p-values and error rates would be beneficial. Similarly, using the same set of features to produce different protein-level summaries may affect multiple testing correction by introducing correlation between p-values describing proteins from the same cluster. In this work, we focused on comparisons limited to a single cluster without multiple membership correction, so these issues were not as important as in large-scale investigations, but additional research is required to understand this aspect of differential abundance analysis for protein isoforms.

We believe that the proposed approach is already useful for practical considerations. We showed that it improves the precision of estimation and inference in the protein quantification problem. Moreover, we provided an efficient implementation of the approach that can be used with arbitrary experimental designs and input files from various MS data processing tools. Hence, the proposed model can already improve the quantification results and enrich the conclusions about protein isoforms based on various types of studies and processing choices.

Chapter 4

Estimation of segment-level H/D exchange probabilities from spectra of overlapping peptides

4.1 Introduction

In this chapter, we introduce a statistical model to infer segment-level hydrogen-deuterium exchange probabilities from observed peptide-level spectra. The model uses a data-driven definition of a segment instead of enforcing single-residue resolution without sufficient information. Moreover, it is capable of estimating uncertainty of estimated probabilities. We describe approaches to fitting the model under homoscedasticity and heteroscedasticity. We implemented the proposed model in a free and open source R package `IsoHDX` that enables data pre-processing, model fitting, and visualization. We evaluated the proposed approach in simulations and case studies in terms of its ability to recover observed isotopic peaks and precision of estimation of segment-level exchange probabilities.

4.2 HDX-MS data structure and notation

In this section, we provide notation necessary for introducing the proposed approach. In particular, we expand on the intepretation of MS1 data provided in Section 2.3.2.4 by including a hydrogen-deuterium exchange perspective which connects intensities of observed peaks to exchange probabilities at a given time point .

4.2.1 Peptide-level data

We assume that HDX-MS spectra are collected for time points $T_k, k = 1, \ldots, K$. We use this notation to stress the discrete nature of observed time values. Additionally, most studies include measurements for undeuterated peptides. We will refer to those as time $T_0 = 0$ observations. In each MS run, spectra describe peptide ions $P_i, i = 1, \ldots, I$. While various charge states of the same peptide are regarded as different units, we will refer to them as simply peptides instead of peptide ions for simplicity. Each spectrum $W_{i,k}$, recorded at time $T_k > 0$ for a peptide P_i, describes partial (or complete) deuteration. Hence, it consists of observed peaks $O_{i,k,j}, j = 0, \ldots, N_i^{ex} + N_i^0 - 1$, where N_i^0 denotes the number of peaks observed for the i-th peptide at time 0 (control MS run describing undeuterated state) and N_i^{ex} denotes the number of exchangeable hydrogens for this peptide. In the case of $T_k = 0$, the number of observed peaks is simply N_i^0. As the first peak correponds to the monoisotopic mass, it is convenient to count these peaks starting from 0. Such indices label each peak based on its mass shift compared to

the monoisotopic one. Let us note that the index $j = j(i,k)$ is nested in time and peptide identifiers (labels).

To summarize, for simplicity we will refer to a set of isotopic peaks for a given peptide and a fixed time as a spectrum, even though a true MS1 or MS2 spectrum consists of peaks that describe many peptides. In general, it is possible to perform multiple runs per time point. Such runs may correspond to either technical or biological replicates. Then, the i-th peptide at the k-th time point is described by multiple spectra $W_{i,k,r}, r = 1, \ldots, R_k$, where R_k is the number of replicates per time point. This number may be equal for all times, but a different number of technical replicates may be obtained for different time points. However, to fix attention and simplify the notation, we will only describe the case of a single replicate, with a multi-replicate case being a simple extension. For a fixed peptide at a given time, all observed replicate spectra are compared to the same expected isotopic distribution. Thus, the model extension is a matter of additional summation. From a statistical perspective, experimental design with multiple replicates is desirable, as it increases the amount of available independent observations.

Finally, let us note that some peptides may be identified only in a subset of MS runs. This means a complete lack of information about a these peptides in some runs. In such cases, fewer spectra are available to estimate model parameters related to the peptides. At the same time, in runs where a given peptide was identified, some isotopic peaks may be missing. We discuss the latter issue further in Section 4.3.3.

4.2.2 Interpretation of MS1 data in HDX-MS studies

Isotopic patterns of peptide ions observed in MS1 spectra have a fundamental role in quantifying the H/D exchange in HDX-MS experiments. As hydrogen atoms are exchanged for deuterium atoms, the mass of peptides increases due to the approximately 1 Da difference in mass between the two isotopes. This causes a change in isotopic patterns of peptide ions, which can be used to extract the exchange information.

Let us consider the isotopic distribution of a peptide ion P_i, $i = 1, \ldots, I$, which contains N_i^{ex} exchangeable hydrogens. The distribution is measured at discrete time points T_k, $k = 1, \ldots, K$.

Since peptides with different charge states exhibit distinct physicochemical properties, we treat each charge variant of a peptide separately. For readability, however, we will often refer to these ions simply as *peptides*.

The observed isotopic pattern of a peptide results from the convolution of two independent probability distributions:

1. **Natural isotopic distribution of the undeuterated peptide.**
   This describes how the presence of heavy isotopes (^{13}C, ^{15}N, etc.) changes the observed peptide mass. Formally,
   $$F_{i,n} = \mathbb{P}\Big(\text{the monoisotopic mass of peptide } i \text{ is shifted by } n \text{ Da}\Big),$$
   where $n = 0, \ldots, N_i^0 - 1$ and N_i^0 is the number of peaks in the isotopic envelope of the *undeuterated* peptide. In other words, $F_{i,n}$ is the probability of observing peptide i in a form that differs from its monoisotopic mass by n Daltons.

2. **Hydrogen–deuterium exchange distribution.**
   During the experiment, some of the N_i^{ex} exchangeable hydrogens of peptide i may be replaced by deuterium. At time T_k, the probability of exactly n exchanges is
   $$\delta_{i,k,n}^P = \mathbb{P}\Big(\text{peptide } i \text{ has exchanged } n \text{ hydrogens at time } T_k\Big),$$
   for $n = 0, \ldots, N_i^{ex}$.

Thus, for each peptide i and time point T_k, we obtain a multinomial probability distribution

$$\Delta_{i,k}^{P} = \left(\delta_{i,k,n}^{P}\right)_{n=0}^{N_i^{ex}},$$

which corresponds to the random variable

$$X_{i,k} = \text{number of exchanged hydrogens in peptide } i \text{ at time } T_k.$$

By construction,

$$\sum_{n=0}^{N_i^{ex}} \delta_{i,k,n}^{P} = 1.$$

This stochasticity arises from a large population of peptide molecules undergoing hydrogen–deuterium exchange.

Similarly, the distribution

$$(F_{i,n})_{n=0}^{N_i^0-1}$$

is associated with the random variable

$$U_i = \text{mass shift of peptide } i \text{ relative to its monoisotopic mass},$$

and satisfies

$$\sum_{n=0}^{N_i^0-1} F_{i,n} = 1.$$

Combining both effects, the observed peak index can be written as

$$J_{i,k} = U_i + X_{i,k}.$$

The probability of observing peak j is therefore given by the convolution

$$\mathbb{P}(J_{i,k} = j) = \sum_{d} \mathbb{P}(U_i = d)\, \mathbb{P}(X_{i,k} = j - d) = \sum_{d} F_{i,d}\, \delta_{i,k,\,j-d}^{P},$$

where the sum is effectively restricted to values of d for which both terms are defined.
Since the expected intensity is proportional to this probability, we obtain

$$E_{i,k,j} = O_{i,k} \sum_{d=\max(0,\, j-N_i^{ex})}^{\min(N_i^0-1,\, j)} F_{i,d}\, \delta_{i,k,\,j-d}^{P}, \qquad j = 0, \ldots, N_i^{ex} + N_i^0 - 1, \tag{4.1}$$

where $O_{i,k}$ denotes the total observed intensity of all isotopic peaks of peptide i at time T_k.

This representation of isotopic envelopes has been employed in, for example, Claesen, 2013, Babić, Kazazić, and D. M. Smith, 2019, and Z. Zhang, 2020. Equation 4.1 can also be reformulated by interchanging the roles of the indices that correspond to hydrogen exchanges and natural isotopic shifts.

To summarize, the distribution of isotopic peak positions for a deuterated peptide is given by the convolution of the distributions of U_i and $X_{i,k}$. The expected intensities are then obtained by rescaling this distribution with the total observed intensity, as expressed in Equation 4.1. In practice, the convolution can be computed directly from Equation 4.1, or by alternative methods such as the generating function approach, where convolution corresponds to polynomial multiplication.

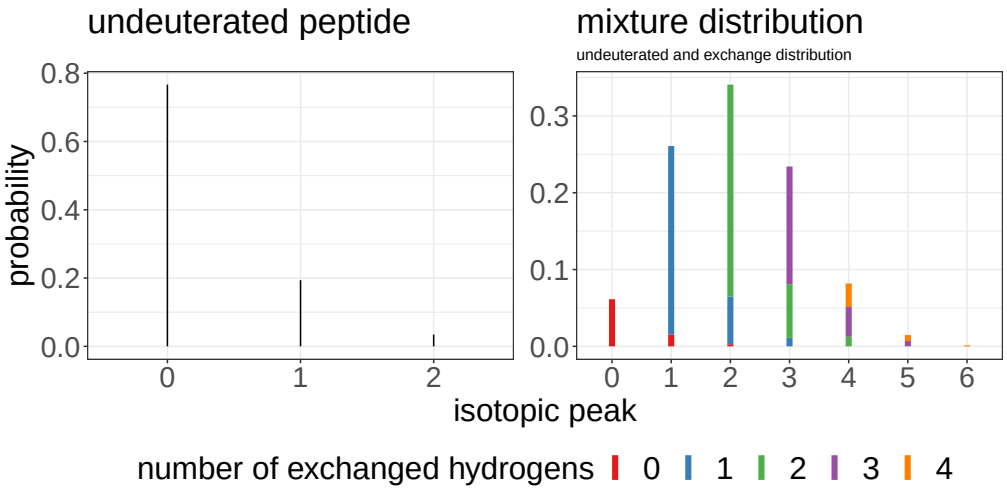


Figure 4.1: Left panel: isotopic distribution of undeuterated peptide DKLV restricted to three peaks. Right panel: isotopic distribution of a convolution of undeuterated distribution and hydrogen exchange distribution.

Example We illustrate Equation 4.1 in Figure 4.1 which presents an example based on a short amino acid sequence DKLV. The left-hand side panel shows the isotopic distribution $F_{1,n}, n = 0, 1, 2$, of an undeuterated peptide. The right-hand side panel shows a convolution of this distribution with a distribution of exchanged hydrogens such that $\delta^P_{1,1,0} = 0.08$, $\delta^P_{1,1,1} = 0.32$, $\delta^P_{1,1,2} = 0.36$, $\delta^P_{1,1,3} = 0.2$, $\delta^P_{1,1,4} = 0.04$. Each peak is a sum of contributions from different numbers of exchanged hydrogens multiplied by the corresponding isotopic probabilities of an undeuterated peptide, as shown by the colors of bars. For example, let us consider the peak shifted by 2 Da relative to the monoisotopic peak. Table 4.1 presents summands that appear in the application of Equation 4.1 for this peak. These summands correspond to lines by red, blue, and green colors, respectively. Total expected height of the peak is equal to the sum of terms $F_{1,d}\delta_{1,1,2-d}$.

d	$F_{1,d}$	$\delta_{1,1,2-d}$	$F_{2-d}\delta_{1,1,2-d}$
0	0.7665	0.3600	0.2760
1	0.1937	0.3200	0.0620
2	0.0345	0.0800	0.0028

Table 4.1: Summands of the expected peak height $E_{1,1,2}$ for peptide DKLV and example exchange probabilities. Values in the column $F_{2-d}\delta_{1,1,2-d}$ sum up to the probability of 2 Da shift. Here, with a total intensity of a spectrum equal to 1, this is exactly the expected intensity of the related peak.

4.2.3 Segment-level data

We propose using data-driven segments instead of fixed, single-residue segments. We define a data-driven segment as a subsequence of the protein sequence such that its first amino acid is either at the first position in the protein sequence, one position before an observed peptide, or one position after an observed peptide. Conversely, the last position in the amino acid sequence of a segment may be either the last amino acid in the protein sequence, or a final amino acid of an observed peptide, or one amino acid before a start of an observed peptide. In other words, segments constitute the largest subsequences of observed peptides such that they are disjoint and they sum to all observed peptide sequences. Data-driven segments coincide with single residues in case of perfectly overlapped data where each segment is of length 1.

Moreover, we can define a cluster in way analogous to the protein inference case. A set of peptides with overlapped sequences that do not share segments with other peptides will be referred to as a cluster. As peptides from different clusters are disjoint, clusters can be treated independently from the perspective of exchange rate estimation. Let us note that at the same time, disjoint peptides occur in spectra together, so they cannot be treated separately in terms of normalization and quality control.

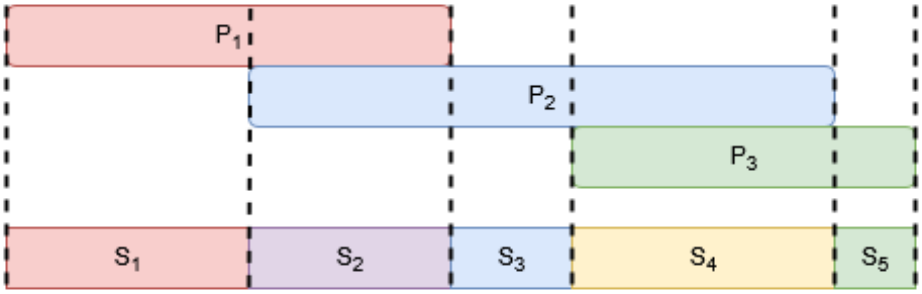


Figure 4.2: A hypothetical example of 3 peptides P_1, P_2, and P_3 that overlap in a way that generates 5 disjoints segments. Dashed lines denote positions in protein sequence that are a start or an end of a segment.

Figure 4.2 presents an example of data-driven segments creation. Dashed lines indicate first and last AAs in the sequences of peptides. As described in the previous paragraph, segments are contiguous sub-sequences of AAs such that the first segment starts at the first AA of an identified peptide, and consecutive segments are separated by starting and ending AAs of subsequent peptides. Figure 4.3 presents an example of observed segments based on a part of a sequence of protein from the HVEM case study.

Hydrogen–deuterium exchange (HDX) is not observed for the first, and sometimes the second, amino acid (AA) of a peptide due to digestion and back-exchange effects (Walters et al., 2012). This introduces an important distinction between a peptide's full amino acid sequence and its *exchangeable sequence*.

The exchangeable sequence is defined as the peptide sequence with its initial residue(s) removed, since their exchange cannot be measured. Consequently, if the same amino acid residue appears in different peptides, its exchangeability may depend on the peptide context. For example, an amino acid that is the first residue in one peptide is considered non-exchangeable, whereas the same residue can be exchangeable if it appears internally in another peptide.

In addition, proline residues do not undergo exchange. Thus, the number of exchangeable hydrogens in a peptide (and in its subsequences) is calculated as the peptide length minus one (to exclude the initial AA) minus the number of prolines it contains.

Figure 4.4 illustrates this point. The segment containing residue K is the N-terminal amino acid of the second (middle) peptide, so its exchange is unobservable in that context. However, the same residue K appears as the second amino acid in the first (top) peptide, where it is exchangeable. From the first peptide's perspective, this segment contributes one exchangeable hydrogen, whereas from the second peptide's perspective it contributes none.

This motivates our definition of an exchangeable sequence: only those segments that start and end within the exchangeable region of each peptide are considered. In particular, the single-residue segment "K" is excluded from the exchangeable sequence of the second peptide. As a result, no artificial overlap is introduced between the two peptides at this position, and the observed exchange of residue K is attributed solely to the first peptide.

Exchange probabilities for a single residue can be estimated only when it is a data-driven segment. Otherwise, a common exchange distribution may be estimated for a group of residues that constitute a

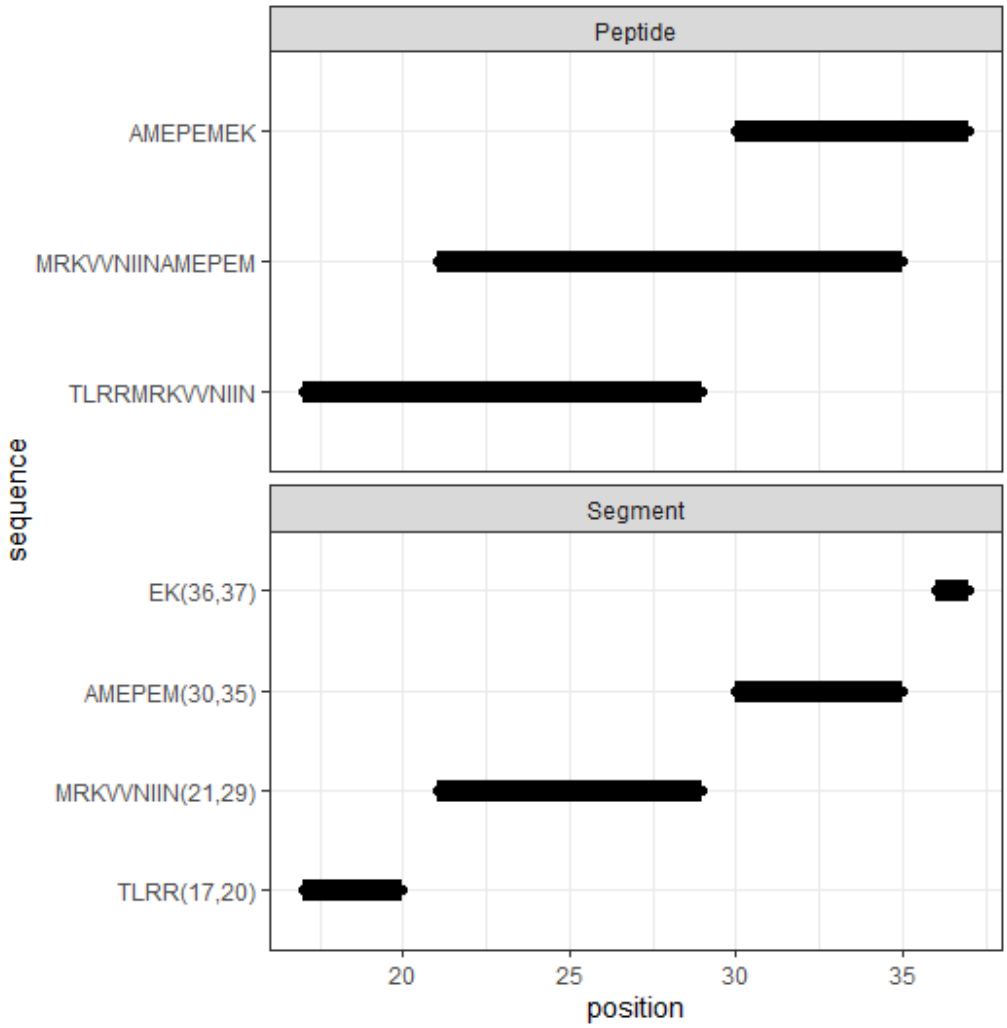


Figure 4.3: **HVEM case study: data-driven segments derived from overlapping peptides**. The upper panel presents observed peptides. These overlapping peptides can be decomposed into disjoint segments shown on the lower panel. Numbers in parentheses in segment annotations and on the x-axis denote positions in the amino acid sequence of source protein.

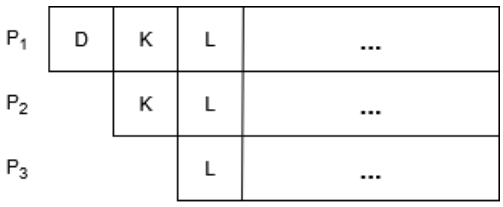


Figure 4.4: A hypothetical example of 3 peptides P_1, P_2, P_3, that overlap in a way which produces segments that include residues with hydrogens that are exchangeable in some peptides, but not in others, such as segment K in P_2 and P_1, respectively.

segment. In this case, a segment of length m is modeled with m binary random variables that describe whether the exchange occurred. Then, the H/D exchange probabilities for a segment can be modeled in the same way as the peptide-level distribution (Section 4.2.2). A general way of specifying this probability distribution is to use a multinomial distribution with probabilities

$$\delta^S_{l,n}(k) = \mathbb{P}\left(\text{segment } l \text{ exchanged } n \text{ hydrogens at time } T_k\right) \tag{4.2}$$

for $n = 0, \ldots, n_j^{ex}$. We will denote such a distribution by $\Delta^S_{l,k}$ and the corresponding random variable

by

$$Y_{l,k} = \text{number of hydrogens exchanged by segment } l \text{ at time } T_k.$$

4.3 Proposed method

We aim to recover the segment-level exchange rates (probabilities) from peptide-level spectra. We propose a single-stage approach which models peptide-level probabilities only implicitly, and uses data-driven segment definitions rather than enforcing single residue segments.

4.3.1 Peptide- and segment-level modeling

We propose searching for an optimal set of segment-level probabilities that result in the lowest prediction error of the peptide-level isotopic peaks. We use the standard representation of expected isotopic peaks given by Equation 4.1 and a model that connects observed peptide-level peak data to unobserved segment-level information. Let us consider a peptide i with N_i^{ex} exchangeable hydrogens which is composed by segments $S_j, j = 1, \ldots, N_i$. Let us define sets $V(i)$ to describe this relationship:

$$V(i) = \{j : \text{Segment } S_j \text{ is a subsequence of Peptide } P_i\}. \tag{4.3}$$

Then, for each peptide i at time T_k, $X_{i,k} = \sum_{j \in V(i)} Y_{j,k}$. Hence, the probability distribution $\Delta_{i,k}^P$ can be expressed as a convolution of probability distributions $\Delta_{j,k}^S, j = 1, \ldots, N_i$ (see Equation 4.2) under the assumption that all segments exchange hydrogens independently in the probabilistic sense. Hence, to model observed peaks in terms of segment-level probabilities, it is sufficient to fix segment-level probabilities, compute a convolution of relevant distributions for each peptide, and then calculate expected peak intensities based on peptide-level probability distributions. Such a procedure enables a comparison of observed and predicted intensities. In what follows, we introduce the details of the proposed approach.

4.3.1.1 Segment-level modeling

Modeling the exchange distribution of segment j, $\Delta_j^S(T_k)$, requires a parametrization of the probabilities $\delta_{j,k,n}^S$ at time $t = T_k$ (see Equation 4.2). We propose the following multinomial logit-type model:

$$\delta_{j,k,n}^S(\beta) = \begin{cases} \dfrac{1}{1 + \exp(\exp(\beta_{j,n_j^{ex}})t + \beta_j) + \sum_{m=1}^{n_j^{ex}-1} \exp(\beta_{j,m}t + \beta_j)}, & n = 0, \\ \dfrac{\exp(\beta_{j,n}t + \beta_j)}{1 + \exp(\exp(\beta_{j,n_j^{ex}})t + \beta_j) + \sum_{m=1}^{n_j^{ex}-1} \exp(\beta_{j,m}t + \beta_j)}, & 0 < n < n_j^{ex}, \\ \dfrac{\exp(\exp(\beta_{j,n_j^{ex}})t + \beta_j)}{1 + \exp(\exp(\beta_{j,n_j^{ex}})t + \beta_j) + \sum_{m=1}^{n_j^{ex}-1} \exp(\beta_{j,m}t + \beta_j)}, & n = n_j^{ex}. \end{cases} \tag{4.4}$$

Here, β_j is a segment-specific intercept, and parameters $\beta_{j,n}$ ($n = 1, \ldots, n_j^{ex}$) control the dynamics of exchanging n hydrogens at time t. The form for $n = 0$ ensures normalization $\sum_{n=0}^{n_j^{ex}} \delta_{j,k,n}^S = 1$.

To capture the biological constraint that the probability of exchanging all hydrogens tends to 1 as $t \to \infty$, we parametrize the corresponding coefficient as $\exp(\beta_{j,n_j^{ex}})$, which guarantees positivity and dominance in the limit. *Let us note, however, that this additional constraint is not strictly required for modeling. It corresponds to the assumption that all segments eventually reach full deuteration, whereas in practice it may be reasonable to allow that some segments plateau below complete exchange.*

4.3.1.2 Peptide-level modeling

We now turn to the relationship between segment-level and peptide-level exchange probabilities. To this end we use probability generating functions. For each segment j and time point T_k, let $Y_{j,k}$ denote the random variable giving the number of exchanged hydrogens in that segment, and let $q_{j,k}$ be its probability generating function:

$$q_{j,k}(x) = \sum_{n=0}^{n_j^{ex}} \delta_{j,k,n}^S \, x^n.$$

Analogously, for peptide i we define the random variable $X_{i,k}$ and its generating function

$$Q_{i,k}(x) = \sum_{n=0}^{N_i^{ex}} \delta_{i,k,n}^P \, x^n.$$

Because a peptide consists of several segments, the total number of exchanged hydrogens in peptide i is the sum of the contributions from its segments,

$$X_{i,k} = \sum_{j \in V(i)} Y_{j,k}.$$

Under the assumption that different segments exchange independently, the corresponding generating functions satisfy

$$Q_{i,k}(x) = \prod_{j \in V(i)} q_{j,k}(x). \tag{4.5}$$

Thus, peptide-level probabilities $\delta_{i,k,n}^P$ are obtained directly from the coefficients of the polynomial $Q_{i,k}(x)$, which in turn are determined by the segment-level probabilities $\delta_{j,k,n}^S$, themselves parametrized through the coefficients $\beta_j, \beta_{j,1}, \ldots, \beta_{j,n_j^{ex}}$.

4.3.2 Model fitting

In this section, we describe methods that can be used to fit the proposed model to data.

4.3.2.1 Loss functions

Let us denote observed intensities of isotopic peaks for peptide $P_i, i = 1, \ldots, I$ at time $T_k, k = 1, \ldots, K$ by $O_{i,k,j}, j = 0, \ldots, n_i^{ex} + n_i^0 - 1$. Here, index j denotes the mass shifts of each peak compared to the monoisotopic mass. Let the total intensity of a spectrum be denoted by $O_{i,k} = \sum_{j=0}^{n_i^{ex}+n_i^0-1} O_{i,k,j}$. Finally, let us denote the set of parameters $\beta_{j,m}$ that define probabilities $\delta_{j,m}^S, j = 1, \ldots, J, m = 0, \ldots, M_J$ as $\boldsymbol{\beta} = (\beta_1, \beta_{1,0}, \ldots, \beta_{1,M_1}, \beta_2, \beta_{2,0}, \ldots, \beta_{J,M_J})^T$.

The proposed approach compares observed intensities $O_{i,k,j}$ to the expected intensities given by Equation 4.1 to evaluate the similarity of observed peptide-level isotopic distributions to predicted distributions derived from the segment-level probabilities. Loss function that can be used for this comparison depends on the particular statistical model and its assumptions. In this section, we discuss two variants: ordinary least-squares (OLS) and pseudolikelihood generalized least-squares (PL-GLS).

First, let us consider a standard homoscedastic Gaussian error model:

$$O_{i,k,j} = E_{i,k,j} + \varepsilon_{i,k,j}, \varepsilon_{i,k,j} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2). \tag{4.6}$$

This assumption naturally leads to the following loss function:

$$\ell_{OLS}(\boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{i}^{I} \sum_{j=0}^{n_i^{ex}+n_i^0-1} \left\{ E_{i,k,j}(\boldsymbol{\beta}) - O_{i,k,j} \right\}^2. \tag{4.7}$$

The resulting minimization problem is complex and non-convex due to products of segment-level probabilities. The following Section 4.3.2.2 describes an application of global optimization methods to solving this problem. First, however, we discuss an alternative noise structure.

Model given by Equation 4.6 assumes that the variance associated with each peak is the same and independent of its intensity. To better model the noise structure of HDX-MS data, it is also useful to consider a model where the variance of the random error depends on the intensity of a peak (Q. Zhu, Valkenborg, and Burzykowski, 2010; Morris et al., 2008). Thus, we also consider a power-of-the-mean variance model, which is special case of the model given by Equation 2.9 with $g\left(\beta,\theta\right)=E_{i,k,j}^{\theta}$:

$$O_{i,k,j}=E_{i,k,j}+\varepsilon_{i,k,j},\varepsilon_{i,k,j}\sim\mathcal{N}\left(0,\sigma^{2}E_{i,k,j}^{2\theta}\right). \tag{4.8}$$

Errors $\varepsilon_{i,k,j}$ are assumed to be independent. Parameter θ denotes an unknown power of the expected intensity which controls the variance of the observed peak-intensity.

Under these assumptions, the full likelihood function $L(\beta)$ is given by

$$L(\beta)=\prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{j=1}^{n_i^{ex}+n_i^0-1}\left(\frac{1}{\sqrt{2\pi}\sigma E_{i,k,j}^{\theta}}\exp\left(\frac{(O_{i,k,j}-E_{i,k,j}(\boldsymbol{\beta}))^{2}}{2\sigma^{2}E_{i,k,j}^{2\theta}(\boldsymbol{\beta})}\right)\right).$$

For simplicity, let us omit the dependence of $E_{i,k,j}$ on $\boldsymbol{\beta}$. Then, the negative log-likelihood function ℓ_{GLS} can be written as

$$\ell_{GLS}(\beta)=N\log\sigma+\theta\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\log\left(E_{i,k,j}\right)+\frac{1}{2\sigma^{2}}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\left(\frac{O_{i,k,j}-E_{i,k,j}}{E_{i,k,j}^{\theta}}\right)^{2}. \tag{4.9}$$

Hence, the full likelihood function is a complex function of model parameter $\boldsymbol{\beta}$, unknown standard deviation σ, and an unknown mean-variance relationship parameter θ. Let us note that the final last term can be seen as an example of a weighted least squares, in which squared residuals $(O_{i,k,j}-E_{i,k,j})^{2}$ are weighted by the variance-related terms $(\sigma E_{i,k,j})^{-2}$.

A standard approach to solving this problem is an application of the PL-GLS approach given by Algorithm 2 (Davidian and Giltinan, 1995). The first step of this approach requires finding an explicit estimator of σ^2 as a function of other parameters. Let us denote the total number of observations (all observed peaks in all spectra for all peptides in a cluster) as N. Then, the estimator $\widehat{\sigma}^2$ is equal to

$$\widehat{\sigma}^{2}(\boldsymbol{\beta})=\frac{1}{N}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\left(\frac{O_{i,k,j}-E_{i,k,j}}{\sigma E_{i,k,j}^{\theta}}\right)^{2}.$$

Substituting this estimator into Equation 4.9 yields

$$\tilde{\ell}^{*}{}_{GLS}=\frac{N}{2}\left[\log\left(\frac{1}{n}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\left(\frac{O_{i,k,j}-E_{i,k,j}}{E_{i,k,j}^{\theta}}\right)^{2}\right)+\frac{2\theta}{n}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\log\left(E_{i,k,j}\right)+1\right] \tag{4.10}$$

where the constant 1 appears because the summation terms that define $\widehat{\sigma}^2$ are the same as the last term in Equation 4.9. Using the properties of the logarithmic function, the middle term can be re-written as a product of powers of the expected intensity $E_{i,k,j}$. Then, the profile log-likelihood function 4.10 becomes equal, up to constants,

$$\begin{aligned}\tilde{\ell}^{*}{}_{GLS}&=\log\left(\frac{1}{N}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\left(\frac{O_{i,k,j}-E_{i,k,j}}{E_{i,k,j}^{\theta}}\right)^{2}\right)+\log\left(\prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{j=0}^{n_i^{ex}+n_i^0-1}\left(E_{i,k,j}^{\frac{2\theta}{N}}\right)\right)\\&=\log\left(\frac{1}{N}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=0}^{n_i^{ex}+n_i^0-1}\left(\frac{O_{i,k,j}-E_{i,k,j}}{E_{i,k,j}^{\theta}}\right)^{2}\prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{j=0}^{n_i^{ex}+n_i^0-1}\left(E_{i,k,j}^{\frac{2\theta}{N}}\right)\right).\end{aligned}$$

Denoting $\widehat{\mu} = \left[\prod_{i=1}^{I} \prod_{k=1}^{K} \prod_{j=0}^{n_i^{ex}+n_i^0-1} (E_{i,k,j}) \right]^{\frac{1}{N}}$, we can see that optimizing $\tilde{\ell}^*{}_{GLS}$ is equivalent to minimizing

$$\tilde{\ell}_{GLS}(\beta) = \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{j=0}^{n_i^{ex}+n_i^0-1} \left[(O_{i,k,j} - E_{i,k,j}) \left(\frac{\widehat{\mu}}{E_{i,k,j}} \right)^{\theta} \right]^2. \tag{4.11}$$

In some contexts, we will also denote $\tilde{\ell}_{GLS}(\boldsymbol{\beta})$ as $\tilde{\ell}_{GLS}(\boldsymbol{\beta}, \theta)$ to emphasize the dependence on θ parameter. Minimization of (4.11) can be viewed as a weighted least-squares (WLS) problem for estimating $\boldsymbol{\beta}$, with weights

$$W_{i,k,j}(\boldsymbol{\beta}, \theta) = \left(\frac{\widehat{\mu}}{E_{i,k,j}(\boldsymbol{\beta})} \right)^{\theta}. \tag{4.12}$$

As a result, estimates of $\boldsymbol{\beta}$ and θ can be obtained by a simple iterative procedure (Davidian and Giltinan, 1995; Q. Zhu, Valkenborg, and Burzykowski, 2010), which applies the general Algorithm 2 to this choice of a mean-variance function. The following section connects optimization of both loss functions ℓ_{OLS} and $\tilde{\ell}_{GLS}$, and describes a proposed approach to model fitting.

4.3.2.2 Optimization of proposed loss functions

The proposed model can be fitted to data by using the loss function ℓ_{OLS} or $\tilde{\ell}_{GLS}$. The latter has better properties in terms of estimation of variance of model parameters, which we will discuss later. Minimization of both loss functions uses the same building block of finding the minimum of

$$\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{j=0}^{n_i^{ex}+n_i^0-1} \left[(O_{i,k,j} - E_{i,k,j})\, W_{i,k,j} \right]^2 \tag{4.13}$$

where $W_{i,k,j} = 1$ for ℓ_{OLS} and $W_{i,k,j} = \left(\frac{\widehat{\mu}}{E_{i,k,j}} \right)^{\theta}$ for $\tilde{\ell}_{GLS}$. As the PL-GLS iterative algorithm treats weights as fixed while optimizing function given by Equation 4.13, the same optimization approach can be used in both cases, with the only difference in terms of the definition of $W_{i,k,j}$.

Let J be the total number of segments identified in an experiment, M denote the maximum number of iterations, and tol denote the tolerance for comparing consecutive estimates of β. Algorithm 4 summarizes the proposed method of minimizing the loss function $\tilde{\ell}_{GLS}$.

The optimization problem in Step (4) is a simple univariate least-squares problem, and it can be solved with any global optimization method. Optimization problems involving both loss functions ℓ_{OLS} and Step (6) of the PL-GLS approach for $\tilde{\ell}_{GLS}$ can be solved using standard optimization routines. Example approaches include gradient descent, Newton-Raphson method variants (such as the Broyden-Fletcher-Goldfarb-Shanno method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) or its limited-memory modification Byrd et al., 1995 for constrained problems), or the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). We optimize both functions using a variant of the Netwon-Raphson method. In our implementation, we use several random starting points for the OLS fit to account for non-convexity of the problem, and select the one that produces a solution with the smallest value of the loss function.

4.3.2.3 Analytical gradient for proposed loss function

Gradient of the loss function Let us consider the OLS problem given by Equation 4.7. Gradient with respect to the vector of β parameters is given by

$$\frac{\partial}{\partial \beta} \ell_{OLS}(\boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{i}^{I} \sum_{j=0}^{n_i^{ex}+n_i^0-1} 2 \left\{ E_{i,k,j}(\boldsymbol{\beta}) - O_{i,k,j} \right\} \frac{\partial}{\partial \beta} E_{i,k,j}(\boldsymbol{\beta}).$$

Algorithm 4: Iterative method of estimating the proposed model with the $\tilde{\ell}_{GLS}$ loss function.

Input: $O_{i,k,j}$ - observed intensities of isotopic peaks,
$V(i), i = 1, \ldots, I$ - segment memberships.

Output: $\delta^S_{j,k,n}\ j = 1, \ldots, J, k = 1, \ldots, K, n = 1, \ldots, n^{ex}_j$: estimated segment-level exchange probabilities,
$\widehat{\theta}$: estimated mean-variance relationship parameter,
$\widehat{\sigma}^2$: estimated variance σ^2.

Initialize $\widehat{\beta}^{(0)} = 0$, $\widehat{\beta}^{(1)} = \arg\min \ell_{OLS}(\beta)$ [Equation 4.7]
$i \leftarrow 1$
while $||\widehat{\beta}^{(i)} - \widehat{\beta}^{(i-1)}||^2 > tol$ *and* $i \leq M$ **do**
  $\widehat{\theta}^{(i+1)} = \arg\min_\theta \tilde{\ell}_{GLS}(\widehat{\beta}^{(i)}, \theta)$ [Equation 4.11 as a function of θ]
  $\widehat{W}^{(i+i)} \leftarrow W(\widehat{\beta}^{(i)}, \widehat{\theta}^{(i+1)})$ [Equation 4.12]
  $\widehat{\beta}^{(i+i)} \leftarrow \arg\min_\beta \tilde{\ell}_{GLS}(\beta, \widehat{\theta}^{(i+1)}, \widehat{W}^{(i+1)}$ [Equation 4.11 as a function of $\boldsymbol{\beta}$]
  $i \leftarrow i + 1$
end

Similarly, for the loss function $\tilde{\ell}_{GLS}$ with fixed weights $W_{i,k,j}$,

$$\frac{\partial}{\partial \beta}\tilde{\ell}_{GLS}(\boldsymbol{\beta}|\theta, W_{i,k,j}) = \sum_{k=1}^{K}\sum_{i}^{I}\sum_{j=0}^{n^{ex}_i+n^0_i-1} 2\left\{E_{i,k,j}(\boldsymbol{\beta}) - O_{i,k,j}\right\} W_{i,k,j}\frac{\partial}{\partial \beta}E_{i,k,j}(\boldsymbol{\beta}).$$

Hence, it is sufficient to consider the derivatives of the expected peak intensities for each peptide i, time T_k, and peak $j = j(i,k)$.

Gradient with respect to the expected peak heights Let us consider a single peak $E_{i,k,j}$ and derivatives of $E_{i,k,j}$ with respect to parameters that describe a fixed segment S_j. If $j \notin V(i)$, all these derivatives are equal to 0, so it is enough to consider segments S_j such that $j \in V(i)$. Each expected peak intensity $E_{i,k,j}$ involves probabilities derived from all matching segments. However, while taking the derivative with respect to parameter $\beta_{j,n}$, only probabilities describing segment j will result in a non-trivial value. Hence, we can decompose each peptide-level probability $\delta^P_{i,k,n}$ in the following way:

$$\begin{aligned}
&\mathbb{P}\left(\text{peptide } i \text{ exchanged } n \text{ hydrogens at time } T_k\right)\\
&= \mathbb{P}\left(\text{Segment } j \text{ exchanged } 0 \text{ hydrogens}\right)\mathbb{P}(\text{Segments other than } j \text{ exchanged } n \text{ hydrogens})\\
&+ \mathbb{P}\left(\text{Segment } j \text{ exchanged } 1 \text{ hydrogen}\right)\mathbb{P}(\text{Segments other than } j \text{ exchanged } n-1 \text{ hydrogens})\\
&\ldots\\
&+ \mathbb{P}\left(\text{Segment } j \text{ exchanged } n \text{ hydrogens}\right)\mathbb{P}(\text{Segments other than } j \text{ exchanged } 0 \text{ hydrogens})
\end{aligned}$$

Let us denote the probability $\mathbb{P}$(Segments other than j exchanged n hydrogens) as $\delta^S_{-j,k,n}$. Each probability $\delta^S_{-j,k,n}$ is a constant with respect to parameters $\beta_j, \beta_{j,n}, n = 0, \ldots, n^{ex}_j$. To summarize, for each peptide-level probability $\delta^P_{i,k,n}$, and a fixed segment S_j, we have

$$\delta^P_{i,k,n} = \sum_{m=\max(0,n-(N^{ex}_i)-n^{ex}_j)}^{\min(n,n^{ex}_j)} \delta^S_{j,k,m}\delta^S_{-j,k,n-m}. \tag{4.14}$$

It follows that it is enough to find an explicit formula for $\frac{\partial}{\partial \beta_{j,m}}\delta^S_{j,k,n}$, and aggregate the resulting derivatives over time points, peaks, and decomposed probabilities $\delta^P_{i,k,n}$.

Gradient with respect to the segment-level probabilities Let us define

$$Tot_{j,k} = 1 + \exp(\exp(\beta_{j,n_j^{ex}})t + \beta_j) + \sum_{n=1}^{n_j^{ex}-1} \exp(\beta_{j,n}t + \beta_j).$$

Then, a direct calculation shows that for a given segment j, time $t = T_k$, and number of exchanged hydrogens n the derivatives of interest are given by

$$\frac{\partial}{\partial \beta_{j,m}} \delta^S_{j,k,n} = \begin{cases} \frac{-t\exp(\beta_{j,m}t+\beta_j)}{Tot^2_{j,k}}, m \leq n_j^{ex}, n = 0 \\ \frac{-t\exp\left(\exp(\beta_{j,n_j^{ex}})t+\beta_{j,n_j^{ex}}+\beta_j\right)}{Tot^2_{j,k}}, m = n_j^{ex}, n = 0, \\ \frac{((Tot_{j,k}-\exp(\beta_{j,m}t+\beta_j))t\exp(\beta_{j,m}t+\beta_j)}{Tot^2_{j,k}}, m = n, 0 \leq n \leq n_j^{ex} \\ \frac{((Tot_{j,k}-\exp(\beta_{j,n_j^{ex}}t+\beta_j))t\exp(\beta_{j,n_j^{ex}}t+\beta_j)\exp(\exp(\beta_{j,n_j^{ex}}))}{Tot^2_{j,k}}, m = n = n_j^{ex} \\ \frac{-t\exp(\beta_{j,n}t+\beta_j)\exp(\beta_{j,m}t+\beta_j)}{Tot^2_{j,k}}, m \neq n, 0 \leq n \leq n_j^{ex} \\ \frac{-t\exp(\beta_{j,n}t+\beta_j)\exp(\exp(\beta_{j,n_j^{ex}})t+\beta_j)\exp(\exp(\beta_{j,n_j^{ex}}))}{Tot^2_{j,k}}, m \neq n, m = n_j^{ex} \end{cases} \tag{4.15}$$

Moreover, for the intercept β_j we have:

$$\frac{\partial}{\partial \beta_j} \delta^S_{j,k,n} = \begin{cases} \frac{-(Tot_{j,k}-1)}{Tot^2_{j,k}}, n = 0 \\ \frac{\exp(\beta_{j,n}t+\beta_j)}{Tot^2_{j,k}}, 0 \leq n \leq n_j^{ex} \\ \frac{\exp(\exp(\beta_{j,n})t+\beta_j)}{Tot^2_{j,k}}, n = n_j^{ex} \end{cases} \tag{4.16}$$

After aggregating over times, peptides, and peaks, these derivatives can be used to search for the optimal set of β parameters using either gradient-based methods or second order methods.

4.3.2.4 Covariance of estimated parameters

Uncertainty associated with estimated segment-level probabilities may be of interest, both to evaluate the precision of point estimates and to compare exchange probabilities in different biological conditions.

In OLS case, the standard estimator of the asymptotic variance-covariance matrix of parameters V_β in non-linear least squares problems can be approximated by $(J^T J)^{-1}$, where J denotes the Jacobian of the loss function (Gavin, 2019), or by $s(\widehat{\boldsymbol{\beta}})H^{-1}$, where $s(\beta)$ denotes the value of the objective function at the solution, and H^{-1} is the inverse of Hessian matrix of the loss function calculated at the solution.

In case of the $\tilde{\ell}_{GLS}$ loss, a formula for the variance-covariance matrix of $\boldsymbol{\beta}$ parameters is given by Davidian and Giltinan, 1995:

$$\Sigma(\beta) = \left[\sigma^2 \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{U_m} E^{2\theta}_{m,k,j}(\beta)\frac{\partial E_{m,k,j}(\beta)}{\partial \beta}\left[\frac{\partial E_{m,k,j}(\beta)}{\partial \beta}\right]^T\right]^{-1}. \tag{4.17}$$

In both cases, uncertainties of parameters β can be used to approximate the uncertainties of exchange probabilities using the Delta method. Towards this aim, consider the fixed exchange probability $\delta^S_{j,k,n}(\beta)$ and the transformation $F(\beta) : \beta \mapsto \delta^S_{j,k,n}(\beta)$. Then, the variance of $\delta^S_{j,k,n}(\beta)$ is given by $\nabla F^T V_\beta \nabla F$, where the gradient of F is computed at the estimated parameters.

4.3.3 Data processing

Typical HDX-MS data processing tools return or use aggregated deuteration values estimated based on masses of isotopic peaks in observed spectra (Seetaloo, Kish, and Phillips, 2022b; Z. Zhang, A. Zhang, and Xiao, 2012; Liu et al., 2011; Pascal et al., 2012). Estimating the proposed model requires alternative data processing to extract complete spectral data.

We assume that peptide identification has been performed and begin with a list of all identified peptide ions, possibly with post-translational modifications. Additionally, information about specific spectra in which peptides were identified or retention time ranges can help reduce the amount of data that needs to be searched to extract the isotopic peaks.

General approach to extracting isotopic peaks Let us consider a single scan (MS1 spectrum) measured in an HDX-MS study. Such a spectrum consists of pairs (m_i, I_i), where m_i denotes an m/z value, while I_i is the associated intensity, $i = 1, \ldots, N$, where N is the total number of peaks in this particular spectrum. Let us fix our attention on a single peptide ion. The task of determining whether isotopic peaks originating from this peptide are found in the spectrum of interest may be based on a simple comparison of theoretical and observed masses. Theoretical isotopic masses can be calculated based on the elemental composition of the peptide, as the isotopic distributions of atoms are known. We used the BRAIN algorithm (Dittwald, Claesen, et al., 2013; Dittwald and Valkenborg, 2014) for this task. Expected m/z values can then be calculated by using the formula

$$mz_i^{exp} = m_{prot} + m_i^{exp}/z,$$

where mz_i^{exp} denotes the mass-to-charge ratio for i-th peak $i = 0, \ldots, N_0$ (where N_0 is the selected maximum mass shift), m_i^{exp} denotes the calculated theoretical isotopic mass of a peak corresponding to a mass shift of i, and z denotes the charge value. Moreover, m_{prot} is the mass of a proton, approximately equal to $1.007Da$ (higher precision was used for actual computations). Then, m/z values of all observed peaks within a reasonable range that includes all m/z values mz_i^{exp} are compared to theoretical m/z values and selected as isotopic peaks if the relative difference (as defined by Equation 2.10) is smaller than a chosen tolerance value. We used a tolerance of 30 ppm. When identification data provide detailed information about retention times or spectra where a given peptide was identified, this procedure requires only checking a small number of spectra. When such information is lacking, it may be necessary to iterate over a wide range of spectra. After identifying isotopic peaks, we usually require the presence of a monoisotopic peak among the selected peaks and a minimum number of peaks per spectrum. Moreover, for undeuterated peptides, it is possible to apply filtering based on a comparison between relative peak intensities and theoretical isotopic probabilities.

Undeuterated peptides As indicated earlier, to obtain the range of mass shifts (equivalently: isotopic masses) at which isotopic peaks may be expected, the number of observable peaks in the isotopic distribution of an undeuterated peptide is required. If a given experiment included an additional MS run with undeuterated peptides, it is possible to search the spectra for peptides identified in MS runs for deuterated samples. Then, the number of isotopic peaks for the undeuterated peptide can be deduced from such spectra. Otherwise, an assumption is required on the number of peptides. One of the possible heuristics in this case can be based on selecting a number of peaks that provides a fixed isotopic probability coverage. However, in general, a lack of information about undeuterated isotopic distributions may lead to the addition of spurious, noisy peaks at the tail of the isotopic distribution, or conversely, missing peaks in the tail of the distribution.

If a given experiment included an additional MS run with undeuterated peptides, the number of undeuterated isotopic peaks for each peptide can be derived from data by applying the aforementioned principle of detecting isotopic peaks and counting them.

Deuterated peptides In case of deuterated peptides, it is necessary to search the entire range of mass shifts between 0 (monoisotopic peak) and $N_i^{ex} + N_i^0 - 1$. The method of finding isotopic peaks is the same as described above, but the number of peaks is larger. Additionally, depending on the

rate of exchange, at specific time points, peaks corresponding to a very low or a very high number of exchanged hydrogens may have too small intensities to be identified. Hence, on one hand, the number of detected peaks may not achieve the theoretically possible number. On the other hand, peaks may be missing due to low intensities.

Again, extracting isotopic peaks of deuterated peptides may require iterating the procedure over a large number of spectra if no additional information about the identifications is provided. However, in this case, as opposed to the undeuterated case, the exchange probabilities are unknown, and no filtering based on a comparison to expected relative intensities can be performed. Hence, quality control of such spectra remains challenging.

4.3.3.1 Case study: application of the proposed data processing

In this section, we provide details of the data processing steps required to create a proper input to the proposed statistical model based on an example peptide identified in the **msHDX** case study. This data set exhibited higher complexity, both in terms of the segment-peptide graph structure and data processing, due to its size. Let us consider peptide YMGRTLQNT. It was identified in two charge states (1 and 2) in several runs of the experiment. Let us focus on the doubly charged variant.

As previously discussed, peptide identifications are subject to various types of errors. Hence, it is appropriate to verify the presence of the actual signal in the reported m/z range and retention time. First, let us consider the undeuterated peptide. The number of isotopic peaks in its envelope is not known. We considered 10 possible peaks. As a consequence, for each peptide P_i, we searched for a maximum number of $9 + N_i^{ex}$ peaks. Using such a wide range of candidate m/z values may lead to the identification of spurious peaks. However, it has two benefits. Firstly, it leads to more identified peaks in cases where the undeuterated distributions truly consist of a large number of peaks. Secondly, it provides information about possible overlaps with isotopic envelopes of other peptide ions. Using an appropriate tolerance for mass errors can offset the problem with detecting spurious peaks. Here, we used a tolerance of 30 ppm. Figure 4.5 presents a plot of the total intensity of peaks over such a range of m/z values across different retention times. Each time corresponds to a single spectrum. A possibly slightly skewed bell curve pattern, such as this one, is typical for the elution of a peptide. It indicates the presence of a signal in the region of an identified peptide; however, we verify its compatibility with the expected isotopic distribution associated with this sequence.

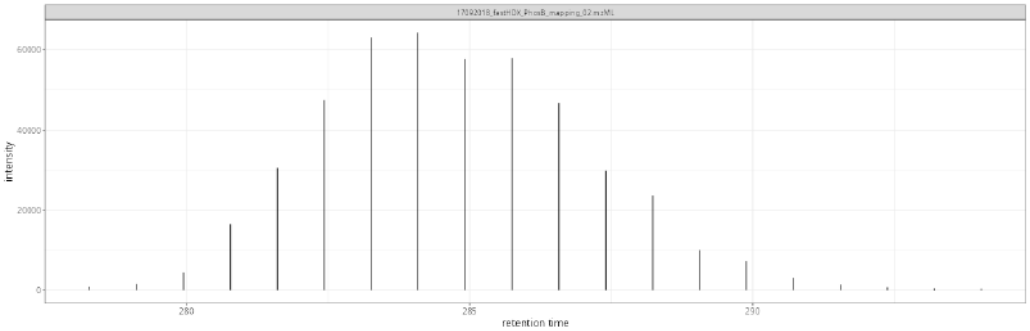
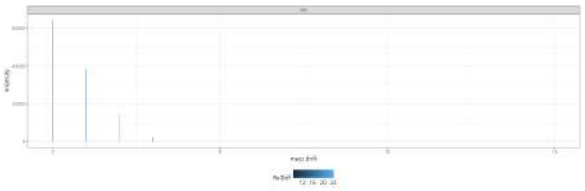


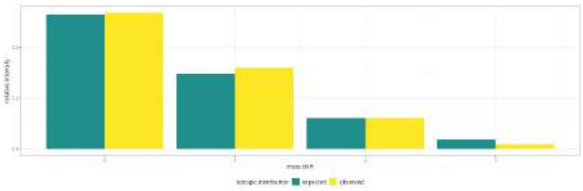
Figure 4.5: **msHDX case study: example plot of total intensity over a range of m/z values associated with peptide YMGRTLQNT**. The length of lines indicates total intensity at a given retention time.

In MS data analysis, higher-intensity peaks are commonly associated with higher-quality data. Hence, we focused on spectra corresponding to the highest total intensity, and each peptide ion at a given time was represented by a single spectrum per MS run. Hence, a single spectrum was selected for each peptide, MS run, and time point. Figure 4.6 (a) displays such a spectrum. As the range of m/z values is wide, it includes some very low intensity peaks in the right tail. A simple condition

that removes such noisy peaks is to require no gaps between consecutive peaks. Hence, Figure 4.6 (b) compares the first four peaks in this spectrum that correspond to the monoisotopic peak and three consecutive mass shifts to an expected isotopic distribution of this peptide. Observed intensities were scaled by their sum to make them comparable to isotopic probabilities.



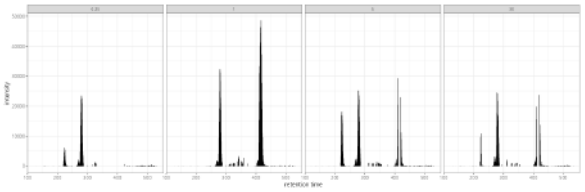
(a) Raw spectrum for the undeuterated peptide YMGRTLQNT.



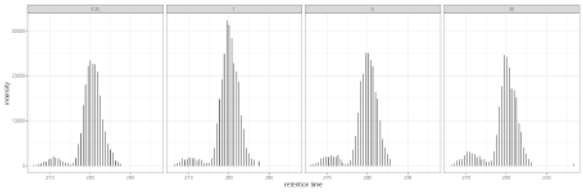
(b) A comparison of relative intensities of the first four peaks in the spectrum presented in (a) and the expected isotopic distribution for this peptide.

Figure 4.6: **msHDX study: spectrum of the undeuterated peptide YMGRTLQNT exhibited a pattern similar to theoretical isotopic probabilities**.

Based on visual inspection, expected and observed relative intensities were similar. Automated quality control can utilise statistical tests to compare true and observed multinomial distributions. However, such an approach would face two challenges: low sample size and an unknown number of true peaks in the spectrum. Hence, we only used Pearson's chi-squared-type tests for rough exploratory analysis rather than a rigorous analysis. In the case of the peptide YMGRTLQNT, we conclude that there were four peaks in its undeuterated spectrum based on mass errors and similarity of isotopic patterns. It includes 8 exchangeable hydrogens. Hence, a maximum mass shift of interest was 11.



(a) Total intensities related to YMGRTLQNT peptide across a full range of retention times at 4 experimental time points.



(b) Total intensities related to YMGRTLQNT peptide in a shorter retention time range in which signal was found for each experimental time point.

Figure 4.7: **msHDX study: total intensity of signal related to peptide YMGRTLQNT fluctuated over retention time, but there was a common range of retention times at which high intensities were observed**.

Identifying the isotopic peaks of the undeuterated peptide enabled us to narrow down the range of m/z values, as the gap between consecutive peaks is approximately 1 Da divided by the charge value, and the maximum number of peaks was known. Moreover, the retention time at which this spectrum was measured was known; hence, it would be possible to limit the search for related spectra at later experimental time points to retention times close to this value. However, we searched for a signal in the entire set of MS1 spectra. Figure 4.7 (a) presents total intensities in the relevant m/z range across all retention times for 4 examples of experimental time points presented. This enabled us to narrow down a range of high-intensity spectra that occurred at the same times in different runs of the experiment, which corresponded to different experimental time points. Figure 4.7 (b) presents total intensities in that range. Again, a tolerance of 30 ppm was used.

Finding the range of high intensities presented in Figure 4.7 (b), we were able to find a set of spectra that potentially included the signal related to the peptide of interest at each experimental time point.

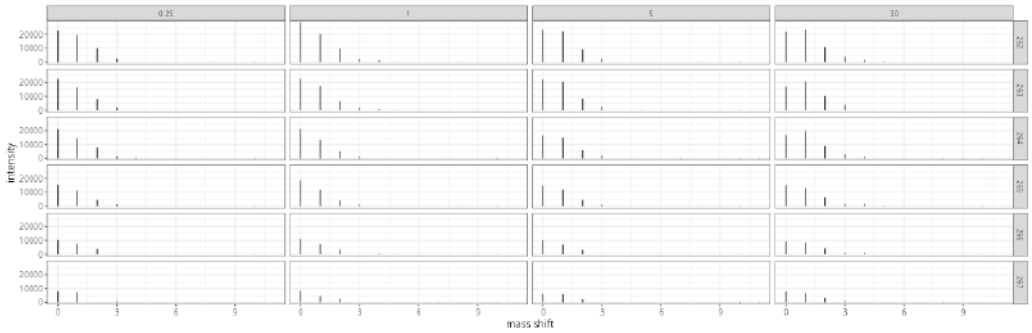


Figure 4.8: **msHDX case study: isotopic patterns in several consecutive spectra at each experimental time point**. Columns indicate experimental time points while rows denote MS1 spectra. Each panel presents a single spectrum for the YMGRTLQNT peptide.

Figure 4.8 presents a set of consecutive MS1 spectra that corresponded to the highest total intensities. Each spectrum consisted of a set of contiguous peaks starting with the monoisotopic peak and a group of small peaks in the right tail of the distribution that were likely noise peaks, but potentially could be a result of fast exchange in some residues of the peptide. Finally, Figure 4.9 presents a set of spectra selected for the peptide of interest. Potentially noisy peaks can be removed from such spectra based on their low relative intensity, the presence of gaps between mass shifts, or other criteria.

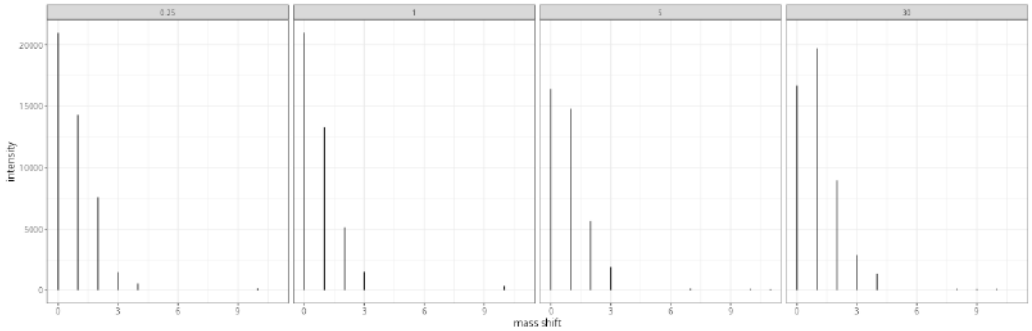


Figure 4.9: **msHDX case study: selected spectra that describe the changes in isotopic distribution of the YMGRTLQNT peptide over time**. Columns indicate experimental time points.

We performed this data extraction using a custom R/Shiny application (W. Chang et al., 2022) with a simple graphical user interface, which enabled the assessment of spectra in the presented manner. However, this approach still requires manual curation of the data. The development of a fully automated, open-source workflow for processing isotopic envelopes of peptides measured in HDX-MS studies remains an open problem. We discuss this issue further in Section 4.6

4.4 Evaluation

We applied the proposed approach to two biological data sets with different levels of complexity of the segment-peptide structures, and a set of simulated data. In this section, we describe the design of a simulation study and introduce metrics which will be used to evaluate the proposed approach. All simulated data and fitted models can be accessed in a reproduction repository https://github.com/m-staniak/HDX_reproduction.

4.4.1 Simulated HDX-MS spectra

In order to simulate HDX-MS data for a fixed segment-peptide structure of data it is sufficient to select model parameters β to determine segment-level exchange probabilities, compute peptide-level exchange probabilities using Equation 4.5, and finally calculate expected peptide intensities based on Equation 4.1. In this section, we provide details of the data-generating process. In each simulation study, we used 50 repetitions of the experiment.

4.4.1.1 Cluster structure

We considered two types of data: low-resolution and high-resolution. The latter case follows the segment-peptide structure found in the msHDX case study, with short segments, typically consisting of a single residue. The former case resembles the HVEM case study, with longer segments, and consequently more probabilities required to describe each segment.

We generated three variants of a peptide-segment cluster. Two of them represented the case of slightly longer segments, while one represented the case of many single residue segments. Table 4.2 presents descriptive statistics for all three data sets, including information about simulated spectral data which we discuss in Section 4.4.1.3. Data sets 1-2 consisted of 5-8 peptides with 6-8 segments. Each segment included 2-4 exchangeable hydrogens. Data set 3 consisted of 19 peptides and 20 segments, where each segment was a single residue. Sequence of peptides, and therefore segments, were sampled uniformly from the set of amino acids excluding the non-exchanging proline. For simplicity, we calculated the number of exchangeable hydrogens as the number of AAs in a sequence, ignoring the standard assumption that the H/D exchange for the first AA is unobservable.

Data set	No. peptides	No. segments	No. ex. range	Ave. no. peaks	N_{tot}
1	5	6	2 - 3	9.47 (8 - 11)	288
2	8	8	2 - 4	11.27 (7 - 15)	552
3	19	20	1 - 1	11.34 (6 - 14)	1326

Table 4.2: **Simulated data sets**: characteristics of simulated scenarios both in terms of cluster and spectral data structure. For each data set 1, 2, and 3 we provide the number of peptides and segments, range of numbers of exchangeable hydrogens per segments (*No. ex. range* column), average number of non-zero peaks per spectrum (*Ave. no. peaks* column, with the range of peaks per spectrum given in parentheses. Column N_{tot} denotes the total sample size (the number of observed peaks in all spectra).

4.4.1.2 Segment-level modeling

Generating segment-level data requires a choice of $\boldsymbol{\beta}$ parameters in Equation 4.4 which is related to the range of values of time T_k. Together, they determine the range of probabilities observable in a given experiment. The time variable can be expressed in various units, and we selected an arbitrary sequence of values $T_k \in \{0.1, 0.5, 1, 2.5, 5, 10\}$. The small number of time points limited the total data size without restricting the range of probabilities or introducing any practical estimability issues. With fixed values of experimental times, it is possible to find a range of β parameter values that result in a wide enough range of observable probabilities. We found that β values ranging from about -1 to an arbitrarily small value provided a good range of possible probabilities. Hence, we sampled β parameters from the range of $[-0.5, -0.01]$. Additionally, we split that interval into three sub-intervals $[-1, -0.5]$, $[-0.5, -0.1]$, $[-0.1, -0.01]$ to create set of consecutive segments with similar rates of exchange. Resulting values of β parameters can be found in the reproduction repository. Data sets 1, 2, nd a3 were characterized by 20, 30, and 40 parameters, respectively. Figure 4.10 presents an example of simulated segment-level probabilities for data set 1. Most segments reached full deuteration,

and differed mostly in the behavior of probabilities of exchanging a lower than maximum number of hydrogens.

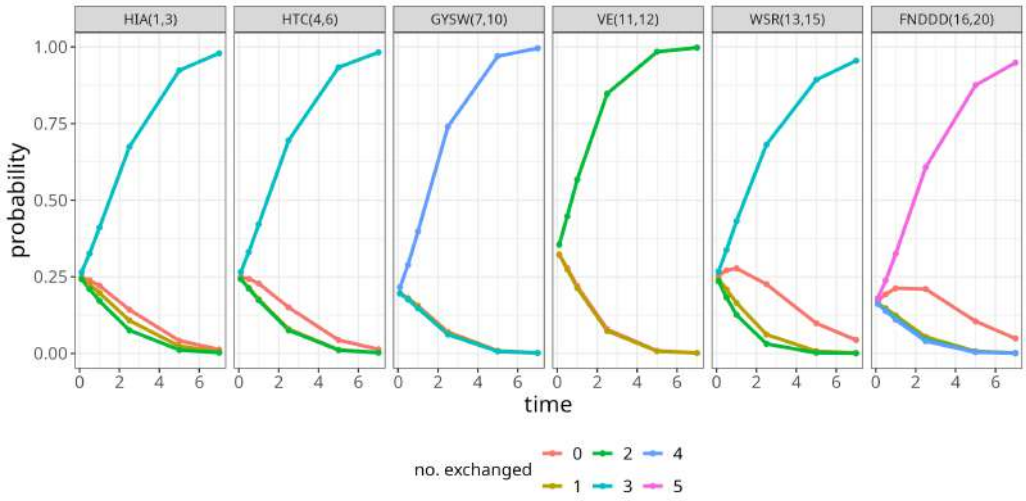


Figure 4.10: **Simulated HDX-MS data**: example set of segment-level probabilities from simulated data set 1.

4.4.1.3 Peptide-level modeling and simulated spectral data

Peptide-level exchange probabilities were calculated via a convolution based on segment-level probabilities. Expected intensities of isotopic peaks were computed using Equation 4.1 with a total intensity of a spectrum $O_{i,k} = 100$. We considered two types of random error: homoscedastic and heteroscedastic. In the former case, noise was added to simulated intensities $O_{i,k,j}$ using the formula

$$O_{i,k,j} = E_{i,k,j} + \varepsilon_{i,k,j}, \varepsilon_{i,k,j} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{hom}) \tag{4.18}$$

for $\sigma_{hom} \in \{0.5, 1\}$. In the heteroscedastic case, the noise was added using the following formula:

$$O_{i,k,j} = E_{i,k,j} + E_{i,k,j} \cdot \varepsilon_{i,k,j}, \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{het}). \tag{4.19}$$

We used values of $\sigma_{het} \in \{0.01, 0.05\}$. Let us note that taking the logarithm of both sides of Equation 4.19 implies that a model where log-intensities $\log O_{i,k,j}$ are represented as a function of $\log E_{i,k,j}$ and a homoscedastic noise term $\log(1 + \varepsilon_{i,k,j})$ provided that $1 + \varepsilon_{i,k,j} > 0$. However, as the proposed model uses raw peak-intensities, simulations based on Equation 4.19 describe its properties under heteroscedastic error.

Figure 4.11 displays intensities of isotopic peaks generated by the probabilities shown in Figure 4.10 with a heteroscedastic error (Equation 4.19). We used this model structure for most simulation studies as it is more representative of the noise patterns in biological studies. Table 4.2 provides a summary of resulting data sets. The number of peptides ranged from 5 to 19, while the total sample sizes (the number of observed peaks in the spectra) varied between 288 and 1326. Hence, the simulated data sets covered both the case of a small cluster, and a more complex peptide-segment structure.

Let us note that as the OLS criterion corresponds to the homoscedastic errors assumption, each simulation study with data generated with Equation 4.19 tests the proposed model's precision in case of a misspecified variance structure.

Moreover, we considered two modifications of the described data structure. Firstly, to evaluate the variance estimation quality, we generated data with additional technical replicates. In this case, for each time point and peptide ion we generated two spectra with independent random noise. Secondly, to

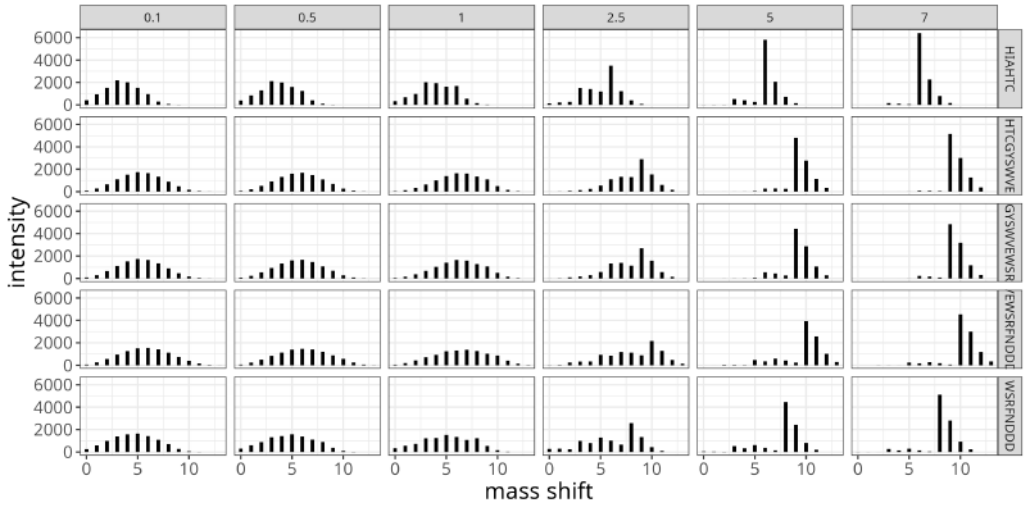


Figure 4.11: HDX-MS spectra simulated from the proposed model generated based on probabilities shown in (a).

evaluate the model with noisier data, we generated a variant of the data with additional noisy peaks in the tails of isotopic distributions. This variant of the simulated data reflected the issues with identifying relevant peaks in biological case studies due to measurement noise and overlaps between isotopic distributions, or presence of post-translation modifications such as oxidation. For data set 2, we replace last three intensities of isotopic peaks with intensities from the interval $[5, 10]$, with the total intensity equal to 100. Figure 4.12 presents an example of noisy and *clean* (no additional peaks) spectra for data set 2.

4.4.2 Evaluation strategy

HVEM case study The HVEM case study corresponds to the the case of smaller clusters of overlapped peptides with longer average segment. Table 4.3 summarizes quantitative information about these clusters. Segments in non-trivial clusters consisted of 2-5 exchangeable hydrogens on average. As no ground truth was available for this case study, we evaluated the ability of the proposed approach to recover observed isotopic patterns and display fitted segment-level probabilities.

	Cluster ID	no. peptides	no. segments	mean no. ex.
1	1	1	1	14.00
2	2	3	3	4.29
3	3	1	1	4.00
4	4	20	14	2.88
5	5	12	9	2.04
6	6	2	2	5.67
7	7	1	1	4.00
8	8	1	1	7.00

Table 4.3: **HVEM case study**: clusters of overlapped peptides. There were 4 non-trivial clusters of peptides in this study, with two small clusters (2-3 peptides) and two larger clusters (12-20 peptides). The last column indicates the average number of exchangeable hydrogens per segment in a given cluster.

msHDX case study msHDX case study (Kish et al., 2023b) exemplifies a different structure of observed overlapped peptides. In this case, all peptides overlap and the typical segment consists of a

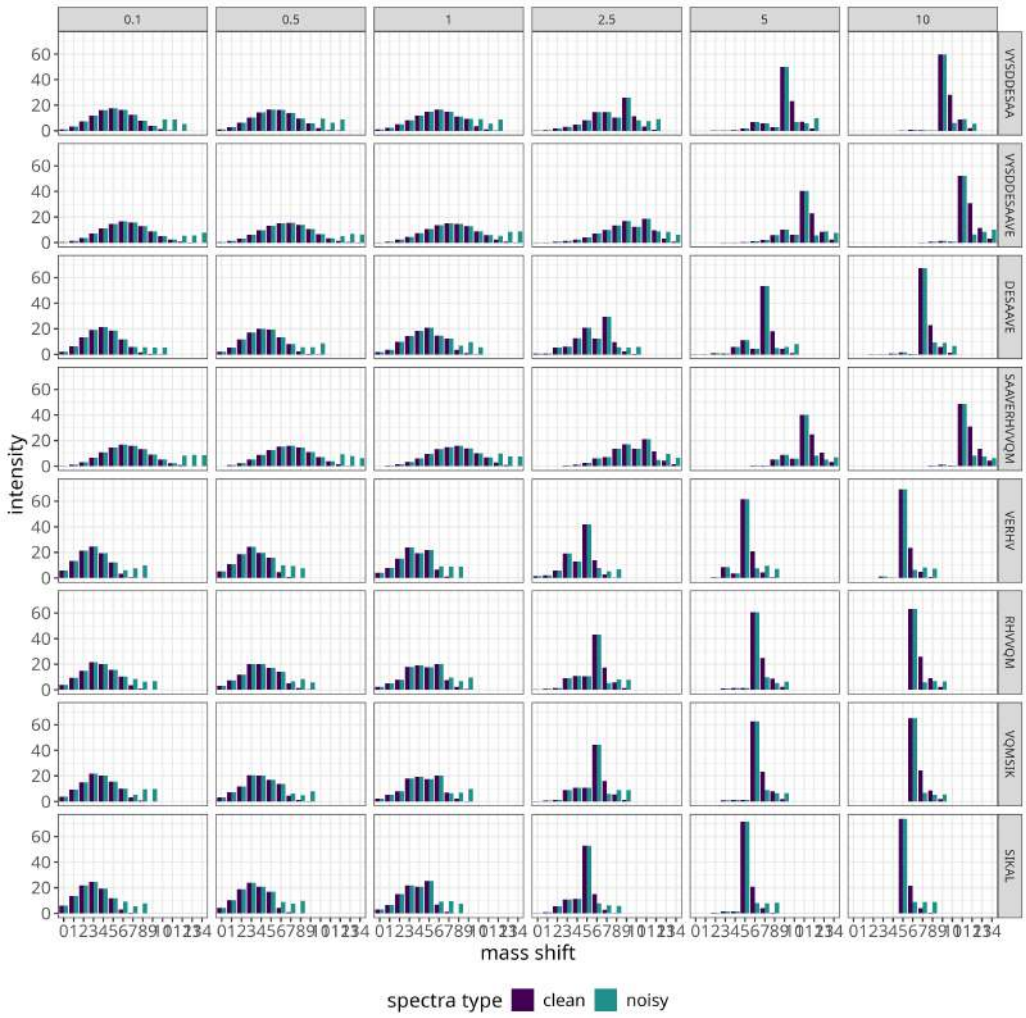


Figure 4.12: **Simulated data set 2: a comparison of *clean* spectra from the regular noise model 4.19 and *noisy* spectra where last three peaks in each spectrum were replaced with random values to mimick the issues with peak selection. Colors indicate the types of spectra.**

single AA, indicating very high possible resolution of deuterium exchange rates. Table 4.4 summarizes this quantitative information. Again, this data set did not include ground truth information, so we compared the observed peak intensities to intensities predicted by the proposed approach and plotted fitted segment-level probabilities.

	Cluster ID	no. peptides	no. segments	mean no. ex.
1	1	969	659	1.19

Table 4.4: **msHDX case study**: a single cluster of overlapped peptides with an average number of exchangeable hydrogens close to 1, as indicated in the last column.

4.4.3 Evaluation metrics

In bottom-up studies, only peptide-level spectral data are observed. From such data, most approaches extract average deuteration values per peptide. Hence, quality of a model fit can only be assessed by comparing either the raw peak intensities or calculated peptide-level deuteration values to their fitted counterparts. As our approach uses full observed isotopic distributions, we will evaluate its ability to correctly recover the observed isotopic patterns. As segment-level information is of interest, we

will focus our analysis of the simulation studies of the accuracy of estimation of H/D probabilities for segments.

1. At the peptide-level, we compared observed and predicted peak intentisities. Consider a single peptide $i, i = 1, \ldots, I$ and time point $T_k, k = 1, \ldots, K$. We consider both absolute and relative mean-squared errors in m-th repetition of the simulation study, $m = 1, \ldots, M$: $AMSE_{i,k,m} = \sum_{j=0}^{N_i^0+N_i^{ex}} (E_{i,k,j} - O_{i,k,j})^2$ and $RMSE_{i,k,m} = \sum_{j=0}^{N_i^0+N_i^{ex}} \frac{(E_{i,k,j}-O_{i,k,j})^2}{E_{i,k,j}}$. These errors measures can be aggregated over time and peptides.

2. At the segment-level, we compared observed and predicted exchange probabilities. Consider a single segment $j, j = 1, \ldots, J$, time point $T_k, k = 1, \ldots, K$, and number of exchanged hydrogens $n, n = 0, \ldots, n_j^{ex}$. We consider both absolute and relative mean-squared errors in m-th repetition of the simulation study, $m = 1, \ldots, M$: $AMSE_{j,k,n,m} = \sum_{n=0}^{n_j^{ex}} (\delta_{j,k,n}^S - \hat{\delta}_{j,k,n}^S)^2$ and $RMSE_{j,k,n,m} = \sum_{j=0}^{n_j^{ex}} \frac{(\delta_{j,k,n}^S - \hat{\delta}_{j,k,n}^S)^2}{\delta_{j,k,n}^S}$, where $\delta_{j,k,n}^S$ and $\hat{\delta}_{j,k,n}^S$ denote simulated and fitted probabilities, respectively. Again, such error measures can be aggregated over segments and time points.

4.4.4 Model fitting details

In this section, we briefly refer important details of model fitting for both simulated data and case studies.

4.4.4.1 Starting point importance

As the loss functions are not convex, optimization routines with different starting points may lead to different local minima or fail to converge. For example, for data set 1, optimization via searching for zeros of the analytical gradient converged 13 and 9 times out of 50, with standard deviations of the error term equal to 0.01 and 0.05, respectively. Thus, in practice, the procedure should be used while trying many starting points. Then, the set of β parameters that leads to the smallest value of the chosen loss function across all starting points should be selected. Hence, for case studies where no ground truth was available, we used various starting points such that each element of the β vector was sampled uniformly from the interval $[-0.9, -0.001]$. For the relevant sets of time points, such interval covers probabilities both close to 0, and close to 1. In simulation studies, we evaluated the proposed model under the assumption that starting point is close the solution. We initialized optimization algorithms for noisy simulated data with true values of underlying β parameters. Hence, the conclusions are applicable to results of model fitting when a suitable starting point was found. This means that for practical application, using a large number of starting points is recommended.

4.4.4.2 Optimization routines

As indicated previously, both optimization problems involving ℓ_{OLS} and $\tilde{\ell}_{GLS}$ loss functions can be solved using standard optimization tools. We fitted the proposed model using the Broyden method - a quasi-Newton approach (Reichel and Gragg, 1990) as implemented in the R package `nleqslv` (Hasselman, 2022). This approach used the analytical gradient of relevant loss functions as computed in Section 4.3.2.3. For the PL-GLS fitting algorithm, the maximum number of iterations was set to 50, and the tolerance to 0.01.

4.5 Results

4.5.1 Fitting the segment-level model

Simulated data: small cluster case Table 4.5 summarizes the results of a simulation study for data sets 1 and 2 with heteroscedastic errors (Equation 4.19). The maximum number of iterations was set 100 for the simpler cluster, and 150 for the slightly more complex cluster. With the standard deviation of random noise equal to 0.01, all fitting procedures reached convergence. At standard deviation equal to 0.05, the routine did not converge in a two examples for both data sets. Complexity of the loss functions resulted in Hessian matrices characterized by condition numbers typically larger than $1,000$. Hence, the problem is sensitive to noise and numerical accuracy issues, and discerning local minima from saddle points was challenging.

At the lowest level of data and cluster complexity, the optimization routine required $7-21$ iterations, with an average of 15 and 30 for σ equal to 0.01 and 0.05, respectively. Here, all numerical Hessians computed at solutions indicated minima of the loss function. With a slightly more complex cluster and a higher number of parameters, optimization routine required $10-99$ steps, with and average of around 14 and 42 steps for σ_{het} equal to 0.01 and 0.05, respectively.

Data set	σ	no. converged	ave. no. steps	min. no. steps	max. no. steps
1	0.01	50	10.84	7	15
2	0.01	50	13.80	10	21
1	0.05	48	25.83	14	50
2	0.05	48	41.85	18	99

Table 4.5: **Simulated data (small cluster case): Newton-type optimization approach enabled fitting the proposed approach**. Table summarizes 50 repetitions of the simulation study for two clusters, one describe by 20 parameters, and one described by 30 parameters. Columns *ave. no. steps*, *min. no. steps*, and *max. no. steps* refers to average, minimum, and maximum number of iterations of the optimization routines required to reach convergence.

PL-GLS estimation required 3-9 iterations to converge on average. The fitting algorithm converged in all examples with $\sigma_{het} = 0.01$. For $\sigma_{het} = 0.05$ it converged to a stationary point in 45 and 48 repetitions of the simulation study for data sets 1 and 2, respectively.

4.5.2 Recovery of peptide-level isotopic patterns

Simulated data: large cluster case Figure 4.13 shows simulated noisy spectra and predicted peaks based on estimated segment-level parameters in a single repetition of the simulated study. The proposed approach correctly recovered peptide-level isotopic patterns.

HVEM case study Let us consider a cluster of 11 peptides and 8 segments from the HVEM case study. The best fit of the proposed model based on OLS across 20 starting points sampled from the interval $[-0.9, -0.001]$ for each parameter resulted in the predicted peak intensities displayed in Figure 4.14. Predicted spectra exhibited high similarity to the observed isotopic patterns despite varying rates of exchange for different peptides, evident in differences in centers of the isotopic envelopes.

msHDX case study Figure 4.15 presents a comparison between estimated and observed peaks for a cluster of four peptides that consisted of 5 segments found in the msHDX case study. A smaller number of segments compared to the full data set, and larger lengths of segments (2-7 AAs) are due to quality restrictions and manual curations of spectral data. Despite the presence of both missing peaks, and potentially noisy peaks in the tails of isotopic distributions, the proposed approach recovered the bulk of isotopic patterns.

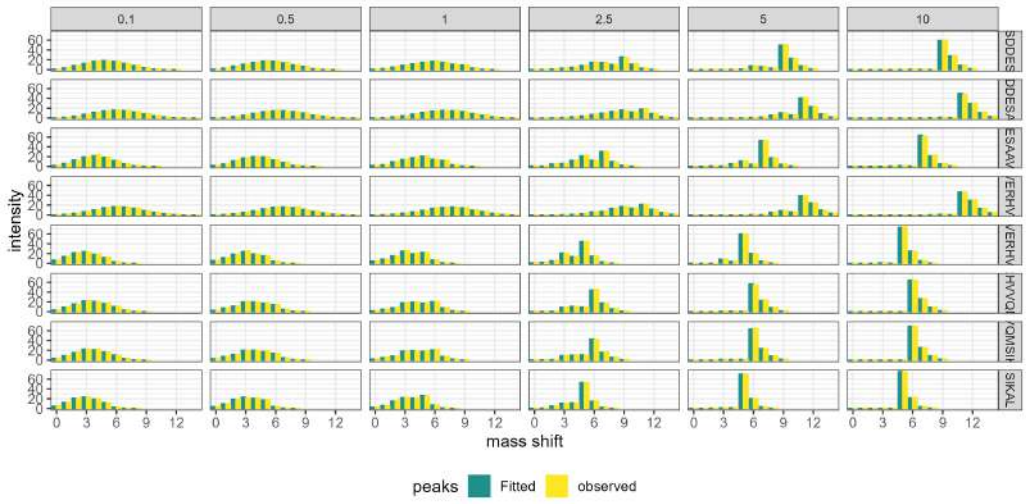


Figure 4.13: **Simulated data: proposed approach recovered peptide-level isotopic patterns using segment-level exchange probabilities**. Colors differentiate simulated and fitted peaks for a single repetition of the experiment with a small cluster of 8 peptides described by 30 parameters.

4.5.3 Estimation of segment-level probabilities

Simulated data: small cluster case Figure 4.16 summarizes 50 repetitions of the simulation study based on data set 1 with boxplots of estimated probabilities for each segment and numbers of exchanged hydrogens indicated by panels over time indicated by the x-axis. While starting at the true parameters, the proposed approach recovered true segment-level probabilities from noisy peptide-level data with high accuracy. Boxplots indicate that, on average, probabilities were unbiased, with larger variance when σ_{het} increased.

Figure 4.17 summarizes analogous results for Data set 2. The increase in model complexity did not affect model's performance. Again, the estimated probabilities were unbiased with variability that increased with σ_{het}. Overall, in terms of both convergence and precision of estimates, the influence of σ_{het} under the heteroscedastic model was stronger than the effect of growing model complexity.

Figure 4.18 summarizes simulation results for a cluster of 19 peptides characterized by 40 parameters. In this simulated scenario, each segment was length 1. OLS estimation resulted in unbiased estimates characterized by low variance across repetitions.

HVEM case study Figure 4.19 presents segment-level probabilities for a cluster of peptides from the HVEM case study. These fitted probabilities generated predicted spectra presented in Figure 4.14. The rate of exchange varied by segment. According to the model, some segments quickly reached full deuteration, while three segments did not reach it at all at time points used to fit the model.

4.5.4 Improved estimation precision in the presence of technical replicates

We evaluated the precision of estimation of both noise term variance and variances of model parameters in two simulation studies. Firstly, we considered the homoscedastic noise structure with $\sigma_{hom} \in \{0.5, 1\}$, and one and two technical replicates. Then, we evaluated the estimator of variance-covariance matrix given by Equation 4.17 for PL-GLS model in a heteroscedastic case based on data sets 1 and 2.

Simulated data: small cluster case Figure 4.20 displays estimated probabilities of $\sigma_{hom} = 0.5$. Boxplots indicate reduced variability of estimates with an added technical replicate. Table 4.6 summarizes these results across noise levels and numbers of simulated technical replicates. Again reduced variability of estimates may be observed, with virtually no bias.

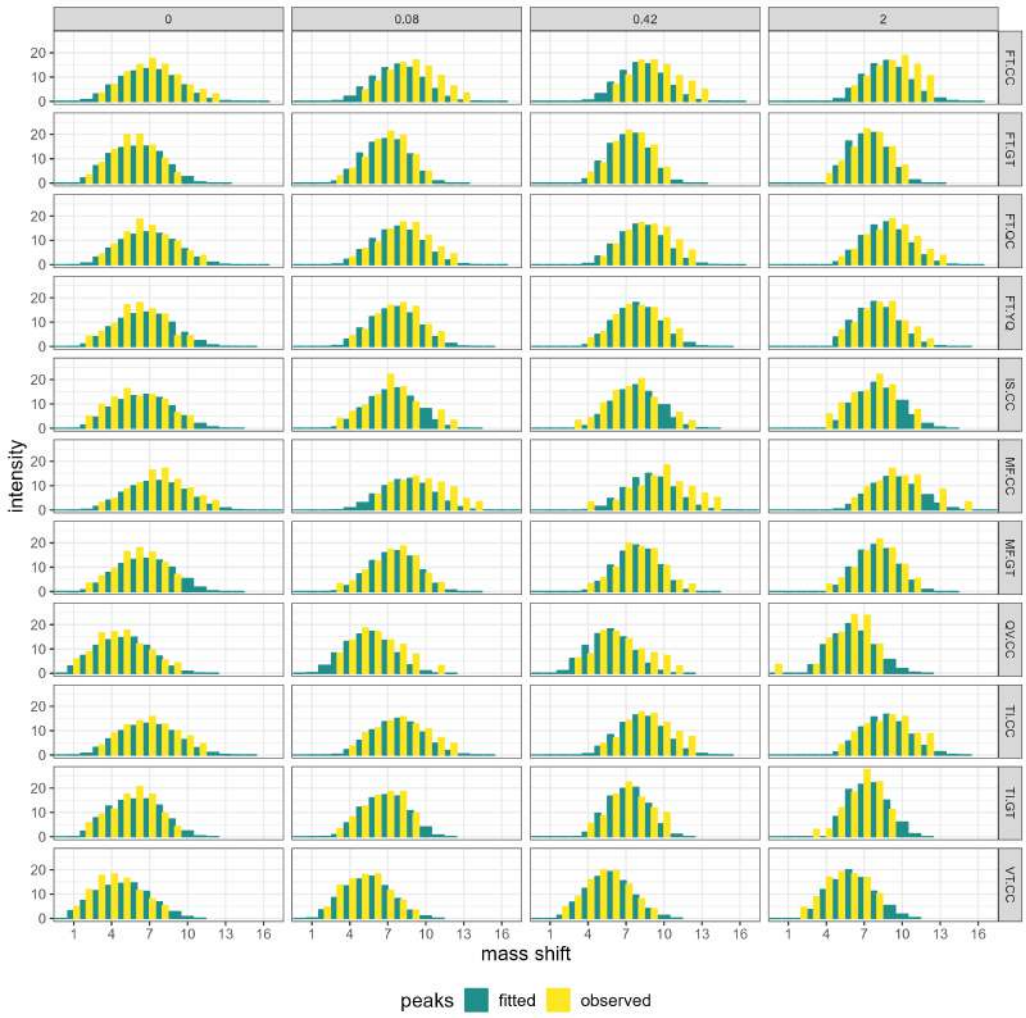


Figure 4.14: **HVEM case study: proposed approach enabled recovering observed peak intensities based on estimated segment-level exchange probabilities.** Rows indicate experimental time points while columns indicate peptides. Full sequences of peptide were abbreviated using first two and last two amino acids for readability.

Moreover, additional independent replicates positively affected convergence of the proposed approach. Table 4.7 summarizes 50 repetitions of the simulations study for different values of $\sigma_{hom} \in \{0.5, 1\}$ for one or two technical replicates. For both values of σ_{hom}, addition of technical replicates increased the number of converged fits, both in terms of finding the zero of a gradient of the loss function, and it being a local minimum rather than a saddle point. Moreover, the average number of iterations of the optimization routine (Broyden method based on analytical gradient) required for convergence was also reduced.

Moreover, let us consider the task of estimating σ and the variances of estimators of exchange probabilities. The standard estimator of σ in nonlinear least squares is given by $\frac{\sum_{i=1}^{n} r_i^2}{n-p}$ where r_i denotes a residual for i-th observation, n denotes the number of observations, and p denotes the number of model parameters. An estimator of σ can be calculated by taking a square root of this expression. Let us consider the model 4.18, for which this estimator was derived. In this case, we found that it its average values across repetitions of the simulations study were close to theoretical values used in simulations (see Table 4.8).

However, estimating the variances of model parameters by using the asymptotic formula for a variance-covariance matrix $s(\widehat{\boldsymbol{\beta}})H^{-1}$ and, consequently, the variances of estimators of exchange probabilities, remained challenging. Figure 4.21 summarizes results of the simulation study for a small

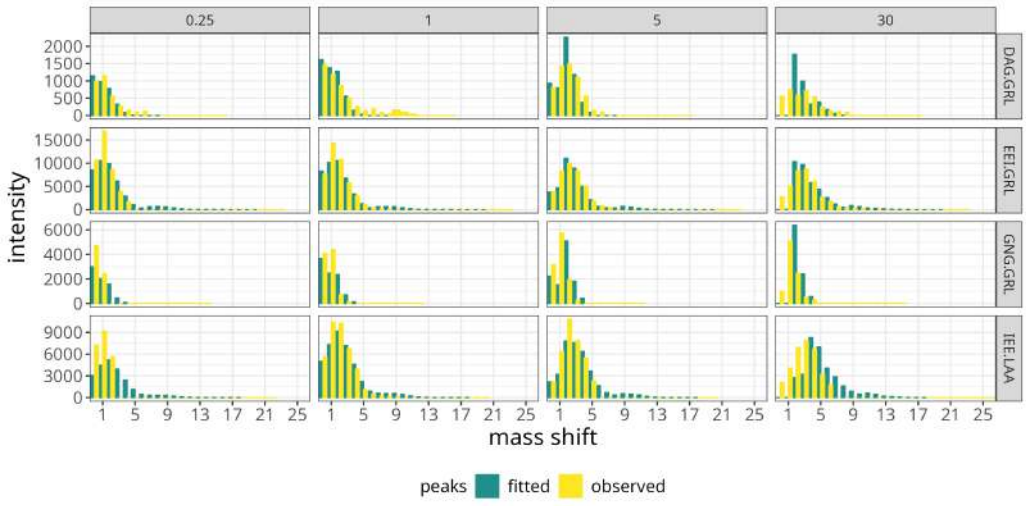


Figure 4.15: **msHDX case study: proposed approach enabled recovering observed peak intensities based on estimated segment-level exchange probabilities.**

σ	no. tech. rep.	ave. bias	ave. var.
0.5	1	$< 10^{-17}$	0.00009
0.5	2	$< 10^{-17}$	0.00005
1.0	1	$< 10^{-17}$	0.00048
1.0	2	$< 10^{-17}$	0.00038

Table 4.6: **Simulated data: added technical replicates reduced the variance of parameter estimates**. Parameter σ denotes standard deviation of Gaussian noise in a homoscedastic error model, *no. tech. rep.* indicates the number of simulated technical replicates (independent copies of a spectrum for each peptide ion and charge), *ave. bias* summarizes bias of exchange probabilities estimation across all segments, time and numbers of exchanged hydrogens, while *ave. var.* summarizes the variance of these estimates.

cluster and homoscedastic errors (Equation 4.18). The model-based variance underestimated the empirical variance of the estimated parameters.

Hence, even with a proper variance structure specification, the standard asymptotic estimator did not perform well in the simulation study. Let us know evaluate the variance-covariance matrix estimator for the PL-GLS model with the same samples sizes, but heteroscedastic variances. Figure 4.22 compares the average model-based variance of β parameters, calculated by using the variance-covariance matrix defined by Equation 4.17 to empirical variances calculated over 50 repetitions of the simulation study for data set 1. Each point corresponds to a single parameter. Typically, the estimated variances were similar to the empirical ones, as indicated by proximity of the points to the $y = x$ line.

Figure 4.23 present analogous results for data set 2. With a larger number of model parameters, the deviations from empirical variances were larger than in the simpler cluster, but the overall trends still indicates good estimation of variances of model parameters.

4.5.5 Influence of peak-picking quality on estimation precision

As indicated in Section 4.3.3, processing spectral HDX-MS data is challenging due to the influence of unknown exchange probabilities on isotopic patterns. Hence, distinguishing overlapping isotopic patterns of various peptides including variants originating from post-translational modifications or noisy peaks from true isotopic patterns is challenging. For example, a group of high-intensity peaks

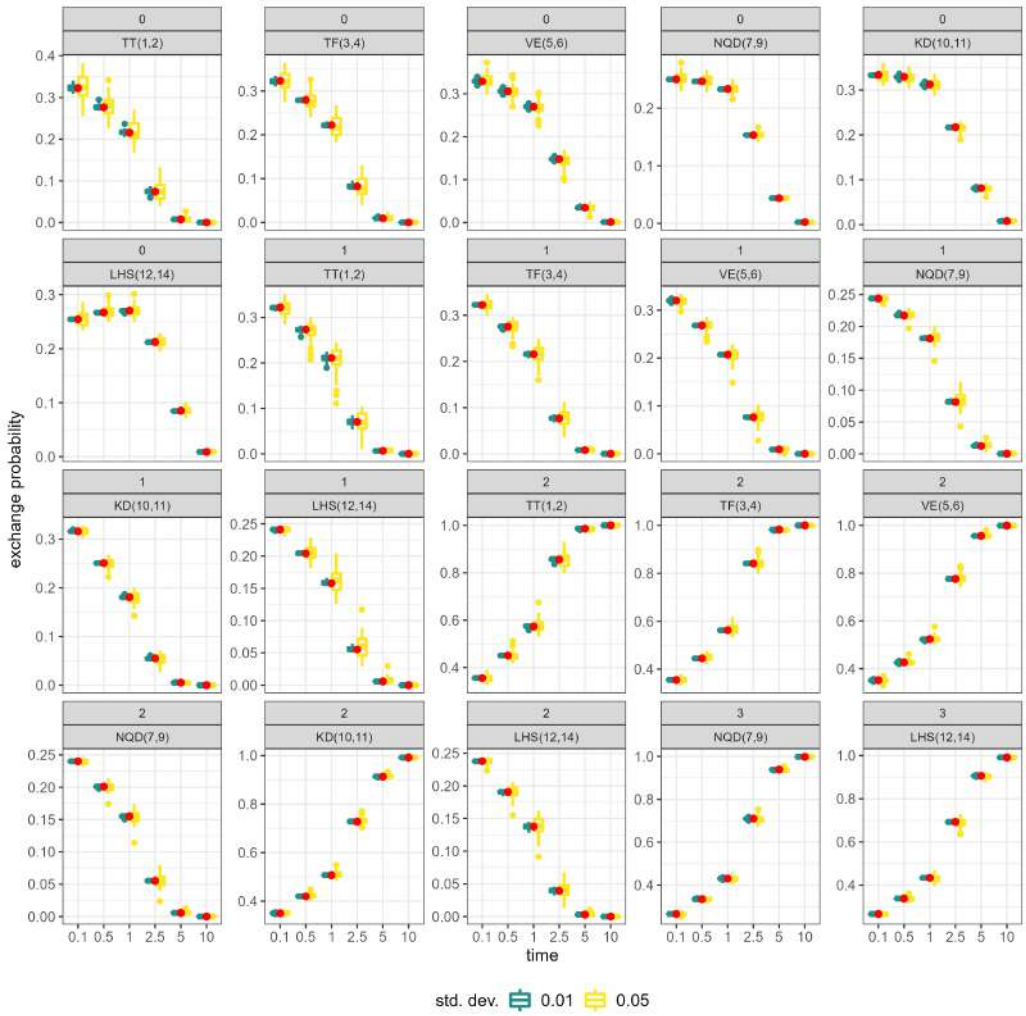


Figure 4.16: **Simulated data: proposed approach estimated segment-level probabilities from peptide-level data.** Red dots indicate true values of probabilities. Boxplots summarize 50 repetitions of the simulation study with two sizes of random σ_{het}.

in the tail of a given isotopic distribution may indicate a quickly-exchanging segment of a peptide, or simply high variance of random noise. Hence, we evaluated the ability of the proposed approach to estimate exchange probabilities in the presence of spurious isotopic peaks. Such peaks may occur on both left and right tails of isotopic distributions, and all peaks in a spectrum are noisy to an unknown degree. To fix attention, we focused on spurious peaks in the right tail of a spectrum, which are generated by higher numbers of exchanged hydrogens. As the maximum number of isotopic peaks that may be observed for a given peptide is known, we refer to as peaks characterized by extraordinarily large variance as spurious.

Figure 4.24 compares the results of model fitting for data set 2 with $\sigma_{het} = 0.01$ in two variants: one without the spurious peaks (*clean*), and one with noisy peaks in the tail of each isotopic distribution (*noisy*). As previously shown in Section 4.5.3, clean data enabled the proposed model to correctly recover H/D exchange probabilities for each segment. Additional errors in the tail peaks resulted in biased estimates. Both underestimation and overestimation of exchange probabilities was observed, depending on a segment and number of exchanged hydrogens. Moreover, spurious peaks led to issues with convergence. With spurious, only 34 out of 50 examples converged before reaching the maximum number of iterations (150). On average, it took almost 86 iterations, compared to 12 iterations for the *clean* data set. Moreover, finding the right starting point was more difficult, and we used the BFGS-based optimization with numerical gradient to explore the parameter space before applying the

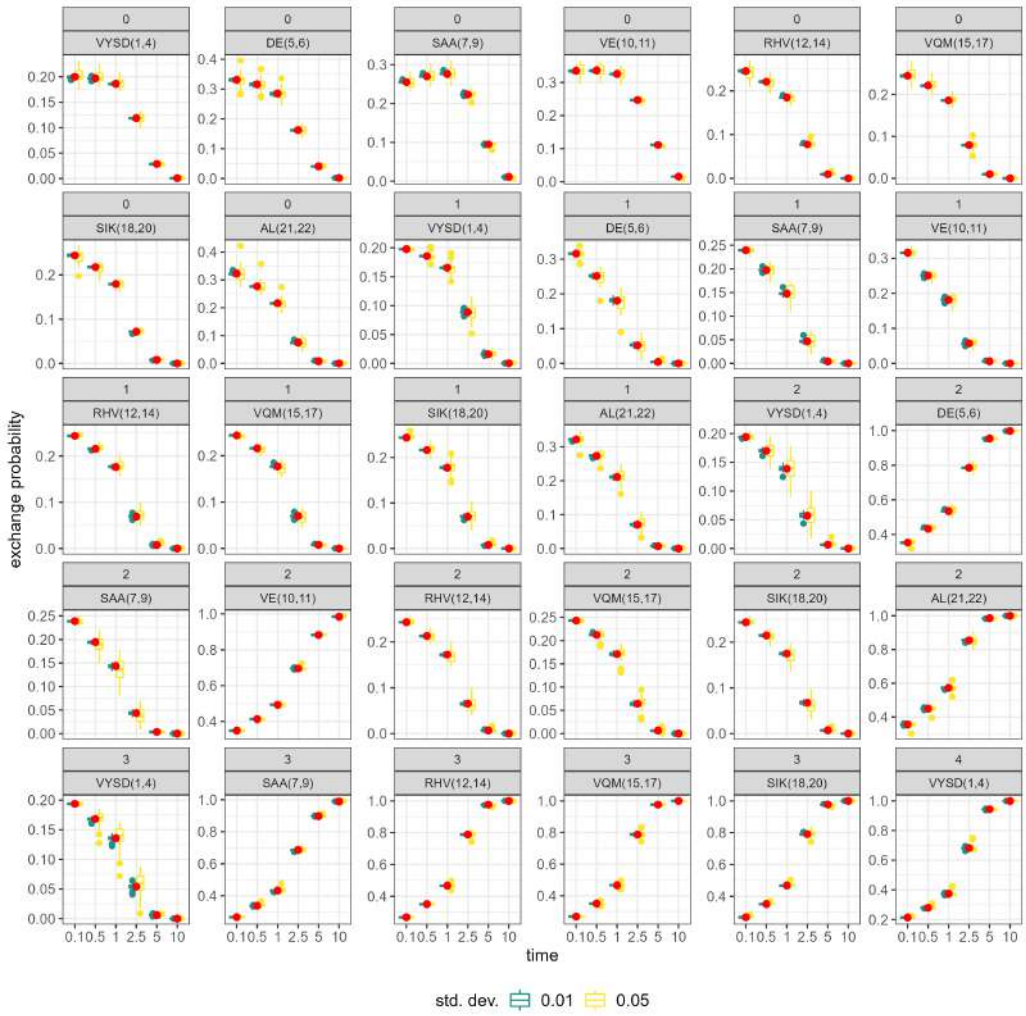


Figure 4.17: **Simulated data: proposed approach correctly estimated segment-level probabilities from peptide-level data.** Red dots indicate true values of probabilities. Boxplots summarize 50 repetitions of the simulation study with two sizes of random σ_{het}.

Newton-type optimization using the analytical gradient. Average relative error of estimating exchange probabilities was equal to 8% for the noisy spectra, compared to 0.02% for clean spectra. Hence, the quality of spectra is crucial to the analysis of HDX-MS data and high resolution data analysis, as it significantly affects estimated segment-level probabilities.

4.6 Discussion

We proposed and implemented a statistical model to infer implicit segment-level H/D exchange probabilities from observed peptide-level isotopic patterns. Based on simulation studies and applications to biological investigations, we verified its ability to provide unbiased estimates of exchange probabilities. Moreover, we proposed two estimation methods that are capable of adjusting to different noise structures. We offered an open-source implementation of the approach in the form of an R package.

A single residue resolution is not achievable in every study. Hence, working with data-driven segments is a reasonable alternative to existing approaches that parametrize each residue, such as Babić, Kazazić, and D. M. Smith, 2019 or Stofella, Skinner, et al., 2022. Moreover, not all existing approaches provide a free and accessible implementation or a method for obtaining parameter uncertainty. However, a widespread adoption of this approach requires both developments on the modelling side and the data processing side.

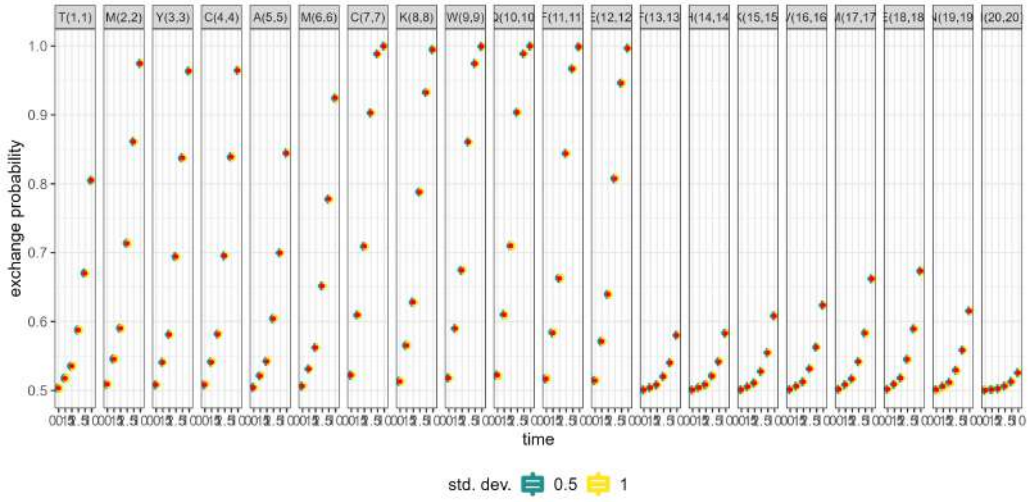


Figure 4.18: **Simulated data: proposed approach correctly estimated segment-level probabilities from peptide-level data.** Red dots indicate true values of probabilities. Boxplots summarize 50 repetitions of the simulation study with two sizes of σ_{hom} for a cluster with residue-level resolution.

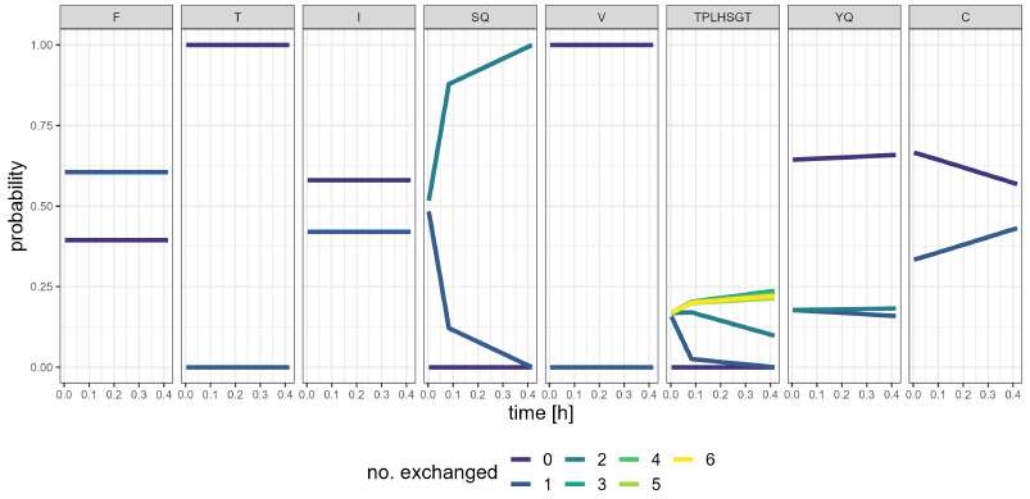


Figure 4.19: **HVEM case study: proposed approach estimated segment-level probabilities from peptide-level data.** Colors indicate numbers of exchanged hydrogens, while each panel presents estimated dynamics of H/D exchange for a different segment over time.

Let us begin with modeling concerns. Firstly, the proposed model made a simplifying assumption that all segments exchange hydrogens independently. This assumption enabled a relatively straightforward calculation of exchange probabilities at the peptide level via a convolution of probability distributions based on a simple generating functions approach. This way, the task of finding aggregated peptide-level exchange probabilities is reduced to simple polynomial multiplication. However, as the primary goal of studying H/D exchange dynamics is to identify regions in the 3D protein structure that are protected from exchange due to folding or other factors, it is reasonable to expect that some sets of consecutive segments will exhibit similar behaviour. Thus, it would be beneficial to consider a model that does not assume independence and instead treats contiguous segments as possibly related, either due to similar exchange patterns or vice versa. This can be achieved in different ways. Previously, constraints on differences between exchange rates of closely related residues were used Stofella, Skinner, et al., 2022. Alternatively, it is possible to search for clusters of similarly behaving segments using a k-means-type approach or to specify a more flexible model for aggregating segment-level exchanges compared to a simple convolution, allowing for a degree of correlation.

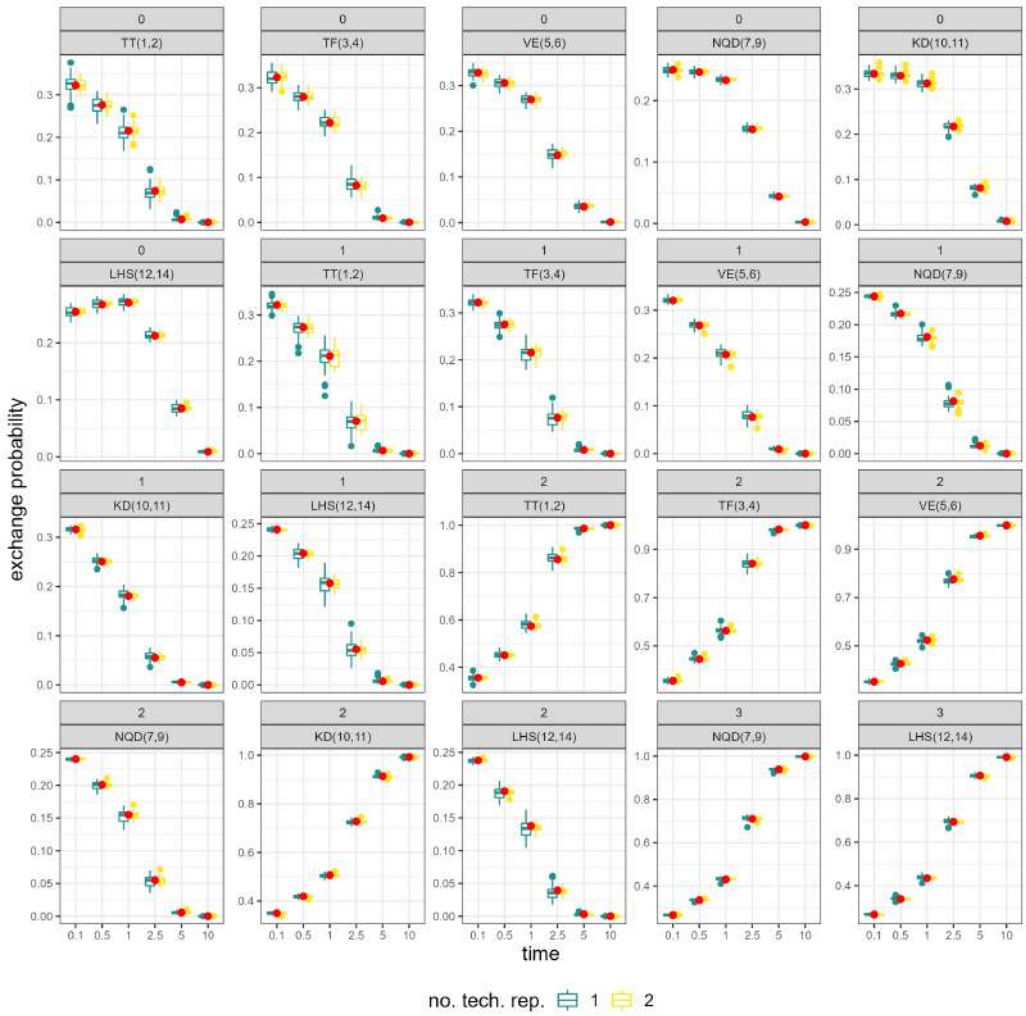


Figure 4.20: **Simulated data: inclusion of technical replicates reduced the variance of estimates.** Boxplots summarize 50 repetitions of the simulation study for Data set 1. Colors indicate the numbers of technical replicates, while each panel presents estimated probabilities as a function of experimental time points for a given number of exchanged hydrogens and a segment.

Such modifications to the proposed approach would be a valuable direction for future research.

Secondly, an empirical study into noise patterns in HDX-MS spectra would enable better modeling of noise structure in isotopic peaks. We proposed optimization methods for homoscedastic errors and mean-dependent errors; however, alternative noise structures may be helpful in modeling isotopic peaks. Understanding and properly modeling random errors in the intensities of isotopic peaks would benefit spectral data preprocessing, quality control, and downstream statistical modeling.

Computational complexity is the third major challenge for isotopic distribution-based modeling of HDX-MS data and high-resolution estimation of exchange rates. Modeling exchange rates across a long sequence of residues requires roughly as many parameters as there are exchangeable hydrogens, with some additional intercept parameters. In the example **msHDX** case study, a single cluster consisted of almost a thousand residues. Hence, the number of parameters required for complete analysis of such data sets is very large, and both analytical and numerical optimization of relevant models, such as the proposed approach, become challenging. Let us recall that the computation of the analytical gradient for the proposed model involved iterating over all segments and computing exchange probabilities of all remaining segments. Even with a dynamic programming-type strategy to reduce the number of repeated / redundant computations, such calculations for a very large model would require a substantial number of iterations just to compute the gradient for a single vector of parameters.

σ	no. tech. reps.	no. converged	no. local min.	ave. steps	total
0.5	1	48	48	26.58	49
1.0	1	40	36	40.25	47
0.5	2	50	49	21.20	50
1.0	2	45	40	34.60	48

Table 4.7: **Simulated data: additional technical replicates improved convergence of the proposed approach**. Table summarizes 50 repetitions of a simulation study with homoscedastic Gaussian errors with standard deviation indicated by the σ column. Columns *no. tech. reps*, *no. converged*, *no. local min.*, *ave. steps*, *total* describe the number of simulated technical replicates, number of converged results (norm of the gradient sufficiently close to 0), number of proper local minima found (positive-definite Hessian at the solution), average number of iterations until convergence, and the number of fits that stopped before reaching the maximum number of iteratios, respectively.

σ	no. tech. rep.	mean $\hat{\sigma}$
0.5	1	0.4828
0.5	2	0.4829
1.0	1	0.9461
1.0	2	0.9616

Table 4.8: **Simulated data: standard deviation of random noise was well approximated by a standard estimator**. Column σ denotes the standard deviation of homoscedastic random noise used in the simulation study, *no. tech. rep.* indicates number of simulated technical replicates, while *mean* $\hat{\sigma}$ presents estimators of standard deviation of the noise averaged over 50 repetitions of the simulation study.

Numerical approximations to the gradient would face similar problems with repeated evaluations of a complex function that connects model parameters to expected isotopic peaks. Hence, the development of efficient optimization routines tailored to isotopic distribution-based HDX-MS data is a crucial research direction for future research.

Additionally, the proposed parametrization of segment-level exchange probabilities ensures that as time increases, the probability of no exchange approaches 0, and the probability of complete exchange approaches 1. However, no monotonicity constraints were added to the model, as they would further complicate optimization. Hence, estimated probabilities may differ from prior expectations that the probability of no exchange never increases. While this non-monotonicity could be exploited to model back-exchange, an explicit model for this phenomenon would be a better solution. Moreover, the current parametrization does not properly account for time 0 (undeuterated state). The optimal parametrization would result in the probability of no exchange at $t = 0$ being equal to 1. While this is not required, as the isotopic distribution of an undeuterated peptide (time 0) is explicitly used to generate peptide-level exchange probabilities, and only exchange probabilities at subsequent time points enter the model. However, a parametrization that follows both the monotonicity constraints and leads to a distribution with certain non-exchange at time 0 would better reflect the physicochemical description of HDX-MS experiments. Hence, alternative approaches to modelling exchange probabilities, such as hidden Markov model-based methods, or estimating smooth curves that describe exchange probabilities over time with proper constraints, or alternative per-time-point parametrizations, could improve the proposed approach.

Most importantly, statistical methods for modeling the complete dynamics of H/D exchange based on MS data require high-quality spectral data as a base for statistical inference. Many existing methods utilize aggregated exchange data derived from isotopic envelopes, thereby abstracting from the quality of spectral data through aggregation and replication. However, such aggregated measures do

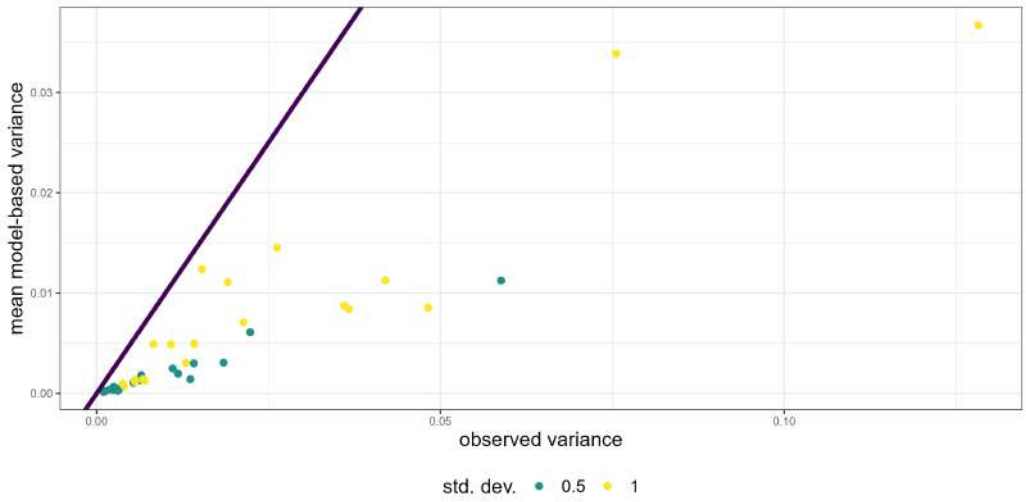


Figure 4.21: **Simulated data: asymptotic covariances matrix underestimated the variability of model parameters.** Each point corresponds to a model parameter in the small cluster case with homoscedastic errors. Colors indicate standard deviations of the noise term. X-axis corresponds to an empirical variance of parameter estimates across 50 repetitions of the simulation study, while y-axis presents average (over 50 repetitions) variance inferred from the variance-covariance matrix given by Equation 4.17.

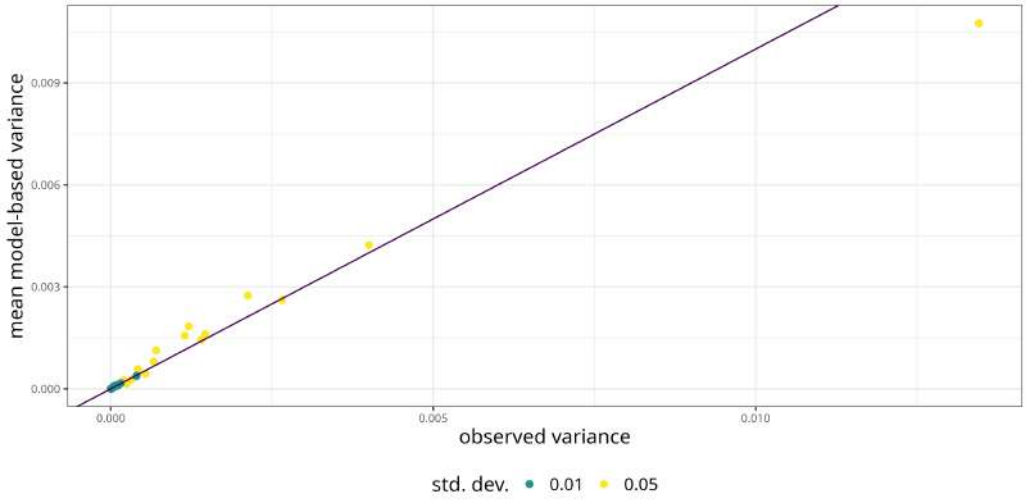


Figure 4.22: **Simulated data: covariance matrix derived from the PL-GLS approach enabled precise estimation of variances of model parameters** β. Each point corresponds to a model parameter in the small cluster case (Data set 1) with heteroscedastic errors. Colors indicate standard deviations of the noise term. X-axis corresponds to an empirical variance of parameter estimates across 50 repetitions of the simulation study, while y-axis presents average (over 50 repetitions) variance inferred from the variance-covariance matrix given by Equation 4.17.

not fully capture the kinetics of the exchange, as they are unable to differentiate between EX1 and EX2 regimes. Multimodal isotopic patterns that are lost during the aggregation step can provide valuable insight into the protein structure. Hence, more data analysis methods should utilize complete isotopic distributions. However, there are two significant challenges: data availability for method development and quality control of spectral data. Firstly, HDX-MS data are more challenging to obtain compared to quantitative proteomics MS data, which are frequently shared via platforms such as ProteomeXchange (Vizcaíno et al., 2014). Secondly, the availability of controlled mixtures or externally annotated data sets would simplify the evaluation and benchmarking of various modeling approaches.

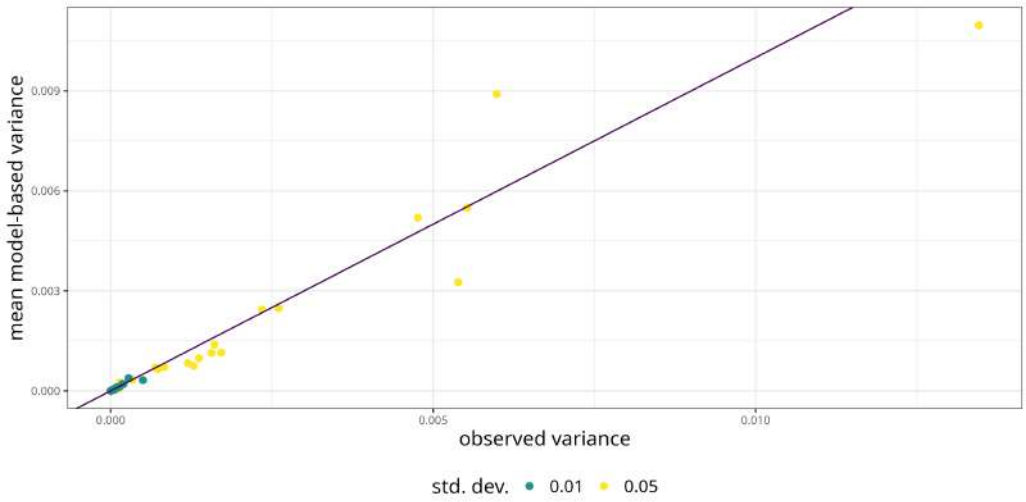


Figure 4.23: **Simulated data: covariance matrix derived from the PL-GLS approach enabled precise estimation of variances of model parameters** β. Each point corresponds to a model parameter in the small cluster (Data set 2) case with heteroscedastic errors. Colors indicate variances of the noise term. X-axis corresponds to an empirical variance of parameter estimates across 50 repetitions of the simulation study, while y-axis presents average (over 50 repetitions) variance inferred from the asymptotic variance-covariance matrix

Additionally, data preprocessing that uses closed-source software or produces aggregated data that are not shared with raw spectra impedes reproducible research and fair comparisons of downstream statistical analysis methods. Hence, improvements are needed in data sharing infrastructure and tools for benchmarking HDX-MS data analysis methods and software.

Moreover, as previously indicated, the quality control of isotopic peaks is crucial for the quality of input data for statistical analysis, and consequently, for the precision of estimating exchange rates. Both the proper selection of peaks belonging to a given peptide and the modeling of noise in those peaks are important. Overlapping isotopic envelopes that include unidentified post-translational modifications or other peptides may skew the results of peptide quantification in a given spectrum. Novel quality control methods tailored to isotopic distribution-based analyses are needed to ensure that spectra from which exchange probabilities are derived are accurately quantified and reflect the H/D exchange, rather than other artifacts of the data. HDX-MS experiments typically involve replication, allowing both the time-varying structure of the data and replication to be used to assess the quality of spectra and differentiate noise from signal. Hence, quality control is an important direction for future research.

Let us note that in addition to spurious or noisy peaks, spectra may be characterized by a lack of identified peaks at certain mass shifts, either at the tail of the distribution (which may be related to lack of full deuteration or, conversely, very fast exchange), or in the center. Both cases, but particularly the latter, require proper treatment in data preprocessing and statistical analysis. A study on the influence of missing peaks on exchange rate estimation would be a significant contribution to finding the appropriate methods for modeling HDX-MS data.

Finally, the proposed model can be used for differential analysis of the dynamics of H/D exchange, rather than just estimating exchange probabilities. When measurements are made for samples under various biological conditions, comparisons can be made in terms of H/D exchange rates in selected regions of the amino acid sequences of proteins, depending on those conditions. The proposed model can be extended to this problem by jointly modeling exchange probabilities from different conditions and constructing tests (equivalently, confidence intervals) for differences in probabilities. However, due to the complex nature of the function that transforms segment-level probabilities into expected peptide-level spectra, a deeper understanding of the likelihood function is required for proper differen-

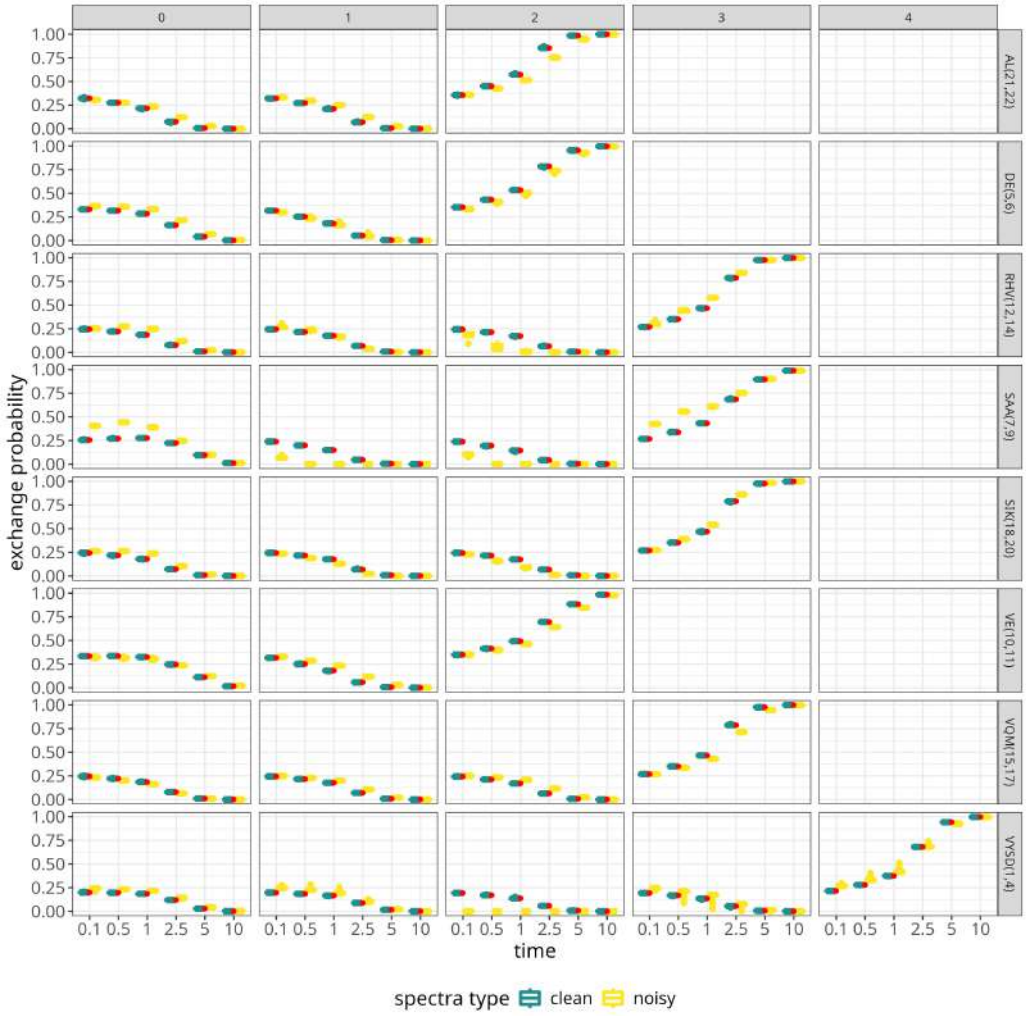


Figure 4.24: **Simulated data: proposed approach correctly estimated segment-level probabilities from peptide-level data, but the precision depended on peak selection quality**. Boxplots summarize 50 repetitions of a simulation study for data set 2 with $\sigma_{het} = 0.01$. Colors indicate spectra which were used as model input: *clean* denotes spectra without spurious peaks, while *noisy* denotes spectra with spurious peaks in the right tail.

tial analysis, particularly in light of previous discussion of possible improvements in parametrization or constraints on the estimated probabilities.

The proposed model can be readily applied to HDX-MS studies to improve the resolution of estimated exchange rates. The presented results indicate that the model can accurately recover the unknown segment-level behavior of a protein and provide uncertainty estimates. The provided software tool can be used to fit the proposed model and explore the results of HDX-MS experiments.

Chapter 5

Software contributions

In this chapter, we present our work in the field of statistical software design and implementation. First, we describe a redesign of the MSstats family of packages (msstats.org). Then, we describe two software packages written in the open-source language R that implement the proposed statistical methods.

5.1 Contributions to the MSstats family of packages

We begin by summarizing contributions to the established software package for discovering differentially abundant proteins based on MS studies called MSstats, proposed in Kohler, Staniak, Tsai, et al., 2023. We begin with a brief overview of the history of development of the MSstats packages, proposed by the Olga Vitek Lab at Northeastern University (originally at Purdue University, US). We discuss the general challenges in developing statistical software for researchers from other fields of study, and then describe our contributions to the MSstats package, which addresses these challenges.

5.1.1 Introduction: MSstats family of packages

The MSstats family of packages for the analysis of mass spectrometry-based proteomics data began with a release of the MSstats package on the Bioconductor platform W. Huber et al., 2015 in 2013, followed by a publication of its description Choi, C.-Y. Chang, et al., 2014, which unified several previously published approaches to analysis of data generated with different experimental approaches and protocols (C.-Y. Chang et al., 2012; Clough et al., 2012; Surinova et al., 2013), and added tools for working with outputs of various signal processing tools. This constituted MSstats 2.0. The original article garnered 970 citations, according to the publisher's information, while the package is downloaded hundreds of times every month. It is used by both academic researchers in proteomics and the industry. A major update to the package and associated statistical modeling approach was presented in Meena Choi's dissertation in 2016 (Choi, 2016), which introduced the MSstats statistical model as described in Sections 2.4.2.2 and 2.4.2.3 (MSstats version 3.0). Moreover, the MSstats was consistently updated, both in terms of compatibility with new experimental workflows (Huang, Choi, et al., 2020; Kohler, Tsai, et al., 2023), and additional functionalities such as filtering of spectral features based on their quality (Tsai et al., 2020), or quality control tools (Dogu et al., 2018).

The following section is based in large part on paper Kohler, Staniak, Tsai, et al., 2023 and the related tutorial paper Kohler, Staniak, F. Yu, et al., 2024. The former introduced a major update to the MSstats package (MSstats version 4.0) while the latter described the data analysis workflow and its capacity for processing large data. Proposed updates facilitated the creation of a new graphical user interface for MSstats, as described in Kohler, Kaza, et al., 2023, and affected related packages, such as MSstatsTMT (version 2.0) or MSstatsPTM. My contributions to the MSstats family of packages described in this chapter were developed during my work in Olga Vitek's Lab, funded by the

Chan-Zuckerberg Essential Open-Source Software Award granted to Prof. Olga Vitek. The proposed changes did not modify the statistical methods implemented in the relevant packages. Instead, they improved their implementation, extensibility and maintainability by simplifying the user interface, reducing the number of dependencies and overall size of the codebase, creating a modular design of a family of packages that perform particular data analysis tasks, and increasing modularization of code within each package, extracting general functions for re-formatting outputs of various signal processing tools and data processing.

5.1.2 Design of statistical software for proteomics

The characteristics of MS-based proteomics data sets change as the technology advances. Labelling techniques, DIA experiments, and related developments generate increasingly large and complex data sets. In this section, we provide a brief overview of the challenges in statistical software development, related both to these advancements and to the context in which the software is used.

Firstly, software for mass spectrometry data analysis is typically used by practitioners with a biological rather than statistical or programming background. Hence, user interface and documentation are of crucial importance. In R-based applications, there are two major aspects of the interface. At a higher level, interfaces can be divided into graphical user interfaces (GUIs) and command-line interfaces (CLIs). Software written in R naturally gravitates towards a focus on CLIs, but GUIs are more desirable from the user's perspective. Hence, even when designing the CLI, it is essential to consider its potential for GUI usage. At a lower level, each type of interface should be kept as simple as possible, with well-defined tasks for all components, easy-to-understand outputs, and minimal, clear parameters.

Software for statistical analysis of MS data is typically used within a larger data analysis pipeline. Downstream statistical analysis is always preceded by peptide identification and quantification based on collected spectra. Its outputs may be used to generate reports or as inputs for other statistical tasks, so backwards-incompatible changes to both the software itself and its dependencies may affect larger existing data analysis infrastructures in addition to affecting the reproducibility of old results. Hence, it is important to minimize the likelihood of such disruptive changes by both minimizing the number of external software dependencies and ensuring that any changes are backwards compatible.

Moreover, the peptide identification and signal processing tools used to obtain feature-level data from raw spectra are numerous and include popular programs such as MaxQuant (Jürgen Cox and Matthias Mann, 2008; Tyanova, Temu, and Juergen Cox, 2016), Proteome Discoverer (Orsburn, 2021), DIA-NN (Demichev et al., 2020), OpenMS (Röst et al., 2016), Spectronaut (Bernhardt et al., 2012), and Skyline (B. MacLean et al., 2010; Pino et al., 2020). Each tool outputs data in a different format, with varying sets of information. Additionally, some MS groups or companies may use custom processing tools. However, statistical analysis of MS data requires a fixed and predictable format, depending on the assumed statistical model. Thus, it is essential to provide users with the tools necessary for reshaping data from arbitrary data structures into the required format and to verify the correctness of the input.

Despite the consistent format, input data may represent various experimental designs and require different processing steps, such as normalization, aggregation of fractionated data or treatment of missing data. Thus, the software for MS data analysis needs to recognize elements of the experimental design that differentiate or influence the applied statistical models and automatically apply necessary data transformations.

Lastly, MS data sets are typically characterized by high volume. Most of the size reduction occurs at the peptide identification and quantification steps; however, in extreme cases, such as extensive clinical studies, quantitative MS data may reach well over 100 GB. Even in smaller cases, both statistical operations and pre-processing steps may have significant memory and time requirements. Hence,

it is essential to design the MS software in a manner that is both efficient in terms of memory and computation time usage, and enables users to handle large computations.

To summarize, we identified four key factors required in the design of statistical software for MS data analysis: user interface, stability, adaptability to varying inputs (both in terms of data format and statistical properties), and computational efficiency. From the perspective of software development, an additional challenge is managing the complexity (size and dependencies) of these tools. We will address all of these concerns while describing both our contributions to the established MSstats framework and the implementations of proposed approaches.

In our contribution to the MSstats family of packages, we addressed the four concerns in the following manner:

- **user interface** by simplifying and unifying parameters of functions, and modularizing the workflow (decomposing complex functions that implemented large portions of the workflow into smaller functions)
- **stability** by reducing the number of external dependencies and dividing parts of the MSstats workflow into more independent packages,
- **adaptability** by creating general functions for converting any output of a signal processing tool into the MSstats-compatible format, moving related functions to a separate package `MSstatsConvert`,
- **computational efficiency** by using efficient implementations of data processing steps based on tidy tabular formats, and re-implementing parts of the workflow in a lower-level programming language, C++, using the Rcpp (Eddelbuettel and François, 2011) infrastructure.

We evaluated proposed changes to the MSstats family of packages qualitatively and quantitatively. From a qualitative perspective, we were interested in changes to the overall design of the core MSstats package and the entire family of packages, specifically in terms of external package dependencies, modularity, extensibility, and user interface changes. All these factors contribute to the long-term sustainability and scalability of software tools.

From a performance perspective, we compared the running times and memory consumption of previous versions of MSstats with the proposed version 4.0. The complete MSstats workflow comprises data import, protein-level summarization, and differential abundance analysis. As a part of the refactoring process, some data transformations were moved to functions responsible for other parts of this workflow. Hence, a fair comparison of the two versions requires evaluating the running time of a complete workflow. We evaluated it by using the *microbenchmark* package (Mersmann, 2023), which measures the time needed to execute pieces of R code based on multiple repetitions. This ensures that additional periodic operations, such as the garbage collector clearing the memory, do not bias the measurements. To assess the changes in memory consumption, we used the *profvis* package, which measures time spent in various function calls in a given R program, along with the memory requested and freed by them. As assignments of memory usage to particular lines or small pieces of code may not be entirely accurate, we again used measurements for the entire workflow to ensure a fair comparison.

5.1.3 Results

5.1.3.1 Reduction in number of dependencies

Some R packages, such as *data.table* or *ggplot2*, provide state-of-the-art open-source implementations of data analysis tasks, including tabular data processing in the former case and methodical (*grammar of graphics*-based) visualization in the latter case. Hence, it is impossible and impractical to avoid

external dependencies altogether. However, R packages by various authors differ in their approach to backwards-incompatible changes, the number of their own dependencies, and other matters related to long-term software maintenance. Hence, limiting the number of dependencies is beneficial to the software's stability.

MSstats v3 relied on multiple packages, including two distinct frameworks for tabular data transformations (*data.table* and *dplyr*). MSstats v4 reduced the number of direct dependencies, as displayed in Figure 5.1. Here, by direct dependencies, we mean packages explicitly named as a dependency in the *Imports* section of the package's *DESCRIPTION* file. The original release of MSstats v4 did not include the *plotly* dependency, which was added later to enhance the package's graphical capabilities.

The number of indirect dependencies (dependencies of direct dependencies) does not necessarily decrease with a lower number of direct dependencies. However, these cannot be controlled by a developer and may evolve over time. MSstats v4 removed the dependence on some packages known for changes in the interface that favor convenience over backwards compatibility and dependencies related to the same task (such as packages for parallel processing or tabular data transformations from the tidyverse family of packages). Hence, the reduction in the number of direct dependencies was primarily achieved by introducing a new, consistent method for processing data, which will be expanded upon in Section 5.1.3.3. Newly introduced dependencies and most remaining dependencies were either small packages designed for stability (such as the `tinytest` package (van der Loo, 2020)) or popular and stable packages such as Rcpp (Eddelbuettel and François, 2011).

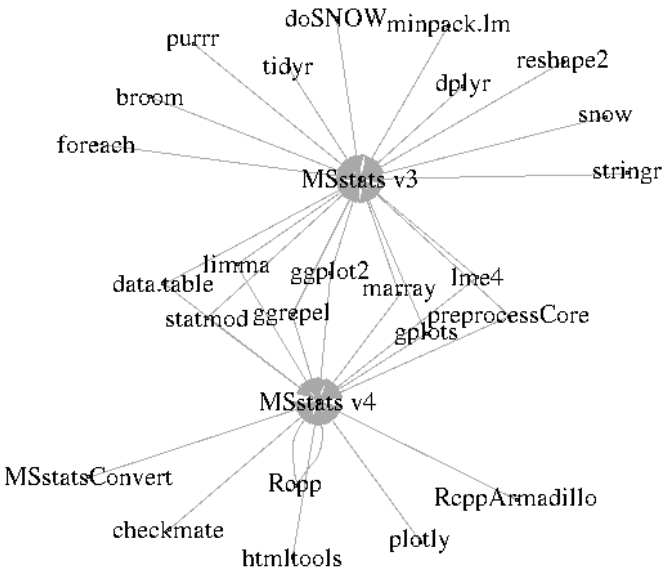


Figure 5.1: **MSstats v4 reduced the number of direct dependencies**. Major nodes labeled as MSstats v4 and MSstats v3 denote the two MSstats version of interest. Edges of the graph denote dependency of each version on various packages indicated by their names.

5.1.3.2 Modular design of MSstats packages

Increased modularization addresses multiple challenges in software design, including extensibility, testability, and the creation of user interfaces.

Package-level modularization Two additional packages were extracted from MSstats to increase coherence of each piece of software: MSstatsLOBD and MSstatsConvert. The former implemented a self-contained set of methods for estimating the limit of blank and the limit of detection in a given MS experiment (Galitzine et al., 2018). These measures characterize assays (procedures for measuring

specific target analytes) based on samples with known peptide concentrations by estimating concentrations at which the upper bound of a prediction interval of the background noise intersects either the mean intensity of peptide samples (LOB) or a lower bound of the prediction interval of samples with the peptide (LOD). The MSstatsConvert package implements all operations related to converting a set of MS measurements from the output of a signal processing tool to the standard MSstats format. It includes new, more general functions for pre-processing the outputs of various signal processing tools. Hence, the MSstats package was designated to implement basic summarization and quantification, which can be used by both end users for the analysis of label-free and targeted experiments, as well as other packages designed for different types of experimental workflows, such as isotope labeling (MSstatsTMT package) or PTM quantification (MSstatsPTM package). Hence, with the decomposition of MSstats into several packages, most importantly MSstatsConvert, updates regarding tasks such as interfacing MSstats and external tools became independent of the development of the statistical core part of the package. Similarly, various workflows exist independently in different packages that share the common core of functions implemented in MSstats.

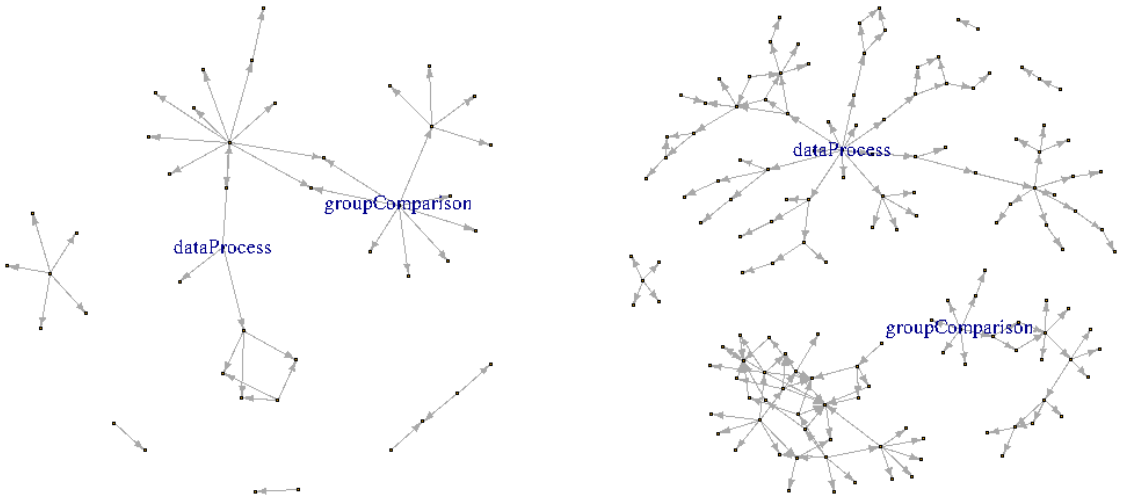
Task-level modularization The MSstatsConvert package was an original redesign of the MSstats preprocessing workflow. Originally, the MSstats package offered a converter function for each supported signal processing tool. For example, a function which re-formatted Skyline output into a tabular format assumed by MSstats was called *SkylinetoMSstatsFormat*. These functions repeated large portions of the code related to specific data transformations, such as aggregating multiple measurements describing the same feature in a given run. MSstatsConvert abstracted these functions into three new major components: MSstatsClean, MSstatsPreprocess, and MSstatsBalancedDesign. The latter two functions implemented steps common to processing outputs from all signal processing tools. These steps include filtering (for example by q-values (Storey and Tibshirani, 2003) estimated to control errors of peptide identification from spectra or by removing contaminants), removing features characterized by at most two measurements across MS runs, removing shared peptides, aggregating multiple measurements per feature and run, optional removal of proteins identified by a single feature, and merging annotation data with the quantitative data. Annotation connects each MS run to the experimental design of a given study, matching identifiers of biological or technical replicates and experimental conditions to MS runs (in the case of label-free data) or runs and channels (in the case of labeled data). Additionally, the resulting data are re-formatted into a long tabular format with a row for intensity values for each run (or run and channel) and each spectral feature (potentially with missing values).

Hence, the aforementioned functions can be used to re-format arbitrary data, including custom in-house spectral data processing workflows, into an MSstats-compatible format. The MSstatsClean function, on the other hand, implements (via S4 class inheritance) re-formatting steps unique to each tool. For example, MS quantification data may be output in long or wide format. In the latter version, measurements from different runs are stored in separate columns. The MSstatsClean methods ensure that a table suitable for input to the MSstatsPreprocess function is in long format, meaning that feature and run identifiers are stored in columns, with quantified intensity values in a single column. The benefits of using this format are described in the Section 5.1.3.3.

Similarly, the two major functionalities of MSstats, i.e., summarization and differential analysis, implemented in the *dataProcess* and *groupComparison* functions, respectively, were abstracted into multiple functions that implement logical steps of the analysis. For protein-level summarization, these steps included input verification and cleaning, feature-level normalization, handling fractions and imputation, feature selection, estimating the summary for each protein, and formatting the output. Each step was implemented in a separate function with a standardized name (for example *MSstatsNormalize*) and output format (typically a table). Differential analysis consisted of input verification and cleaning, checking or creating contrasts, proper per-protein model fitting, and post-processing. Again, each step was implemented in a separate function.

Such a design provides several benefits. Firstly, small functions were easier to test, as with such a modular design, introducing changes requires working with only a small subset of the codebase. Sec-

ondly, users gained the ability to save and analyze partial results of the analysis, which is particularly beneficial for large data sets. Lastly, the new functions were designed with parallelization in mind, allowing for the simultaneous processing of multiple proteins. In the proposed implementation, no built-in parallelization was provided. In general, the use of parallel backends depends on the operating system and hardware choices. Hence, MSstats 4.0 originally provided only functions that could be used to parallelize computations, leaving the choices regarding particular parallelization framework up to the user. However, this functionality was added to MSstats in a later update by another developer (Anthony Wu, MSstats version 4.12).



(a) Internal dependencies between functions in MSstats v3.

(b) Internal dependencies between functions in MSstats v4.

Figure 5.2: **MSstats v4 increased the modularity of the package by extracting small functions implementing specific sub-tasks from functions that implemented high-level data analysis tasks**. Dependency data were extracted using the *pkgdepR* package in R (Peyton, 2022).

Figure 5.2 illustrates the difference in design of functions between MSstats v3 and refactor MSstats v4. In this plot, the vertices of each graph correspond to functions in the package. Edges denote a call of a given function by another function. A significant increase in the number of vertices and edges indicates a more granular design, with smaller functions performing specific tasks that are reused when appropriate.

Example 6 (Case study in modularization) *As an example of task-level modularization, let us consider the* `PDtoMSstatsTMT` *function, which converts the outputs of Proteome Discoverer (Orsburn, 2021) into the MSstats format. In MSstats v3, it consisted of over 200 lines of code. Many operations are performed by the function we shared with other converter functions. In MSstats v4, the function consists of about 40 lines of code, mainly due to formatting. Changes that facilitate modularization can be observed in Code Listing 5.1. The . . . symbol denotes passing arguments to functions or minor operations such as displaying log messages on the console. The pre-processing workflow was generalized in the following manner. Functions* `MSstatsMakeAnnotation`, `MSstatsPreprocess`, *and* `MSstatsBalancedDesign` *implement general operations required to clean quantitative data and reshape them into a format. All converters use them. Moreover, as functions exported by the package, they can be used to create custom pre-processing workflows that convert outputs of in-house or highly customized signal processing tools to the MSstats format.* `MSstatsMakeAnnotation` *extracts annotation information that is later merged with quantitative data.* `MSstatsPreprocess` *performs filtering and aggregation operations to ensure that the data satisfies the assumptions of the MSstats*

format and the user's criteria for data quality. Examples include q-value filtering, removing modified peptides, or taking the maximum intensity when multiple values are found for the same feature and run. In particular, lines 6-7 provide an example of a data structure that implements a general filter for removing modifications specified by the user. `MSstatsBalancedDesign` *function reshapes data into a format that includes a single row for each feature and run, according to the assumptions of MSstats. Hence, it converts data into a proper long format. On the other hand, the functions* `MSstatsImport` *and* `MSstatsClean` *are specific to each signal processing tool.* `MSstatsClean` *is a generic function that provides methods for data types created by the* `MSstatsImport` *function, which simply reads output files from tools such as PD. These functions can be extended by creating new methods for additional tools. Again, this can be done easily by users who require MSstats data pre-processing for non-standard signal processing workflows.*

```
MSstatsConvert::MSstatsLogsSettings(...)
input = MSstatsConvert::MSstatsImport(...)
input = MSstatsConvert::MSstatsClean(...)
annotation = MSstatsConvert::MSstatsMakeAnnotation(...)

oxidation_filter = list(col_name = "PeptideSequence", pattern = "Oxidation",
    filter = removeOxidationMpeptides, drop_column = FALSE)

feature_columns = c("PeptideSequence", "PrecursorCharge")
input = MSstatsConvert::MSstatsPreprocess(...)
input = MSstatsConvert::MSstatsBalancedDesign(...)
...
input
```

Listing 5.1: Example of a re-factor converter function `PDtoMSstatsFormat`

5.1.3.3 Tidy data format

The concept of tidy data was described in Wickham, 2014 and introduced to the R environment via the tidyverse family of packages (Wickham et al., 2019). However, this general concept does not require any of the tidyverse package for a proper implementation. In short, a tidy data set is a table or a group tables such that column corresponds to a variable (in a statistical sense), and each row corresponds to an observation. These simple assumptions are often violated in practice, for example in case of MS data when measurements from different runs are stored in a single cell of a table with a text separator such as a comma, or when a row definition does not match the definition of an observation (outcome a single measurement, smallest observational unit).

In parallel to the proposed refactoring of the MSstats package, usefulness of the tidy data concept in processing of MS data was observed by Kumler and Ingalls, 2022. Let us notice that tidy data-based design may use the long format of data to express standard data transformations as basic operations of tabular data. In case of the MSstats format, an observation is defined as a single $\log_2$-intensity value for each feature (defined by peptide sequence and ion charge) and run. Additionally, features are matched to proteins that they originated from. On the other hand, each run is annotated by at least biological replicate and condition. Two examples of non-tidy format for such data are a table with multiple feature intensities per run, or a table in which each run is described by a separate column.

To illustrate the difference between wide and long format, let us consider the the example task of normalizing feature-level data by subtracting run-specific mean from all $\log_2$-intensities of features. In the wide format, this requires iterating over columns that denote MS runs, calculating and possibly storing their means, and subtracting them from the raw values. In the long format, the same operation can be performed by grouping by the run column, calculating means across such groups and subtracting them from a column that stores raw intensities. The latter approach uses syntax that is common to tabular operations and databases such as SQL, does not require iterating, and is implemented efficiently in tools such as the data.table package (Barrett et al., 2025).

Hence, we proposed using the long, tidy format for data and implementing all processing steps, including summarization, in terms of basic tabular operations: filtering rows, modifying columns, grouping and aggregating data. This made it possible to use the most efficient implementations of such operations (data.table package) and to create an interface that is highly consistent with external database capable of processing out-of-memory data such as spark (Zaharia et al., 2016; Zaharia et al., 2016).

5.1.3.4 Reduction in data processing time

Table 5.1 summarizes the differences between three versions of MSstats: v2, which implemented an earlier version of the statistical methods, v3, based on Choi, 2016, and v4, which refactored v3 code to make it more efficient and facilitate working with large data sets. Data sets are labelled as in Table 5.2 below. These examples covered cases from small (64 MB) to reasonably large (1.04 GB) data sets. In some cases, the proposed approach limited total memory consumption by over 50%, while maximum allocations were reduced even further. In terms of a total running time of the workflow (data conversion, summarization, and differential analysis), refactored MSstats improved by 20-50%.

		Processing Time [s]			Mem. Total [MB]			Mem. Sum Max [MB]		
ID	File Size	v2	v3	v4	v2	v3	v4	v2	v3	v4
1.	200 MB	831.6	231.8	164.4	16,583.6	40,315.1	13,542.0	109.3	147.2	20.2
2.	1.04 GB	3,895.7	1,428.8	963.2	20,726.5	17,920.8	7,090.2	291.1	243.6	13.8
3.	63 MB	243.5	97.10	42.4	20,341.5	15,176.9	6,042.4	100.6	111.6	20.5
4.	257 MB	1,077.0	636.6	339.0	16,948.5	24,507.4	10,665.8	297.5	335.4	16.2
5.	315 MB	4,411.8	2,258.3	1,897.6	85,845.1	26,531.1	18,125.7	412.4	234.2	29.8

Table 5.1: **Refactored MSstats version reduced both computation time and memory usage across data sets of varying sizes and properties.** Taken from Kohler, Staniak, Tsai, et al., 2023.

These improvements were made possible by the data processing strategy described in the previous section and the use of efficient implementations of basic operations, such as Tukey Median Polish, written in C++ using the Rcpp framework (Eddelbuettel and François, 2011). Basic information about data sets used in this evaluation is given in Table 5.2.

ID	Name	Mode	Processing	No. conditions	No. bio. rep.	No. tech. rep.
1.	Controlled mixture	DDA	MaxQuant	4	1	3
2.	Controlled mixture	DIA	Spectronaut	2	1	3
3.	Controlled mixture	DDA	Skyline	5	1	3
4.	Mouse	DDA	MaxQuant	2	6	2
5.	*S. cerevisiae*	DIA	Skyline	6	3	1

Table 5.2: **Experimental design of data sets used to evaluate the performance of a proposed implementation of the MSstats package.** Adapted from Kohler, Staniak, Tsai, et al., 2023. Data sets 1-3 represented a group comparison design, data set 4 - paired design, data set 5 - time course design. References to original studies and locations of the data are given in Kohler, Staniak, Tsai, et al., 2023. Column *mode* denotes acquisition mode (data-dependent or data-independent), while the *processing* column indicates the signal processing tools which were used to extract quantitative data from raw spectra. Data sets were originally published: 1 - in Choi, Eren-Dogu, et al., 2017, 2 - in Navarro et al., 2016, 3 - inChiva, Ortega, and Sabido, 2014, 4 - in Meierhofer et al., 2016, 5 - in Selevsek et al., 2015.

5.1.4 Discussion

We presented a redesign and refactoring of an existing software suite, MSstats. Proposed structure of R packages in the MSstats family and changes to handling operations on tabular data, along with updated implementations of parts of the model fitting-related tasks, enabled analysis of larger data sets and easy extension of pre-processing functionalities to accommodate outputs of additional signal processing tools. Moreover, it improved long-term maintainability and extensibility of all relevant packages.

The MSstats approach to differential analysis of protein abundances derived from MS data is both reliable and flexible enough to accommodate new experimental protocols that produce data with varying characteristics and requiring diversified statistical models to represent all relevant sources of variation, such as quantification of post-translational modifications (Kohler, Tsai, et al., 2023) or limited proteolysis MS (Malinovska et al., 2023). Moreover, MSstats provides tools for intermediate data analysis steps such as missing data imputation or data reduction (selection of a subset of spectral features), quality control (Dogu et al., 2018), and visualization of both feature-level data and the outcomes of hypothesis testing. The future development of MSstats-based data analysis tools aligns with the extensions of statistical methods proposed in this dissertation in several ways.

Firstly, a major bottleneck for the performance of MSstats, in terms of both memory consumption and processing time, is data imputation. Currently, MSstats uses a state-of-the-art implementation of the AFT model from the `survival` package (Therneau, 2023; Terry M. Therneau and Patricia M. Grambsch, 2000). However, large numbers of spectral features matching some proteins (possibly more than a hundred), combined with increasing numbers of runs, significantly increase the total size of data that needs to be processed during imputation. Reduced computation time can be achieved by reducing the number of features or by modifying the imputation method, either in terms of the optimization routine or the statistical approach. In the simplest scenario, the former can be achieved by simply selecting a fixed number of features characterized by the highest average intensity (the popular *top N* approach). At the same time, the statistical model assumed by MSstats is not compatible with multi-protein cluster data. As the AFT model assumes homogeneity of quantitative patterns, it requires a significant update to accommodate data with shared peptides. Moreover, multi-protein clusters naturally include a larger number of peptides, sharing the large data issue with some of the unique peptides-based data. Hence, the development of an efficient imputation approach compatible with a data structure that includes shared peptides is a crucial topic for future research, with a significant impact on the overall performance of MSstats data processing.

Data imputation is not the only processing step in MSstats that might benefit from an alternative modeling approach. The modularized design of MSstats enables using various summarization methods. The currently used approach, Tukey median polish, is fast and robust; however, it is only an approximation to the L_1 norm-based optimization problem, which could provide additional information about the uncertainty of estimating protein-level expression profiles. It provides point-wise estimates with no way of evaluating their credence based on sample size, estimated noise levels or other criteria.

The statistical approach to joint summarization of proteins proposed in this dissertation improves on TMP-based optimization by utilizing an explicit statistical model, which is estimated by minimizing an associated loss function. In principle, this facilitates the computation of confidence or prediction intervals for estimated abundances of proteins. Moreover, estimation for a single protein is a natural special case of this approach. In practice, to enable complete description of variability of abundance estimates, more theoretical work related to the likelihood function for the model is required due to constraints on weights with the possibility of finding solutions on the border of these constraints, and the fact that we search for minima of the objective function in the biconvex terms, before the model can be fully utilized in this context.

Moreover, the MSstats suite now includes an experimental tool `MSstatsBioNet` (Wu and Vitek, 2025) for obtaining information about protein interactions from the INDRA database (Gyori et al.,

2017; Bachman, Gyori, and Sorger, 2023). Further developments in the area of connecting databases that describe known relationships between proteins, along with outputs from both protein inference and differential analysis, would provide additional biological insights in studies involving protein isoforms identified by shared peptides. Hence, developing tools for integrating information from publicly available databases, combined with an effort to incorporate prior knowledge in the joint summarization of protein clusters, would be beneficial for both the applications of the proposed statistical approach and the MSstats family of packages.

Lastly, MSstats is an example of fostering good practices in the field of MS data analysis. It includes a well-defined input format with tools for re-shaping outputs of popular data processing tools into this format, proper statistical models that take into account the experimental design, graphical and command line interfaces to all methods, and an associated repository for sharing quantitative data that facilitates re-analyses of data, and thus benchmarking of MS data analysis strategies (Choi, Carver, et al., 2020). The development of related tools for HDX-MS data, particularly a platform for sharing and re-analyzing data, would be immensely beneficial. Similarly, standardized open formats and general statistical models would simplify both the analysis and re-analysis of HDX data sets.

With technological and conceptual advancements in the field of mass spectrometry-based proteomics, the size and complexity of the data may continue to increase. The MSstats framework possesses the flexibility required to adapt to such developments and continue serving as a good tool for the interpretation of quantitative MS data. The proposed changes in both structure and implementation of MSstats packages were an important step towards this goal.

5.2 Implementation of proposed statistical models

Proposed methods described in Chapters 3 and 4 were implemented in open source R packages.

- *MSstatsWeightedSummary* package provides tools for weighted summarization of labeled MS data. It is currently available at

  https://github.com/Vitek-Lab/MSstatsWeightedSummary.

- *IsoHDX* package implements the HDX exchange rates localization. It is currently available at https://github.com/mstaniak/IsoHDX.

In this section, we describe the design, workflow and functionalities of both packages.

5.2.1 MSstatsWeightedSummary package

The weighted summarization approach to joint quantification of clusters of proteins was implemented in a free and open source R package `MSstatsWeightedSummary`. The goal was to seamlessly extend the unique peptides-oriented workflow of `MSstatsTMT`. An analysis that uses shared peptides requires the following steps

- identification of clusters of proteins
- (optional) re-formatting data into the MSstats format
- (optional) processing such as normalization
- summarization
- re-formatting into the MSstatsTMT format.

In this section, we describe details of each step of the analysis.

5.2.1.1 Data structure induced by the proposed approach

Restricting analysis to unique peptides induces a data structure in which observations collected across mixtures, MS runs, biological samples and spectral features can be arranged in a hierarchical structure with protein as the highest level of grouping. The inclusion of shared peptides changes this structure, as shared peptides are crossed between proteins. This complicates the statistical analysis with the MSstats workflow. Firstly, proteins are no longer the highest level of grouping, with clusters replacing them in this role. Secondly, the information from peptide-protein graph has to be stored in some way to ensure that every peptide is included in summarization of matching proteins. A simple way to address these changes with minimal changes to the MSstats data format is to add a column with cluster identifiers, and duplicate data concerning each shared peptide as many times as there are proteins that match to it. Such implementation uses the long data format to store peptide-protein graph information, and enables easy aggregation of information both per protein and per cluster. However, it requires extra care while performing certain data processing operations such as normalization. In that case, only intensities of each feature need to be used only once in median computation. Hence, proposed implementation uses non-duplicate feature-level values, and then matches transformed data back to the complete set of inferred proteins.

5.2.1.2 MSstatsWeightedSummary workflow

Identification of clusters of proteins The first and crucial step in the analysis of protein isoforms data consists of finding cluster of proteins that share peptides. Then, as each cluster can be analyzed independently of others except for purposes such as normalization, other analysis steps, in particular summarization, can be performed by iterating over identified clusters.

We used the R package igraph (Csardi and Nepusz, 2006; Csárdi et al., 2025) for graph-related operations. The simplest way to identify clusters is implemented in the `createPeptideProteinGraph` function, which takes as input quantitative data with user-specified columns that describe proteins and peptides, and passes them to igraph's algorithm for constructing bivariate graphs from edges represented by rows of R's tabular data structure `data.frame`. Output of this function can be then passed the `addClusterMembership` function, which uses igraph's functionality for detecting connected subgraphs of a graphs to assign unique integer IDs to each cluster. This function decomposes proteins into clusters based on complete data set. As discussed previously, this may lead to different clusters compared to a per-run analysis, as some subsets of proteins in a cluster may share peptides or be identified by unique peptides only in some runs.

Let us reiterate that in order to properly account for shared peptides, analysts need to select a proper size of a protein database at the peptide identification step. However, for purposes of exploration of the influence of database sizes, shared peptides, and related factors, or benchmarking quantitative approaches, such peptide search may be infeasible or impossible. In these cases, we propose using the `adjustProteinAssignments` function, which searches for all proteins that contain a given set of (unmodified) peptide sequences in a selected fasta database of protein sequences. Additionally, functions `getClusterStatistics` and `plotClusterStats` provide an option to calculate and visualize simple summary statistics such as counts of unique or shared peptides per cluster, or sizes of clusters.

Moreover, operations on protein IDs described in Section 3.2.1 were implemented in a function called `processIsoforms`. It is capable of finding proteins identified by the same set of peptides based on peptide-protein graphs of clusters and merging them. It also provides an option to remove proteins identified solely by a shared peptide from the analysis. Processing subset proteins is currently limited to removing all subset proteins due to previously discussed conceptual challenges. Each filtering step is followed by re-computing clusters due to possible changes in uniqueness of peptides.

Re-formatting into the MSstats format In principle, any data set in the MSstatsTMT 11-column format is suitable for analysis with the `MSstatsWeightedSummary` package. Whenever labels of protein clusters were not provided by the user, they are added automatically as a part of the workflow.

However, some details of data pre-processing change with the addition of shared peptides. The user may either use one of the standard `MSstatsTMT` converters while keeping shared peptides, or a custom pre-processing pipeline. The latter option provides more control over the order of operations and choice whether some options should include shared peptides or not. For example, filtering by a minimum number of features per protein may use only unique peptides or both unique and shared peptides. As the order of operations is also up to user's prefence, we do not provide any additional converter functions.

Data processing As discussed in Section 3.2.1, including shared peptides in summarization requires careful normalization. We implemented it in the `normalizeSharedPeptides` function. If the data were normalized earlier (for example by the signal processing tool) or no normalization is needed, this step of the analysis can be skipped. Currently, the package only supports median normalization, unlike the main `MSstats` package which also provides options such as quantile normalization. Moreover, a simplistic method of imputing missing data under the assumption that they are missing at random (median imputation) is given. However, by default missing values are ignored.

Cluster summarization The core `getWeightedProteinSummary` function which takes as input a table in the same format as MSstatsTMT. Previous normalization needs to be indicated by the presence of a column named `log2IntensityNormalized`, while pre-calculated clusters should be indicated by a `Cluster` column. If not provided, the function will add cluster labels before summarization. Development version of this function included options of selecting different loss functions and their hyperparameters, modifying constraints on weights, and weights penalization, but user-oriented version removed those options in favor of simplicity of the interface and reducing the *researcher's degrees of freedom*. As a consequence, only one parameter related to loss function remained (`norm_param` which stores the value of a hyperparameter of the Huber loss, with a default value of 10^{-6}. Two parameters describe the convergence of iterative optimization routine: `tolerance` which denotes the average squared difference between consecutive sets of weights required to declare convergence (default value: 0.01) and `max_iter` which defines the maximum number of iterations (default value: 10). Two additional parameter control information about the iterative optimization process which will be stored: `save_weights_history`, `save_convergence_history`. Both are logical parameters with a default false of `FALSE`. If the former is set to `TRUE`, weights estimated at each step will be saved in a list of tables. If the latter is set to `TRUE`, values of the convergence criterion at each iteration will be reported.

As the summary for every cluster is estimated separately in each run of the experiment, this core task is performed by another function exported for the user: `getWeightedSummarySingleRun`. It can be called directly to speed up the analysis by skipping data correctness checks and other minor steps included in the `getWeightedProteinSummary` function, or implicitly by running the main function. Proposed implementation of weighted summarization uses the `CVXR` package (Fu, Narasimhan, and Boyd, 2020) internally to optimize convex sub-problems of full optimization problem.

The step with fixed weights is a simple unconstrained problem in which abundances of proteins in each channel are the main parameters of interest, and the design matrix of an associated linear model is populated with previously estimated weights for the main parameters, and $0/1$ values for the intercept and feature effect.

The step with fixed protein abundances is a constrained optimization problem. `CVXR` package accepts information about constraints in the form of binary matrices that indicate which parameters are subjected to a given constrained (with different constraints stored in different rows of such matrix), symbols that indicate the type of a constraints (equality / inequality), and a vector of values of the constraints. This way we ensure that the weights are non-negative and sum to 1 for each feature.

MSstatsTMT format conversion The main output of the `getWeightedProteinSummary` function is a table that provides abundances of all proteins in each run. Additionally, we return information about estimated weights, convergence for each cluster, and feature-level data which were used as input. The package provides accessor functions for returning all that information: `featureData` and

`proteinData` for input and output data, `convergenceSummary` and `convergenceHistory` for detailed or minimum convergence information. Finally, `weightsHistory` and `featureWeights` functions return feature-specific weights from all iterations or only the final weights, respectively. Moreover, `plotSummary` method enables plotting profile plots of both feature-level data and protein-level summaries for selected clusters.

An additional utility function `makeMSstatsTMTInput` converts the results of weighted summarization into an exact format accepted by the `groupComparisonTMT` function from the `MSstatsTMT` package which performs differential analysis. Its optional parameter `msstatstmt_output` enables easy merging of summarization results for clusters that included shared peptides with results from the `MSstatsTMT` package based only on unique peptides. Optionally, protein abundances can be normalized after summarization using the `normalizeProteins` function.

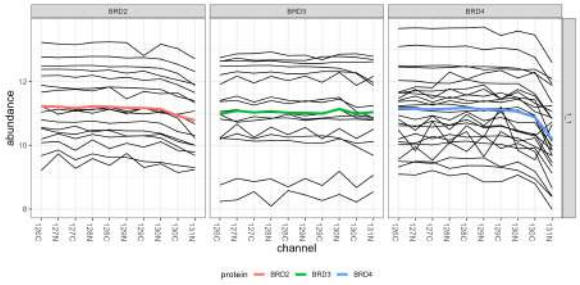
Code listing 5.2 presents an example of the `MSstatsWeightedSummary` workflow. Lines 1-3 identify clusters of peptides and remove subset proteins after merging proteins identified by identical sets of proteins, according to the proposed data processing scheme. Lines 5-10 use a function from the standard MSstats workflow to pre-process data, while retaining shared peptides. Following code selects a single cluster of BRD proteins and indicates that peptide intensities were already normalized. Line 20 fits the proposed model using a `getWeightedProteinSummary` function with Huber loss ($M = 10^{-6}$). Finally, lines 21-22 use utility functions `plotSummary` and `featureWeights` to produce a plot of input feature-level intensities and fitted protein-levels summaries, and a table of estimated peptide-protein weights. Outputs of these functions are displayed in Figure 5.3.

```
1  peptide_protein_graph = createPeptideProteinGraph(orig_long_by_prot, "
       ProteinName", "PeptideSequenceNew")
2  quant_with_cls = addClusterMembership(orig_long_by_prot, peptide_protein_graph)
3  quant_no_subsets = processIsoforms(quant_with_cls, remove_single_shared = T,
       merge_identical = T, remove_subsets = T)
4
5  quant_no_subsets[, Intensity := 2^log2Intensity]
6  quant_all_procd = MSstatsConvert::MSstatsPreprocess(quant_no_subsets,
7   annotation = unique(quant_no_subsets[, .(Run, Channel, BioReplicate, Condition,
        Mixture, TechRepMixture)]),
8    feature_columns = c("PeptideSequence", "PrecursorCharge"),
9    remove_shared_peptides = FALSE, remove_single_feature_proteins = F,
10   list(remove_features_with_few_measurements = FALSE, summarize_multiple_psms =
       max))
11
12 final_graph = createPeptideProteinGraph(quant_all_procd)
13 quant_all_procd = addClusterMembership(quant_all_procd, final_graph)
14
15 brd_cluster = quant_all_procd[Cluster == 529]
16 brd_cluster[, log2IntensityNormalized := log(Intensity, 2)]
17
18 ch_order = annot[, .(Channel, Time, Condition, Group)][order(-Group, Time),
       unique(Channel)]
19
20 full_summary = getWeightedProteinSummary(brd_cluster, "Huber", 1e-6)
21 plotSummary(full_summary2, proteins = c("BRD2", "BRD3", "BRD4"), channel_order =
        ch_order)
22 featureWeights(full_summary2)
```

Listing 5.2: Example application of the MSstatsWeightedSummaryPackage

5.2.2 IsoHDX package

The proposed model for estimating segment-level H/D exchange probabilities was implemented in the free and open-source R package `IsoHDX`. The primary goal of the package was to provide tools for



(a)

	ProteinName	PSM	Run	Weight
1	BRD2	K.R	1_1	0.026
2	BRD4	K.R	1_1	0.974
3	BRD3	R.L	1_1	0.413
4	BRD4	R.L	1_1	0.587
5	BRD2	R.L	1_1	0.606
6	BRD3	R.L	1_1	0.000
7	BRD4	R.L	1_1	0.394
8	BRD2	R.L	1_1	0.624
9	BRD3	R.L	1_1	0.000
10	BRD4	R.L	1_1	0.376
11	BRD3	R.V	1_1	0.855
12	BRD4	R.V	1_1	0.145

(b)

Table 5.3: Example outputs of the `MSstatsWeightedSummary` workflow. (a) A profile plot of input $\log_2$-intensities of features with protein-level summaries indicated by colored lines. (b) A table of fitted

fitting and visualizing the segment-level probabilities. However, as most HDX-MS data processing workflows use average deuteration levels instead of isotopic data, it also had to provide basic tools for extracting complete isotopic distributions from raw spectral data.

5.2.2.1 Proposed data structure

Spectral data used to fit the proposed model have a straightforward structure. Required information consists of: peptide sequences, time values, information about replicates per peptide and time, mass shift and intensity. Using mass shifts to label isotopic peaks has an important advantage over using continuous m/z values. Fitting the proposed approach requires comparing observed and predicted intensities, which is achieved by merging tables that describe the two sets of intensities. Comparisons between discrete mass shift values are more practical than comparisons between m/z values that are subject to rounding accuracy errors. Table 5.4 presents an example of such a table.

Peptide	Charge	Time	Rep	IntDiff	Intensity
MFTISQVTPLHSGT	2	0.0027	3	2	3.52
MFTISQVTPLHSGT	2	0.0027	3	3	6.46
MFTISQVTPLHSGT	2	0.0027	3	4	9.45
MFTISQVTPLHSGT	2	0.0027	3	5	16.44
MFTISQVTPLHSGT	2	0.0027	3	6	17.97
MFTISQVTPLHSGT	2	0.0027	3	7	16.19

Table 5.4: **HVEM case study:** example of a tabular data structure that stores spectral data.

Additional information is required to fit the proposed model: segments must be defined and matched to observed peptides, the number of exchangeable hydrogens and, consequently, model parameters must be given, and isotopic distributions of undeuterated peptides are needed to calculate expected intensities of peaks. We store segment information in a table that lists matching peptides, their starting and ending positions in the protein sequences, and the number of exchangeable hydrogens for each segment. Isotopic distributions are stored in a named list, with peptide sequences serving as identifiers. Moreover, when time 0 data is available, a table with counts of isotopic peaks observed

for undeuterated peptides can be used to obtain a precise number of maximum observable isotopic peaks for each peptide.

5.2.2.2 Model-fitting details

The `IsoHDX` package implements the proposed model in a function called `fitIsoSegmentModel`. Peptide-level data in a format described in Section 5.2.2.1 are a primary input to this function (parameter `observed_spectra`).

Moreover, information about the clusters is required (parameter `peptides_cluster`). In principle, the proposed approach can estimate probabilities for arbitrarily defined segments, provided that the related problem is identifiable and a good starting point for iterative optimization is found. For example, calling the function with segments identical to peptides can produce a peptide-level analysis of exchange probabilities. Data-driven segments can be identified by using the `getPeptidesCluster` function, which extracts segments from HDX-MS peptide identification data that consist of peptide sequences and indices of starting and ending positions in the protein sequence for each peptide.

Information about undeuterated peptides must be extracted or prepared by the user in the form of a table (parameter `time_0_data`) with columns `Peptide`, `Charge`, `NumPeaks`, where the last column indicates the number of isotopic peaks in the undeuterated spectrum of each peptide ion defined by the remaining columns. This information is used to determine the maximum number of isotopic peaks per spectrum and to find the isotopic probabilities of undeuterated peptides by using the BRAIN package (Dittwald, Claesen, et al., 2013; Dittwald and Valkenborg, 2014).

Users can choose between OLS and PL-GLS approaches (parameter `method`) and between using analytical or numerical gradient (parameter `use_analytical_gradient`). Optimization using numerical gradient is implemented based on the BFGS algorithm as implemented in the `optim` function in R (R Core Team, 2023), while optimization using the analytical gradient is implemented based on a variant of the Newton method, the Broyden method, as implemented in the package `nleqslv` (Hasselman, 2022). Additional parameters control the convergence of PL-GLS and OLS minimization algorithms. User may provide a set of starting points for optimization or select the number of starting points that will be sampled from the interval $[-0.9, -0.001]$ (parameter `starting_point`). The package can be used to produce plots of estimated probabilities, such as in Figure 4.19, and comparisons of fitted and observed spectra, such as in Figure 4.14

Bibliography

Abzalimov, Rinat R and Igor A Kaltashov (2006). "Extraction of local hydrogen exchange data from HDX CAD MS measurements by deconvolution of isotopic distributions of fragment ions". In: *Journal of the American Society for Mass Spectrometry* 17, pp. 1543–1551.

Babić, Darko, Saša Kazazić, and David M Smith (2019). "Resolution of protein hydrogen/deuterium exchange by fitting amide exchange probabilities to the peptide isotopic envelopes". In: *Rapid Communications in Mass Spectrometry* 33.15, pp. 1248–1257.

Bachman, John A, Benjamin M Gyori, and Peter K Sorger (2023). "Automated assembly of molecular mechanisms at scale from text mining and curated databases". In: *Molecular Systems Biology* 19.5, e11325.

Bai, Mingze et al. (2023). "LFQ-based peptide and protein intensity differential expression analysis". In: *Journal of Proteome Research* 22.6, pp. 2114–2123.

Barrett, Tyson et al. (2025). *data.table: Extension of 'data.frame'*. R package version 1.17.8. URL: https://CRAN.R-project.org/package=data.table.

Beausoleil, Sean A et al. (2006). "A probability-based approach for high-throughput protein phosphorylation analysis and site localization". In: *Nature biotechnology* 24.10, pp. 1285–1292.

Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Bernhardt, Oliver M et al. (2012). "Spectronaut: a fast and efficient algorithm for MRM-like processing of data independent acquisition (SWATH-MS) data". In: *Biognosys. ch*.

Bertsekas, Dimitri P (1997). "Nonlinear programming". In: *Journal of the Operational Research Society* 48.3, pp. 334–334.

Blake, Robert A (2019). "GNE-0011, a novel monovalent BRD4 degrader". In: *Cancer Research* 79.13_Supplement, pp. 4452–4452.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Blein-Nicolas, Mélisande et al. (2012). "Including shared peptides for estimating protein abundances: A significant improvement for quantitative proteomics". In: *PROTEOMICS* 12.18, pp. 2797–2801.

Bludau, Isabell et al. (2021). "Systematic detection of functional proteoform groups from bottom-up proteomic datasets". In: *Nature Communications* 12.1, pp. 1–18.

Brenton, A Gareth and A Ruth Godfrey (2010). "Accurate mass measurement: terminology and treatment of data". In: *Journal of the American Society for Mass Spectrometry* 21.11, pp. 1821–1835.

Broyden, C (1969). "A new double-rank minimisation algorithm. Preliminary report". In: *American Mathematical Society, Notices* 16, p. 670.

Bukhman, Yury V et al. (2008). "Design and analysis of quantitative differential proteomics investigations using LC-MS technology". In: *Journal of Bioinformatics and Computational Biology* 6.01, pp. 107–123.

Byrd, Richard H et al. (1995). "A limited memory algorithm for bound constrained optimization". In: *SIAM Journal on scientific computing* 16.5, pp. 1190–1208.

Chang, Ching-Yun et al. (2012). "Protein significance analysis in selected reaction monitoring (SRM) measurements". In: *Molecular & Cellular Proteomics* 11.4.

Chang, Winston et al. (2022). *shiny: Web Application Framework for R*. R package version 1.7.2. URL: https://CRAN.R-project.org/package=shiny.

Chiva, Cristina, Mireia Ortega, and Eduard Sabido (2014). "Influence of the digestion technique, protease, and missed cleavage peptides in protein quantitation". In: *Journal of proteome research* 13.9, pp. 3979–3986.

Choi, Meena (2016). "A flexible and versatile framework for statistical design and analysis of quantitative mass spectrometry-based proteomic experiments". dissertation. Purdue University. URL: https://docs.lib.purdue.edu/open_access_dissertations/636/.

Choi, Meena, Jeremy Carver, et al. (2020). "MassIVE.quant: a community resource of quantitative mass spectrometry–based proteomics datasets". In: *Nature methods* 17.10, pp. 981–984.

Choi, Meena, Ching-Yun Chang, et al. (May 2014). "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments". In: *Bioinformatics* 30.17, pp. 2524–2526. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu305. eprint: https://academic.oup.com/bioinformatics/article-pdf/30/17/2524/9963847/btu305.pdf. URL: https://doi.org/10.1093/bioinformatics/btu305.

Choi, Meena, Zeynep F Eren-Dogu, et al. (2017). "ABRF Proteome Informatics Research Group (iPRG) 2015 Study: detection of differentially abundant proteins in label-free quantitative LC–MS/MS experiments". In: *Journal of proteome research* 16.2, pp. 945–957.

Claesen, Jürgen (2013). "Statistical models for high-throughput proteomic and genomic data". en. Phd thesis. Hasselt University.

Clough, Timothy et al. (2012). "Statistical protein quantification and significance analysis in label-free LC-MS experiments with complex designs". In: *BMC bioinformatics* 13.Suppl 16, S6.

Cox, Jürgen and Matthias Mann (2008). "MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification". In: *Nature biotechnology* 26.12, pp. 1367–1372.

Csardi, Gabor and Tamas Nepusz (2006). "The igraph software package for complex network research". In: *InterJournal* Complex Systems, p. 1695. URL: https://igraph.org.

Csárdi, Gábor et al. (2025). *igraph: Network Analysis and Visualization in R*. R package version 2.1.4. DOI: 10.5281/zenodo.7682609. URL: https://CRAN.R-project.org/package=igraph.

Davidian, Marie and David M Giltinan (1995). *Nonlinear Models for Repeated Measurement Data*. Vol. 62. CRC Press.

Davis, Mindy I et al. (2011). "Comprehensive analysis of kinase inhibitor selectivity". In: *Nature biotechnology* 29.11, pp. 1046–1051.

De Leeuw, Jan (1994). "Block-relaxation algorithms in statistics". In: *Information Systems and Data Analysis: Prospects—Foundations—Applications*. Springer, pp. 308–324.

Demeulemeester, Nina et al. (2024). "msqrob2PTM: Differential Abundance and Differential Usage Analysis of MS-Based Proteomics Data at the Posttranslational Modification and Peptidoform Level". In: *Molecular & Cellular Proteomics* 23.2.

Demichev, Vadim et al. (2020). "DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput". In: *Nature methods* 17.1, pp. 41–44.

Dermit, Maria et al. (2020). "Peptide correlation analysis (PeCorA) reveals differential proteoform regulation". In: *Journal of Proteome Research* 20.4, pp. 1972–1980.

Dittwald, Piotr, Jürgen Claesen, et al. (2013). "BRAIN: a universal tool for high-throughput calculations of the isotopic distribution for mass spectrometry". In: *Analytical chemistry* 85.4, pp. 1991–1994.

Dittwald, Piotr and Dirk Valkenborg (2014). "BRAIN 2.0: time and memory complexity improvements in the algorithm for calculating the isotope distribution". In: *Journal of the American Society for Mass Spectrometry* 25.4, pp. 588–594.

Dogu, Eralp et al. (2018). "MSstatsQC 2.0: R/Bioconductor package for statistical quality control of mass spectrometry-based proteomics experiments". In: *Journal of proteome research* 18.2, pp. 678–686.

Domahidi, Alexander, Eric Chu, and Stephen Boyd (2013). "ECOS: An SOCP solver for embedded systems". In: *2013 European control conference (ECC)*. IEEE, pp. 3071–3076.

Eddelbuettel, Dirk and Romain François (2011). "Rcpp: Seamless R and C++ Integration". In: *Journal of Statistical Software* 40.8, pp. 1–18. DOI: 10.18637/jss.v040.i08.

Eid, Sameh et al. (2017). "KinMap: a web-based tool for interactive navigation through human kinome data". In: *BMC bioinformatics* 18, pp. 1–6.

Eng, Jimmy K, Tahmina A Jahan, and Michael R Hoopmann (2013). "Comet: an open-source MS/MS sequence database search tool". In: *Proteomics* 13.1, pp. 22–24.

Erosheva, Elena, Stephen Fienberg, and John Lafferty (2004). "Mixed-membership models of scientific publications". In: *Proceedings of the National Academy of Sciences* 101.suppl_1, pp. 5220–5227.

Feller, William (1991). *An introduction to probability theory and its applications*. John Wiley & Sons.

Figueroa-Navedo, Amanda (2023). "Development and Evaluation of Data Analysis Approaches to Increase the Specificity and Performance of Thermal Shift Assays for Assessment of Protein-Small Molecule Interactions". English. PhD thesis. Northeastern University, p. 124. ISBN: 9798381184303. URL: https://www.proquest.com/dissertations-theses/development-evaluation-data-analysis-approaches/docview/2905662282/se-2.

Fink, A. M. (1988). "How to Polish off Median Polish". In: *SIAM Journal on Scientific and Statistical Computing* 9.5, pp. 932–940. DOI: 10.1137/0909064. eprint: https://doi.org/10.1137/0909064. URL: https://doi.org/10.1137/0909064.

Fletcher, Roger (1970). "A new approach to variable metric algorithms". In: *The computer journal* 13.3, pp. 317–322.

Fu, Anqi, Balasubramanian Narasimhan, and Stephen Boyd (2020). "CVXR: An R Package for Disciplined Convex Optimization". In: *Journal of Statistical Software* 94.14, pp. 1–34. DOI: 10.18637/jss.v094.i14.

Gałecki, Andrzej and Tomasz Burzykowski (2012). "Linear mixed-effects model". In: *Linear mixed-effects models using R: a step-by-step approach*. Springer, pp. 245–273.

Galitzine, Cyril et al. (2018). "Nonlinear regression improves accuracy of characterization of multiplexed mass spectrometric assays". In: *Molecular & Cellular Proteomics* 17.5, pp. 913–924.

Gavin, Henri P (2019). "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems". In: *Department of Civil and Environmental Engineering Duke University August* 3, pp. 1–23.

Gerster, Sarah, Taejoon Kwon, et al. (Feb. 1, 2014). "Statistical Approach to Protein Quantification". In: *Molecular & Cellular Proteomics* 13.2, pp. 666–677.

Gerster, Sarah, Ermir Qeli, et al. (2010). "Protein and gene model inference based on statistical modeling in k-partite graphs". In: *Proceedings of the National Academy of Sciences*.

Goeminne, Ludger JE, Andrea Argentini, et al. (2015). "Summarization vs peptide-based models in label-free quantitative proteomics: performance, pitfalls, and data analysis guidelines". In: *Journal of proteome research* 14.6, pp. 2457–2465.

Goeminne, Ludger JE, Kris Gevaert, and Lieven Clement (2018). "Experimental design and data-analysis in label-free quantitative LC/MS proteomics: A tutorial with MSqRob". In: *Journal of Proteomics* 171, pp. 23–36.

Goldfarb, Donald (1970). "A family of variable-metric methods derived by variational means". In: *Mathematics of computation* 24.109, pp. 23–26.

Gorski, Jochen, Frank Pfeuffer, and Kathrin Klamroth (2007). "Biconvex sets and optimization with biconvex functions: a survey and extensions". In: *Mathematical Methods of Operations Research* 66, pp. 373–407.

Guo, Tiannan, Judith A Steen, and Matthias Mann (2025). "Mass-spectrometry-based proteomics: from single cells to clinical applications". In: *Nature* 638.8052, pp. 901–911.

Gyori, Benjamin M et al. (2017). "From word models to executable models of signaling networks using automated assembly". In: *Molecular systems biology* 13.11, p. 954.

Hasselman, Berend (2022). *nleqslv: Solve Systems of Nonlinear Equations*. R package version 3.3.3. URL: https://CRAN.R-project.org/package=nleqslv.

Hermann, Juliane, Leon Schurgers, and Vera Jankowski (2022). "Identification and characterization of post-translational modifications: Clinical implications". In: *Molecular Aspects of Medicine* 86. Impact of Post-Translational Modification on the Genesis and Progression of Diseases, p. 101066. ISSN: 0098-2997. DOI: https://doi.org/10.1016/j.mam.2022.101066. URL: https://www.sciencedirect.com/science/article/pii/S0098299722000036.

Hogg, Robert V, Joseph W McKean, and Allen T Craig (2005). *Introduction to mathematical statistics*. Pearson Education International.

Huang, Ting, Meena Choi, et al. (2020). "MSstatsTMT: Statistical detection of differentially abundant proteins in experiments with isobaric labeling and multiple mixtures". In: *Molecular & Cellular Proteomics* 19.10, pp. 1706–1723.

Huang, Ting, Haipeng Gong, et al. (2013). "ProteinLasso: A Lasso regression approach to protein inference problem in shotgun proteomics". In: *Computational Biology and Chemistry*.

Huang, Ting, Mateusz Staniak, et al. (2023). "Statistical detection of differentially abundant proteins in experiments with repeated measures designs and isobaric labeling". In: *Journal of Proteome Research* 22.8, pp. 2641–2659.

Huang, Ting, Jingjing Wang, et al. (2012). "Protein inference: a review". In: *Briefings in Bioinformatics* 13.5, pp. 586–614.

Huber, Peter J (1992). "Robust estimation of a location parameter". In: *Breakthroughs in statistics: Methodology and distribution*. Springer, pp. 492–518.

Huber, W. et al. (2015). "Orchestrating high-throughput genomic analysis with Bioconductor". In: *Nature Methods* 12.2, pp. 115–121. URL: http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Hvidt, Aase and Sigurd O Nielsen (1966). "Hydrogen exchange in proteins". In: *Advances in protein chemistry* 21, pp. 287–386.

Jacob, Laurent, Florence Combes, and Thomas Burger (June 2018). "PEPA test: fast and powerful differential analysis from relative quantitative proteomics data using shared peptides". In: *Biostatistics* 20.4, pp. 632–647. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxy021. eprint: https://academic.oup.com/biostatistics/article-pdf/20/4/632/30161002/kxy021.pdf. URL: https://doi.org/10.1093/biostatistics/kxy021.

James, Ellie I et al. (2021). "Advances in hydrogen/deuterium exchange mass spectrometry and the pursuit of challenging biological systems". In: *Chemical reviews* 122.8, pp. 7562–7623.

Jin, Shuangshuang et al. (Jan. 1, 2008). "The Effects of Shared Peptides on Protein Quantitation in Label-Free Proteomics by LC/MS/MS". In: *Journal of Proteome Research* 7.1, pp. 164–169.

Kirkpatrick, Donald S et al. (2013). "Phosphoproteomic characterization of DNA damage response in melanoma cells following MEK/PI3K dual inhibition". In: *Proceedings of the National Academy of Sciences* 110.48, pp. 19426–19431.

Kish, Monika et al. (2023a). "Online fully automated system for Hydrogen/Deuterium-exchange mass spectrometry with millisecond time resolution". In: *Analytical Chemistry* 95.11, pp. 5000–5008.

— (2023b). "Online fully automated system for Hydrogen/Deuterium-exchange mass spectrometry with millisecond time resolution". In: *Analytical Chemistry* 95.11, pp. 5000–5008.

Kohler, Devon, Maanasa Kaza, et al. (2023). "MSstatsShiny: a GUI for versatile, scalable, and reproducible statistical analyses of quantitative proteomic experiments". In: *Journal of proteome research* 22.2, pp. 551–556.

Kohler, Devon, Mateusz Staniak, Tsung-Heng Tsai, et al. (2023). "MSstats version 4.0: statistical analyses of quantitative mass spectrometry-based proteomic experiments with chromatography-based quantification at scale". In: *Journal of Proteome Research* 22.5, pp. 1466–1482.

Kohler, Devon, Mateusz Staniak, Fengchao Yu, et al. (2024). "An MSstats workflow for detecting differentially abundant proteins in large-scale data-independent acquisition mass spectrometry experiments with FragPipe processing". In: *Nature Protocols* 19.10, pp. 2915–2938.

Kohler, Devon, Tsung-Heng Tsai, et al. (2023). "MSstatsPTM: Statistical Relative Quantification of Posttranslational Modifications in Bottom-Up Mass Spectrometry-Based Proteomics". In: *Molecular & Cellular Proteomics* 22.1.

Kumler, William and Anitra E Ingalls (2022). "Tidy Data Neatly Resolves Mass-Spectrometry's Ragged Arrays." In: *R Journal* 14.3.

Kuncewicz, Katarzyna et al. (2019). "A structural model of the immune checkpoint CD160–HVEM complex derived from HDX-mass spectrometry and molecular modeling". In: *Oncotarget* 10.4, p. 536.

Levenberg, Kenneth (1944). "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of applied mathematics* 2.2, pp. 164–168.

Lin, Miao-Hsia et al. (Apr. 2022). "Benchmarking differential expression, imputation and quantification methods for proteomics data". In: *Briefings in Bioinformatics* 23.3, bbac138. ISSN: 1477-4054. DOI: 10.1093/bib/bbac138. eprint: https://academic.oup.com/bib/article-pdf/23/3/bbac138/43745440/bbac138.pdf. URL: https://doi.org/10.1093/bib/bbac138.

Linderstrom-Lang, K (1955). "The pH-dependence of the deuterium exchange of insulin". In: *Biochimica et biophysica acta* 18.2, p. 308.

Liu, Sanmin et al. (2011). "HDX-analyzer: a novel package for statistical analysis of protein structure dynamics". In: *BMC bioinformatics* 12, pp. 1–10.

Lukasse, Pieter NJ and Antoine HP America (2014). "Protein inference using peptide quantification patterns". In: *Journal of Proteome Research* 13.7, pp. 3191–3199.

MacLean, Brendan et al. (2010). "Skyline: an open source document editor for creating and analyzing targeted proteomics experiments". In: *Bioinformatics* 26.7, pp. 966–968.

Maculins, Timurs et al. (2021). "Multiplexed proteomics of autophagy-deficient murine macrophages reveals enhanced antimicrobial immunity via the oxidative stress response". In: *Elife* 10, e62320.

Madeira, Fábio et al. (2022). "Search and sequence analysis tools services from EMBL-EBI in 2022". In: *Nucleic acids research* 50.W1, W276–W279.

Madhira, Raviteja (2016). *The effects of parsimony logic and extended parsimony clustering on protein identification and quantification in shotgun proteomics*.

Malinovska, Liliana et al. (2023). "Proteome-wide structural changes measured with limited proteolysis-mass spectrometry: an advanced protocol for high-throughput applications". In: *Nature Protocols* 18.3, pp. 659–682.

Marco, Nicholas et al. (2024). "Functional mixed membership models". In: *Journal of Computational and Graphical Statistics* 33.4, pp. 1139–1149.

Marquardt, Donald W (1963). "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* 11.2, pp. 431–441.

Matthiesen, Rune and Jakob Bunkenborg (2020). "Introduction to mass spectrometry-based proteomics". In: *Mass spectrometry data analysis in proteomics*. Springer.

Mayya, Viveka and David K Han (2009). "Phosphoproteomics by mass spectrometry: insights, implications, applications and limitations". In: *Expert review of proteomics* 6.6, pp. 605–618.

Meierhofer, David et al. (2016). "Ataxin-2 (Atxn2)-knock-out mice show branched chain amino acids and fatty acids pathway alterations". In: *Molecular & cellular proteomics* 15.5, pp. 1728–1739.

Mersmann, Olaf (2023). *microbenchmark: Accurate Timing Functions*. R package version 1.4.10. URL: https://CRAN.R-project.org/package=microbenchmark.

Miller, Rachel M and Lloyd M Smith (2023). "Overview and considerations in bottom-up proteomics". In: *Analyst* 148.3, pp. 475–486.

Mnatsakanyan, Ruzanna et al. (2018). "Detecting post-translational modification signatures as potential biomarkers in clinical mass spectrometry". In: *Expert Review of Proteomics* 15.6. PMID: 29893147, pp. 515–535. DOI: 10.1080/14789450.2018.1483340. eprint: https://doi.org/10.1080/14789450.2018.1483340. URL: https://doi.org/10.1080/14789450.2018.1483340.

Morris, J.S. et al. (2008). "Bayesian analysis of mass spectrometry proteomics data using wavelet based functional mixed models". In: *Biometrics* 64.2, pp. 479–489.

Munoz, Javier and Albert JR Heck (2014). "From the human genome to the human proteome". In: *Angewandte Chemie International Edition* 53.41, pp. 10864–10866.

Navarro, Pedro et al. (2016). "A multicenter study benchmarks software tools for label-free proteome quantification". In: *Nature biotechnology* 34.11, pp. 1130–1136.

Nesterov, Yurii et al. (2018). *Lectures on convex optimization*. Vol. 137. Springer.

Nesvizhskii, Alexey I and Ruedi Aebersold (2005). "Interpretation of shotgun proteomic data". In: *Molecular & Cellular Proteomics* 4.10, pp. 1419–1440.

Noor, Zainab et al. (2021). "Mass spectrometry–based protein identification in proteomics—a review". In: *Briefings in bioinformatics* 22.2, pp. 1620–1638.

Ong, Shao-En and Matthias Mann (2005). "Mass spectrometry–based proteomics turns quantitative". In: *Nature chemical biology* 1.5, pp. 252–262.

Orsburn, Benjamin C. (2021). "Proteome Discoverer—A Community Enhanced Data Processing Suite for Protein Informatics". In: *Proteomes* 9.1. ISSN: 2227-7382. DOI: 10.3390/proteomes9010015. URL: https://www.mdpi.com/2227-7382/9/1/15.

Pascal, Bruce D et al. (2012). "HDX workbench: software for the analysis of H/D exchange MS data". In: *Journal of the American Society for Mass Spectrometry* 23.9, pp. 1512–1521.

Peyton, Ed (2022). *pkgdepR: A package for tracking and visualizing package function dependencies*. URL: https://pkgdepr.org/.

Pino, Lindsay K et al. (2020). "The Skyline ecosystem: Informatics for quantitative mass spectrometry proteomics". In: *Mass spectrometry reviews* 39.3, pp. 229–244.

Plubell, Deanna L, Lukas Käll, et al. (2022). "Putting humpty dumpty back together again: what does protein quantification mean in bottom-up proteomics?" In: *Journal of Proteome Research* 21.4, pp. 891–898.

Plubell, Deanna L, Phillip A Wilmarth, et al. (2017). "Extended multiplexing of tandem mass tags (TMT) labeling reveals age and high fat diet specific proteome changes in mouse epididymal adipose tissue". In: *Molecular & Cellular Proteomics* 16.5, pp. 873–890.

Price, Thomas S. et al. (2007). "EBP, a Program for Protein Identification Using Multiple Tandem Mass Spectrometry Datasets". In: *Molecular & Cellular Proteomics* 6.3, pp. 527–536.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rao, V Srinivasa et al. (2014). "Protein-protein interaction detection: methods and analysis". In: *International journal of proteomics* 2014.1, p. 147648.

Reichel, Lothar and William B Gragg (1990). "Algorithm 686: FORTRAN subroutines for updating the QR decomposition". In: *ACM Transactions on Mathematical Software (TOMS)* 16.4, pp. 369–377.

Röst, Hannes L et al. (2016). "OpenMS: a flexible open-source software platform for mass spectrometry data analysis". In: *Nature methods* 13.9, pp. 741–748.

Saltzberg, Daniel J et al. (2017). "A residue-resolved Bayesian approach to quantitative interpretation of hydrogen–deuterium exchange from mass spectrometry: application to characterizing protein–ligand interactions". In: *The Journal of Physical Chemistry B* 121.15, pp. 3493–3501.

Schork, Karin et al. (2022). "Characterization of peptide-protein relationships in protein ambiguity groups via bipartite graphs". In: *Plos one* 17.10, e0276401.

Seetaloo, Neeleema, Monika Kish, and Jonathan J Phillips (2022a). "HDfleX: software for flexible high structural resolution of hydrogen/deuterium-exchange mass spectrometry data". In: *Analytical Chemistry* 94.11, pp. 4557–4564.

— (2022b). "HDfleX: software for flexible high structural resolution of hydrogen/deuterium-exchange mass spectrometry data". In: *Analytical Chemistry* 94.11, pp. 4557–4564.

Selevsek, Nathalie et al. (2015). "Reproducible and consistent quantification of the Saccharomyces cerevisiae proteome by SWATH-mass spectrometry". In: *Molecular & Cellular Proteomics* 14.3, pp. 739–749.

Serang, Oliver et al. (2012). "Recognizing uncertainty increases robustness and reproducibility of mass spectrometry-based protein inferences". In: *Journal of Proteome Research* 11.12, pp. 5586–5591.

Shanno, David F (1970). "Conditioning of quasi-Newton methods for function minimization". In: *Mathematics of computation* 24.111, pp. 647–656.

Shen, Xinyue et al. (2017). "Disciplined multi-convex programming". In: *2017 29th Chinese Control and Decision Conference (CCDC)*. IEEE, pp. 895–900.

Shuken, Steven R (2023). "An introduction to mass spectrometry-based proteomics". In: *Journal of Proteome Research* 22.7, pp. 2151–2171.

Sivanich, Michael K. et al. (2022). "Recent advances in isobaric labeling and applications in quantitative proteomics". In: *Proteomics* 22.19-20, p. 2100256. DOI: https://doi.org/10.1002/pmic.202100256. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/pmic.202100256. URL: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/pmic.202100256.

Skinner, Simon P et al. (2019). "Estimating constraints for protection factors from HDX-MS data". In: *Biophysical Journal* 116.7, pp. 1194–1203.

Smith, Rob and Annika R Tostengard (2020). "Quantitative evaluation of ion chromatogram extraction algorithms". In: *Journal of proteome research* 19.5, pp. 1953–1964.

Spivak, Marina et al. (2012). "Direct maximization of protein identifications from tandem mass spectra". In: *Molecular & Cellular Proteomics* 11.2.

Staniak, Mateusz et al. (Jan. 2025). "Relative quantification of proteins and post-translational modifications in proteomic experiments with shared peptides: a weight-based approach". In: *Bioinformatics* 41.3, btaf046. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btaf046.

Stastna, Miroslava and Jennifer E Van Eyk (2012). "Analysis of protein isoforms: can we do it better?" In: *Proteomics* 12.19-20, pp. 2937–2948.

Steen, H and M Mann (2004). "The abc's (ans xyz's) of peptide sequencing". In: *Nature*.

Sticker, Adriaan et al. (2020). "Robust summarization and inference in proteome-wide label-free quantification". In: *Molecular & Cellular Proteomics* 19.7, pp. 1209–1219.

Stofella, Michele, Antonio Grimaldi, et al. (2024). "Computational Tools for Hydrogen–Deuterium Exchange Mass Spectrometry Data Analysis". In: *Chemical Reviews* 124.21. PMID: 39481095, pp. 12242–12263. DOI: 10.1021/acs.chemrev.4c00438. eprint: https://doi.org/10.1021/acs.chemrev.4c00438. URL: https://doi.org/10.1021/acs.chemrev.4c00438.

Stofella, Michele, Simon P Skinner, et al. (2022). "High-resolution hydrogen–deuterium protection factors from sparse mass spectrometry data validated by nuclear magnetic resonance measurements". In: *Journal of the American Society for Mass Spectrometry* 33.5, pp. 813–822.

Storey, John D and Robert Tibshirani (2003). "Statistical significance for genomewide studies". In: *Proceedings of the National Academy of Sciences* 100.16, pp. 9440–9445.

Surinova, Silvia et al. (2013). "Automated selected reaction monitoring data analysis workflow for large-scale targeted proteomic studies". In: *Nature protocols* 8.8, pp. 1602–1619.

Terry M. Therneau and Patricia M. Grambsch (2000). *Modeling Survival Data: Extending the Cox Model*. New York: Springer. ISBN: 0-387-98784-3.

The, Matthew et al. (2018). "A protein standard that emulates homology for the characterization of protein inference algorithms". In: *Journal of Proteome Research* 17.5, pp. 1879–1886.

Therneau, Terry M (2023). *A Package for Survival Analysis in R*. R package version 3.5-3. URL: https://CRAN.R-project.org/package=survival.

Truong, Patrick, Matthew The, and Lukas Käll (2023). "Triqler for protein summarization of data from data-independent acquisition mass spectrometry". In: *Journal of Proteome Research* 22.4, pp. 1359–1366.

Tsai, Tsung-Heng et al. (2020). "Selection of features with consistent profiles improves relative protein quantification in mass spectrometry experiments". In: *Molecular & Cellular Proteomics* 19.6, pp. 944–959.

Tsiamis, Vasileios and Veit Schwämmle (Mar. 2022). "VIQoR: a web service for visually supervised protein inference and protein quantification". In: *Bioinformatics* 38.10, pp. 2757–2764. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac182. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/10/2757/43705175/btac182.pdf. URL: https://doi.org/10.1093/bioinformatics/btac182.

Tyanova, Stefka, Tikira Temu, and Juergen Cox (2016). "The MaxQuant computational platform for mass spectrometry-based shotgun proteomics". In: *Nature Protocols* 11.12, pp. 2301–2319.

van der Loo, MPJ (2020). "A method for deriving information from running R code". In: *The R Journal*, Accepted for publication. URL: https://arxiv.org/abs/2002.07472.

Vizcaíno, Juan A et al. (2014). "ProteomeXchange provides globally coordinated proteomics data submission and dissemination". In: *Nature biotechnology* 32.3, pp. 223–226.

Volmer, Dietrich and Andrew Leslie (2007). "Dealing with the masses: a tutorial on accurate masses, mass 32 uncertainties, and mass defects". In: *Spectroscopy*.

Walters, Benjamin T et al. (2012). "Minimizing back exchange in the hydrogen exchange-mass spectrometry experiment". In: *Journal of the American Society for Mass Spectrometry* 23.12, pp. 2132–2139.

Wickham, Hadley (2014). "Tidy data". In: *Journal of statistical software* 59, pp. 1–23.

Wickham, Hadley et al. (2019). "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43, p. 1686. DOI: 10.21105/joss.01686.

Wilmarth, Phil (2020). *https://pwilmart.github.io/blog/2020/09/19/shotgun-quantification-part2*. URL: https://pwilmart.github.io/blog/2020/09/19/shotgun-quantification-part2 (visited on 03/18/2024).

Wilson, RJ (1972). *Introduction to graph theory*. Edinburgh:[sn].

Wu, Anthony and Olga Vitek (2025). *MSstatsBioNet: Network Analysis for MS-based Proteomics Experiments*. https://github.com/Vitek-Lab/MSstatsBioNet.

Xu, Yingrong et al. (2021). "A Comparison of Two Stability Proteomics Methods for Drug Target Identification in OnePot 2D Format". In: *ACS Chemical Biology* 16.8. PMID: 34374519, pp. 1445–1455. DOI: 10.1021/acschembio.1c00317. eprint: https://doi.org/10.1021/acschembio.1c00317. URL: https://doi.org/10.1021/acschembio.1c00317.

Zaharia, Matei et al. (Oct. 2016). "Apache Spark: a unified engine for big data processing". In: *Commun. ACM* 59.11, pp. 56–65. ISSN: 0001-0782. DOI: 10.1145/2934664. URL: https://doi.org/10.1145/2934664.

Zhang, Bo et al. (2017). "Covariation of peptide abundances accurately reflects protein concentration differences". In: *Molecular & Cellular Proteomics* 16.5, pp. 936–948.

Zhang, Zhongqi (2020). "Complete Extraction of Protein Dynamics Information in Hydrogen/Deuterium Exchange Mass Spectrometry Data". In: *Analytical Chemistry* 92.9. PMID: 32309934, pp. 6486–6494. DOI: 10.1021/acs.analchem.9b05724. eprint: https://doi.org/10.1021/acs.analchem.9b05724. URL: https://doi.org/10.1021/acs.analchem.9b05724.

Zhang, Zhongqi, Aming Zhang, and Gang Xiao (2012). "Improved protein hydrogen/deuterium exchange mass spectrometry platform with fully automated data processing". In: *Analytical chemistry* 84.11, pp. 4942–4949.

Zhu, Qi, Dirk Valkenborg, and Tomasz Burzykowski (2010). "Markov-chain-based heteroscedastic regression model for the analysis of high-resolution enzymatically 18O-labeled mass spectra". In: *Journal of proteome research* 9.5, pp. 2669–2677.

Zhu, Yafeng et al. (2020). "DEqMS: a method for accurate variance estimation in differential protein expression analysis". In: *Molecular & Cellular Proteomics* 19.6, pp. 1047–1057.

Zhuang, Guanglei et al. (2013). "Phosphoproteomic analysis implicates the mTORC2-FoxO1 axis in VEGF signaling and feedback activation of receptor tyrosine kinases". In: *Science signaling* 6.271, ra25–ra25.