Identification of variant peptides using mass spectrometry

Dissertation thesis

Miroslav Hruška

Department of Computer Science
Faculty of Science
Palacký University Olomouc

Olomouc, 2022

Author
Miroslav Hruška
Department of Computer Science
Faculty of Science
Palacký University Olomouc
17. listopadu 1192/12
779 00 Olomouc
Czech Republic
hruska.miro@gmail.com

Supervisor
doc. Ing. Petr Sosík, Dr.

Keywords
peptide detection, mass spectrometry, peptide prior probability, variant peptides, computational proteomics

Declaration
I hereby declare that the thesis has been composed solely by myself and that work contained herein is my own except where explicitly stated otherwise. Several parts of the work are an outcome of a collaboration with my colleagues from the Institute of Molecular and Translational Medicine of Palacky University Olomouc, notably: Marián Hajdúch, Petr Džubák, Dušan Holub, and Lakshman Varanasi.

Miroslav Hruška

Abstract

Detection of peptides from mass spectrometric data lies at the core of computational proteomics. In our research, we focus on detecting *variant* peptides—a large class of unlikely but highly-informative peptides with rich biomedical applications. Common peptide detection methods typically result in a small number of variant peptides detected, along with a high rate of false positives, hence preventing utilizing the full potential of variant peptides in follow-up applications. Herein, we argue that one reason for the inefficient detection is the neglect of peptide prior probabilities—the probabilities of the presence of the peptides in the sample before the mass spectrometric analysis itself. In accordance, we develop theoretical and algorithmic methods based on Bayes' theorem to probabilistically incorporate peptide prior probabilities into detection. Afterward, we show that our methods derive accurate error rates under multiple circumstances and substantially improve the detection performance over several popular peptide variant detection algorithms. Finally, we develop computational methods that process the detected peptide variants and illustrate their applications in medicine, research reproducibility, and forensics.

Acknowledgments

The journey behind this research was a rather complex one. During this period, the people closest to me have witnessed it firsthand, and because the circumstances were, at times, not particularly easy, I want to herein express my gratitude. I want to thank Emilia Wądzyńska for supporting me before and while writing the thesis, especially during the more demanding situations. I want to thank Candace Hathaway for giving me a different perspective on the purpose of the work—it came at the right moment. I want to thank Michal Cisárik for repeated encouragement and for seeing the bigger picture of the research—the one that I often lost along the way. I want to thank Lakshman Varanasi for numerous professional and friendly discussions—no matter what they were about, they were always uplifting. Last but not least, I want to thank my parents, Vladimir and Vlasta, for their general support.

From a more professional perspective, I want to thank my supervisors for their help. First, I want to thank Petr Sosík for his guidance in writing the thesis and for helping me in various ways during my doctoral studies. Second, I want to thank Marián Hajdúch, my supervisor at the Institute of Molecular and Translational Medicine, for his ideas and guidance in the research.

This work was supported in parts by the Ministry of Education, Youth and Sports of the Czech Republic (CZ.02.1.01/0.0/0.0/16_019/0000868, CZ.01.1.02/0.0/0.0/16_084/0010360, LM2015064, LM2015047, LM2018130, LM2018131), Technology Agency of the Czech Republic (TE02000058, TN01000013), Ministry of Health of Czech Republic (NV16-32318A, NV16-32302A), and the European Union's Horizon 2020 (EOSC-Life Grant agreement no. 824087).

Preface

The thesis deals with probabilistic detection of variant peptides from data measured using modern mass spectrometers. In our research, we specifically investigate the commonly overlooked notion of peptide prior probability and argue that it plays a significant role in peptide detection. In accordance, we develop several models of peptide prior probabilities to capture the *a priori* knowledge about an experiment and develop computational methods based on Bayes' theorem to utilize such knowledge in peptide detection. Notably, the use of peptide prior probabilities is orthogonal to many developments in the field, allowing their natural integration with existing peptide detection approaches.

The content of the thesis is in parts based on the following articles:

[1] Hruska, M. & Holub, D. A complete search of combinatorial peptide library greatly benefited from probabilistic incorporation of prior knowledge. *International Journal of Mass Spectrometry* **471,** 116723. ISSN: 13873806 (Jan. 2022)

[2] Hruska, M. & Holub, D. Evaluation of an integrative Bayesian peptide detection approach on a combinatorial peptide library. *European Journal of Mass Spectrometry,* 146906672110667. ISSN: 1469-0667 (Jan. 2022)

[3] Hruska, M. *et al.* Deep probabilistic search detects protein variants in shotgun proteomics data independently of DNA/mRNA sequencing. *eLife* (Submitted)

In [1], we introduced a Bayesian method for calculating posterior probabilities of peptides in complete searches of fragment mass spectra. Therein, we investigated detection performance for various prior distributions and scoring metrics. The core of the approach is presented in the sections 3.1.3, and its extended adaptation for peptide detection in section 4.4.1.2. Finally, several results from the article are presented in the section 5.1.

In [2], we extended the Bayesian model to integrate additional match-based models applicable to peptide detection while considering more involved peptide prior probability models. Therein, we also discussed a more computationally tractable *tail-complete* search strategy and showed that the error rates derived using this strategy are highly similar to those calculated from the complete search. Partial results from the article are presented in the section 5.1.

In [3], we investigated the detection of peptide variants in several large-scale computational proteomics datasets. Therein, we developed a more realistic model of peptide prior probabilities, which we described here in an extended form in the section 3.2.4. The theoretical and computational methods related to this work are presented in sections 3.1, 3.2, 4.4.2, and 4.3. Finally, several results of the work are presented in section 5.3.

Besides the previous works, we have also the following European Patent application:

[4] Hruska, M. *et al. Method of identification of entities from mass spectra.* European Patent Application (EP 18184710.4), 2018

The patent application [4] protects the detection of variant peptides using methods developed in [3], and presents several downstream applications of these methods.

Contents

Chapter 1

Introduction

The thesis deals with the computational detection of *peptides*, molecules of a certain linear structure, from their data measured using mass spectrometry. In particular, we develop mathematical and computational methods allowing probabilistic detection of a peptide from its *fragment mass spectrum*—measurement of its mass and the masses of its fragments (**Fig. 1.1**). Although computational detection of peptides is a central and routine procedure within the field of *computational proteomics*, existing methods are often inapplicable for detecting *variant peptides*—a large class of highly-informative but unlikely peptides. Such inapplicability is of concern because the detection of variant peptides has rich biomedical applications and might play a crucial role in diagnosing severe health disorders, including cancers.

Even though we developed these methods primarily for peptide detection, the core methods remain rather general and serve to probabilistically analyze candidate causes of observed data using both the candidate's agreement with the data and its prior probability. Importantly, we developed these methods with a particular intention—to allow reliable identification of unlikely causes. Although this posed relatively minor problems theoretically, detection of unlikely causes can translate to substantial challenges in practice, which was also evident in our applications. For instance, the detection of variant peptides using our methods sometimes requires testing up to million candidates per fragment mass spectrum, which in turn requires corresponding algorithmic developments. Our research thus also illustrates the rather non-trivial process of translating the theoretical approach for detecting unlikely causes into an applied one for detecting unlikely peptides.

Finally, we present the importance of variant peptides in downstream applications and investigate the possibility to computationally verify the correctness of their detection, a problem in itself. Altogether, the thesis presents several theoretical and computational methods, shows their adaptation to detecting variant peptides from fragment mass spectra, and illustrates their follow-up applications.

Research aims Having introduced the core topics, let us now specify our research aims. Overall, our primary aim is the creation and implementation of fast computational methods for reliable detection of variant peptides from fragment mass spectra. In doing so, we develop a theoretical approach for probabilistic analysis of candidate causes of observed data and then translate it into the problem of peptide detection within computational proteomics.

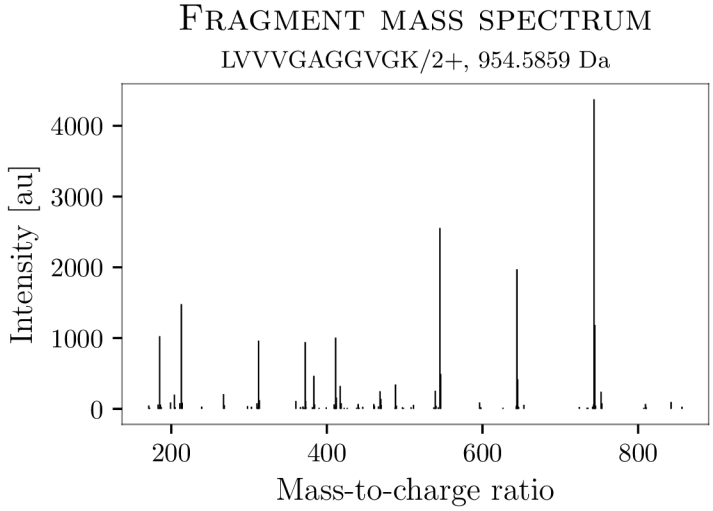


Figure 1.1: An example of a mass spectrum.

The figure depicts an observed fragment mass spectrum of a doubly charged peptide LVVVGAGGVGK, with a measured parental mass of approximately 954.5859 Dalton.

1.1 Research problem

Let us now introduce the research problem that we aim to address in more detail. In particular, we study the computational detection of human variant peptides from *typical* mass spectrometric data. Even though a variety of computational peptide detection methods exist [5], detection of variant peptides is still not a routine procedure and often results in largely incorrect error rate estimates in typical experiments [6–9]. For non-typical experiments, the detection of variant peptides is approached by first performing additional biochemical and computational analyses [9]. In particular, the researchers obtain the sample's DNA or mRNA, derive a small set of expected variant peptides, and identify the mass spectra against such a set of peptides. Although the approach is generally reliable [7, 9], it does not apply to typical experiments because most of them do not have the corresponding DNA or mRNA data. Furthermore, the approach is more resource-consuming and provides a potentially biased view of peptide variation. Consequently, detection of variant peptides without performing an additional biochemical analysis is desirable.

As computational detection of variant peptides lies at the core of our research aims, let us briefly elaborate on some of the problems that affect, in general, the detection of unlikely peptides. One of the fundamental problems is that the scoring metrics relating a peptide with a fragment mass spectrum are not powerful enough. For instance, selecting a peptide with a maximal agreement with the spectrum among all possible peptides is generally inadequate—such a peptide is usually not the correct one [1]. Still, the correct peptides often do have an agreement close to the maximal one, allowing, for instance, the use of statistical significance of such an agreement to identify the correct peptides [5]. However, although such an approach works reasonably well for likely peptides, it can be largely insufficient for the unlikely ones [9]. In our research, we argue that one of the reasons for such behavior is the neglect of *prior probability* of a peptide—the probability that a randomly selected peptide from a sample is the peptide of interest [1, 2], which particularly affects the detection of unlikely peptides. The inability to associate the correct peptide to a mass spectrum based only on its agreement is thus a comparably

small problem for likely peptides but can become a serious problem otherwise [7, 8].

Research questions To shed light on the research problem at hand, we decomposed the research problem into more manageable subproblems. We formulated the following research questions that we aim to answer in our research:

- Q_1 How effective are the computational peptide detection methods in detecting variant peptides?
- Q_2 What factors impact the precision and recall of the variant peptide detection?
- Q_3 What factors impact the detection of individual, i.e., sequence-specific, variant peptides?
- Q_4 What are the ways to validate variant peptide detection methods?
- Q_5 To what degree do peptide prior probabilities influence peptide detection?

The answers to the individual research questions thus provide grounds for resolving the central research problem.

1.2 Research contribution

The thesis contributes knowledge to multiple scientific fields—both theoretical and practical. On the theoretical side, the thesis develops novel theoretical and computational methods for computer science and bioinformatics, with insights, detailed approaches, and implementations in computational proteomics. On the practical side, the thesis presents novel computational methods for processing the detected peptide variants and shows their applications in medicine, forensics, and research reproducibility.

1.2.1 Computer science

The thesis contributes to computer science by developing theoretical methods for cause identification and more flexible data analysis. The first method allows assigning a maximal posterior probability ($\Pr_{\max}$) to a candidate cause of observed data while considering only prior probabilities of causes that have at least as good agreement with the data. In practice, the method can reanalyze causes identified using other approaches to remove provably unlikely causes—those with low $\Pr_{\max}$. For instance, one can utilize the method to filter out causes that had a statistically significant agreement with the data yet are unlikely correct. The method thus allows utilizing prior information in cause-identifying approaches and helps improve their precision.

As our second method, we develop a Bayesian model for probabilistically identifying a cause of observed data while considering *all* causes, their agreement with the data, and their prior probabilities. The method allows to derive posterior probabilities of all candidate causes and is thus applicable for standalone cause detection. Its application, however, depends on the problem domain because the need to consider all candidate causes might be hard to implement in practice—especially if the number of candidate causes is very high. Nevertheless, the Bayesian model allows probabilistic detection of causes, and the posterior probabilities provide guarantees over expected rates of correct identifications in the long run.

Besides the methods mentioned above, the thesis also contributes to computer science by implementing various general-purpose data analysis functions created along the way to resolve the original problems. These mainly include a functional programming library `fplib` and a library for analyzing tabular data (i.e., `pandas' DataFrame`), both implemented in `python`. The methods developed for resolving the primary problem have thus value on their own and contribute to more practical aspects of computer science.

1.2.2 Computational proteomics

The thesis contributes to computational proteomics by applying the computer-scientific methods to peptide detection, by developing algorithmic methods for fast matching of peptides with mass spectra, by recognizing the importance of peptide prior probabilities, and by implementing an open-source variant peptide detection system CLAIRE.

Peptide detection We show that the use of our method for calculating the maximal posterior probability substantially improved the detection performance of four popular variant peptide detection systems (section 5.2). The result thus provided evidence for the broad utility of the approach and showed the importance of peptide prior probabilities in peptide detection. Similarly, the application of our Bayesian method for calculating posterior probabilities resulted in estimates of probabilities that corresponded well with their expected long-term behavior (section 5.1), albeit only on a dataset restricted to 10^8 candidate peptides per spectrum. The method nevertheless estimated the error rates accurately, presenting a potential resolution of the problem with incorrect error rates affecting the detection of variant peptides [6–9].

Algorithmic developments To translate the theoretical methods into computational proteomics, we developed a fragment-indexation algorithm that allows fast calculation of agreement of multiple peptides with one spectrum (section 3.2.6). Although a similar algorithm was developed around the same time by another research group [10], its implementation, primary use, and purposes differ. Further, our two additional levels of indexing allow fast and memory-efficient matching of peptides against large peptide databases in the size of hundreds of gigabytes and, likely, much more. In turn, these allow testing a large number of hypothetical peptides per spectrum required to calculate the maximal posterior probability of a peptide ($\mathrm{Pr}_{\mathrm{max}}$) in typical computational proteomics circumstances.

Insights for detecting unlikely peptides In our research, we have also evidenced that unlikely technical artifacts start to show up when focusing exclusively on unlikely peptides. For instance, what might look like a fragment spectrum of a correctly detected unlikely peptide can turn out to be a fragment spectrum of a likely peptide but with the parental mass of the molecule incorrectly determined by the mass spectrometer's operating system. If neglecting such an unlikely possibility, some incorrect detections will slip through the analysis and worsen the precision of the detection method.

Open-source system CLAIRE Finally, we implemented the methods in a cross-platform open-source system CLAIRE applicable for detecting variant peptides, making it directly usable by

researchers in bioinformatics and computational proteomics. The online version of the system, along with its source code, is available at `https://claire.imtm.cz`.

1.2.3 Computational mass spectrometry

As mass spectrometry is the principal technology for the current proteomics research, the methods developed in the thesis also contribute to computational mass spectrometry. The problems with detecting unlikely peptides are likely to transfer directly to detecting other unlikely molecules in mass spectrometry, e.g., those in the related field of metabolomics. By a direct generalization of the findings, the thesis thus also contributes to the field of computational mass spectrometry.

1.2.4 Other fields

Besides the methods involved directly in peptide detection, we have also developed several methods that exploit the high informational content of biological variation (section 4.3). For instance, the application of CLAIRE to patients' tumor samples recognized tumors suitable for immunotherapy, showing a clinically-relevant application in biomedicine (section 5.3.1). Utilizing the expected prevalence of human variants, we developed a method to determine the origin of a proteomics sample by matching the detected peptide variants against a DNA database. The method assigns probabilities for each candidate DNA origin, and its application resolved identities of genetically-related members against DNA database—showing a potential application in forensics (section 5.3.3). Similarly, we have developed a more general variant-based method for calculating statistical significance between samples from large-scale datasets to detect the presence of genetic relationships. Our analysis of public datasets revealed several samples of different origins, showing an application in research reproducibility (section 5.3.2). Finally, we note that this list of applications is not exhaustive, and we refer the interested reader to our patent application for further details [4]. In summary, the follow-up methods have thus additional and relevant applications in respective scientific fields.

1.3 Overview of the thesis

Let us now provide a structural overview of the thesis. To get a visual idea of how the main sections of the thesis correspond to each other, we also present a graphical summary of its most relevant parts on **Fig. 1.2**. We now proceed by describing several higher-order sections in individual chapters.

In chapter 2, we review the literature relevant to our research problem. First, we briefly introduce the field of proteomics (2.1), situate our research, and describe peptide detection approaches, including those applicable for variant peptides (2.2). Afterward, we describe methods applicable for estimating error rates, along with their insufficiencies for detecting variant peptides (2.2.1). Finally, we review how researchers utilized peptide prior probabilities in peptide detection (2.2.2), revealing a gap that we aim to fill by our research.

In chapter 3, we develop the core methods for probabilistic analysis of causes of observed data (3.1), and translate the approach into computational proteomics (3.2). In the latter, we introduce several simple models of peptide prior probabilities (3.2.3), along with a more realistic prior model

applicable for peptide detection in typical circumstances (3.2.4). For the more realistic prior model, we develop an algorithm that enumerates all peptides that are above a specific minimal prior probability (3.2.5) and discuss some aspects of their storage. Afterward, we describe in detail an algorithm for fast calculation of spectral matches, along with its additional optimizations for large databases (3.2.6). Finally, we specify the calculation of $\mathrm{Pr}_{\mathrm{max}}$ for all candidate peptides of fragment spectra using our methods (section 3.2.7). Altogether, these methods allow us to build a highly-optimized deep database of peptides and their prior probabilities, allowing in-depth interpretation of fragment spectra.

In chapter 4, we describe less central methods that serve to provide additional grounds to answer our research questions. First, we briefly describe the proteomics and genomics datasets we employed for the computational analyses in the thesis (4.1). Afterward, we define performance metrics to externally evaluate the peptide detection performance—both in idealized conditions (4.2.1) and in typical ones applicable when we have DNA or mRNA data of the corresponding sample available (4.2.2). We then develop several mathematical and computational methods for downstream applications of detected variant peptides (4.3). Afterward, we provide detailed description of the software used in comparisons and several adjustments and extensions of our approach (4.4). We conclude the chapter with description of CLAIRE—our software system that implements the methods presented in thesis (4.5).

In chapter 5, we show direct and downstream applications of our methods. First, we focus on the peptide detection in idealized conditions of a combinatorial peptide library, which allows us to directly use our Bayesian model as the number of candidate peptides per spectrum is reasonably low (5.1). Therein, we show that the posterior probabilities calculated using our Bayesian model were close to their desired long-term behavior (5.1.1), and that even weak prior models substantially improved peptide detection (5.1.3). Afterward, we shift our focus to more typical experiments by analyzing 61 samples from NCI_{60} cancer cell line panel [11]. Therein, we show that the use of $\mathrm{Pr}_{\mathrm{max}}$ and its various extensions substantially improved detection of variant peptides when used for post-processing results of four variant peptide detection approaches (5.2). Finally, we illustrate the downstream applications in medicine, research reproducibility, and forensics (5.3).

In chapter 6, we discuss several aspects of the developed methods. In particular, we discuss the extension of our deep database search method for standalone detection of unlikely peptides (6.1), the improvements to the prior probability models (6.2), the utility of large databases in peptide detection (6.3), and further applications (6.4). We conclude the chapter by providing summarized answers to the research questions (6.5).

At the very end of the thesis, we conclude with a brief and conceptual summary of the most important findings. Therein, we argue that the value of peptide prior probabilities is underexplored in computational proteomics and their orthogonality to many approaches is likely to allow substantial improvements in peptide detection in the future.

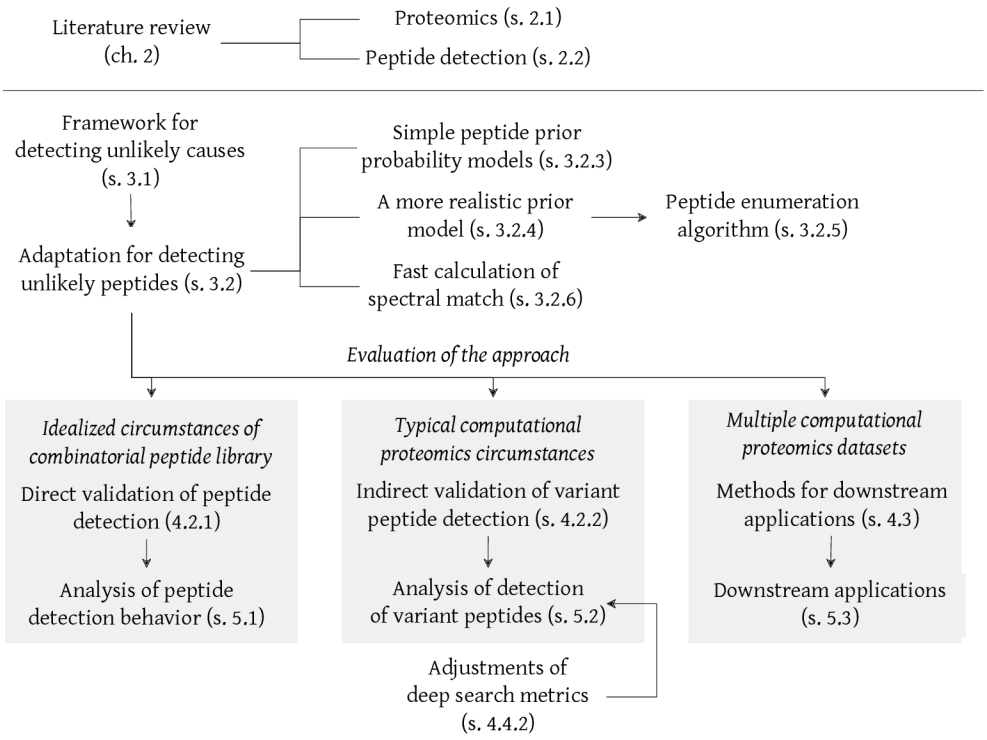


Figure 1.2: Overview of the most relevant parts of the thesis and their relationships.

Chapter 2

Literature review

This chapter reviews the literature relevant to our research problem. First, we briefly introduce the field of proteomics in section 2.1, wherein we describe the principal approaches for protein analysis and situate our research. In section 2.2, we narrow our focus to detecting peptides and particularly variant peptides—the primary interest of our research. We describe the available approaches, methods for the assignment of confidence measures, and the complications associated with detecting variant peptides. Further, we review how researchers employed prior probabilities in peptide detection, revealing a gap that we aim to fill by our research. Finally, we summarize the major points of the review in a concluding section 2.3.

2.1 Proteomics

Proteomics is an interdisciplinary field that investigates large-scale behavior of *proteins* and their interactions in complex biological systems [12, 13]. Proteins are biomolecules that, on the one hand, serve a structural role of being cellular building blocks and, on the other hand, perform a wide range of biological functions [14]. For instance, proteins known as *enzymes* accelerate the rates of chemical reactions, *transport proteins* allow selective transport of molecules across cells boundaries and *antibodies* act in immune responses [14–16]. Although the behavior of proteins follows from their higher-order structure [17], for purposes of their detection, we focus on their primary structure, i.e., a linear sequence of *amino acids*—their elementary components. Note that although the cellular machinery builds proteins using 20 canonical amino acids, the proteins can undergo many *post-translational modifications* (*PTMs*) and more than 600 types of such PTMs were categorized as of 2021 [18]. As a result, proteins are diverse molecules, and their intrinsic complexity also complicates their detection.

Modern proteomics approaches utilize *mass spectrometry* to detect proteins—an analytical technique that detects molecules based on the measurement of their mass spectra [5, 12, 19]. In a mass spectrometer, molecules are first ionized [20], allowing the device to influence them using electric and magnetic fields. Although the principles of operation vary greatly [21], let us briefly mention the rather simple design of a time-of-flight (*ToF*) instrument [22]. In ToF, the ionized molecules are accelerated by an electric field, fly through a field-free path and the time of their arrival at the detector is measured. Lighter molecules arrive earlier, the heavier ones later, allowing one to calculate the molecule's mass from the time of their flight, hence the name (note that the situation is more complicated due to multiply charged ions). For a

detailed description of various instrumental designs, we refer the reader to [21]. Nevertheless, as typical proteomics samples are rather complex, the molecules are first separated using *liquid chromatography* [12, 23], which allows their gradual introduction into the mass spectrometer over an adequate time period (ranging in hours for complex samples). As a result, modern proteomics experiments interface liquid chromatography (LC) to a mass spectrometer (MS), in a configuration commonly abbreviated as *LC/MS*.

Mass spectrometers are, in general, highly versatile devices [24], and allow multiple modes of operation and data acquisition. In our research, we focus on configuration for *shotgun proteomics* [12], a common experimental setup suitable also for detecting variant peptides. In shotgun proteomics, the proteins are first biochemically cut into *peptides*—short protein subsequences, which are then analyzed using LC/MS [25]. In addition, the measurement of mass spectra happens on two levels: MS^1 and MS^2. The MS^1 level measures the mass spectra of the intact ionized molecules, and such spectra are called *precursor spectra* and the individual ions as *precursor ions*. Afterward, precursor ions of a specific narrow mass range are isolated, fragmented, and masses of their fragments are measured on the MS^2 level, giving rise to *fragment spectra*. Although there are multiple precursor isolation strategies, we focus on one wherein a single precursor ion species is a target for fragmentation—such a strategy is called *Data-Dependent Acquisition* (*DDA*) [26]. In DDA, one can thus roughly assume that a single molecular species produced the fragment spectrum, and such spectrum then constitutes the primary data from which we aim to detect the peptide that produced it. Finally, once the fragment spectra are interpreted using a suitable computational method for peptide detection, the detected peptides are assigned to parental proteins [27], concluding the detection part of the analysis.

2.2 Peptide detection

The detection of peptides from fragment spectra lies at the core of computational proteomics [5, 28]. In principle, there are two major approaches for peptide detection and their various hybridizations: a database search and *de novo* sequencing. In a database search [29–31], fragment spectra are matched against predicted fragment spectra of peptides from an appropriate database (e.g., reference proteins of a studied organism). Similarly, one can match the fragment spectra against known fragment spectra of peptides [32, 33], which is generally more discriminative but spectral libraries are limited in their extent. In *de novo* sequencing, the fragment spectra are interpreted directly, without the use of a sequence database—utilizing just the masses of amino acids and, potentially, their various modifications [34, 35]. Even though *de novo* sequencing is fast [36], and allows large-scale modification search [37], it achieves only around 35 % agreement with a database search, making it impractical for routine analyses [34]. Hybrid approaches typically utilize partial *de novo* sequencing to extract *sequence tags* [38, 39], short sequences of amino acids (e.g., 3–6 residues), which filter out unviable peptide candidates when matching against peptide database. Because our research concerns the detection of peptides in typical circumstances, we will focus on a database search and some of its hybrid versions.

Database search is the most popular method for peptide detection [40], with more than 30 search engines available in 2017 [41]. Overall, the database search engines work as follows. For each fragment spectrum, the search engine selects peptides of appropriate precursor mass from

a supplied database and calculates the matching score of each peptide's theoretical spectrum with the measured fragment spectrum [29–31]. Usually, only the peptide with the best match per spectrum is retained [5], and each such assignment of a peptide to spectrum is then called a *peptide-spectrum match* (*PSM*). Once fragment spectra are interpreted, the PSMs undergo post-processing to establish confidence measures [42, 43], and these measures are generally reliable as long as one is interested in detecting reference peptides of an organism but are problematic otherwise [7–9].

We now shift our focus to detecting variant peptides, wherein we also review the applicable hybrid peptide detection approaches. The most common approach for detecting variant peptides is the so-called *sample-specific database search* employed in proteogenomics [6, 9, 44], a field studying the interplay of genomics and proteomics. Therein, the researchers first sequence DNA or mRNA of the sample, construct a sample-specific protein database from the DNA/mRNA variants, and match the mass spectra against the protein database using any database search engine [45–47]. Although the approach successfully detects variant peptides [7, 9], the obvious disadvantage is the need to perform the DNA or mRNA sequencing, which makes the approach inapplicable to typical proteomics experiments. Furthermore, it is advised that the researchers incorporate only highly confident genomic events in the sample-specific database because the detection of variant peptides tends to result in much higher than estimated error rates [6]. Nonetheless, the sample-specific database approach is well established and has shown multiple biomedical applications [6, 48].

To allow DNA/mRNA-independent detection, one can perform the database searches against a peptide database constructed from a globally observed DNA/mRNA variants [3, 49, 50], and we refer to such a search as *global peptide-variant* (*GPV*) search. GPV search, however, results in high rates of false positives—even at stringent confidence criteria [7, 8]. A partial reason for this behavior is that many peptides are *homologous* to the variant peptides [9], meaning they are of similar sequence and fragment spectra. However, as we argue in [1, 2], that itself is only a partial explanation. The critical and often neglected fact is that the variant peptides are unlikely a *priori*—as a result, the interpretation of fragment spectra by these homologous peptides is generally more preferable. Consistent with our argumentation, restricting the GPV search to a limited number of curated variants that are likely *a priori* allows their confident detection [51], albeit at the cost of low sensitivity. Further, although the GPV search generally results in high error rates [7–9], we have shown that a deep Bayesian re-analysis of claimed variant peptides makes the approach reliable [3], and we provide further evidence in the thesis.

Another option for detecting variant peptides is to use some of the database-guided and hybrid detection methods [29, 30, 52, 53]. The most straightforward possibility is to use the exhaustive substitution of amino acids per peptide [29, 30]. For instance, the *point mutation search* in X!Tandem [29] or the *error-tolerant search* in MASCOT [30] match the fragment spectra against peptides with amino acid substitutions incorporated into the peptides from the supplied search database. However, such approaches substantially increase the search space, and any detection method based on the statistical significance of a spectral match quickly loses sensitivity [9], resulting in a rather small number of variant peptides detected. To improve on the situation, some approaches, e.g., BICEPS [54] or TagGraph [55], utilize sequence tags to prefilter the search space to candidate peptides that match the sequence tag—a method that can

decrease the search space over several orders of magnitude depending on the length of the tag [56]. Nevertheless, in our former work [1], we have shown that although sequence tags substantially improve peptide detection, they provide only a limited advantage for discriminating homologous peptides—unless the sequence tags are very long and of high certainty. One can also resort to approaches that aim to solve a more general problem—detecting peptides shifted by a mass of unknown modification. For instance, the open search approach [57] implemented in the fast MSFragger algorithm [10], utilizes a standard database search against very wide precursor mass window (e.g., 500 Da instead of typical range on the order of $\approx$ 0.01 Da), allowing detection of peptides with modifications of unknown masses. Some less common approaches include a pair-wise comparison of measured fragment spectra to detect mass shifts corresponding to amino acid variants [58], or open searches against spectral libraries [52]. Afterward, the mass shifts are localized, and if the mass difference corresponds to an amino acid substitution, it is interpreted as such. Nevertheless, most of these approaches were developed for the detection of PTMs, and their large-scale validation on variant peptides is missing, complicating the establishment of their applicability for this purpose.

2.2.1 Assignment of confidence measures

A crucial aspect of peptide detection using database searches is the assignment of confidence measures [5]. The most popular method is the target-decoy approach (TDA) [59], which aims to control the False Discovery Rate (FDR) of the peptide detection results [60]. FDR has an obvious interpretation: for instance, setting an FDR threshold of 1% should ideally result in 1% of incorrect peptide-spectrum matches. Note, however, that the FDR refers to a *set* of PSMs instead of a single PSM, and is thus inadequate to establish the confidence of a particular PSM. Nevertheless, TDA also allows calculating posterior error probabilities (PEP) for individual PSMs [60], which should ideally express the probability that a peptide-spectrum match is incorrect. In TDA, a search engine matches the fragment spectra against two databases: the target database containing the expected peptides and a decoy database containing reversed or shuffled peptide sequences that are assumed to be incorrect [61]. Then, depending on the number of decoy PSMs at a given score, posterior error probabilities of PSMs are calculated, and a set of peptides with the desired FDR is retained. Further, one can also separate the target and decoy matches using multiple criteria besides the spectral match (e.g., peptide length, or a deviation from the expected mass), and the popular Percolator software uses more than 40 features employed in a support vector machine for the purpose [62]. Nevertheless, although TDA is popular [40], its use is also controversial [63], and some researchers advised against it because its error estimates can be largely incorrect [64, 65]. In line with this, we have shown that the error rates calculated using Percolator can be largely underestimated when using certain types of homologous databases [2]—a situation that resembles detecting variant peptides. In accordance, other researchers have shown that the TDA approach is inadequate for detecting variant peptides [7, 8, 66]. Thus even though the TDA approach is popular and applicable in certain circumstances, it is not universal, as illustrated by incorrectly calculated error rates on multiple occasions—including, notably, the detection of variant peptides.

An alternative, decoy-free approach to establish error rates employs Bayesian mixture models to categorize PSMs into correct and incorrect detections [5]. The approach is implemented in the

PeptideProphet post-processing system [42] which forms an integral part of the popular Trans-Proteomic Pipeline [67]. Employing the expectation-maximization algorithm, PeptideProphet iteratively updates the posterior probabilities that individual PSMs are correct while adjusting the parameters of score distributions for correct and incorrect PSMs; the process then continues until convergence. Further, the posterior probabilities calculated using PeptideProphet can be aggregated in meta-processing system iProphet [68], which integrates PSMs from multiple search engines and fragment spectra, allowing to improve the overall detection performance. Nevertheless, our previous analyses had shown that PeptideProphet is problematic for detecting homologous peptides [2], and can assign largely inaccurate posterior probabilities, even though we found that its behavior was substantially better than that of Percolator [43] based on the target-decoy approach.

Another approach to calculate error rates is to directly calculate spectrum-specific confidence measures such as p-values and E-values of a single PSM [65, 69]. The p-value p for a PSM with a score s equals the probability that a randomly selected peptide has a score at least as good as s. In turn, the E-value e accounts for multiple testing [60], and, for a given size n of database, refers to the expected number of peptides with a score at least as good as s ($e = p \cdot n$). Ideally, the E-value should then reflect the expected number of false hits in search results [70]. Several search engines [29, 30, 71] provide estimates of E-values or p-values, allowing to assign decoy-free spectrum-specific confidence measures. Furthermore, if the scoring metric is additive, one can also quickly calculate the score histogram of all theoretical peptides *exactly* using dynamic programming, and thus also obtain exact p-values and E-values [31, 69]. However, the fact that a peptide-spectrum match is highly significant does not necessarily mean that the peptide was identified correctly. For instance, we have shown that even exact E-Values can be largely misleading in database searches if peptides more likely *a priori* are not included in such a search [1]—a situation that often happens when detecting variant peptides. As a result, for some types of database searches—including global peptide-variant searches—these confidence measures do not adequately capture the notion that a peptide was detected correctly. In summary, an accurate calculation of the probability that a peptide was detected correctly, applicable to broad experimental circumstances and for discrimination of homologous peptides was not yet conclusively established in computational proteomics.

2.2.2 The use of peptide prior probabilities

In our former articles [1, 2], we gave evidence that one of the reasons for the discrepancy between calculated error rates and the true error rates in database searches is the neglect of *peptide prior probabilities*. By prior probability of a peptide p, we mean the probability that a randomly selected peptide molecule from a sample of interest is the peptide p [1, 3]. For instance, suppose selecting a random peptide molecule from a shotgun proteomics sample of a random human. In general, it is much more likely that the selected peptide is a reference peptide—a peptide present among the vast majority of humans—compared to a rare peptide variant present in a tiny fraction of the human population. From a different perspective, the spectral match metrics are, by far, not discriminative enough to uniquely detect the best peptide among all theoretical candidates at a given mass of precursor [1], which also translates to the low practical efficiency of *de novo* sequencing [34]. As a result, we argued that the high dynamic range of peptide

prior probabilities plays a substantial role in peptide detection [1, 2], evident especially when detecting peptides unlikely *a priori*—such as variant peptides. Further, we have shown that the use of our Bayesian approach allows accurate estimation of posterior error probabilities in highly-homologous searches of combinatorial peptide library [1, 2], and also allows detailed probabilistic modeling of prior knowledge.

The use of peptide prior probabilities in shotgun proteomics, nevertheless, remains rather marginal. As an early example from 2002, the ProbID algorithm [72] employed a prior probability model categorizing peptides into three categories based on their conformance to the expected peptide-cutting pattern (unexpected, partially expected, and fully expected). The peptide prior probabilities, however, can be modeled in a sequence-dependent manner and thus be much more granular—potentially assigning a unique prior probability to every single peptide depending on its detailed characteristics [2]. In this respect, the Paragon algorithm [73] uses more granular peptide prior probabilities as peptide hypothesis probabilities to reduce the search space but does *not* utilize them in the scoring itself. Thus, in the Paragon algorithm, if some reference peptide and a rare variant peptide have the same spectral match, both are considered equally likely—this, however, does not correspond to our intuition that the reference peptide is indeed more likely (and often substantially). On the other hand, the Bayesian approach BICEPS [54] utilized prior probabilities and assigned penalties to non-reference peptides, capturing the notion that peptides that are less likely *a priori* require more evidence for their correct detection. However, BICEPS considers only a very small number of potential post-translational modifications, many of which are more likely *a priori* than the nucleotide change resulting in a variant peptide. As we have shown previously [1, 2], incomplete database searches in which peptides more likely *a priori* are not included in the search are prone to substantial errors in establishing error rates.

Furthermore, none of the approaches considered the important fact that the prior probabilities of *individual* variant peptides also range over many orders of magnitude. For instance, the dbSNP [74] and ExAC [75] databases indicate that the prevalence of DNA/mRNA variants ranges at least over six orders of magnitude. Thus, it is reasonable to expect that the prior probabilities of the most likely class of variant peptides—those resulting from a single nucleotide variant—varies similarly. As a result, criteria for detecting frequent variant peptides, e.g., those present in 10% of humans, are very unlikely to be sufficient for detecting rare variants estimated to be present in one human per million. Further, the differences are even more pronounced as some variant peptides might be present in a subpopulation of cells, thus further lowering their prior probabilities [3].

Overall, our research thus aims to fill the gap by thoroughly investigating the role and the importance of peptide prior probabilities in peptide detection. Finally, we note that utilizing a proper peptide prior probability model is likely to improve any peptide detection approach and allows researchers to also independently focus on what is known about the sample in advance.

2.3 Conclusions

Herein, we summarize the principal conclusions of the literature review concerning the detection of peptide variants. Because our research deals with the detection of peptide variants in typical shotgun proteomics experiments, we focused on detection methods based on database search—*de*

novo sequencing is still impractical for ordinary circumstances [34, 35]. When detecting variant peptides using large databases of homologous peptides, database search methods often result in largely incorrect error rates [7–9] or low sensitivity [5], and we argued that one of the reasons is the neglect of peptide prior probabilities [1–3]. Peptide prior probabilities span—even for the most likely class of peptide variants—over six orders of magnitude [74, 75], and because spectral match is not discriminative enough to uniquely detect the correct peptide, such prior probabilities play a substantial role in peptide detection [2]. Further, using peptide prior probabilities, we can efficiently capture what is known about the sample in advance and detect variant peptides with accurately estimated error rates [1, 2]. As a result, our thesis expands on our previous results and aims to fill the research gap by investigating the relevance of prior probabilities in peptide detection—an underexplored topic in computational proteomics.

Chapter 3

Theoretical framework

The chapter deals with the theoretical core of the thesis and consists of two parts. In the first part, we develop theoretical methods for probabilistic analysis of causes of observed data, wherein we utilize prior probabilities of individual causes and their agreement with the data (section 3.1). In the second part, we develop a framework within computational proteomics that allows us to apply these theoretical methods to the detection of peptides from fragment mass spectra (section 3.2).

3.1 Computer science

The section focuses on a probabilistic analysis of candidate causes of observed data based on their agreement with the data and their prior probabilities. For this purpose, we introduce particular types of functions that have certain desirable probabilistic properties over the data of interest. First, these functions allow us to rather easily calculate an upper bound on the posterior probability of a cause—allowing one to reject unlikely causes. Second, using these functions, we formulate a Bayesian approach that calculates the posterior probabilities of all candidate causes for data of interest.

3.1.1 Preliminaries

We start by defining the key terms and concepts.

Notation In what follows, we will always work with a finite set of causes $\mathbb{C}$ and a set $\mathbb{D}$ representing the data. Further, the set $\mathbb{C}$ of causes will be complete in the sense that there will always be a single cause c that caused the data d.

Definition 1 (Cause-agreement function)**.** A cause-agreement function Θ is a function $\Theta\colon \mathbb{C} \times \mathbb{D} \mapsto \mathbb{X}$, where $\mathbb{X}$ is a finite totally-ordered set.

A particular cause-agreement function Θ thus defines the agreement between the cause and the data.

Notation Often, we will work with probabilities expressed in two forms, and we now explicitly state these forms to clarify their meaning. In the first form,

$$\Pr\big(\Theta(c,d) = a\big),$$

the expression denotes the probability that the cause c has the agreement a in the cause-agreement function Θ, wherein the probability is taken over data d. The second form,

$$\Pr\big(\Theta(c,d) = a \,|\, c\big),$$

denotes the conditional probability that the cause c has an agreement a in Θ, taken over data d, once we know that the cause c has occurred (i.e., c is the true cause). We now introduce the notion of a cause-agreement function that behaves in a certain desirable probabilistic way over the data of interest.

Definition 2 (Probabilistically-increasing cause-agreement function)**.** A cause-agreement function Θ is probabilistically-increasing if for all causes $c \in \mathbb{C}$, and agreements $a, b \in \mathbb{X}$, $a \leq b$, the following holds over data d:

$$\Pr\big(\Theta(c,d) = a \,|\, c\big) \leq \Pr\big(\Theta(c,d) = b \,|\, c\big),$$

and

$$\Pr\big(\Theta(c,d) = a\big) \geq \Pr\big(\Theta(c,d) = b\big).$$

Intuitively, a probabilistically-increasing cause-agreement function tends to assign a higher agreement to the true causes while doing the opposite for the random causes. For illustration, suppose that the agreement function Θ assigns only two agreements: high (1) and low (0). If the function generally assigns the high agreement to a rather small number of causes, often including the true one, and at the same time assigns the low agreement to a rather high number of causes, often excluding the true one, it is probabilistically increasing.

Definition 3. The cause-agreement function Θ is called true-cause normalized if for any causes $a, b \in \mathbb{C}$, and any agreement $x \in \mathbb{X}$, the following holds over data d:

$$\Pr\big(\Theta(a,d) = x \,|\, a\big) = \Pr\big(\Theta(b,d) = x \,|\, b\big).$$

The true-case normalized agreement function thus behaves such that it is equally likely to observe a particular agreement x with the data d if either cause caused the data.

Definition 4. The cause-agreement function Θ is called random-cause normalized if for any causes $a, b \in \mathbb{C}$, and any agreement $x \in \mathbb{X}$, the following holds over data d:

$$\Pr\big(\Theta(a,d) = x\big) = \Pr\big(\Theta(b,d) = x\big).$$

The random-cause normalized agreement thus behaves such that it is equally likely to observe a particular agreement at random for different causes.

Example Suppose a set of causes $\mathbb{C} = \{a, b\}$, and a set of agreements $\mathbb{X} = \{0, 1\}$. Now, suppose Θ behaves with respect to the data as follows:

$$\Pr\big(\Theta(a,d) = 0\big) = \Pr\big(\Theta(b,d) = 0\big) = \frac{3}{4},$$

$$\Pr\big(\Theta(a,d) = 1\big) = \Pr\big(\Theta(b,d) = 1\big) = \frac{1}{4}.$$

Such Θ is then random-cause normalized because the probability of a particular agreement $x \in \mathbb{X}$ is the same for all true causes. Further, suppose that Θ behaves as follows:

$$\Pr\big(\Theta(a,d) = 1 \,|\, a\big) = \Pr\big(\Theta(b,d) = 1 \,|\, b\big) = \frac{4}{5},$$

$$\Pr\big(\Theta(a,d) = 0 \,|\, a\big) = \Pr\big(\Theta(b,d) = 0 \,|\, b\big) = \frac{1}{5}.$$

Such a Θ is then true-cause normalized because the probability of a particular agreement $x \in \mathbb{X}$ is the same for all causes, once these causes happen. Furthermore, Θ is probabilistically increasing because for any $c \in \mathbb{C}$,

$$\Pr\big(\Theta(c,d) = 0 \,|\, c\big) \leq \Pr\big(\Theta(c,d) = 1 \,|\, c\big)$$

and

$$\Pr\big(\Theta(c,d) = 0\big) \geq \Pr\big(\Theta(c,d) = 1\big).$$

Notation In what follows, we will denote $\Pr(c)$ the prior probability of a cause c. Note that because we work with a complete set of exclusive causes $\mathbb{C}$, the sum of prior probabilities over the whole set will always equal one, thus

$$\sum_{c \in \mathbb{C}} \Pr(c) = 1.$$

Now, suppose a cause c and its prior probability $\Pr(c)$. Then, we denote $c^{\Pr}$ the set of causes that are at least as likely *a priori* as c, thus

$$c^{\Pr} = \{a \in \mathbb{C} \,|\, \Pr(c) \leq \Pr(a)\}.$$

Now suppose a cause c, data d, and a cause-agreement function Θ. Let us denote c^{Θ_d} the set of all causes that have at least as high agreement with d as c, thus

$$c^{\Theta_d} = \{a \in \mathbb{C} \,|\, \Theta(c,d) \leq \Theta(a,d)\}.$$

Finally, let us denote $c^\star$ the set of *at-least-as-good causes* as c both in terms of agreement and prior probability, thus

$$c^\star = c^{\Pr} \cap c^{\Theta_d},$$

where the $\Pr, \Theta$, and d are assumed to be clear from the context. With these preliminary definitions, we now turn to the probabilistic analysis of individual causes.

3.1.2 Calculation of maximal posterior probability ($\Pr_{\max}$)

Herein, we establish upper bounds on the maximal posterior probability of a candidate cause given prior probabilities of all at-least-as-good causes. The primary reason for calculating such bounds is to analyze causes identified using other approaches (e.g., using statistical significance of the agreement). In practice, such analysis allows rejecting causes whose posterior probabilities are low once we take the prior probabilities of causes into account.

Theorem 1 (Tighter bound on maximal posterior probability)**.** *Suppose data $d \in \mathbb{D}$, a candidate cause $c \in \mathbb{C}$, prior probabilities $\Pr(a)$ for all $a \in c^\star$, and a cause-agreement function Θ that is probabilistically increasing, true-cause normalized, and random-cause normalized. Then*

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) \leq \frac{\Pr(c)}{\sum\limits_{a \in c^\star} \Pr(a)}.$$

Proof. From Bayes Theorem, we have:

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) = \frac{\Pr\big(\Theta(c,d) = x \,|\, c\big) \cdot \Pr(c)}{\Pr\big(\Theta(c,d) = x\big)}.$$

For simplicity, we first prove the result for a special case when all the causes have the same agreement x with the data. Thus, suppose that $\Theta(c,d) = \Theta(a,d) = x$ for all $a \in c^\star$. Then

$$\frac{\Pr\big(c \,|\, \Theta(c,d) = x\big)}{\Pr\big(a \,|\, \Theta(a,d) = x\big)} = \frac{\Pr(c)}{\Pr(a)},$$

because the agreement is the same and Θ is true-cause and random-cause normalized. Now, the sum of posterior probabilities over all causes equals one when $c^\star = \mathbb{C}$. In such case, the following holds:

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) = \frac{\Pr(c)}{\sum\limits_{a} \Pr(a)}.$$

In general, the sum can be less than one, therefore

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) \leq \frac{\Pr(c)}{\sum\limits_{a} \Pr(a)}.$$

Now suppose $\Theta(a,d) = y \geq x$. Then

$$\frac{\Pr\big(c \,|\, \Theta(c,d) = x\big)}{\Pr\big(a \,|\, \Theta(a,d) = y\big)} \leq \frac{\Pr(c)}{\Pr(a)}$$

because

$$\Pr\big(\Theta(c,d) = x \,|\, c\big) \leq \Pr\big(\Theta(a,d) = y \,|\, a\big)$$

as Θ is probabilistically increasing and true-cause normalized, and

$$\Pr\big(\Theta(c,d) = x\big) \geq \Pr\big(\Theta(a,d) = y\big)$$

as Θ is probabilistically increasing and random-cause normalized. As $c^\star \subseteq \mathbb{C}$, the sum of posterior

probabilities over all at-least-as-good causes is at most one. It follows that

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) \leq \frac{\Pr(c)}{\sum\limits_{a} \Pr(a)}.$$

□

In other words, the posterior probability of a cause is at most the proportion of its prior probability among the at-least-as-good causes, for this particular type of cause-agreement functions.

Corollary 1 (Looser bound on maximal probability)**.**

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) \leq |c^{\star}|^{-1} \tag{3.1}$$

The theorem also provides a weaker result. Herein, the maximal posterior probability is at most the inverse of the number of at-least-as-good causes. Such a bound might be more meaningful in practice when one focuses on establishing the order of prior probabilities rather than their numerical values.

3.1.3 Calculation of posterior probability

Let us now turn to the calculation of posterior probabilities of candidate causes. Overall, we are interested in using the Bayes' Theorem in the following form:

$$\Pr\big(c \,|\, \Theta(c,d) = x\big) = \frac{\Pr\big(\Theta(c,d) = x \,|\, c\big) \cdot \Pr(c)}{\Pr\big(\Theta(c,d) = x\big)}. \tag{3.2}$$

Thus, given a particular agreement x of the cause c with the data d, we are interested in the posterior probability of the cause c. Similarly as we did previously, we will utilize the true-cause and random-cause normalized agreement functions such that it is straightforward to specify both $\Pr\big(\Theta(c,d) = x \,|\, c\big)$ and $\Pr\big(\Theta(c,d) = x\big)$ from a training dataset. Note that we intentionally use the Bayes' theorem to include $\Pr(c)$—the prior probability of the cause c because of our intended applications. In particular, we expect the prior probabilities to vary substantially, and we plan to model their values based on the available prior knowledge.

3.1.3.1 Model training

We now discuss how to specify the parts of the equation (3.2) to allow calculating the posterior probabilities. Suppose a training dataset of data $D = \langle d_1, \ldots, d_n \rangle$, corresponding true causes $C = \langle c_1, \ldots, c_n \rangle$, and an agreement function Θ that is both true-cause normalized and random-cause normalized.

Agreement for true causes

Because Θ is true-case normalized, we set the probability that a true cause c has an agreement x with the data d to the overall proportion of the agreement x for the true causes from the dataset

D, thus:

$$\Pr\big(\Theta(c,d) = x \,|\, c\big) = \frac{|\{i \in \mathbb{I} \,|\, \Theta(c_i, d_i) = x\}|}{n}, \tag{3.3}$$

where $\mathbb{I} = \{1, \ldots, n\}$ is the set of indexes over the dataset. Note that we do that because the true causes are interchangeable with respect to the agreement and the data for true-cause normalized cause-agreement functions.

Agreement for random causes

We now do the analogous for the behavior of random causes. Let $\Theta^d \colon \mathbb{X} \mapsto \mathbb{N}$ denote the distribution of agreement x with data d calculated using Θ over all candidate causes, thus:

$$\Theta^d(x) = \big|\big\{c \in \mathbb{C} \,|\, \Theta(c,d) = x\big\}\big|.$$

Now let us define the same but over the whole dataset:

$$\Theta^D(x) = \sum_{d \in D} \Theta^d(x).$$

Because Θ is random-cause normalized, we set the probability that a random cause c has an agreement x with the data d to the overall proportion of the agreement x in the dataset D, thus:

$$\Pr\big(\Theta(c,d) = x\big) = \frac{\Theta^D(x)}{\sum\limits_x \Theta^D(x)}. \tag{3.4}$$

The equations 3.3 and 3.4 then allow us to calculate the posterior probability using the equation 3.2 once we specify the prior probability of a particular cause.

Note for practical applications Note that in applications, the actual cause-agreement function Θ will often be only roughly true-cause and random-cause normalized. Nevertheless, because we work with a complete set of causes, we calculate the posterior probabilities for each cause and normalize the posterior probabilities to sum to one.

3.2 Computational proteomics

The section deals with the principal methods and algorithms required to apply the probabilistic cause-detection approach to computational proteomics. First, we introduce a simple cause-agreement function that evaluates the similarity between peptide and fragment mass spectrum (section 3.2.2). Afterward, we introduce various peptide prior probability models that aim to model the prior knowledge about the experiment—both in idealized situations (section 3.2.3) and in a more realistic one (section 3.2.4). For the more realistic model, we develop an algorithm that enumerates peptides with their relative prior probabilities above a particular threshold and then discuss some aspects of their storage (section 3.2.5). We then describe a fast spectral match algorithm that quickly calculates the agreement of all relevant peptides for a fragment spectrum (section 3.2.6). Utilizing all the developed notions, we then present the calculation of $\Pr_{\max}$ of all candidate peptides for a particular fragment spectrum (section 3.2.7). Finally, we conclude with

the summary of the developed methods (section 3.2.8), which also contains a visual diagram of their relationships (**Fig. 3.2**).

3.2.1 Preliminaries

Let us start by introducing the key concepts relevant to our application to peptide detection. In general, we introduce the notions of *fragment mass spectrum* and *peptide* that correspond to the notions of data and cause, respectively, within the computer-scientific framework (section 3.1).

In the context of our research, a *fragment mass spectrum* or simply a *fragment spectrum*, is a measurement of fragment masses of a parental molecule (**Fig. 1.1**). We model a fragment spectrum m as a set $\{m_1, \ldots, m_n\}$ of fragment masses, such that $n \geq 1$ and each $m_i \in \mathbb{R}^+$. In what follows, we will denote the set of all fragment mass spectra as $\mathbb{M}$. Although a fragment spectrum always comes with intensities associated with the corresponding masses, we disregard the intensities to simplify our exposition and refer to them only when these matter for our purposes.

Occasionally, we will require the fragment spectrum to be ordered by mass, and we will refer to such spectra as *mass-ordered fragment spectra*. A mass-ordered fragment spectrum M is thus a vector $M = \langle m_1, \ldots, m_n \rangle$, $n \geq 1$, such that each $m_i \in \mathbb{R}^+$ for $1 \leq i \leq n$ and $m_i < m_{i+1}$ for $1 \leq i < n$.

Notation In the upcoming definition, we introduce the notion of *peptide*. We start by first specifying its building blocks, its *residues*. Foremost, each peptide is terminated on both sides by *terminal residues*. We denote the set of applicable terminal residues on the left as $\mathbb{A}^{\vdash}$ (N-terminal residues), the set of applicable terminal residues on the right as $\mathbb{A}^{\dashv}$ (C-terminal residues), and the set of the remaining non-terminal residues as $\mathbb{A}$. Each pair of these sets has an empty intersection, thus

$$\mathbb{A}^{\vdash} \cap \mathbb{A}^{\dashv} = \emptyset, \mathbb{A}^{\vdash} \cap \mathbb{A} = \emptyset, \mathbb{A}^{\dashv} \cap \mathbb{A} = \emptyset.$$

Further, we denote all residues as

$$\mathbb{A}^{\vdash\dashv} = \mathbb{A} \cup \mathbb{A}^{\vdash} \cup \mathbb{A}^{\dashv}.$$

Each residue $r \in \mathbb{A}^{\vdash\dashv}$ has an associated *mass*

$$\textsc{Mass}(r) \in \mathbb{R}^+.$$

Because we primarily deal with modern mass spectrometric measurements, we will assume that $\textsc{Mass}(r)$ corresponds to the *monoisotopic* mass of residue r.

We now turn to the definition of a *peptide*. Although slightly technical, a peptide is a sequence of non-terminal residues terminated on each side by an appropriate terminal residue.

Definition 5 (Peptide)**.** A peptide is a sequence $\langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$, $n \geq 1$, such that $p_{\vdash} \in \mathbb{A}^{\vdash}$, $p_{\dashv} \in \mathbb{A}^{\dashv}$, and $p_i \in \mathbb{A}$ for $1 \leq i \leq n$.

Because the mass measurement is at the core of mass spectrometric measurements, let us also define the mass of a peptide. The mass of a peptide $p = \langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$, denoted $\textsc{Mass}(p)$,

is the sum of its residues, thus

$$\textsc{Mass}(p) = \textsc{Mass}(p_{\vdash}) + \sum_{1 \leq i \leq n} \textsc{Mass}(p_i) + \textsc{Mass}(p_{\dashv}).$$

Notation We denote the set of all peptides as $\mathbb{P}$. Although the set of peptides $\mathbb{P}$ is countably infinite, we will always work with its finite subsets in peptide detection. In particular, we assume that we can always measure the true mass m_{p} of a parental molecule within a tolerance $\epsilon_{\mathrm{p}} \geq 0$. The subscript in m_{p} and ϵ_{p} refers to the fact that such mass measurements are performed on the *precursor* level. In accordance, we will typically work with the subset $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$ of peptides, whose parental mass is within the mass range $\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}$, thus

$$\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}} = \left\{ q \in \mathbb{P} \,\middle|\, |\hat{m}_{\mathrm{p}} - \textsc{Mass}(q)| \leq \epsilon_{\mathrm{p}} \right\}.$$

Note that the set $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$ is especially relevant in data acquired using data-dependent acquisition (section 2.1), because besides the fragment spectrum, we always have the measurement $\hat{m}_{\mathrm{p}}$ of a mass of the non-fragmented, parental molecule.

3.2.2 Agreement between peptides and fragment mass spectra

We now describe a particular cause-agreement function that links peptides (causes) and fragment mass spectra (data). Peptide-spectrum agreements are typically defined in terms of the match between a theoretical fragment spectrum predicted for a particular peptide and an observed fragment spectrum. In accordance, we first describe a model for predicting theoretical mass spectra of individual peptides (3.2.2.1). Afterward, we define a simple scoring metric that calculates the match between two fragment spectra and briefly discuss the possibilities of its extensions (3.2.2.2).

3.2.2.1 Prediction of fragment mass spectra

We now turn to the description of a simple peptide fragmentation model [76]. In doing so, we will assume that each peptide molecule is fragmented only once, in one out of several possible locations along the peptide backbone. Overall, our aim is to calculate the masses of these fragments.

Residue mass ladders We first describe two notions that help us specify the masses of individual fragments of a peptide $p = \langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$. The *prefix residue mass ladder*, shortly *PRM ladder*, of peptide p is the set $\{m_1, \ldots, m_n\}$ such that

$$m_i = \textsc{Mass}(p_{\vdash}) + \sum_{j \leq i} \textsc{Mass}(p_j).$$

The PRM ladder thus consists of masses for all prefixes of the peptide p (including the mass of the N-terminal). Analogously, we have a *suffix residue mass ladder*, shortly *SRM ladder*, which

is a set $\{m_1, \ldots, m_n\}$ of masses such that

$$m_i = \textsc{Mass}(p_{\dashv}) + \sum_{j \geq i} \textsc{Mass}(p_j).$$

The SRM ladder thus consists of masses for all suffixes of a peptide.

Shifts from the residue mass ladders The mass ladders introduced above are not yet the actual masses of the fragments. Depending on the fragments formed, these masses are further shifted based on the specific chemical bond where the fragmentation occurred. For simplicity, we will focus on the common CID and HCD fragmentation techniques, which result predominantly in b and y ions. Given our notion of the PRM ladder, the shift for the b ion fragment is then -1.007825 Da (mass of hydrogen), and the corresponding symmetric shift for the y ion fragment is then $+1.007825$ Da. For simplicity, we will not consider neutral losses but refer the reader to [76] for their discussion, including discussion of other fragment types.

Conversion to mass-to-charge ratio Mass spectrometry does not directly measure masses but *mass-to-charge* ratios (MZs) of individual fragment ions. For instance, if we want to calculate the agreement between a peptide and some raw experimental fragment spectrum, the peptide's theoretical fragment spectrum should contain MZs of individual fragments and not their masses. To allow the transformation between the two, we will thus introduce a function

$$\textsc{Mass-to-mz}\colon \mathbb{R}^+ \times \mathbb{N} \mapsto \mathbb{R}^+$$

that assigns MZ to a mass m at a given charge z. The MZ of a mass at charge z, denoted $\textsc{Mass-to-mz}(m, z)$, is then

$$\textsc{Mass-to-mz}(m, z) = \frac{m + z \cdot \textsc{Mass}(p)}{z},$$

where $\textsc{Mass}(p) = 1.00727647$ Da is the mass of a proton. We will also use Mass-to-mz to calculate MZs over sets of masses, thus

$$\textsc{Mass-to-mz}(\{m_1, \ldots, m_n\}, z) = \{\textsc{Mass-to-mz}(m_1, z), \ldots, \textsc{Mass-to-mz}(m_n, z)\}.$$

With these notions introduced, we now define the theoretical fragment spectrum of a peptide p at a maximal fragment charge z.

Definition 6 (Theoretical fragment spectrum of a peptide p)**.** Suppose a PRM ladder P of a peptide p, and SRM ladder S of a peptide p. Further, suppose the appropriate mass shifts Δ_p and Δ_s for prefix and suffix masses, respectively, depending on the fragment type. Let $L = (P + \Delta_p) \cup (S + \Delta_s)$ be the union of the mass ladders. Then, the theoretical mass spectrum of a peptide p at a maximal fragment charge z is a set

$$\bigcup_{1 \leq i \leq z} \textsc{Mass-to-mz}(L, z).$$

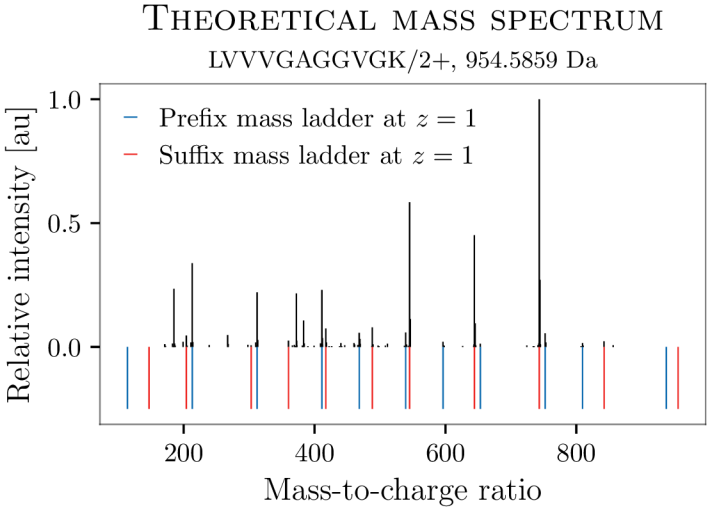


Figure 3.1: Correspondence between experimental and theoretical mass spectra.

The plot shows an experimental spectrum (black) and the corresponding prefix (blue) and suffix mass ladders (red) at maximal charge $z = 1$.

Example A particular theoretical spectrum along with a corresponding experimental spectrum is shown on a figure 3.1.

Notes on the theoretical mass spectra Let us reiterate that the cause-agreement function between peptides and mass spectra is of the form $\Theta\colon \mathbb{P} \times \mathbb{M} \mapsto \mathbb{X}$. As we mentioned above, the calculation of the agreement Θ is generally approached by predicting a theoretical fragment spectrum and then by calculating its match with the experimental fragment spectrum. Let us thus denote $\Lambda\colon \mathbb{P} \mapsto \mathbb{M}$ the function that gives the theoretical spectrum. Then, for some function $\Gamma\colon \mathbb{M} \times \mathbb{M} \mapsto \mathbb{X}$ that calculates match between two spectra, it holds that

$$\Theta(p, m) = \Gamma(\Lambda(p), m).$$

Note, however, that the spectrum prediction model, denoted Λ herein, is non-injective. In particular, there are $a \neq b \in \mathbb{A}$ such that $\textsc{Mass}(a) = \textsc{Mass}(b)$. As a result, there are $p \neq q \in \mathbb{P}$ such that $\Lambda(p) = \Lambda(q)$, and thus $\Theta(p, m) = \Theta(q, m)$ for all mass spectra m in any scoring metric Γ. Such an approach is thus fundamentally incapable of differentiating between some peptides without resorting to additional knowledge.

3.2.2.2 Calculation of spectral match

Having defined the model predicting theoretical fragment spectra, we now turn to the calculation of an agreement between two fragment mass spectra. For simplicity, we will calculate the *the number of matching fragments* (NMF) between the spectra, and afterward briefly discuss some of its extensions. Nevertheless, as matching the spectra up to tolerance $\epsilon \geq 0$ has a potentially undesirable behavior when $\epsilon > 0$, let us briefly study its properties.

Convention In what follows, we will often refer to two types of fragment spectra depending on their source. The theoretical fragment spectrum, predicted from a mass fragmentation model, will

be denoted by T. The experimental fragment spectrum, measured using the mass spectrometer, will be denoted by E.

Notation Let us denote $\boxtimes_\epsilon(T, E)$ the set of indices of corresponding matching fragments between mass spectra T and E up to tolerance ϵ, where $|T| = n$, $|E| = m$. Thus,

$$\boxtimes_\epsilon(T, E) = \big\{\langle i, j\rangle \in \{1, \ldots, n\} \times \{1, \ldots, m\} \,\big|\, |t_i - e_j| \leq \epsilon\big\}.$$

We start with the definition of the total count of fragments that match a fragment in the other spectrum.

Definition 7 (The total count of matching fragments)**.** Suppose a mass-ordered mass spectrum $T = \langle t_1, \ldots, t_n\rangle$, a mass-ordered mass spectrum $E = \langle e_1, \ldots, e_m\rangle$, and a match tolerance $\epsilon \geq 0$. The total count of matching fragments between a mass spectrum T and E up to a tolerance ϵ, denoted $\textsc{tcmf}_\epsilon(T, E)$, is

$$\textsc{tcmf}_\epsilon(T, E) = |\boxtimes_\epsilon (T, E)|.$$

The TCMF has a certain desirable property, namely that the order of the spectra in such a metric does not matter, thus

$$\textsc{tcmf}_\epsilon(T, E) = \textsc{tcmf}_\epsilon(E, T).$$

However, the TCMF has also an undesirable property: it can happen that $\textsc{tcmf}(T, E) > |T|$. For instance, suppose

$$T = \langle 0\rangle, E = \langle 0, 1\rangle, \epsilon = 1.$$

Then, $\textsc{tcmf}(T, E) = 2$. Such behavior is undesirable because we would prefer that the matching fragments are counted only once.

To overcome this problem, we slightly adjust the metric by considering a particular spectrum as a reference spectrum and count how many of its fragments are matched by a fragment in the other spectrum.

Definition 8 (The number of fragments in a mass spectrum T matching a fragment in a mass spectrum E)**.** Suppose a mass-ordered mass spectrum $T = \langle t_1, \ldots, t_n\rangle$, a mass-ordered mass spectrum $E = \langle e_1, \ldots, e_m\rangle$, and a match tolerance $\epsilon \geq 0$. The number of fragments in a mass spectrum T matching a fragment in a mass spectrum E within match tolerance ϵ, denoted $\textsc{nmf}_\epsilon(T, E)$, is then

$$\big|\big\{i \in \{1, \ldots, n\} \,\big|\, \langle i, j\rangle \in \boxtimes_\epsilon(T, E)\big\}\big|.$$

The previous definition of NMF thus specifies a simple agreement between the theoretical and the experimental spectrum. Further, the NMF metric will be our primary metric which we will utilize in peptide detection in the course of the thesis. Nonetheless, we should remain cautious about its use because it can happen that

$$\textsc{nmf}_\epsilon(T, E) \neq \textsc{nmf}_\epsilon(E, T).$$

For instance, suppose again

$$T = \langle 0 \rangle, E = \langle 0, 1 \rangle, \text{ and } \epsilon = 1.$$

Then $\text{NMF}_\epsilon(T, E) = 1$, while $\text{NMF}_\epsilon(E, T) = 2$. The order of spectra in calculating the number of fragments is thus relevant. Nevertheless, for all our purposes, we are interested in the number of fragments in the theoretical spectrum matching a fragment in the experimental spectrum. With that being said, we now show under what conditions the order does not matter.

Lemma 1. *Suppose a mass-ordered mass spectrum $T = \langle t_1, \ldots, t_n \rangle$, a mass-ordered mass spectrum $E = \langle e_1, \ldots, e_m \rangle$, and a match tolerance $\epsilon \geq 0$. Now suppose that*

$$|t_{i+1} - t_i| > 2\epsilon, 1 \leq i < n,$$

and

$$|e_{j+1} - e_j| > 2\epsilon, 1 \leq j < m.$$

Then,

$$\text{NMF}_\epsilon(T, E) = \text{NMF}_\epsilon(E, T).$$

Proof. The idea is that each fragment in one spectrum can match at most one fragment in the other spectrum. Suppose a fragment t_i is within ϵ of fragment e_j. Then t_i can not match e_{j+1} because $t_i + \epsilon < e_{j+1}$. To see this, first suppose $e_j = t_i - \epsilon$. Then $t_i + \epsilon = e_j + 2\epsilon < e_{j+1}$. Now suppose $e_j > t_i - \epsilon$. Then, $t_i + \epsilon < e_j + 2\epsilon < e_{j+1}$. Using a similar reasoning in the opposite direction, we can show that $t_i - \epsilon > e_{j-1}$. Analogously, we can exchange the roles of t_i and e_j, showing that each fragment in one spectrum can match at most one fragment in the other spectrum. The result then follows. □

The lemma thus shows that as long as there are sufficiently large differences in the consecutive mass fragments (i.e., $> 2\epsilon$), the order of the spectra does not matter.

Extensions of NMF The NMF metric is a simple metric with straightforward interpretation and can be thus considered as a starting reference point. In our research [1], we considered several direct extensions of the metric by various uses of fragment intensities, normalizations, and suppression of noise peaks—such extensions generally improved the detection performance. Nevertheless, as our overall focus is on the importance of peptide prior probabilities in peptide detection, we leave the metric as is—knowing that we are likely to obtain better performance with a more involved scoring metric.

3.2.3 Simple peptide prior probability models

The section describes various simple models of peptide prior probabilities. These models illustrate several ways to express the prior knowledge about an experiment and also serve as an introduction to the more realistic model developed in the next section (3.2.4). In practice, we utilize these models for the analysis of peptide detection in idealized circumstances (5.1), wherein we directly use the Bayesian model for calculating posterior probabilities of all candidate peptides because the number of such candidates is reasonably low (i.e., $\leq 10^8$). Although the prior models are

simple, they nevertheless aim to capture a particular aspect of situations encountered in the computational detection of peptides.

We now specify what we mean by peptide prior probability models. Note that in assigning the prior probabilities to peptides, we always refer to the finite set $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ of peptides within the corresponding mass range.

Definition 9 (Peptide relative prior probability model)**.** A peptide relative prior probability model is a function

$$\Pr^\star\colon \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \mapsto \mathbb{R}^+.$$

Definition 10 (Peptide prior probability model)**.** A peptide prior probability model is a peptide relative prior probability model $\Pr^\star$ such that

$$\sum_p \Pr^\star(p) = 1.$$

Note that it often suffices to work with a relative prior probability model. For instance, such a model is enough to calculate the maximal posterior probability of a peptide (section 3.1.2). In addition, we can often normalize the relative prior probabilities to obtain a prior probability model. As a result, we often consider these models interchangeable and focus on their differences only when these matter for intended purposes.

3.2.3.1 Uniform prior

The uniform prior refers to a situation when essentially no prior knowledge about expected peptides is available, or its use is not desirable. In such case, for all $p \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, we have

$$\Pr^\star(p) = 1.$$

The use of such a model then refers to a completely-unaware peptide sequencing *de novo*.

3.2.3.2 Residue distribution prior

The model captures a situation when some residues are more likely *a priori* than others. Thus suppose a distribution of residues

$$\Pr_\beta\colon \mathbb{A}^{\vdash\dashv} \mapsto \langle 0, 1\rangle,$$

such that

$$\sum_{r\in\mathbb{A}^{\vdash\dashv}} \Pr_\beta(r) = 1.$$

In this model, we define the relative prior probability of a peptide p as follows:

$$\Pr^\star_\beta(\langle p_1, \ldots, p_n\rangle) = \prod_i \Pr_\beta(p_i).$$

The model then corresponds to sequencing *de novo* with expectations over the distribution of residues for correct peptides.

3.2.3.3 Prior based on expected cutting after a residue

The model is motivated by the properties of enzymes used in bottom-up proteomics. In particular, many such enzymes cut a protein sequence with a certain probability *after* a specific residue. Thus, let us have a function

$$\alpha \colon \mathbb{A} \mapsto \langle 0, 1 \rangle,$$

which gives the probability of an enzyme cutting a sequence after encountering a particular non-terminal residue. We define the relative prior probability of peptide p based on the cleavage model α, denoted $\Pr^\star_\alpha(p)$, as

$$\Pr^\star_\alpha(\langle p_\vdash, p_1, \ldots, p_n, p_\dashv \rangle) = \left(\prod_{i=1}^{n-1} 1 - \alpha(p_i) \right) \cdot \alpha(p_n).$$

In other words, it is the multiplication of probabilities that a peptide was cut after the last residue and never before.

Note If a peptide is to be produced in typical bottom-up proteomics experiments, the cutting happens over some parental sequence. As a result, one could also consider the unknown residue just before p_1. However, we would require additional knowledge about the experiment, for instance, the expected parental sequences or the expected distribution of residues. In this model, we do not consider such situations; we do so, however, in a more realistic model introduced later (section 3.2.4). The use of the model corresponds to sequencing *de novo*, however, with preferences for some sequences based on the expected behavior of the enzyme.

3.2.3.4 Distance to a single sequence

The model corresponds to a situation when we expect a particular sequence q and assume that sequences p closer to the expected sequence q are more likely a priori. We define the relative prior probability $\Pr^\star(p)$ through a distance

$$\Delta \colon \mathbb{P} \times \mathbb{P} \mapsto \mathbb{R}^+$$

between a peptide p and the expected peptide q. We then define the prior probabilities in such a model as

$$\Pr^\star(p) = c^{\Delta(p,q)},$$

for some $c \in \langle 0, 1 \rangle$ that specifies how the prior probability of a peptide decreases with its increase in distance to q. The model corresponds to a situation when we expect a particular reference peptide and aim to detect its deviations.

3.2.3.5 Minimal distance to multiple sequences

The following prior model generalizes the previous model by considering multiple sequences and a *minimal* distance to any of them. Thus, suppose a distance function

$$\Delta \colon \mathbb{P} \times \mathbb{P} \mapsto \mathbb{R}^+$$

and a set P of peptides. For each $p \in \mathbb{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$, let us have the minimal distance of a peptide p to any peptide q in P,

$$\textsc{Dist}(p, P) = \min_{q \in P} \Delta(p, q).$$

Then, the prior model is

$$\Pr^\star(p) = c^{\textsc{Dist}(p,P)},$$

for some $c \in \langle 0, 1 \rangle$ specifying how the prior probability of a peptide decreases with its increase in minimal distance. Such a prior model then corresponds to a reference-guided search, applicable, for instance, to detecting variants of reference peptides.

3.2.4 A more realistic prior probability model

Herein, we develop a more realistic model of peptide prior probabilities, which aims to be usable in analyzing typical computational proteomics data. In this model, we assume that individual peptides originate from a set of reference proteins through modification, substitution, and cleavage events. Further, we assume that these events are statistically independent, allowing us to derive some aspects of the relative prior probabilities. Still, the model only aims to be realistic to a certain degree; as a result, we will make several assumptions to simplify both the model and the calculation of the relative prior probabilities.

Notation Let us first introduce some additional notation to simplify the exposition. In general, we assume that the parental sequences consist only of a subset $\mathbb{A}_\wedge^{\vdash\dashv}$ of all residues $\mathbb{A}^{\vdash\dashv}$. We refer to such a subset as *reference residues*. The $\mathbb{A}_\wedge^{\vdash\dashv}$ consists of twenty amino acids $\mathbb{A}_\wedge$ used by cells during the synthesis of proteins and of standard non-modified terminals: $\vdash$ and $\dashv$. We have

$$\mathbb{A}_\wedge^{\vdash\dashv} = \mathbb{A}_\wedge \cup \{\vdash, \dashv\}.$$

For completeness, let us also specify the $\mathbb{A}_\wedge$ by using one-letter code for amino acids, thus

$$\mathbb{A}_\wedge = \{\mathtt{A}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}, \mathtt{I}, \mathtt{K}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{P}, \mathtt{Q}, \mathtt{R}, \mathtt{S}, \mathtt{T}, \mathtt{V}, \mathtt{W}, \mathtt{Y}\}$$

Furthermore, each *non-reference residue* $r \in \mathbb{A}^{\vdash\dashv} \setminus \mathbb{A}_\wedge^{\vdash\dashv}$ corresponds to a *single* reference residue $b \in \mathbb{A}_\wedge^{\vdash\dashv}$, denoting such a reference residue $\downarrow r = b$. For each reference residue $r \in \mathbb{A}_\wedge^{\vdash\dashv}$, we also let $\downarrow r = r$ to simplify the presentation.

Example Suppose there is a non-reference residue $\mathtt{M} \oplus \mathtt{Oxidation}$, representing an oxidized methionine. Then, the corresponding reference residue of the non-reference residue $\mathtt{M} \oplus \mathtt{Oxidation}$ is $\mathtt{M}$ and thus $\downarrow \mathtt{M} \oplus \mathtt{Oxidation} = \mathtt{M}$.

Definition 11 (Modified form of a residue)**.** A residue $b \in \mathbb{A}^{\vdash\dashv}$ is a modified form of a residue $a \in \mathbb{A}_\wedge^{\vdash\dashv}$ if $\downarrow b = a$.

Definition 12 (Substituted form of a residue)**.** A residue b is a substituted form of a residue $a \in \mathbb{A}_\wedge$ if $b \in \mathbb{A}_\wedge$.

Note that we always consider the non-modified form of a residue as one of its modified forms, and we do analogously for the substituted form of a residue.

3.2.4.1 Modification of a residue

We now introduce some additional notation for modified residues. Let us denote

$$\mathcal{M}(a) = \{b \in \mathbb{A}^{\vdash\dashv} \mid \downarrow b = a\},$$

the set of modified forms of a residue $a \in \mathbb{A}_{\wedge}^{\vdash\dashv}$. Further, let us denote $\mathcal{M}_a(b)$ the expected proportion of a modified form $b \in \mathbb{A}^{\vdash\dashv}$ of a residue $a \in \mathbb{A}_{\wedge}^{\vdash\dashv}$. In general, we will assume that we consider all modified forms. Finally, because we also consider the absence of modification, the sum over all forms b of a normalizes to one, thus:

$$\sum_b \mathcal{M}_a(b) = 1.$$

Example For instance, suppose there are only two possible forms of amino acid Methionine (M): its non-modified form (M) and its oxidized form ($\mathtt{M} \oplus \mathtt{Oxidation}$). For simplicity, suppose we would expect to see both forms in equal proportions. Then,

$$\mathcal{M}(\mathtt{M}) = \{\mathtt{M}, \mathtt{M} \oplus \mathtt{Oxidation}\},$$

and

$$\mathcal{M}_{\mathtt{M}}(\mathtt{M}) = \frac{1}{2} = \mathcal{M}_{\mathtt{M}}(\mathtt{M} \oplus \mathtt{Oxidation}).$$

3.2.4.2 Substitution of a residue

Similarly as we did for the modified forms, let us denote $\mathcal{S}(a)$ the set of substituted forms of a residue a. Actually, $\mathcal{S}(a) = \mathbb{A}_{\wedge}$. Analogously as for the modified forms, we denote $\mathcal{S}_a(b)$ the expected proportion of a substituted form $b \in \mathbb{A}_{\wedge}$ of $a \in \mathbb{A}_{\wedge}$. Because we also include no substitution and because we assume that we consider all reference residues, the proportion of all substituted forms over each $a \in \mathbb{A}_{\wedge}$ sums to one, thus:

$$\sum_b \mathcal{S}_a(b) = 1.$$

Example Suppose a reference amino acid $r = \mathtt{M}$. Then the $\mathcal{S}(r) = \mathbb{A}_{\wedge}$. Let us specify, for instance, the expected proportion of I substituted from M to be 10^{-4}, thus $\mathcal{S}_{\mathtt{M}}(\mathtt{I}) = 10^{-4}$.

3.2.4.3 Expected proportion of a residue form

We now combine the notions of a modification and a substitution of a residue. Let us denote the expected proportion of a residue $b \in \mathbb{A}^{\vdash\dashv}$ originating from a residue $a \in \mathbb{A}_{\wedge}^{\vdash\dashv}$ as $\Pr(a \to b)$. Then we have the following:

Lemma 2 (Expected proportion of a residue form)**.** *Suppose that the events of modification and substitution of residues are statistically independent. Then*

$$\Pr(a \to b) = \mathcal{S}_a(\downarrow b) \cdot \mathcal{M}_{\downarrow b}(b).$$

Proof. From the statistical independence. □

Example For instance, the expected proportion of an oxidized Methionine ($\mathtt{M} \oplus \mathtt{Oxidation}$) originating from Cysteine ($\mathtt{C}$) then equals

$$\mathcal{S}_{\mathtt{C}}(\mathtt{M}) \cdot \mathcal{M}_{\mathtt{M}}(\mathtt{M} \oplus \mathtt{Oxidation}).$$

Notation Similarly, as we did for modifications and substitutions, we now introduce the notation of all forms $\mathcal{F}$ of a reference residue a. Then $\mathcal{F}(a) = \mathbb{A}$ and $\mathcal{F}_a(b) = \Pr(a \to b)$.

3.2.4.4 Expected proportion of a sequence form

We now expand the notion of the expected proportion of a residue form over a sequence of residues. Let us denote

$$\Pr(\langle s_1, \ldots, s_l \rangle \to \langle p_1, \ldots, p_l \rangle)$$

the expected proportion of a sequence form $\langle p_1, \ldots, p_l \rangle$ originating from a sequence $s = \langle s_1, \ldots, s_l \rangle$ of the same length, such that that each residue p_i originated from s_i.

Lemma 3 (Expected proportion of a sequence form)**.** *Suppose that the events of modifications and substitutions over individual residues are statistically independent. Then*

$$\Pr(\langle s_1, \ldots, s_l \rangle \to \langle p_1, \ldots, p_l \rangle) = \prod_i \Pr(s_i \to p_i).$$

Proof. From the statistical independence. □

3.2.4.5 General structure of cutting

The notions introduced in the previous sections give the expected proportion of a particular form of a peptide. However, to obtain such a peptide, we also require that some parental sequence was first cut accordingly. Let us first consider the situation in general without resorting to an actual sequence cutting model. For simplicity, we will also ignore the terminal residues. Thus, given a parental sequence

$$s = \langle s_1, \ldots, s_n \rangle, n \geq 1,$$

we need to specify the expected proportion of each cut of s starting at i and ending at j, denoted $i \overset{s}{\leftrightarrow} j$, for $1 \leq i \leq j \leq n$. We will denote the expected proportion of such a cut as

$$\Pr(s \dashrightarrow i \overset{s}{\leftrightarrow} j).$$

Overall, it must hold that

$$\sum_{1 \leq i \leq j \leq n} \Pr(s \dashrightarrow i \overset{s}{\leftrightarrow} j) = 1.$$

In other words, we thus need to specify the proportions of all possible cuts.

3.2.4.6 Expected proportions of cuts in a cleavage-after-residue model

Let us now focus on a cutting model that cuts after a residue, similarly as we did in our cut-after-residue model in the section 3.2.3.3. For simplicity, we will again ignore the terminal residues. Thus, suppose we have the expected proportion of cuts after a residue $\alpha \colon \mathbb{A} \mapsto \langle 0, 1 \rangle$ and a

parental sequence $s = \langle s_1, \ldots, s_n \rangle$. We will assume a particular expected behavior of the relative prior probabilities of individual cuts and then specify a model that conforms to such behavior.

Independence of cut position We start with the independence of the expected proportion of a cut on the position of a subsequence within the parental sequence. Suppose that a sequence s contains a sequence $p = \langle p_1, \ldots, p_n \rangle$ at two positions, starting at f_a and f_b. We assume that if the residue before each of these subsequences is the same, thus $s_{f_a-1} = s_{f_b-1}$, the relative prior probabilities of the cuts are the same as well, thus

$$\Pr^\star_\alpha(s \dashrightarrow f_a \overset{s}{\leftrightarrow} f_{a+n-1}) = \Pr^\star_\alpha(s \dashrightarrow f_b \overset{s}{\leftrightarrow} f_{b+n-1}). \tag{3.5}$$

If the residue just before either subsequence differs, we assume the following relationship among the relative prior probabilities:

$$\frac{\Pr^\star_\alpha(s \dashrightarrow f_a \overset{s}{\leftrightarrow} f_{a+n-1})}{\Pr^\star_\alpha(s \dashrightarrow f_b \overset{s}{\leftrightarrow} f_{b+n-1})} = \frac{\alpha(s_{f_a-1})}{\alpha(s_{f_b-1})}. \tag{3.6}$$

In other words, the ratio of the relative prior probabilities of the whole cuts equals the ratio of the expected proportions of cuts after the residues just before each subsequence.

Single residue difference within same-length subsequences Now we consider a situation when two subsequences differ in a single residue. Thus, suppose that the sequence s contains subsequences p and q of the same length, starting at positions f_a and f_b, respectively, and that the residue just before them is the same, thus $s_{f_a-1} = s_{f_b-1}$. Further, suppose that $p = \langle p_1, \ldots, p_n \rangle$ and $q = \langle q_1, \ldots, q_n \rangle$ differ in their i-th residue *only*.

Now, suppose that $i = n$, thus i corresponds to the last residue. We assume that

$$\frac{\Pr^\star_\alpha(s \dashrightarrow f_a \overset{s}{\leftrightarrow} f_{a+n-1})}{\Pr^\star_\alpha(s \dashrightarrow f_b \overset{s}{\leftrightarrow} f_{b+n-1})} = \frac{\alpha(f_{a+n-1})}{\alpha(f_{b+n-1})}. \tag{3.7}$$

In other words, the ratio of the cuts equals the ratio of the expected proportion of cuts after the last residue.

Now suppose that the different residue is any residue except the last one, thus $i < n$. Then, we assume that

$$\frac{\Pr^\star_\alpha(s \dashrightarrow f_a \overset{s}{\leftrightarrow} f_{a+n-1})}{\Pr^\star_\alpha(s \dashrightarrow f_b \overset{s}{\leftrightarrow} f_{b+n-1})} = \left(\frac{(1 - \alpha(p_i))}{(1 - \alpha(q_i))}\right)^{-1}. \tag{3.8}$$

In other words, the ratio of the cuts equals the inverse of the expected proportion of not cutting.

Cleavage model Having specified the expected behavior of cuts, we now provide a model that satisfies the equations 3.5–3.8. In this model, the relative prior probability of a cut $f \overset{s}{\leftrightarrow} t$ from a sequence s, is then

$$\Pr^\star_\alpha(s \dashrightarrow f \overset{s}{\leftrightarrow} t) = \Pr_\alpha(s_{f-1}) \cdot \prod_{f \le i < t} (1 - \Pr_\alpha(s_i)) \cdot \Pr_\alpha(s_t).$$

3.2.4.7 Expected proportion of a cut of a particular form

We now combine the notions of modification, substitution, and cleavage events. Thus, suppose a parental sequence $s = \langle \vdash, s_1, \ldots, s_m, \dashv \rangle$, and a peptide $p = \langle p_\vdash, p_1, \ldots, p_n, p_\dashv \rangle$, such that $n \leq m$. In what follows, we define the expected proportion of a cut of form p, from parental sequence s, starting at position i, denoting it as $\Pr^\star(s \xrightarrow[i]{} p)$.

Definition 13 (Expected proportion of a cut of form p of s starting at position i)**.** The expected proportion of a cut of form p of parental sequence s starting at position i, denoted $\Pr^\star(s \xrightarrow[i]{} p)$, is defined as follows:

$$\Pr^\star(s \xrightarrow[i]{} p) = \Pr^\star_\alpha(s \dashrightarrow i \overset{s}{\leftrightarrow} i+n-1) \cdot \Pr(\langle s_i, \ldots, s_{i+n-1} \rangle \to \langle p_1, \ldots, p_n \rangle) \cdot \mathcal{M}_\vdash(p_\vdash) \cdot \mathcal{M}_\dashv(p_\dashv).$$

Clarification In other words, the $\Pr^\star(s \xrightarrow[i]{} p)$ is equal to the multiplication of the following:

- the expected proportion of the cut of s of length n, starting at position i;
- the expected proportion of sequence form $\langle p_1, \ldots, p_n \rangle$ of sequence $\langle s_i, \ldots, s_{i+n-1} \rangle$;
- the expected proportion of N-terminal form $p_\vdash$; and
- the expected proportion of C-terminal form $p_\dashv$.

3.2.4.8 Maximal expected proportion of a sequence form

To further simplify our model and calculations, we will focus only on the maximal expected proportion of a sequence form. Let us denote $\Pr^\star_{\max}(s \dashrightarrow p)$ the maximal expected proportion of a sequence form $p = \langle p_\vdash, p_1, \ldots, p_n, p_\dashv \rangle$ originating from a parental sequence $s = \langle \vdash, s_1, \ldots, s_m \dashv \rangle$ at some starting position i.

Lemma 4 (Maximal expected proportion of a sequence form p originating from a sequence s)**.**

$$\Pr^\star_{\max}(s \dashrightarrow p) = \max_{1 \leq i \leq m-n+1} \Pr^\star(s \xrightarrow[i]{} p).$$

Proof. The only indices over which $\Pr^\star(s \xrightarrow[i]{} p)$ is defined are $i \in \{1, \ldots, m-n+1\}$. □

Similarly, let us denote $\Pr^\star_{\max}(S \dashrightarrow p)$ the maximal expected proportion of a sequence form p originating from sequences $= \{S_1, \ldots, S_n\}$.

Theorem 2 (Maximal expected proportion of a sequence form p originating from a sequence in S)**.**

$$\Pr^\star_{\max}(S \dashrightarrow p) = \max_{s \in S} \Pr^\star_{\max}(s \dashrightarrow p).$$

Proof. Straightforward. □

The model Finally, we set the relative prior probability of p as the maximal expected proportion of a sequence p originating from a sequence in S, thus

$$\Pr^\star(p) = \Pr^\star_{\max}(S \dashrightarrow p). \tag{3.9}$$

Note As is evident from our relative prior probability model, we focus on the maximal proportion of a sequence form p. Such a model creates some imprecision, e.g., if a particular sequence form p can originate from multiple parental sequences or multiple positions, its expected proportion should be correspondingly higher. Nevertheless, we neglect the imprecision for the sake of simplicity.

3.2.5 Enumeration of peptides

Herein, we introduce an algorithm that enumerates peptides and their relative prior probabilities according to the more realistic prior probability model (section 3.2.4). Overall, we utilize the algorithm to obtain all peptides whose minimal relative prior probability is above some prespecified threshold $p_{\min}$. In turn, this allows us to calculate the maximal posterior probability $\Pr_{\max}$ for all peptides with prior probabilities above $p_{\min}$ given their agreements with the fragment spectrum. In what follows, we first describe the algorithm itself (section 3.2.5.1), illustrate its behavior for simplified parameters of the prior model (section 3.2.5.2) and then discuss the organization of the peptides by means of their parental mass (section 3.2.5.3).

3.2.5.1 Peptide enumeration algorithm

We now introduce the peptide enumeration algorithm for the more realistic prior probability model (section 3.2.4). Although the algorithm's operation is quite simple, a few technical aspects require consideration. Altogether, the algorithm consists of three procedures, and is presented in a detailed pseudocode on listings 1 and 2. Let us now provide a brief overview of its functioning.

We start with the high-level procedure Build-Peptides, whose output is the desired vector of peptides and their relative prior probabilities (listing 1). Build-Peptides takes a set S of parental sequences, and for each sequence $s \in S$, obtains peptides and their relative prior probabilities using Build-Peptides-From-Seq procedure. Afterward, it retains each peptide's maximal relative prior probability by aggregating over its relative prior probabilities (over individual parental sequences or multiple positions within the sequence). The algorithm also takes two additional parameters: the minimal relative prior probability $p_{\min}$ and the desired mass range $\langle m_{\min}, m_{\max} \rangle$ of peptides. These parameters specify the desired depth of the peptide database ($p_{\min}$), along with its width ($\langle m_{\min}, m_{\max} \rangle$).

We now turn to the mid-level procedure Build-Peptides-From-Seq, which works on the level of a single parental sequence s (listing 1). For each starting position i of s, the procedure initializes the relative prior probability of the peptide to be constructed, based on the cleavage probability of the previous residue s_{i-1} and the expected proportions of forms of its terminal residues. Once initialized, it invokes the recursive Enumerate procedure, responsible for the actual construction of the peptides.

The Enumerate procedure, in essence, recursively adds any applicable form of the next residue from the parental sequence while keeping track of its relative prior probability (listing 2). The procedure also calculates the peptide's relative prior probability if cut after the currently incorporated residue (p_{cleaved}) and if extended (p_{extended}). If the peptide q is of a sufficiently high relative prior probability (i.e., $p_{\text{cleaved}} \geq p_{\min}$) and of appropriate mass (i.e., $m_{\min} \leq \textsc{Mass}(q) < m_{\max}$), it stores the peptide and its relative prior probability. On the other hand, if the relative prior probability is already too low (i.e., $p_{\text{cleaved}} < p_{\min}$ and $p_{\text{extended}} < p_{\min}$), or if the mass of a

peptide is already too high, the procedure abandons the search. Once completed, the procedure thus returns all peptides that start at the position i within sequence s, and are of appropriate relative prior probabilities and masses.

3.2.5.2 Illustration of peptide enumeration

Let us show some examples of the output of the algorithm for peptide enumeration. In what follows, we will always consider the same parental sequence $s = \mathsf{LVVVMKGVGK}$, expressed as a sequence of one-letter amino acid codes, minimal prior probability $p_{\min} = 0.1$, and a complete mass range ($m_{\min} = 0$, and $m_{\max} = \infty$). To increase the clarity of the exposition, we ignore the non-terminal residues. Let us denote $f(s)$ the result of the procedure Build-Peptides-From-Seq(s, $p_{\min}$, $\langle m_{\min}, m_{\max} \rangle$). We now show $f(s)$ for several examples.

No events allowed Suppose that no modifications, no substitutions, and no cleavage events are allowed. Thus, for all $a \in \mathbb{A}_\wedge$, $\mathcal{F}_a(a) = 1$, specifying that only non-modified forms are allowed. Furthermore, for each $b \in \mathbb{A}$, $\alpha(b) = 0$, specifying that no cleavage is allowed. Then

$$f(s) = \langle \langle s, 1.0 \rangle \rangle,$$

because nothing can happen to the parental sequence.

Cleavage always after a residue Suppose the configuration is as in the previous example but let us specify that the cleavage always happens after a residue K, thus $\alpha(\mathsf{K}) = 1.0$. Then

$$f(s) = \langle \langle \mathsf{LVVVMK}, 1.0 \rangle, \langle \mathsf{GVGK}, 1.0 \rangle \rangle.$$

Relaxed cleavage after a residue Now let us relax the cleaving, and suppose $\alpha(\mathsf{K}) = 0.9$. Then

$$f(s) = \langle \langle \mathsf{LVVVMK}, 0.9 \rangle, \langle \mathsf{GVGK}, 0.9 \rangle, \langle \mathsf{LVVVMKGVGK}, 0.1 \rangle \rangle.$$

Note that the relative prior probability of the last peptide is lower because it contains a residue K that was not cleaved.

A single applicable modification Finally, let us consider a single applicable modification, and again, no cleavage is allowed. Suppose $\mathcal{F}_{\mathsf{M}}(\mathsf{M}^{\mathsf{Oxidation}}) = 0.5$. Then,

$$f(s) = \langle \langle \mathsf{LVVVMKGVGK}, 0.5 \rangle, \langle \mathsf{LVVVM}^{\mathsf{Oxidation}}\mathsf{KGVGK}, 0.5 \rangle \rangle.$$

The number of modified forms Finally, we note that the number of modifications is rather large in practice. For instance, as of 2021, the average number of modifications applicable to a residue $r \in \mathbb{A}_\wedge^{\vdash\dashv}$ is around 140, as derived from the Unimod database [77]. Thus, if the $p_{\min}$ for the generation of peptides is low, the number of generated peptides can get high.

Listing 1: Enumeration of peptides above minimal relative prior probability (part 1)

```
/* High-level procedure */
/* Produces peptides and their maximal relative prior probabilities from a set of reference
   sequences. */
Function Build-Peptides(S, p_min, ⟨m_min, m_max⟩):
    Data: Reference sequences S = {s_1, ..., s_n}
          Minimal relative prior probability p_min
          Peptide mass range ⟨m_min, m_max⟩
    Result: Vector Q of peptides and their relative prior probabilities
    begin
        foreach s ∈ S do
            Q_s ← Build-Peptides-From-Seq(s, p_min, ⟨m_min, m_max⟩)
        end
        /* Concatenate the results, Q = ⟨P, R⟩ */
        Q ← ⨁_s Q_s
        /* Retain the maximal relative prior probability per peptide */
        Q ← Unique-Peptides-With-Max-P(Q)
        return Q
    end

/* Mid-level procedure */
/* Produces peptides and their relative prior probabilities from a given reference
   sequence. */
Function Build-Peptides-From-Seq(s, p_min, ⟨m_min, m_max⟩):
    Data: Reference sequence s = ⟨⊢, s_1, ..., s_k, ⊣⟩
          /* See the explanations of the following parameters in the algorithm above */
          p_min, ⟨m_min, m_max⟩
    Result: Vector Q = ⟨P, R⟩ of peptides P and their relative prior probabilities R
    begin
        /* Initialize result */
        Q ← ⟨⟩
        /* For each starting position excluding N- and C-termini */
        foreach i ∈ ⟨1, ..., k⟩ do
            /* Cleavage required before the previous residue */
            /* Note: We set α(⊢) = 1 because the cleavage does not happen (and let s_0
               be ⊢) */
            p_initial ← α(s_{i−1})
            foreach n ∈ M(⊢) do /* For each form of N-terminal */
                foreach c ∈ M(⊣) do /* For each form of C-terminal */
                    /* Include expected proportions of N- and C-termini forms */
                    p ← p_initial · M_⊢(n) · M_⊣(c)
                    /* Create the peptides (see listing 2) */
                    Enumerate(s, i, i, Mass(n) + Mass(c), p, p, n, c, p_min, ⟨m_min, m_max⟩, Q)
                end
            end
        end
        return Q^T
    end
```

Listing 2: Enumeration of peptides above minimal relative prior probability (part 2)

```
/* Low-level procedure */
/* A function generating peptides and their relative prior probabilities */
Function ENUMERATE(s, i, f, m, p_extended, p_cleaved, n, c, p_min, ⟨m_min, m_max⟩, Q):
    Data: Sequence s = ⟨s_0, ..., s_k⟩
          Current position i within s
          Initial position f within s
          Expected proportion p_extended if the current peptide is extended
          Expected proportion p_cleaved if the current peptide is cleaved
          /* The following parameters remain fixed */
          Form n of N-term
          Form c of C-term
          Minimal relative prior probability p_min
          Peptide mass range ⟨m_min, m_max⟩
          Vector Q to store the results

    /* ACCEPTANCE */
    /* If the peptide is of interest */
    if i > f and m ≥ m_min and m < m_max and p_cleaved ≥ p_min then
        APPEND(Q, ⟨⟨n, s_f, ..., s_{i-1}, c⟩, p_cleaved⟩)
    end

    /* REJECTION */
    if i ≥ k then /* Already at protein's C-term */
        return
    end
    if p_extended < p_min and p_cleaved < p_min then /* Peptide probability already too low */
        return
    end
    if m ≥ m_max then /* Peptide mass already too high */
        return
    end

    /* [INCORPORATION OF A NEW RESIDUE] */
    /* Store the original residue */
    e ← s_i
    foreach r ∈ F(e) do /* For each form of e (including the raw form) */
        /* Obtain the expected proportion */
        r_p ← F_e(r)
        if i < k − 1 then /* If still not at the C-term of the parental sequence */
            /* Expected proportion if cleaved after the new residue */
            p_cleaved* ← p_extended · r_p · α(r)
            /* Expected proportion if not cleaved after the new residue */
            p_extended* ← p_extended · r_p · (1 − α(r))
        end
        else /* Otherwise, the cleavage is not happening */
            p_cleaved* ← p_extended · r_p
            p_extended* ← p_extended · r_p
        end

        s_i ← r /* Change the residue */
        ENUMERATE(s, i+1, f, m+MASS(r), p_extended*, p_cleaved*, n, c, p_min, ⟨m_min, m_max⟩, Q)
        s_i ← e /* Change the residue back */
    end
```

3.2.5.3 Indexation by precursor mass

Herein, we will focus on the indexation of the database by parental mass of the peptides. Let us reiterate that the BUILD-PEPTIDES procedure gives a vector

$$\mathbf{Q} = \big\langle \langle p_1, \Pr^\star(p_1)\rangle, \ldots, \langle p_n, \Pr^\star(p_n)\rangle \big\rangle^T$$

such that $\Pr^\star(p_i) \geq p_{\min}$ and $\textsc{Mass}(p_i) \in \langle m_{\min}, m_{\max}\rangle$, for a given range of masses $\langle m_{\min}, m_{\max}\rangle$ and a minimal relative prior probability $p_{\min}$. In computational proteomics, one typically calculates the spectral match only for peptides of a specific mass range. This is because the true mass m_{p} of the non-fragmented molecule is assumed to be within a particular prespecified tolerance ϵ_{p} of the measured mass $\hat{m}_{\mathrm{p}}$. Thus, we generally assume that $m_{\mathrm{p}} \in \hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}$, and thus the correct peptide is within $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$.

As a result, it is meaningful to organize $\mathbf{Q}$ by peptide mass because we will typically retrieve peptides by their masses. Note that one option is to just sort the peptides according to their mass. However, because we will also use a fast search algorithm that calculates spectral match with all peptides in its index, it is beneficial to split the vector $\mathbf{Q}$ into its non-overlapping subvectors. In practice, we set up mass bins of fixed-width w, and let them start at 0 to simplify the formulas. Let us thus denote the mass range corresponding to a bin b as

$$M_b = \langle b \cdot w, (b+1) \cdot w\rangle. \tag{3.10}$$

Then, we will store separately the corresponding subvectors

$$\mathbf{Q}_b = \big\langle \langle p, r\rangle \in \mathbf{Q}^T \;\big|\; \textsc{Mass}(p) \in M_b \big\rangle^T.$$

The partitioning of the database by peptide masses then naturally fits with a fast spectral match which we describe in the following section.

Notes on building the peptide database Note that another way to obtain the database bins $\mathbf{Q}_b$ would be to repeatedly invoke the procedure BUILD-PEPTIDES with the corresponding mass ranges M_b. Although such an approach would be equivalent from the perspective of the resulting data, it would be very inefficient computationally. In particular, in the recursive procedure ENUMERATE, a potentially huge number of peptides would have to be repeatedly dismissed—because of their small mass until reaching the minimal mass $m_{\min}$ acceptable for storing. Instead, it is much more efficient to build the peptides for a wide mass range and then partition the resulting database.

3.2.6 Fast spectral match

Herein, we describe a fragment-indexation method that allows fast calculation of spectral matches between a large number of fragment spectra and a single fragment spectrum. Note that a similar method is implemented in the open-search approach of MSFragger algorithm [10]. For typical applications in computational proteomics, the multiple fragment spectra would correspond to the theoretical fragment spectra of candidate peptides and the single fragment spectrum to an actual measured fragment spectrum. First, we introduce the construction of the fragment-ion

index (section 3.2.6.1), a central structure which allows fast calculation of spectral matches (section 3.2.6.2). Afterward, we adapt the algorithm to return spectral match with all peptides within a specified mass range (section 3.2.6.3) while using a mass-partitioned database. Finally, we describe an algorithmic optimization that loads only a small part of the fragment-ion index—tailored particularly to the measured fragment spectrum (3.2.6.4).

3.2.6.1 Construction of a fragment-ion index

We now turn to the construction of a fragment-ion index. We start first by defining what we mean by a fragment-ion index for a vector of mass spectra $\mathbf{T}$.

Definition 14 (Fragment-ion index)**.** A fragment-ion index for a vector $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$ of mass spectra is a vector $\mathbf{F} = \langle \langle m_1, i_1 \rangle, \ldots, \langle m_l, i_l \rangle \rangle, m_j \leq m_{j+1}$ with no duplicate elements, such that

$$\langle m, k \rangle \in \mathbf{F} \text{ if and only if } m \in T_k.$$

As indicated by the simplicity of the definition, the construction of a fragment-ion index is straightforward. Overall, we concatenate all the fragment spectra from $\mathbf{T}$, while keeping track of the index of their parental spectrum. Finally, we sort the concatenated structure by the fragment mass. The function Build-Fragment-Ion-Index on listing 3 thus constructs the fragment-ion index by the method we just described. Note that we chose the names of the functions in pseudocode to correspond to those in python's NumPy library.

Listing 3: Construction of a fragment-ion index

```
Function Build-Fragment-Ion-Index(T):
    Data: Vector T = ⟨T_1, ..., T_n⟩ of fragment mass spectra
    Result: Fragment-ion index F for fragment mass spectra T
    begin
        /* Linearize the vector */
        L ← Concatenate(T)
        /* Create a vector I of the same length as L such that */
        /* I_j contains an index of the parental mass spectrum */
        /* to which L_j corresponds. */
        I ← Repeat(⟨1, ..., n⟩, Map(Length, T))
        /* Obtain the sorting indices for L */
        A ← ArgSort(L)
        /* Reorder the arrays to create the fragment-ion index */
        F ← Zip(L[A], I[A])
        return F
    end
```

Let us now analyze the complexity of the algorithm depending on the length n of the vector $\mathbf{T}$ of mass spectra. For simplicity, we will assume that the number of fragments in individual mass spectra T_i is constant. The most time-demanding part of the algorithm is the sort of the concatenated array, which can be done in $\mathcal{O}(n \log n)$ time. Finally, we note that even though the fragment-ion index can be constructed efficiently, its construction is relatively infrequent in practice.

3.2.6.2 Matching against the fragment-ion index

We now turn to the calculation of a spectral match with all fragment spectra from the fragment-ion index. In doing so, we will utilize the NMF metric that calculates the number of matching fragments between two spectra, as described in section 3.2.2.2. In what follows, let us have an experimental fragment spectrum $E = \langle e_1, \ldots, e_k \rangle$, vector of fragment spectra $\mathbf{T}$, their fragment-ion index $\mathbf{F}$, and a match tolerance $\epsilon > 0$.

Conceptually, calculating the match of spectrum E against all spectra from $\mathbf{T}$ is straightforward. For each fragment $e \in E$, we use a binary search to locate the fragments that are within tolerance ϵ in the sorted fragment-ion index $\mathbf{F}$. Because the fragment-ion index $\mathbf{F}$ keeps track of the parental indices, we then increase the matches for spectra at these parental indices. Nonetheless, we need to make sure that each fragment from the theoretical spectra $\mathbf{F}$ is counted at most once, such that we indeed calculate $\text{NMF}_\epsilon(T_a, E)$ for each $T_a \in \mathbf{T}$. For this, we utilize an additional array that keeps track of whether a fragment was already counted and increase the match only when it was not. This concludes the description of the algorithm, and we present the pseudocode of the function FAST-MATCH on listing 4.

Let us now prove that the algorithm on listing 4 calculates $\text{NMF}_\epsilon(T_a, E)$ for each spectrum T_a.

Theorem 3 (Correctness of the fast spectral match algorithm)**.** *Suppose a fragment-ion index* $\mathbf{F} = \langle \langle m_1, i_1 \rangle, \ldots, \langle m_l, i_l \rangle \rangle$ *for mass spectra* $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$*, a mass spectrum* E *and a match tolerance* $\epsilon \geq 0$*. The result* M *of the algorithm* FAST-MATCH *on listing 4 contains entries such that* $M_a = \text{NMF}_\epsilon(T_a, E)$*.*

Proof. We prove the theorem for a particular a so that $M_a = \text{NMF}_\epsilon(T_a, E)$. Thus, consider a spectrum $T_a = \langle t_1, \ldots, t_m \rangle \in \mathbf{T}$. Now suppose a fragment $e \in E$. The binary search for $e \in E$ obtains indices $f \leq t$, such that $m_f \geq e - \epsilon$ but $m_{f-1} < e - \epsilon$, and $m_t \leq e + \epsilon$ but $m_{t+1} > e + \epsilon$. Now suppose there is a fragment $t_j \in T_a$ such that $|t_j - e| \leq \epsilon$. If such a fragment was not yet matched, we need to make sure that M_a is increased. Because $\mathbf{F}$ contains mass fragments from all mass spectra in $\mathbf{T}$, it also contains t_j. Note that as $t_j \geq e - \epsilon$ and $t_j \leq e + \epsilon$, then for some $k \in \{f, \ldots, t\}$, $t_j = m_k$ and the index of the corresponding spectrum is $a = i_k$. In the next step, the algorithm checks whether the fragment j was not yet used and if it was not, it increases the match of M_a. Now suppose that no such fragment $t_j \in T_a$ exists such that $|t_j - e| \leq \epsilon$. We need to make sure that M_a is not increased. However, for every $t_j \in T_a$, $|t_j - e| > \epsilon$. As a result, there is no such $k \in \{f, \ldots, t\}$, such that $i_k = a$ and therefore M_a is not increased. The result then follows. □

Let us now analyze the time complexity of the algorithm depending on the number n of theoretical spectra in the fragment-ion index. We express the complexity based on the number of theoretical spectra because the length of the experimental spectrum and the lengths of the individual fragment spectra can be considered constant. In the worst-case scenario, the algorithm has to increase the spectral match for all theoretical spectra; thus, the worst-case time complexity is $\mathcal{O}(n)$. In the best-case scenario, the algorithm does not increase the match for any spectrum. However, we still need to initialize the two vectors M and C whose sizes depend linearly on n, and the best-case time complexity is thus $\Omega(n)$.

Listing 4: Fast calculation of spectral matches using fragment-ion index

```
Function FAST-MATCH(E, F, ε):
    Data: Mass-ordered fragment spectrum E = ⟨e_1, ..., e_k⟩
          Fragment-ion index F = ⟨⟨m_1, i_1⟩, ..., ⟨m_l, i_l⟩⟩ for spectra T = ⟨T_1, ..., T_n⟩
          Match tolerance ε ≥ 0
    Result: A vector M such that M_a = NMF_ε(T_a, E) for 1 ≤ a ≤ n
    begin
        /* Initialize a spectral match vector of size n */
        M ← VECTOR(0, n)
        /* Initialize a vector of size l indicating if a fragment from F was already */
        /* matched */
        U ← VECTOR(false, l)
        /* For each mass from the mass spectrum E */
        for e ∈ E do
            /* Use binary search to retrieve the locations within F at e ± ε */
            f, t ← LOCATE(e ± ε, ⟨m_1, ..., m_l⟩)
            /* NOTE: the locations f, t must be as follows: */
            /* m_f ≥ e − ε but m_{f−1} < e − ε */
            /* m_t ≤ e + ε but m_{t+1} > e + ε */
            for j ∈ ⟨f, ..., t⟩ do
                /* If the theoretical fragment was not matched yet */
                if not U_j then
                    /* Get the parental index a of the theoretical mass spectrum */
                    /* to which m_j belongs */
                    a ← i_j
                    /* Increase the spectral match with the theoretical spectrum a */
                    M_a ← M_a + 1
                    /* Mark the fragment as already matched */
                    U_j ← true
                end
            end
        end
        return M
    end
```

3.2.6.3 Matching against a mass-partitioned database

The FAST-MATCH procedure allows quickly calculating spectral matches of an experimental spectrum with fragment-ion-indexed theoretical spectra of candidate peptides. In computational proteomics, we are typically interested in having spectral matches of peptides that are within a particular precursor mass range $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$. Recall that the peptide enumeration algorithm from section 3.2.5 gives us a vector $\mathbf{Q}$ of peptides and their relative prior probabilities. We partitioned such a dataset $\mathbf{Q}$ into mass-binned datasets $\mathbf{Q}_b$ containing only peptides whose masses overlap with $M_b = \langle b \cdot w, (b+1) \cdot w \rangle$, for some fixed width w of each bin. We now describe an algorithm that uses such mass-binned datasets to calculate the spectral match with all peptides from $\mathbf{Q}$ that are within the mass range $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$.

In what follows, we assume that the fragment-ion indexes $\mathbf{F}_b$ were precomputed for each database portion $\mathbf{Q}_b$ and can be efficiently accessed. To calculate the spectral matches, we

locate the database bins that overlap with the precursor mass range $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$, and for each such bin b, load the fragment-ion index $\mathbf{F}_b$ and calculate the spectral match using the Fast-Match function. As the database bins $\mathbf{Q}_b$ will typically contain peptides outside of the $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$ range, we further restrict the peptides only to those that are within the precursor mass range of interest. In general, this concludes the description of the algorithm, and we provide its pseudocode on the listing 5.

Listing 5: Matching of fragment spectra against mass-binned fragment-ion-indexed database.

Function Match-Against-Database($E, \epsilon, \hat{m}_\mathrm{p}, \epsilon_\mathrm{p}$, $\boldsymbol{Q}$)**:**
  Data: Experimental mass spectrum E
    Fragment match tolerance ϵ
    Precursor mass $\hat{m}_\mathrm{p}$
    Precursor mass tolerance ϵ_p
    Mass-binned database $\mathbf{Q} = \mathbf{Q}_0 \oplus \cdots \oplus \mathbf{Q}_n$, each $\mathbf{Q}_b = \langle \mathbf{P}_b, \mathbf{R}_b \rangle^T$
  Result: A vector $\mathbf{D}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$ of peptides, their prior probabilities and spectral matches
  begin
    /* Initialize a vector that aggregates results over database portions */
    $\mathbf{D}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}} \leftarrow \langle\rangle$
    /* Obtain indices b of database portions such that $M_b \cap \langle \hat{m}_\mathrm{p} - \epsilon_\mathrm{p}, \hat{m}_\mathrm{p} + \epsilon_\mathrm{p}) \neq \emptyset$ */
    $B \leftarrow$ Locate($\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$, $\langle M_0, \ldots, M_n \rangle$)
    /* For each affected database bin b */
    for $b \in B$ **do**
      /* Get the fragment index for the corresponding portion */
      $\mathbf{F}_b \leftarrow$ Load-Fragment-Ion-Index($\mathbf{Q}_b$)
      /* Calculate the spectral match for all peptides within the index */
      $\mathbf{M}_b \leftarrow$ Fast-Match(E, $\mathbf{F}_b$, ϵ)
      /* Obtain indices of peptides that are within $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$ */
      $I \leftarrow$ Mass($\mathbf{P}_b$) $\in \langle \hat{m}_\mathrm{p} - \epsilon_\mathrm{p}, \hat{m}_\mathrm{p} + \epsilon_\mathrm{p} \rangle$
      /* Append the spectral matches */
      Append($\mathbf{D}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$, Zip($\mathbf{P}_b[I]$, $\mathbf{R}_b[I]$, $\mathbf{M}_b[I]$))
    end
    return $\mathbf{D}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$
  end

3.2.6.4 Memory-load optimization

The peptide database constructed for a particular minimal relative prior probability can be considerably large. For instance, in our database of human peptides with minimal relative prior probability $p_\mathrm{min} = 4 \cdot 10^{-6}$, and within the mass range of 700–3 000 Da, there are $2,217,966,178$ peptides. It is therefore often unreasonable to hold the whole fragment-ion index in memory, which is one of the reasons why we also partition the database into multiple portions. Nevertheless, even the size of the fragment-ion index $\mathbf{F}_b$ corresponding to a database portion $\mathbf{Q}_b$ can be quite large (e.g., hundreds of MBs for bin-width of 1 Da). As a result, its loading might thus negatively impact the otherwise fast calculation of the spectral matches. Herein, we describe a memory-load optimization, which often allows loading only a small subset of the fragment-ion index—tailored particularly to the currently analyzed experimental spectrum.

Foremost, the masses of fragments f_i in a fragment-ion index $\mathbf{F} = \langle\langle f_1, i_1\rangle, \ldots, \langle f_l, i_l\rangle\rangle$ repeat, and often to a substantial degree. For instance, in our human peptide dataset, only 5.6 % of fragment masses were unique on average (calculated over individual database portions $\mathbf{Q}_b$). Furthermore, rounding the fragment masses to 2 decimal points, which is adequate for typical fragment mass measurements, resulted in just 0.32 % of unique masses on average. From this perspective, it is, therefore, meaningful to store the fragment masses f_i of a fragment-ion index using run-length encoding. Further, as indicated above, one can also load just the part of the fragment-ion index that can be matched by the spectrum E at the tolerance ϵ. As ϵ is usually low (say, ≤ 0.5), this often allows loading a substantially smaller part of the fragment-ion index. We now proceed with the formal definition of a fragment-ion subindex for a particular spectrum E and a tolerance ϵ.

Definition 15 (Fragment-ion subindex of $\mathbf{F}$ for E and ϵ)**.** A fragment-ion subindex of $\mathbf{F} = \langle\langle f_1, i_1\rangle, \ldots, \langle f_n, i_n\rangle\rangle$ for experimental spectrum E and a match tolerance $\epsilon \geq 0$, denoted $\mathbf{F}^{E,\epsilon}$ is a subvector of $\mathbf{F}$,

$$\mathbf{F}^{E,\epsilon} = \langle\langle s_1, j_1\rangle, \ldots, \langle s_m, j_m\rangle\rangle,$$

such that $s_a \leq s_{a+1}$ and $\langle s, j\rangle \in \mathbf{F}^{E,\epsilon}$ if and only if $\langle s, b\rangle \in \mathbf{F}$ for some b and $|s - e| \leq \epsilon$ for some $e \in E$.

We now show that we can replace the complete fragment-ion index with the fragment-ion subindex on a particular spectrum when calculating the fast spectral match.

Theorem 4. *Suppose a fragment-ion index $\boldsymbol{F}$ for mass spectra $\boldsymbol{T}$, an experimental spectrum E, and a tolerance $\epsilon \geq 0$. Then*

$$\textsc{Fast-Match}(E, \boldsymbol{F}, \epsilon) = \textsc{Fast-Match}(E, \boldsymbol{F}^{E,\epsilon}, \epsilon).$$

Proof. The algorithm on listing 4 only ever accesses the parts of the fragment-ion index that are within the tolerance ϵ of some fragment $e \in E$. Furthermore, the absolute positions of individual entries of the fragment-ion index do not affect the result of the algorithm. The result then follows. □

The previous theorem thus shows that we can calculate the spectral match using a smaller, spectrum-dependent part of the fragment-ion index. To implement the approach, we first load the run-length-encoded fragment masses, and based on individual fragments in the experimental spectrum E, we calculate the indexes of the fragment-ion index which are necessary to load for the calculation. For completeness, we provide a pseudocode of the procedure Load-Fragment-Ion-Subindex that loads the fragment-ion subindex for E and ϵ (listing 6). Once loaded, we then directly use the fragment-ion subindex $\mathbf{F}^{E,\epsilon}$ instead of $\mathbf{F}$ in our mass-binned matching procedure described on listing 5, by replacing the call to Load-Fragment-Ion-Index with Load-Fragment-Ion-Subindex.

3.2.7 Calculation of $\mathrm{Pr}_{\max}$

Herein, we describe the calculation of $\mathrm{Pr}_{\max}$ of candidate peptides using the notions developed in the previous sections. Thus, suppose a fragment spectrum E, its measured precursor mass

Listing 6: Loading of a fragment-ion subindex for a spectrum.

Function LOAD-FRAGMENT-ION-SUBINDEX(E, ϵ, S)**:**
Data: Experimental mass-ordered mass spectrum $E = \langle e_1, \ldots, e_m \rangle$
Match tolerance $\epsilon \geq 0$
Storage system S for fragment-ion indexes
Result: $\mathbf{F}^{E,\epsilon}$, the fragment-ion subindex of $\mathbf{F}$ for spectrum E at tolerance ϵ
begin
/* Initialize the resulting fragment-ion index */
$\mathbf{F} \leftarrow \langle\rangle$
/* Initialize where the previous part of the already loaded */
/* fragment-ion index ended to prevent duplicities */
$t_{\text{previous}} \leftarrow -1$
/* Load run-length encoded (RLE) fragment masses */
/* $\mathbf{M}^r$ will contain runs of unique masses */
/* $\mathbf{M}^l$ will contain the lengths of the corresponding runs */
$\mathbf{M}^r, \mathbf{M}^l \leftarrow$ LOAD-RLE-FRAGMENT-ION-MASSES(S)
/* Calculate cumulative sums over the lengths (to transform the indices) */
$\mathbf{C} \leftarrow$ CUMULATIVE-SUM($\mathbf{M}^l$) $-$ FIRST($\mathbf{M}^l$)
/* $\mathbf{C} = \langle 0, M^l_1, M^l_1 + M^l_2, \ldots, \rangle$ */
/* For each fragment in the experimental spectrum */
for $e \in E$ **do**
/* Use binary search to retrieve the locations of $e \pm \epsilon$ within $\mathbf{M}^r$ */
/* NOTE: The following must hold for the located indices: */
/* $m_f \geq e - \epsilon$ but $m_{f-1} < e - \epsilon$ */
/* $m_t \leq e + \epsilon$ but $m_{t+1} > e + \epsilon$ */
$f, t \leftarrow$ LOCATE($e \pm \epsilon, \mathbf{M}^r$)
/* Set the part to load to at least the end of the previous part */
$f \leftarrow \max(t_{\text{previous}}, f)$
/* Update the end of the already loaded part, for the next iteration */
$t_{\text{previous}} \leftarrow t$
/* Transform RLE indices to linear indices */
$l_f, l_t \leftarrow \langle \mathbf{C}_f, \mathbf{C}_t \rangle$
/* Load the corresponding part of the fragment-ion index */
$\mathbf{F}_e \leftarrow$ LOAD-FRAGMENT-ION-INDEX-SUBSET(S, $l_f : l_t$)
/* Append the part of the fragment-ion index */
APPEND($\mathbf{F}, \mathbf{F}_e$)
end
return ***F***
end

$\hat{m}_\mathrm{p}$, and a precursor tolerance ϵ_p so that the true peptide for E is within $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. The function call MATCH-AGAINST-DATABASE$(E, \epsilon, \hat{m}_\mathrm{p}, \epsilon_\mathrm{p}, \mathbf{Q})$ gives us a vector

$$\mathbf{D}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} = \langle \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}, \mathbf{R}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}, \mathbf{M}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \rangle$$

of peptides, their prior probabilities and their spectral matches. In particular, for each $p \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, we have its relative prior probability $\Pr^\star(p)$ in $\mathbf{R}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, and its match $\Theta(p, E)$ with spectrum E in $\mathbf{M}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ (using NMF at fragment tolerance ϵ). Note that the peptides $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ are of the appropriate mass range $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$. Further, the dataset $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ is closed in the sense that all peptides in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ that are at least likely *a priori* as any peptide in $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ are in $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. In other words, for each $p \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, if $q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ and $\Pr^\star(q) \geq \Pr^\star(p)$ then $q \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. As a result, we have all the necessary ingredients to calculate the $\Pr_\mathrm{max}$ of each peptide $p \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$.

For each $p \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, let us denote $p^\star$ the set of all peptides in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ that are at least as likely *a priori* as p and which have at least as good agreement with E as p, thus

$$\begin{aligned} p^\star &= \{q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \,|\, \Pr^\star(q) \geq \Pr^\star(p) \text{ and } \Theta(q, E) \geq \Theta(p, E)\} \\ &= \{q \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \,|\, \Pr^\star(q) \geq \Pr^\star(p) \text{ and } \Theta(q, E) \geq \Theta(p, E)\}. \end{aligned}$$

To calculate $\Pr_\mathrm{max}$ of p, we assume that Θ is probabilistically-increasing, true-cause normalized, and random-cause normalized. Then, by Theorem 1, the maximal posterior probability of p is

$$\Pr_\mathrm{max}(p, E) = \frac{\Pr^\star(p)}{\sum\limits_{q \in p^\star} \Pr^\star(q)}.$$

3.2.8 Summarization

Let us briefly summarize the relevance and the relationships of the methods developed within the theoretical framework of computational proteomics. To provide a better mental picture, we also provide a graphical summary of these methods on **Fig. 3.2**. As illustrated in the figure, the methods can be categorized based on the frequency of their expected use. The first group of methods is involved in the infrequent construction of the highly optimized database of peptides, relative prior probabilities, and their fragment-ion indexes (left block). In contrast, the second group of methods deals with the repetitive action of matching an experimental fragment spectrum against the database (right block).

Let us reiterate the reasons for the construction of the database. To calculate the $\Pr_\mathrm{max}$ of a peptide p that has an agreement x with a spectrum E, we require all peptides that are at least as likely *a priori* as p, and those that have at least as good agreement x with E. Thus, we first defined the relative peptide prior probabilities (section 3.2.4), and then designed an algorithm that enumerates all peptides above minimal relative prior probability p_min (section 3.2.5). Because the constructed database can be large, we also decided to partition it into non-overlapping precursor-mass portions because it is common to match fragment spectra against peptides of a particular precursor mass range (section 3.2.5.3). Afterward, we described a fast spectral match algorithm (3.2.6), which allows us to quickly calculate matches for multiple peptides against precomputed fragment-ion indexes. In accordance, we constructed a fragment-ion index for each database portion. As the fragment-ion indexes for database portions can be also large, and of-

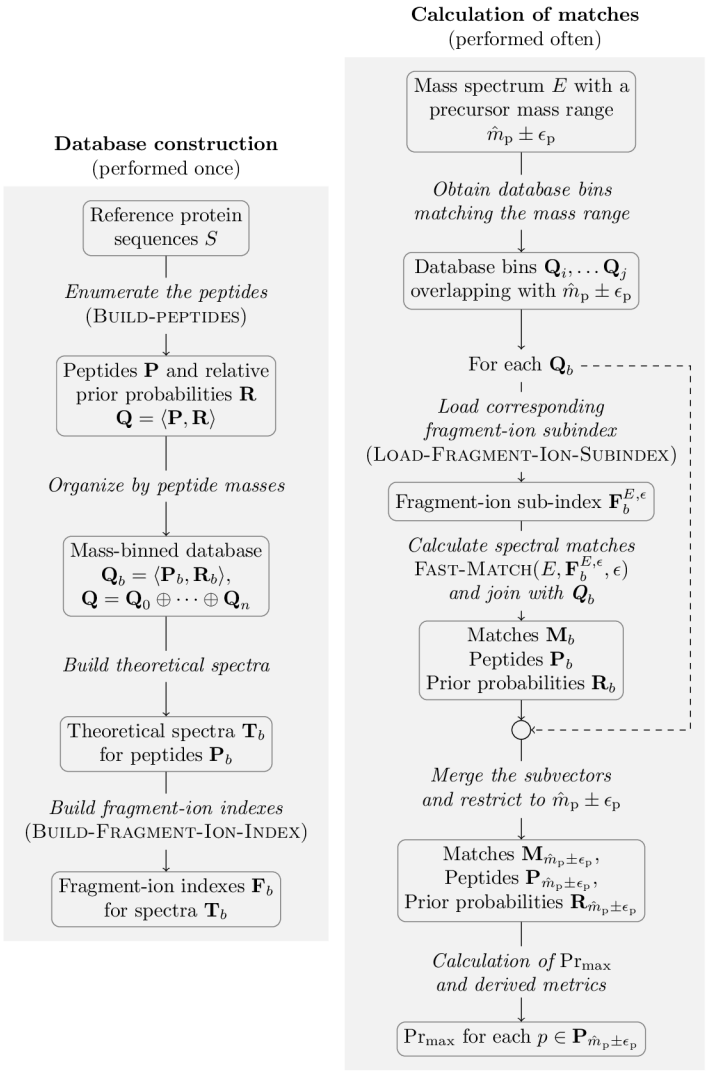


Figure 3.2: Data processing overview

The diagram depicts the schematics of the data processing. Overall, there are two major computational processes that are run at different times. The left part represents the infrequent construction of a deep prior-probability-aware peptide database, along with the prediction of spectra and their indexation. The right part represents the highly repetitive and fast matching of experimental spectra against the constructed database.

ten only a small part of them is needed for calculation of the match for a *particular* spectrum, we introduced memory-load optimization to allow quick loading only the required part of the fragment-ion index (section 3.2.6.4). In general, we were interested in having fragment matches for peptides within a particular precursor mass range $\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}$, and we introduced an algorithm that returns such matches for the mass-binned database (section 3.2.6.3). Finally, we used these notions to calculate Pr_max of all candidate peptides within $\mathbf{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$ (section 3.2.7). Thus, once the database is built, we can quickly access peptides, their relative prior probabilities, calculate matches with experimental spectra, and calculate Pr_max.

Chapter 4

Methods

The chapter builds on the notions introduced in the theoretical framework and develops less central methods required to answer our research questions. First, we describe the datasets employed in our research and discuss their utility in peptide detection (section 4.1). Afterward, we define external metrics for evaluating peptide detection performance, both in idealized circumstances and in more typical ones (section 4.2). Then, we develop several methods for downstream analyses of detected peptide variants, with applications to cancer research, research reproducibility, and forensics (section 4.3). Afterward, we describe adjustments to our peptide detection approach, along with the description of the software used in the comparisons (section 4.4). Finally, we describe CLAIRE—our software system that implements the methods developed in the thesis (section 4.5).

4.1 Datasets

Herein, we briefly describe the datasets used in the development and the evaluation of peptide detection methods. First, we describe a low-precursor-mass combinatorial peptide library that allows direct analysis of peptide detection performance in idealized conditions (section 4.1.1). Afterward, we introduce datasets we have chosen as representatives of proteomics data to investigate the detection performance in typical circumstances (section 4.1.2). For the latter, we also briefly include the description of the corresponding DNA or mRNA datasets, which served us to establish the correctness of detected peptide variants using their DNA/mRNA sequencing support.

4.1.1 Combinatorial peptide library

The synthetic combinatorial peptide library represents a well-defined dataset for the investigation of peptide detection methods. The peptide library, denoted $\mathbb{P}_L$, consists of 400 peptides of sequence LVVVGAxyVGK for all reference residues $x, y \in \mathbb{A}_\wedge$, and of 6,426 fragment mass spectra $\mathbb{M}_L \subseteq \mathbb{M}$. The fragment mass spectra for each peptide were measured independently, and we thus always know the peptide that produced a spectrum $m \in \mathbb{M}_L$. In turn, such knowledge allowed us to evaluate the correctness of their detection directly (section 4.2.1).

Overall, we designed the combinatorial library so that it fits two purposes. First, the library consists of homologous peptides of a relatively low precursor mass (840.54–1 212.70 Da), allowing

us to study their detection by direct application of our Bayesian method for calculation of posterior probabilities for all candidate peptides (section 3.1.3). Nevertheless, to simplify the computational analyses, we restricted the dataset to 2 144 fragment spectra with at most 10^8 candidate peptides per fragment spectrum. Second, the peptides from the library are of biological relevance as variant peptides of this structure are implicated in various cancers [78]. The data for the peptide library are available for download from the PRIDE repository [79] under identifier PXD013421, and several analyses of the data can be found in the Mendeley Data repository (DOI: http://dx.doi.org/10.17632/4jbxwkk5p2.1 and http://dx.doi.org/10.17632/3j95c7tm5t.1). For a more detailed treatment of the library, including its synthesis and various summarizations of its content, we refer the interested reader to our articles [1, 2].

4.1.2 Typical proteomics experiments

Typical computational proteomics datasets contain fragment mass spectra that are substantially harder to interpret than those in the synthetic combinatorial peptide library (section 4.1.1). To investigate peptide detection in typical circumstances, we analyzed three large-scale proteomics datasets comprising samples of cancer cell lines, patients' tumor samples, and blood samples from healthy individuals. Further, we focused on datasets that have the corresponding DNA or mRNA data available to allow us to indirectly establish the correctness of detected variant peptides using an external criterion (section 4.2.2). Overall, for the research presented in the thesis, we analyzed data from 163 biological samples, corresponding to 2 198 LC/MS fractions, amounting to around 1 TB of raw mass spectrometric data. We now turn to a brief description of the individual datasets.

Cancer samples As representative proteomes of cancer samples, we chose mass spectrometric measurements of proteomics samples from NCI_{60} cell line panel [11]. The NCI_{60} panel consists of 59 cancer samples of various types, and besides the measurements of proteomes, contains the data of exomes, transcriptomes, and SNP arrays [80–82] among others. Notably, the proteomics dataset allowed us to study the detection of peptide somatic variants as their presence is often elevated in some types of cancers. We downloaded the proteomics data for the NCI60 panel from PRIDE repository [79] (IDs: PXD005940, PXD005942, and PXD005946), and the DNA and mRNA data [81] from CellMiner [80]. For further description of the datasets, we refer the reader to our article [3].

Patients' tumor samples The samples aim to be representative of tumors of individual patients and are typically less clear-cut than the samples of cancer cell lines. In our analyses, we chose patients with colorectal cancer—the number of somatic variants in such cancers is often higher, making it more suitable for our detection purposes [83, 84]. We downloaded both the proteomics and the DNA data from the Clinical Proteomic Tumor Analysis Consortium (CPTAC) [85]. We note that the downloaded DNA data contained only somatic variants as the germline variants were under restricted access.

Family members To investigate the behavior of germline peptide variants, we studied blood samples of a 7-member Czech family from Moravia. The members of the family were as follows:

father and mother, their three biological daughters, and two biological monozygotic twin sons. We deposited the proteomics dataset into PRIDE archive [79] under identifier PXD013817, and the genetic dataset into European Genome–Phenome Archive [86] (ID: EGAD00001004949). Note that access to the genetic data requires approval from the corresponding committee that grants access to them. For a more detailed treatment of the dataset, we refer the reader to our article [3].

4.2 Performance of peptide detection

The section discusses two approaches for evaluating the performance of peptide detection, depending on the available information. If we know the correct peptide for a fragment spectrum, we can evaluate the performance directly (section 4.2.1). Otherwise, we depend on indirect validation, and we discuss a validation strategy based on evaluating the correspondence between detected peptide variants and the corresponding DNA or mRNA data (4.2.2).

4.2.1 Direct validation

Direct validation refers to a scenario when we know the correct peptide for each analyzed mass spectrum. Thus, suppose a set $\mathbb{M}_a \subset \mathbb{M}$ of analyzed mass spectra, and let us denote $\Gamma(m) = p$ the correct peptide p for a spectrum m. In what follows, we define the notion of precision and recall for a particular assignment of peptides to spectra $\mathbb{M}_a$. For convenience, we also allow the assignment of multiple peptides per spectrum.

Suppose

$$\Omega : \mathbb{M}_a \mapsto \mathcal{P}(\mathbb{P})$$

is a peptide detection approach that assigns peptides to each analyzed mass spectrum. Now, let us denote $\mathbb{M}_a^\Omega$ the set of spectra for which Ω assigned at least one peptide for a spectrum, thus:

$$\mathbb{M}_a^\Omega = \{m \in \mathbb{M}_a \,|\, \Omega(m) \neq \emptyset\}.$$

Further, let us denote $\mathrm{Pr}_\Omega(m)$ the probability that a randomly selected $p \in \Omega(m)$ is the correct peptide $\Gamma(m)$ of m. In other words,

$$\mathrm{Pr}_\Omega(m) = \begin{cases} \frac{1}{|\Omega(m)|} & \text{if } \Gamma(m) \in \Omega(m), \\ 0 & \text{otherwise.} \end{cases}$$

We define the precision of Ω on $\mathbb{M}_a^\Omega$ as

$$\mathrm{Precision}_\Omega = \frac{\sum\limits_{m \in \mathbb{M}_a^\Omega} \mathrm{Pr}_\Omega(m)}{|\mathbb{M}_a^\Omega|}.$$

Similarly, we define the recall of Ω on $\mathbb{M}_a$ as

$$\mathrm{Recall}_\Omega = \frac{\sum\limits_{m \in \mathbb{M}_a} \mathrm{Pr}_\Omega(m)}{|\mathbb{M}_a|}.$$

Note Our definitions of precision and recall extend the usual definitions of precision and recall. We extend them because, in the analysis of computational proteomics data, it sometimes happens that a particular approach provides multiple equally-good peptides for a spectrum. Instead of forcing a random selection of a peptide to calculate the precision or the recall using the usual definition, we allow their multiplicity directly.

4.2.2 Indirect sequencing-based validation

In typical proteomics samples, we usually do not know the correct peptide for a particular spectrum. As a result, the evaluation of peptide detection performance in such circumstances is much less straightforward. Nevertheless, the validation of peptides by an external criterion is possible in practice if we have the DNA or mRNA data corresponding to the proteomics data and design the experiment appropriately. Overall, the idea is to calculate the *sequencing support* of detected peptides. For a variant peptide, the chance of its sequencing support in DNA or mRNA dataset at random is often low, providing indirect evidence of its correct detection.

Let us now provide an overview of the section. In section 4.2.2.1, we define *a protein-coding mRNA* as a central structure for maintaining correspondence between proteins and both DNA and mRNA. Afterward, in section 4.2.2.2, we specify methods that align the detected variant peptides to a set of protein-coding mRNA. Then, we define the sequencing support of variants peptides in section 4.2.2.3 and discuss its relation to correctness of detected peptides in sections 4.2.2.4 and 4.2.2.5. Finally, we define the detection performance metrics in section 4.2.2.6 based on the sequencing support. To get a visual picture of the process, we present an illustration of calculation of sequencing support on figure **Fig. 4.1**, which corresponds to sections 4.2.2.1–4.2.2.3.

4.2.2.1 Protein-coding mRNA

To allow an indirect sequencing validation of detected peptide variants, we need to establish a correspondence between proteins, DNA and mRNA. Herein, we introduce the notion of a *protein-coding mRNA*, which will serve as a central structure for the purpose.

First, we start with a few preliminary notions. Let us denote $\mathbb{DNA}$ the set of DNA nucleotides, thus

$$\mathbb{DNA} = \{A, C, G, T\}.$$

Analogously, let us denote $\mathbb{RNA}$ the set of RNA nucleotides, thus

$$\mathbb{RNA} = \{A, C, G, U\}.$$

Proteins are encoded on DNA molecules called *chromosomes*, which are, for our purposes, sequences of $\mathbb{DNA}$ nucleotides. DNA is a double-stranded molecule, and each nucleotide forms a pair with its *complementary base*. The notion of a complementary base is important because proteins can be encoded on either strand. The complementary base of DNA nucleotide n, denoted

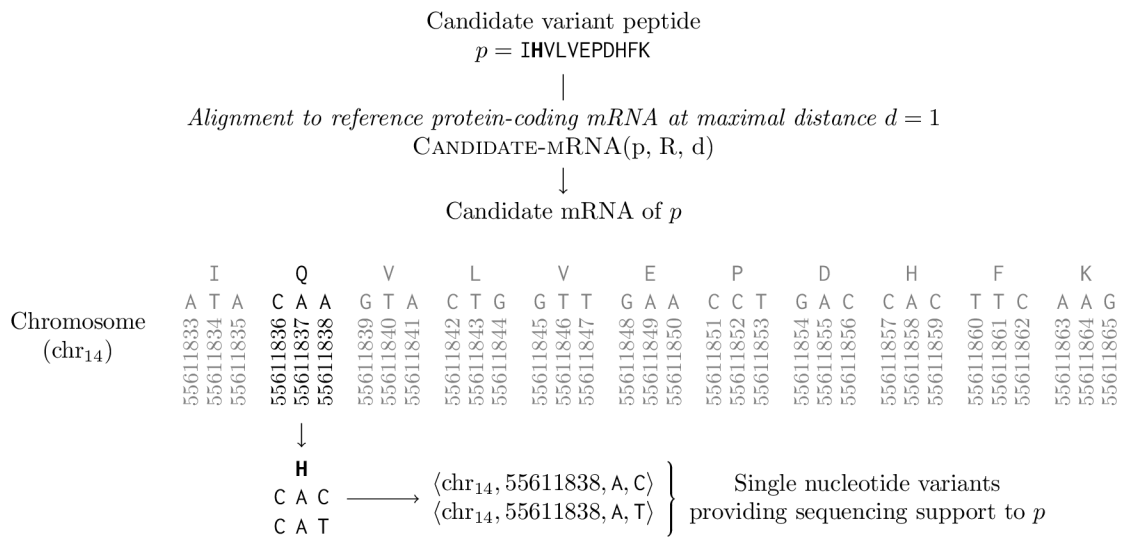


Figure 4.1: The establishment of sequencing support of a variant peptide.

The figure illustrates the process of deriving the sequencing support of a particular candidate variant peptide p. The peptide p is first aligned to the reference protein-coding mRNA R at a maximal nucleotide distance $d = 1$ using the procedure CANDIDATE-MRNA(p, R, d). Note that the variant peptide p differs from the corresponding reference peptide in the second amino acid (Q $\to$ H). At this chromosomal location, the reference amino acid Q is encoded by the RNA codon CAA. Alteration of the third nucleotide within the RNA codon (CA[A $\to$ C] or CA[A $\to$ T]) would then result, during protein translation, in the amino acid H. As a result, the complete chromosomal locations of these single nucleotide variants (SNVs) resulting in the amino acid substitution are then collected. Finally, if any such single nucleotide variant is present in the corresponding DNA/mRNA sequencing dataset V, the peptide p has a sequencing support in V, denoted $p \Leftarrow V$.

as n^{-1}, is as follows:

$$n^{-1} = \begin{cases} A & \text{if } n = T, \\ T & \text{if } n = A, \\ C & \text{if } n = G, \\ G & \text{if } n = C. \end{cases}$$

Note that $(n^{-1})^{-1} = n$.

Let us now introduce the link between DNA and RNA, which we will denote as a function $\nabla\colon \mathbb{DNA} \mapsto \mathbb{RNA}$. The function ∇ behaves as follows:

$$\nabla(n) = \begin{cases} n & \text{if } n \neq T \\ U & \text{if } n = T \end{cases}.$$

Finally, we can introduce the notion of a *protein-coding mRNA*.

Definition 16 (Protein-coding mRNA)**.** A protein-coding mRNA is a tuple $\langle c, s \rangle$, where c is a chromosome and s is a tuple

$$\langle \langle n_1, \ldots, n_m \rangle, \langle l_1, \ldots, l_m \rangle \rangle,$$

such that $m > 0$, $m \equiv 0 \mod 3$, each $n_i \in \mathbb{RNA}$, $l_i \in \mathbb{N}$, and either

a) $l_i < l_{i+1}$ for $1 \leq i < m$, and
$n_i = \nabla(c_{l_i})$ for $1 \leq i \leq m$,

or

b) $l_i > l_{i+1}$ for all $1 \leq i < m$, and
$n_i = \nabla(c_{l_i}^{-1})$ for $1 \leq i \leq m$.

The notion of the protein-coding mRNA is slightly technical, so let us clarify its meaning. First, a protein-coding mRNA sequence has a length $m = 3k$ for some $k \geq 1$. The reason for the multiplier 3 is that individual amino acids are encoded by three consecutive RNA nucleotides—each such triplet is called a *codon*. Further, protein-coding mRNA keeps track of chromosomal location of individual nucleotides, and each l_i represents the corresponding coordinate on the chromosome c. As protein can be encoded either in a forward direction or in a reverse direction of a particular chromosome, we have the respective conditions *a)* and *b)*. Finally, note that if the protein is encoded in the reverse direction, we utilize the complementary DNA nucleotides $c_{l_i}^{-1}$.

As of now, we have introduced the correspondence between DNA and mRNA. Going further, we introduce the link between RNA and amino acids. For convenience, we will use the notion of *RNA codon* c, which is simply a sequence of three RNA nucleotides, thus $c \in \mathbb{RNA}^3$. The RNA codons and amino acids are linked by means of a *genetic code*, which translates an RNA codon into an amino acid. We will represent a genetic code as a function

$$\Phi \colon \mathbb{RNA}^3 \mapsto \mathbb{A}_\wedge \cup \{*\},$$

mapping an RNA codon to an amino acid or to a special symbol $*$ that represents the end of translation. We refer to the codons that result in $*$ as *stop codons*.

Convention In what follows, we will assume that the protein-coding mRNA *does not contain the stop codon* within its sequence. Further, because we will analyze human protein data, the actual Φ will correspond to the human genetic code.

Notation Suppose a protein-coding mRNA $r = \langle c, s \rangle$, such that $s = \langle \langle n_1, \ldots, n_{3m} \rangle, \langle l_1, \ldots, l_{3m} \rangle \rangle$. We will often view s as a sequence of consecutive triples. We let Codons-RNA(s) be the sequence

$$\langle N_1, \ldots, N_m \rangle \in \mathbb{RNA}^{3m}$$

of RNA codons, such that each $N_i = \langle n_{3i-2}, n_{3i-1}, n_{3i} \rangle$. Similarly, we denote Codons-Locs(s) the sequence

$$\langle L_1, \ldots, L_m \rangle \in \mathbb{N}^{3m}$$

of corresponding chromosomal locations, so each $L_i = \langle l_{3i-2}, l_{3i-1}, l_{3i} \rangle$.

Having defined the translation of individual RNA codons, we expand the notion to the whole translation of protein-coding mRNA.

Definition 17 (Protein for a protein-coding mRNA)**.** Suppose a protein-coding mRNA $r = \langle c, s \rangle$ and a genetic code Φ. Let us have N the sequence of RNA codons of s, thus

$$N = \text{Codons-RNA}(s) = \langle N_1, \ldots, N_m \rangle.$$

Then, the protein sequence for r, denoted $\Phi(r)$ is

$$\langle \vdash, \Phi(N_1), \ldots, \Phi(N_m), \dashv \rangle.$$

In other words, the protein sequence for a protein-coding mRNA r is its codon-wise translation by means of the genetic code Φ, terminated by non-modified terminal residues $\vdash$ and $\dashv$.

Convention We defined peptides as finite sequences of amino acids, including their N- and C- termini. However, to simplify the exposition, we will often drop the terminal residues if the peptides (or proteins) have the standard non-modified terminals.

4.2.2.2 Matching peptides to protein-coding mRNA

We now turn to the matching of peptides against protein-coding mRNA to establish their candidate origin. Overall, the rationale is that once we have a variant peptide p, we are interested in the changes in a protein-coding mRNA that result, after its translation, in a protein that contains the peptide p. If such changes are present in the corresponding DNA or mRNA dataset, the variant peptide then has a sequencing support in the dataset.

First, we start with the reversal of the genetic code Φ. We will later utilize such a reversal to calculate the minimal distance between amino acids and RNA codons. Thus, suppose we have an amino acid $a \in \mathbb{A}_\wedge$, and a genetic code $\Phi \colon \mathbb{RNA}^3 \mapsto \mathbb{A}_\wedge \cup \{*\}$. Then, let us denote the inverse of the genetic code Φ as Φ^{-1},

$$\Phi^{-1} \colon \mathbb{A}_\wedge \cup \{*\} \mapsto \mathcal{P}(\mathbb{RNA}^3).$$

Then, the *candidate RNA codons* of amino acid a are $\Phi^{-1}(a)$. For instance, the candidate RNA codons of lysine are $\Phi^{-1}(\mathsf{K}) = \{\mathsf{AAA}, \mathsf{AAG}\}$.

We now turn our focus on calculating the minimal distance between a peptide and a corresponding sequence of RNA codons. First, we start with the Hamming distance of two RNA codons. Thus, suppose RNA codons $a, b \in \mathbb{RNA}^3$, $a = \langle a_1, a_2, a_3 \rangle$ and $b = \langle b_1, b_2, b_3 \rangle$. Then the *RNA codon distance* of a and b, denoted $\Delta(a, b)$ is the number of RNA nucleotides by which these RNA codons differ, thus:

$$\Delta(a, b) = \big|\{i \in \{1, \ldots, 3\} \mid a_i \neq b_i\}\big|.$$

As an example, $\Delta(\mathsf{AGG}, \mathsf{AAG}) = 1$.

Having defined the distance between two RNA codons, we now define a minimal distance between an amino acid and an RNA codon. Thus, suppose an RNA codon $r \in \mathbb{RNA}^3$ and an amino acid $a \in \mathbb{A}_\wedge$. Let us denote the candidate codons of a as C, thus $C = \Phi^{-1}(a)$. Then, *the minimal number of nucleotide changes in RNA codon r to give amino acid a*, denoted $\Delta_{\min}(r, a)$ is

$$\Delta_{\min}(r, a) = \min_{c \in C} \Delta(r, c).$$

We now expand the notion over a peptide and a sequence of RNA codons of the corresponding length. Thus, suppose a peptide $p = \langle p_1, \ldots, p_m \rangle$ and a sequence of RNA codons $C = \langle C_1, \ldots, C_m \rangle$. Then the *minimal number of nucleotide changes in RNA codons C to give*

peptide p, denoted $\Delta_{\min}(p, C)$, is

$$\Delta_{\min}(p, C) = \sum_{1 \leq i \leq m} \Delta_{\min}(p_i, C_i).$$

Having specified the prerequisites, we now turn to the notion of *candidate mRNA of a peptide* in a protein-coding mRNA r. Overall, the idea is to obtain all potential origins of a peptide p up to a particular distance d from the protein-coding mRNA r.

Definition 18 (Candidate mRNA of a peptide p in a protein-coding mRNA r up to a minimal distance d)**.** Suppose a protein-coding mRNA $r = \langle c, s \rangle$, and a peptide $p = \langle p_1, \ldots, p_k \rangle$. Let us have N the RNA codons of s,

$$N = \text{RNA-}\textsc{Codons}(s) = \langle N_1, \ldots, N_m \rangle,$$

and their chromosomal locations L,

$$L = \text{RNA-}\textsc{Locs}(s) = \langle L_1, \ldots, L_m \rangle.$$

Further, let us denote P the positions of the peptide p within s that are up to distance d, thus

$$P = \big\{ i \in \{1, \ldots, m - k + 1\} \,|\, \Delta_{\min}(p, \langle N_i, \ldots, N_{i+m-1} \rangle) \leq d \big\}.$$

Then, the set of candidate mRNA of a peptide p in protein-coding mRNA r up to a minimal distance d, denoted $\textsc{Candidate-mRNA}(p, r, d)$, is

$$\{ \langle c, \langle N_i \oplus \ldots \oplus N_{i+m-1}, L_i \oplus \ldots \oplus L_{i+m-1} \rangle \rangle \,|\, i \in P \},$$

where $\oplus$ denotes the concatenation operation.

Clarification Although a bit technical, the definition corresponds to something simple intuitively. In particular, for a given number d of allowed nucleotide changes, we obtain all consecutive substructures of the protein-coding mRNA r that can result in p by at most d nucleotide changes. For instance, for $d = 0$, the $\textsc{Candidate-mRNA}$ will return only those substructures that directly translate to the peptide p. For $d = 1$, the function will return substructures of r only if they can be translated to p by introducing at most one nucleotide change.

Finally, we expand the notion over a set R of protein-coding mRNA. Thus, suppose a peptide p, a set R of protein-coding mRNAs, and a distance $d \geq 0$. Then the corresponding set of candidate mRNA of p, denoted $\textsc{Candidate-mRNA}(p, R, d)$, is

$$\textsc{Candidate-mRNA}(p, R, d) = \bigcup_{r \in R} \textsc{Candidate-mRNA}(p, r, d).$$

4.2.2.3 Sequencing support of a variant peptide

Having the key prerequisites for the sequencing-based validation established, we now define the sequencing support of a variant peptide. Furthermore, to simplify the task, we will restrict our interest only to variant peptides originating from a single genomic location.

We start with the notion that introduces the most common genetic alteration, *a single nucleotide variant*, which we will often abbreviate as *SNV*. An SNV v is a quadruple $v = \langle c, l, r, a \rangle$, where c is a chromosome, l is the location of the DNA nucleotide change on the chromosome, $r, a \in \mathbb{DNA}$, r is the reference DNA nucleotide at position l on the chromosome c, thus $r = c_l$, and $a \neq r$.

We now turn to the notion of a variant peptide that can result from a single nucleotide variant, given a set of protein-coding mRNA R. A peptide p is called *SNV-peptide* if it is a non-reference peptide, thus

$$\textsc{Candidate-mRNA}(p, R, 0) = \emptyset$$

and can result from a single nucleotide variant, thus

$$\textsc{Candidate-mRNA}(p, R, 1) \neq \emptyset.$$

Further, we introduce the notion of *unique SNV-peptide* that can originate only from a single chromosomal location. Thus, an SNV-peptide p is a *unique SNV-peptide* if

$$|\textsc{Candidate-mRNA}(p, R, 1)| = 1.$$

Note In general, SNV-peptides represent a set of variant peptides that are the closest peptides to the reference sequences. Further, unique SNV-peptides can be aligned only to a single chromosomal location, which allows us to evaluate their sequencing support easily—by considering the presence of applicable SNVs in a single RNA codon.

Notation Having defined the notion of an SNV-peptide, we will also denote $D_{\text{SNV-peptides}}$ the set of all SNV-peptides for a set R of protein-coding mRNA, thus

$$D_{\text{SNV-peptides}} = \{p \in \mathbb{P} \mid p \text{ is an SNV-peptide for } R\},$$

where the set R of protein-coding mRNA will often be clear from the context.

We now turn to the notion of the introduction of a single nucleotide variant into a protein-coding mRNA sequence.

Definition 19 (Protein-coding mRNA r after introducing an SNV v, $r \oplus v$)**.** Suppose a protein-coding mRNA $r = \langle c_r, s \rangle$, $s = \langle \langle n_1, \ldots, n_m \rangle, \langle l_1, \ldots, l_m \rangle \rangle$, and a single nucleotide variant $v = \langle c_v, l, r, a \rangle$. Then a protein-coding mRNA after introducing an SNV v, denoted $r \oplus v$, is $\langle c_r, s \rangle$ if $c_r \neq c_v$, otherwise a tuple $\langle c_r, \langle \langle v_1, \ldots, v_m \rangle, \langle l_1, \ldots, l_m \rangle \rangle \rangle$ such that

$$v_i = \begin{cases} n_i & \text{if } l \neq l_i, \\ \nabla(a) & \text{if } l = l_i \text{ and } n_i = \nabla(r), \text{ and} \\ \nabla(a^{-1}) & \text{if } l = l_i \text{ and } n_i^{-1} = \nabla(r). \end{cases}$$

Note The first case of the definition tells that no change happens to a non-affected nucleotide. The remaining two cases allow a proper introduction of the SNV independently of its original

strand.

Example Let us clarify the definition with an example of the introduction of an SNV into a protein-coding mRNA. Suppose a protein-coding mRNA

$$r = \langle c, \langle \langle A, 100 \rangle, \langle U, 101 \rangle, \langle G, 150 \rangle \rangle^T \rangle$$

for some chromosome c, where A^T denotes the transposition. Now suppose a single nucleotide variant $v = \langle c, 101, T, A \rangle$. Then,

$$r \oplus v = \langle c, \langle \langle A, 100 \rangle, \langle A, 101 \rangle, \langle G, 150 \rangle \rangle^T \rangle.$$

Finally, we define the notion of a unique SNV-peptide being sequencing supported by an SNV. The notion of sequencing support is central for establishing the correctness of detected peptide variants.

Definition 20 (Sequencing support of a unique SNV-peptide p by a SNV v)**.** Suppose a unique SNV-peptide $p = \langle p_1, \ldots, p_n \rangle$, a set R of protein-coding mRNA, and a single nucleotide variant v. Let us have the unique mRNA of the unique SNV-peptide p up to distance 1, thus

$$\{\langle c, \langle N, L \rangle \rangle\} = \textsc{Candidate-mRNA}(p, R, 1).$$

Then the peptide p is sequencing supported by v, denoted $p \Leftarrow v$, if $p = \Phi(\langle c, \langle N, L \rangle \rangle \oplus v)$.

In other words, a unique SNV-peptide is sequencing supported if the sequencing dataset contains the change which gives rise to such a peptide after translation.

Finally, we conclude with the notion of sequencing support of a unique SNV-peptide p by a set V of SNVs. In particular, unique SNV-peptide p is sequencing supported by a set of SNVs V, denoted $p \Leftarrow V$, if there is a $v \in V$ such that $p \Leftarrow v$.

4.2.2.4 Correctness of sequencing-supported peptides

The previous sections established the notions required to derive whether a peptide has a sequencing support in a DNA or mRNA dataset. Now, we discuss the correctness of unique SNV-peptides that have sequencing support. Overall, the idea is to calculate the probability of sequencing support by chance—if such a chance is low, it provides indirect evidence that the variant peptide was detected correctly.

First, we describe a common form of a peptide database search that does not allow establishing the correctness by means of statistical significance of sequencing support. Afterward, we describe a requirement for a peptide database search such that it allows establishing the correctness. Finally, we discuss the notion of a probability of a presence of a particular SNV in a randomly selected sequencing dataset, allowing us to establish the correctness of peptide detection for multiple peptide detection methods.

Sequencing-derived protein database searches

We now describe a scenario in which the sequencing support is inadequate to establish the correctness of detected peptide variants. In practice, such a scenario corresponds to a common approach in proteogenomics [9], in which researchers construct a sample-specific protein dataset derived from the corresponding DNA or mRNA data. The protein dataset is then used in a database search for detecting variant peptides.

Let us now partially formalize the scenario. Suppose a reference set R of protein-coding mRNA and a dataset V of SNVs. Let us denote D_{proteins} the set of proteins that are the result of the introduction of these SNVs into the sequences R, followed by their translation, thus

$$D_{\text{proteins}} = \bigcup_{r \in R, v \in V} \Phi(r \oplus v).$$

Database search engines internally process D_{proteins} to yield a dataset of peptides that they match against the fragment mass spectra. Let us denote such processing as $\Omega : \mathbb{P} \mapsto \mathcal{P}(\mathbb{P})$, and denote the dataset of resulting peptides as D_{peptides}. Then,

$$D_{\text{peptides}} = \bigcup_{p \in D_{\text{proteins}}} \Omega(p).$$

Such processing typically includes cutting the sequences by specified enzyme or including potential modifications. Importantly, let us assume that the database search engine *does not consider any amino acid substitutions.*

Now, let us have D the set of SNV-peptides that a database search approach considers for matching against fragment mass spectra, thus

$$D = D_{\text{peptides}} \cap D_{\text{SNV-peptides}}.$$

In this approach, all peptides $p \in D$ have a sequencing support in V, formally $p \Leftarrow V$. In other words, the probability of sequencing support by chance for all SNV-peptides equals one, thus

$$\Pr(p \Leftarrow V \,|\, p \in D) = 1.$$

Such a form of database search thus fundamentally prevents validating the detected SNV-variant peptides by statistical significance of sequencing support.

Database searches considering a large number of variants

We now turn to a situation when the sequencing support *can* establish the correctness of detected peptide variants. Suppose that a database search approach internally considers a large number of SNV-peptides for matching against the fragment mass spectra. Let us again denote such peptides D as in the previous section. Then, the probability of an SNV-peptide p having sequencing support in V, $\Pr(p \Leftarrow V \,|\, p \in D) = c$, will be typically low. For instance, if such c is around $c \leq 10^{-2}$, the sequencing support is unlikely to occur by chance alone. As a result, we have indirect reasons to conclude that such a peptide was correctly detected.

Sequencing support based on frequency of variants in the population

We now turn to a scenario in which we utilize the probability that a particular variant v is present in a sequencing dataset. Let us denote the probability that an SNV v is present in a randomly selected variant dataset V as

$$\Pr(v \in V).$$

Now, suppose that a peptide detection approach detects a unique SNV-variant peptide p. For simplicity, let us further assume that there is only one applicable SNV v that can result in p. Then,

$$\Pr(p \Leftarrow v) = \Pr(v \in V).$$

Thus, if the nucleotide variant v corresponding to peptide p is unlikely, its sequencing support is also unlikely due to chance. Again, we have indirect reasons to interpret the presence of the sequencing support as evidence that the variant peptide was detected correctly.

Note A unique SNV-peptide p can still be supported by different candidate SNVs $v^\star$ within the same RNA codon. In such case, we let

$$\Pr(p \Leftarrow v) = \max_{v \in v^\star} \Pr(v \in V).$$

4.2.2.5 Incorrectness of sequencing-unsupported peptides

In the previous section, we have discussed the correctness of unique sequencing-supported SNV-peptides. For a suitably designed experiment, the sequencing support of SNV-peptides can be thus often interpreted as evidence for their correctness. However, the situation with incorrectness of detection of SNV-peptides based on lack of sequencing support is more complicated. Overall, a correctly detected SNV-peptide might not have a sequencing support, depending on various circumstances. We first discuss the scenario when the peptide p indeed originated from an SNV v and discuss what affects the presence of v in a sequencing dataset V (and thus whether $p \Leftarrow V$). Afterward, we discuss the possibility that the SNV-peptide p did not originate from an SNV.

SNV-peptide originating from an SNV variant

Suppose that a peptide detection approach has correctly detected a unique SNV-variant peptide p for a particular fragment spectrum m, thus $p = \Gamma(m)$. Furthermore, suppose that the peptide p originated (biologically) from the corresponding SNV. In what follows, let us denote the sequencing dataset as V.

Let us first consider an extreme situation, wherein the variant dataset V is complete in the sense that it contains all nucleotide variants (and that SNV-peptides can not originate by other means). Then, a correct SNV-peptide p will always have a sequencing support, thus

$$\Pr(p \Leftarrow V \,|\, p = \Gamma(m)) = 1.$$

The lack of sequencing support would then imply that the peptide is detected incorrectly.

However, the situation is more complicated in practice. At minimum, we need to consider at least two classes of variants, which have different probabilities that the variant will be in V. The

two types of variants in consideration are *germline* (or inherited) variants and *somatic* variants, and we now turn to their discussion.

Germline (or inherited) variants The inherited variants are, under most circumstances, present in each cell and are thus generally easier to detect by sequencing. As a result, the probability of the presence of a germline variant v in the corresponding sequencing dataset V is thus typically high. Formally,

$$\Pr\big(v \in V \,|\, \textsc{Germline}(v)\big) = g,$$

for g being a rather high value, say $g \approx 0.9$. Therefore, the lack of sequencing support for a germline variant v likely indicates that the variant peptide was detected incorrectly.

Somatic variants The situation with somatic variants is much more complicated. Foremost, the somatic variants can be present in an unknown proportion of cells. The probability of their presence in the sequencing data thus depends on many factors, e.g., the sample itself, its processing, and the computational analysis. Nevertheless, it is reasonable to assume that the probability is lower than for the germline variants. Overall, let us denote the probability as

$$\Pr(v \in V \,|\, \textsc{Somatic}(v)) = s,$$

where s is often unknown. Still, in a sample that contains mostly the tumor with the somatic variant v, it is reasonable to expect that s is reasonably high, say around $s \approx 0.7$. Otherwise, the s might be quite low. In summary, the lack of sequencing support for somatic variants does not necessarily imply the incorrectness of detection of the corresponding peptide variants.

Categorization of variants as germline and somatic Another complication arises because we do not know whether a particular SNV-peptide originated from a somatic or a germline variant. Nevertheless, germline variants are rather common in the human population, while somatic variants are relatively rare. Because we have estimates $\lambda(v)$ on the frequency of individual SNVs v in the human population [74, 75], we categorized the nucleotide variants as follows. If $\lambda(v) \leq 10^{-3}$, or its population frequency was unknown, we categorized v as a somatic variant. Otherwise, we categorized v as a germline variant.

SNV-peptide resulting from another process

The situation is further entangled because a correctly detected SNV-peptide p can result from processes other than an SNV. We describe two processes that might result in the correct detection of an SNV-peptide and its lack of sequencing support.

Protein synthesis errors The first process refers to the synthesis of proteins, which is prone to errors on the order of 1 per 10^{-3} to 10^{-4} synthesized residues [87]. Let us reiterate that the mass spectra we analyze are from the data-dependent acquisition strategy and are thus systematically biased towards more abundant peptides (section 2.1). As protein concentration

ranges over ten orders of magnitude [88], we would expect to see some variant peptides from highly-abundant proteins, which are, however, the result of protein synthesis errors.

RNA editing The second process is RNA editing [89], which happens on the RNA level and thus, the corresponding change would not be present in a DNA-level dataset. Still, one can test whether a particular variant corresponding to an SNV-peptide is present in an RNA editing database [90], which would provide partial evidence for its correct detection.

4.2.2.6 Detection performance metrics

Herein, we establish the metrics for evaluating the performance of variant peptide detection. Suppose that a particular peptide detection approach Ω gives a set $P = \{p_1, \ldots, p_n\}$ of unique SNV-peptides. Further, suppose we have the DNA or mRNA dataset V that corresponds to the analyzed mass spectra. Now, let us denote P^+ the subset of peptides in P that have a sequencing support in V, thus

$$P^+ = \{p \in P \,|\, p \Lleftarrow V\}.$$

Precision We define the precision of the approach Ω as

$$\text{Precision}_\Omega = \frac{|P^+|}{|P|}.$$

Claimed variants The situation with the recall of an approach is less straightforward compared to the direct validation (section 4.2.1). Therein, we know the correct peptide for each fragment spectrum, allowing us to establish the recall of a detection approach easily. In typical experiments, however, we do not know the correct peptides for each spectrum, and we thus also do not know the total number of unique SNV-peptides for a given set of mass spectra. Instead, we will use an absolute metric—the total number of variant peptides claimed by an approach Ω, thus

$$\text{Claimed-Variants}_\Omega = |P|.$$

4.3 Downstream applications

In this section, we will describe three methods for downstream application of the detected peptide variants. First, we introduce a method that calculates the probability that a proteomics sample has originated from a particular DNA origin given their variant overlap (section 4.3.1). Next, we introduce a method for calculating the statistical significance of a variant match applicable to datasets with a large number of analyzed samples, herein employed in the analysis of NCI_{60} datasets (section 4.3.2). Finally, we introduce a method for estimating the rate of protein variation and an analogous measure for genetic variation (section 4.3.3).

4.3.1 Probability of DNA origin

Herein, we propose a method that assigns probabilities to a set of candidate DNA origins for a proteomics sample based on detected peptide variants. Thus, suppose that a peptide detection approach results in a set P of unique SNV-peptides, and let us denote N the set of the SNVs

that correspond to them. Further, let us denote the candidate DNA origins against which we will match N as $\mathbb{O} = \{O_1, \ldots, O_n\}$, wherein each O_i is a set of SNVs. Finally, let us denote the set of all DNA origins, hence the whole population, as $\mathbb{O}^+$ ($\mathbb{O} \subseteq \mathbb{O}^+$).

In what follows, we will first assume a particular behavior over the relative probabilities of the candidate DNA origins, given their match with N. These assumptions allow us to calculate the relative probabilities of individual origins if each origin $O_i \in \mathbb{O}$ has the *same* number of matching variants with N. Afterward, we propose a relationship for the probabilities that extends the previous behavior, allowing us to derive probabilities even when the numbers of matching variants differ.

Equivalent variant overlaps

We start with the assumption of equivalence of probabilities of origins when these have the same overlap with variants N. Thus, suppose that two candidate DNA origins have the same overlap with variants N. We assume that the probability of either origin being the true origin O_t is equal. Thus, if $O_a \cap N = O_b \cap N$, then

$$\Pr(O_t = O_a) = \Pr(O_t = O_b). \tag{4.1}$$

Origins differing each by a single variant

Suppose origins O_a, O_b, and their overlaps with N,

$$O_a \cap N = N_{ab} \cup \{v_a\} \text{ and } O_b \cap N = N_{ab} \cup \{v_b\},$$

such that

$$v_a, v_b \notin N_{ab}, v_a \neq v_b.$$

Thus, the origins O_a, O_b have almost the same match with N except that each matches a different variant in N. We are interested in which of the two origins O_a and O_b is more likely given their variant match with N.

Let us denote the origins in $\mathbb{O}^+$ having variants N as $N^{\triangle}$. We approach the problem indirectly by assuming a particular behavior of probabilities that the true origin O_t is *within* $O_a^{\triangle}$ vs. within $O_b^{\triangle}$. In particular, we assume that such probabilities are equal, thus

$$\Pr(O_t \in O_a^{\triangle}) = \Pr(O_t \in O_b^{\triangle}). \tag{4.2}$$

However, even if such probabilities are equal, the *number* of individual origins within them might differ. In particular, the expected ratio of the number of such origins is inversely proportional to the population frequencies $\lambda(v_a), \lambda(v_b)$ of the corresponding variants, thus

$$\frac{|O_a^{\triangle}|}{|O_b^{\triangle}|} = \left(\frac{\lambda(v_a)}{\lambda(v_b)}\right)^{-1}.$$

For instance, suppose $\lambda(v_a) = 0.1$ and $\lambda(v_b) = 0.5$. Then, we would expect that there are $\frac{\lambda(v_b)}{\lambda(v_a)} = 5$ times more origins having a variant v_b compared to those having a variant v_a.

Finally, we turn to the individual origins *within* $O_a^{\triangle}$ and $O_b^{\triangle}$. Then, assuming that each such individual origin is equally likely *a priori*, we have

$$\frac{\Pr(O_t = O_a)}{\Pr(O_t = O_b)} = \frac{|O_a^{\triangle}|^{-1}}{|O_b^{\triangle}|^{-1}}.$$

Thus, given our assumptions, the probabilities of origins O_a and O_b are inversely proportional to the population frequencies of the additional variant.

Different number of matching variants

The previous assumptions allow us to establish the probability of individual origins if for each O_i,

$$|O_i \cap N| = k. \tag{4.3}$$

We now propose a particular behavior over the relative probabilities of individual origins such that the behavior agrees with equations 4.1 and 4.2 when the number of matching variants is equal but is applicable even when the number of matching variants differs. In particular, for two origins O_a and O_b, we assume that the ratio of their probabilities is the inverse of the product of population frequencies of matching variants, thus

$$\frac{\Pr(O_a)}{\Pr(O_b)} = \left(\frac{\prod_{v \in O_a \cap N} \lambda(v)}{\prod_{v \in O_b \cap N} \lambda(v)} \right)^{-1}.$$

Furthermore, in our applications, we will assume that the true origin is always among the candidate origins, and we will normalize the probabilities to sum to one.

4.3.2 Statistical significance of a variant match

We now describe a method for calculating the statistical significance of a variant match between two samples, applicable if these are a part of large datasets. Further, we allow the variant match to be calculated over nucleotide variants detected using different approaches. Although the method remains rather general, we formulate it directly for the analysis of variant data from the dataset of NCI_{60} cancer cell lines (section 4.1.2).

The task Let us thus denote Γ a variant detection approach that maps an NCI_{60} sample to a set of single nucleotide variants, thus

$$\Gamma \colon \text{NCI}_{60} \mapsto \mathcal{P}(\mathbb{SNV}).$$

Given samples $s, t \in \text{NCI}_{60}$ and approaches Γ_a and Γ_b, our aim is to calculate the significance of their variant match. In what follows, we first define the agreement function and then derive the probability of observing such an agreement by chance.

4.3.2.1 Calculation of variant match

We now specify the calculation of the agreement between detected variants. Overall, we evaluate the agreement on a large set of human germline single nucleotide variants, denoting such set as $\mathbb{SNV}_{\Cap} \subset \mathbb{SNV}$. Now suppose samples $s, t \in \text{NCI}_{60}$ and detection approaches Γ_a and Γ_b. Let us have a function

$$f_s^{\Gamma}(v) \colon \mathbb{SNV}_{\Cap} \mapsto \{0, 1\},$$

indicating whether a variant v was detected using an approach Γ in a sample s, thus

$$f_s^{\Gamma}(v) = \begin{cases} 1 & \text{if } v \in \Gamma(s) \\ 0 & \text{otherwise.} \end{cases}$$

Then, we calculate the agreement $\Theta(s, t)$ as a Spearman's ρ correlation coefficient over all germline variants $\mathbb{SNV}_{\Cap} = \{v_1, \ldots, v_n\}$, thus

$$\Theta(s, t) = \rho(\langle f_s^{\Gamma_a}(v_1), \ldots, f_s^{\Gamma_a}(v_n)\rangle, \langle f_t^{\Gamma_b}(v_1), \ldots, f_t^{\Gamma_b}(v_n)\rangle).$$

4.3.2.2 Calculation of statistical significance

Having established the agreement of a variant match, we now turn to the calculation of its statistical significance. Overall, we will calculate the p-values of the variant match for two related null models and then combine their p-values.

Notation In NCI_{60} samples, there are samples $a \neq b$ that are genetically related. When constructing a null model of the variant match, we will thus not use the agreement between such samples. We introduce the notation

$$\text{NCI}_{60}^{\nleftrightarrow s} \subset \text{NCI}_{60},$$

which represents the set of samples that are not genetically related to a sample $s \in \text{NCI}_{60}$. We refer an interested reader to more information about genetically-related samples among NCI60 samples to our article [3].

Null models of variant match

We now turn to the specification of the two null models of a variant match between s and t; $s, t \in \text{NCI}_{60}$. Herein, we will describe just one such model, noting that we exchange the roles of s and t to obtain the other model.

As the NCI_{60} panel contains many samples (59), we build the null model from variant matches against all genetically unrelated samples. Thus, given samples $s, t \in \text{NCI}_{60}$, let us have a vector X_s that contains agreement of s with each $t_i \in \text{NCI}_{60}^{\nleftrightarrow s}$. Thus, we have

$$X_s = \langle \Theta(s, t_1), \ldots, \Theta(s, t_n)\rangle,$$

where $n = |\text{NCI}_{60}^{\nleftrightarrow s}|$.

Now, we assume that X_s follows the normal distribution for some parameters μ_s, σ_s^2 which we obtain by fitting the distribution, thus $X_s \sim \mathcal{N}(\mu_s, \sigma_s^2)$. We then use the normal distribution to calculate the statistical significance of the match.

Significance of variant match

We now specify the significance of the variant match. For the first null model, the probability of observing a given match x at random is

$$p_s = \Pr(\Theta(s,t) \geq x) = \Pr(\mathcal{N}(\mu_s, \sigma_s^2) \geq x).$$

For the other null model, the probability of observing the match x at random is

$$p_t = \Pr(\Theta(s,t) \geq x) = \Pr(\mathcal{N}(\mu_t, \sigma_t^2) \geq x).$$

We combine the p-values using the harmonic mean procedure [91], which does not require the assumption of independence between the tests. As a result, the p-value p of the match between s and t is then

$$p = \left(\frac{p_s^{-1} + p_t^{-1}}{2} \right)^{-1}.$$

4.3.3 Large-scale rate of variation

In this section, we define two metrics that evaluate the large-scale behavior of variation on both protein and gene level. Our overall aim is to detect so-called *hypermutated tumors*, tumors with a substantially higher rate of somatic variation. In turn, such tumors are treated more efficiently using immunotherapy [83, 84, 92], and thus the knowledge of protein mutation rate might help in selecting preferable cancer treatment.

4.3.3.1 Protein variation rate

We now introduce the notion of a protein variation rate given detected peptide variants and reference protein sequences. In doing so, we will sum up the total number of variant amino acids detected and calculate its fraction to the total number of reference amino acids detected. Although the overall goal is straightforward, there are a few technical aspects to the calculation, and we now describe it in more detail.

Reference amino acids

We start with the calculation of detected reference amino acids. First, we introduce the *sequence coverage* of a protein-coding mRNA $r = \langle c, s \rangle$ by a set of peptides P. Intuitively, the sequence coverage is the number of amino acids of r explained by peptides P. Let us thus denote the protein sequence of r as $q = \Phi(s) = \langle q_1, \ldots, q_k \rangle$. For each peptide p, we denote I_p the indices of amino acids that are covered by the peptide p, thus

$$I_p = \bigcup \left\{ \{i, \ldots, j\} \,\middle|\, p = \langle q_i, \ldots, q_j \rangle \right\}.$$

Then, *the sequence coverage of r by P*, denoted Protein-Coverage(r, P), is the size of the union of all these indices, thus

$$\textsc{Protein-Coverage}(r, P) = \Big| \bigcup_{p \in P} I_p \Big|.$$

In computational proteomics, it sometimes happens that a detected peptide p is a subsequence of multiple reference protein-coding mRNAs. For our notion of protein variation rate, it is beneficial if each such peptide p is assigned to a single protein-coding mRNA. Such an assignment of peptides to a single reference protein is known as *protein inference*, and multiple approaches for this purpose exist [93]. Nevertheless, we will consider just a general function Υ that performs the protein inference. Thus, suppose a set of reference peptides P_{ref} and a set R of reference protein-coding mRNAs. The protein inference is a function $\Upsilon : P_{\text{ref}} \mapsto R$ such that if $\Upsilon(p) = r$, then p is a subsequence of r, thus Candidate-mRNA$(p, r, 0) \neq \emptyset$.

Having introduced the prerequisites, we now turn to the total count of detected reference amino acids. Suppose a set P_{ref} of reference peptides, a set R of reference protein-coding mRNA, and a protein inference $\Upsilon\colon P_{\text{ref}} \mapsto R$. Then, the *total count of reference amino acids*, denoted Ref-Amino-Acids(P_R, S, Υ), is the sum of protein coverage by all these peptides in their respective proteins, thus

$$\textsc{Ref-Amino-Acids}(P_{\text{ref}}, R, \Upsilon) = \sum_{r \in R} \textsc{Protein-coverage}(r, \Upsilon^{-1}(r)).$$

Variant amino acids

We now define an analogous measure for variant amino acids. Because particular variant amino acid can be detected by different (but overlapping) unique SNV-peptides, we will count the number of variant codons corresponding to the unique SNV-peptides. Let us first define the notion of mRNA of a variant codon.

Definition 21 (mRNA of a variant codon for unique SNV-peptide)**.** Suppose a unique SNV-peptide $p = \langle p_1, \ldots, p_m \rangle$, and let us have its single candidate mRNA $\langle c, \langle N_1 \oplus \ldots \oplus N_m, L_1 \oplus \ldots \oplus L_m \rangle \rangle$. As p is an SNV-peptide, let us denote i the only position where $p_i \neq \Phi(N_i)$. Then the mRNA of the variant codon of p, denoted Variant-Codon-mRNA(p), is

$$\textsc{Variant-Codon-mRNA}(p) = \langle c, \langle N_i, L_i \rangle \rangle.$$

The function Variant-Codon-mRNA thus gives, for a unique SNV-peptide p, the particular substructure of protein-coding mRNA, which corresponds to the single amino acid difference. Now, to calculate the total number of variant amino acids detected, we get the size of the set of variant codons for given SNV-peptides. Thus, suppose a set P of unique SNV-peptides. The *total count of variant amino acids*, denoted Var-Amino-Acids(P), is the number of corresponding variant codons, thus

$$\textsc{Var-Amino-Acids}(P) = \big|\{\textsc{Variant-Codon-mRNA}(p) \,|\, p \in P\}\big|.$$

Finally, we define the protein variation rate.

Definition 22 (Protein variation rate)**.** Suppose a set P_{var} of unique SNV-peptides, a set R of protein-coding mRNA, a set P_{ref} of reference peptides, and a protein inference Υ. Then the protein variation rate is

$$\frac{\textsc{Var-Amino-Acids}(P_{\text{var}})}{\textsc{Ref-Amino-Acids}(P_{\text{ref}}, R, \Upsilon)}.$$

4.3.3.2 Gene variation rate

We now turn to the analogous measure of gene variation calculated over a DNA sequencing dataset V. To obtain closer correspondence with the protein variation, we restrict the calculation over chromosomal locations of protein-coding mRNA. Let us thus denote L_R the set of all chromosomal locations over protein-coding mRNAs R:

$$L_R = \{\langle c, l\rangle \mid l = l_i \text{ for some } \langle c, \langle\langle n_1, \ldots, n_m\rangle, \langle l_1, \ldots, l_m\rangle\rangle \in R\}.$$

Definition 23 (Gene variation rate)**.** Suppose a DNA variant dataset V and a protein-coding mRNAs R. Let us denote V_R the set of variants in V that overlap with the chromosomal locations L, thus $V_R = \{\langle c, l, r, a\rangle \in V \mid \langle c, l\rangle \in L_R\}$. Then the gene variation rate is

$$\frac{|V_R|}{|L_R|}.$$

4.4 Data analysis

Herein, we describe methods that we employed for the analysis of data presented in the results (chapter 5). First, we focus on the combinatorial peptide library, wherein we specify prior models and adjustments to the calculation of posterior probabilities (section 4.4.1). Afterward, we focus on the analysis of typical proteomics experiments, describing methods to counteract precursor measurement errors, adjustment of the prior probability model, and relaxation of $\Pr_{\max}$ (section 4.4.2). Finally, we describe the detailed configuration of software utilized in our comparisons, along with the parameters for our more realistic prior model (section 4.4.3).

4.4.1 Combinatorial peptide library

Herein, we accommodate the notions introduced in the theoretical framework to the circumstances applicable in the analysis of our combinatorial peptide library. First, in section 4.4.1.1, we describe peptide prior probability models that are specifications of the general ones introduced in the section 3.2.3, along with an additional prior model of sequence tags. Afterward, in section 4.4.1.2, we adjust the Bayesian detection model from section 3.1.3 to the analysis of the combinatorial peptide library. Therein, we introduce a relativization of peptide-spectrum agreement functions and describe a method for predicting the distribution of true matches for a particular fragment spectrum—allowing us to obtain more precise posterior probabilities.

4.4.1.1 Prior models utilized in the analyses

We now describe the prior models that we utilized to analyze the combinatorial peptide library.

Prior models based on enzymatic cleavage

Herein, we utilize the cut-after-residue model from section 3.2.3.3, and decompose it to see its partial effects on peptide detection. In particular, we consider two partial models: one which models the cutting behavior at the last residue, and another that considers the absence of cuts within the peptide. Multiplication of prior probabilities in these models then gives the model described in the section 3.2.3.3. In what follows, let us have a peptide $p = \langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$ and a particular expected proportion of cuts $\alpha(r)$ after residue r.

Model based on cutting just the last residue The prior model that considers only cutting of the last residue at the C-terminal then behaves as follows:

$$\Pr^{\star}(p) = \alpha(p_n)$$

The relative prior probability of p is thus just the probability of cutting after the last residue p_n.

Model based on the absence of cuts within the sequence The prior model that considers the absence of cuts within the sequence then behaves as follows:

$$\Pr^{\star}(p) = \left(\prod_{i=1}^{n-1} 1 - \alpha(p_i) \right).$$

The relative prior probability of p is thus multiplication of not cutting residues up to the last residue (excluding the last one).

Correct pattern prior

The correct pattern prior models the prior knowledge when we expect a single reference sequence (section 3.2.3.4), and we specify prior probabilities based on the distance to it. In line with the more general model, we employ a distance function $\Delta \colon \mathbb{P} \times \mathbb{P} \mapsto \mathbb{R}^{+}$, i.e., Levensthein distance, to the sequence LVVVGA**XX**VGK. Note that although **X** is not an amino acid, and thus LVVVGA**XX**VGK $\notin \mathbb{P}$, we allow such a sequence just for computing the distance. Then, the relative prior probabilities of peptides $p \in \mathbb{P}$ are

$$\Pr^{\star}(p) = c^{\Delta(p,q)},$$

for a c indicating the multiplicative decrease in prior probability with a unit increase in distance. Note that in our results section 5.1, we will refer to c as distance factor (DF).

Reference proteome prior

The reference proteome prior builds on the notion of a prior model based on the minimal distance to multiple sequences (section 3.2.3.5). In what follows, we derive a particular set of expected peptides and define the minimal distance of a candidate peptide to any such expected peptide. To make the description consistent with our previous exposition, we start with the set R of protein-coding mRNA. Let us have the set S of the corresponding proteins by translation of R,

thus

$$S = \bigcup_{r \in R} \Phi(r).$$

We now specify the expected proportions of cuts $\alpha(r)$ after a residue r in our more realistic prior model (section 3.2.4). Due to the structure of the library, we let α mimic the behavior of trypsin by always cutting after `K` and `R` but never otherwise, thus

$$\alpha(a) = \begin{cases} 1 & \text{if } a \in \{\mathtt{K}, \mathtt{R}\}, \\ 0 & \text{otherwise.} \end{cases}$$

Further, we disallow any modifications and substitution, except for fixed modification of carbamidomethylation of `C`, thus $\mathcal{F}_{\mathtt{C}}(\mathtt{C} \oplus \mathtt{Carbamidomethylation}) = 1$. Finally, we build all the peptides P that are the result of such cuts, thus $\langle P, R \rangle = \textsc{Build-Peptides}(S, 1, \langle 0, \infty \rangle)$.

The model Having specified the set of expected peptides, we now specify the prior model based on the distance to them. For computational efficiency, we will use the Hamming distance instead of the Levenshtein distance (we need to calculate the distances to P for 5.42×10^9 candidate peptides). As a result, we calculate the distance just with the peptides of the same length. Let us thus have the Hamming distance function $\Delta \colon \mathbb{P} \times \mathbb{P} \mapsto \mathbb{N}$. For each p, let us denote the minimal distance to a peptide in P as $\textsc{Dist}(p, P)$, thus

$$\textsc{Dist}(p, P) = \min_{q \in P, |q| = |p|} \Delta(p, q).$$

Then the relative prior probability of p is

$$\Pr^{\star}(p) = c^{\textsc{Dist}(p,P)},$$

for some c specifying the multiplicative decrease in prior probability with unit increase in the minimal distance. Similarly, as for the correct pattern prior, we will refer to the c as distance factor (DF) in the results section 5.1.

Sequence tag prior

The sequence tag prior models the situation when we know a substructure of the correct peptide in a certain probabilistic sense. Such knowledge can be, for instance, derived from the fragment spectra using tag-based approaches [38, 56, 94]. Let us first define what we mean by a sequence tag.

Definition 24 (Sequence tag)**.** A sequence tag is a triple $\langle m_{\vdash}, m_{\dashv}, s \rangle$, such that $m_{\vdash} \in \mathbb{R}^+$ is a missing mass to the N-terminal, $m_{\dashv} \in \mathbb{R}^+$ is a missing mass to the C-terminal, and $s = \langle s_1, \ldots, s_n \rangle, s_i \in \mathbb{A}$ is a sequence of amino acids.

Because we will consider only correct peptide sequence tags, let us specify how to create a sequence tag for a peptide, noting that we would obtain such a sequence tag experimentally in practice. First, to simplify the exposition, let us specify a prefix residue mass ladder that also includes both terminals, thus $\textsc{Prefix-Mass-Ladder}^{\vdash\dashv}(\langle p_0, \ldots, p_{n+1} \rangle) = \langle L_0, \ldots, L_{n+1} \rangle$, such that $L_i = \sum_{j \leq i} \textsc{Mass}(p_i)$, where p_0 and p_{n+1} refer to N-term and C-term, respectively.

Definition 25 (k-length sequence tag for a peptide p starting at residue i)**.** Suppose a peptide $p = \langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$, a length k of a sequence tag, $1 \leq k \leq n$, and a starting position $i \in \{1, \ldots, n - k + 1\}$. Now, denote $L = \langle L_0, \ldots, L_{n+1} \rangle$ the prefix mass ladder of p, thus $L = \langle L_0, \ldots, L_{n+1} \rangle = \textsc{Prefix-Mass-Ladder}^{\vdash\dashv}(p)$. Then, the k-length sequence tag of p starting at residue i, denoted $\textsc{Sequence-Tag}(p, k, i)$, is

$$\langle L_i, \textsc{Mass}(p) - L_{i+k-1}, \langle p_i, \ldots, p_{i+k-1} \rangle \rangle.$$

The previous definition of Sequence-Tag thus allows us to create sequence tags for a peptide. To get a more intuitive idea of the tag, let us show how the sequence tag corresponds to the mass of the peptide. Thus, suppose a peptide p, a length of the sequence tag k, and an applicable starting position i. Let us create a sequence tag as $\langle m_{\vdash}, m_{\dashv}, s \rangle = \textsc{Sequence-Tag}(p, k, i)$. Then, the sum of masses of individual parts equals the mass of the parental peptide, thus

$$\textsc{Mass}(m_{\vdash}) + \textsc{Mass}(m_{\dashv}) + \textsc{Mass}(s) = \textsc{Mass}(p).$$

Thus, in other words, the sequence tag captures a particular substructure of the peptide, while knowing what masses are missing on both sides.

Example Let us illustrate the definition on an example. Suppose a peptide $p =$ LVVVGAGGVGK and let us consider a tag for the substructure LV**VVG**AGGVGK as indicated by the bold typeface. Then,

$$\textsc{Sequence-Tag}(p, 3, 3) = \langle \textsc{Mass}(\vdash) + \textsc{Mass}(\text{LV}), \textsc{Mass}(\text{AGGVGK}) + \textsc{Mass}(\dashv), \text{VVG} \rangle.$$

We now focus on the topic of our primary interest—to define when a peptide matches a sequence tag. Thus, in what follows, we specify under what conditions a peptide p matches a sequence tag t.

Definition 26 (Peptide p matching a tag t at tolerance ϵ)**.** Suppose a peptide $p = \langle p_{\vdash}, p_1, \ldots, p_n, p_{\dashv} \rangle$, a sequence tag $t = \langle m_{\vdash}, m_{\dashv}, s \rangle$, and a tolerance $\epsilon \geq 0$. Let us have the prefix mass ladder L of p,

$$L = \textsc{Prefix-Mass-Ladder}^{\vdash\dashv}(p).$$

Suppose we match L against $m_{\vdash}$ and $\textsc{Mass}(p) - m_{\dashv}$, obtaining the indices M of matching fragments, thus

$$M = \boxtimes_{\epsilon}(L, \langle m_{\vdash}, \textsc{Mass}(p) - m_{\dashv} \rangle).$$

Then a peptide p matches a tag t up to tolerance ϵ, denoted $\textsc{Matches-Sequence-Tag}(p, t, \epsilon)$, if $\langle p_{f+1}, \ldots, p_t \rangle = s$, for some indices f and t, such that $\langle f, 0 \rangle, \langle t, 1 \rangle \in M$.

The prior model Suppose that, for a given spectrum, the correct peptide is q. Further, suppose we are interested in a prior model of sequence tag of length k, starting at position i. We

specify the prior model as follows:

$$\Pr^{\star}_{i,k}(p) = \begin{cases} 1 & \text{if } \textsc{Matches-Sequence-Tag}(p, \textsc{Sequence-Tag}(q, i, k), \epsilon) \\ c & \text{otherwise,} \end{cases}$$

for some c expressing how less likely a priori a peptide p is if not matching the correct sequence tag (of length k, starting at position i). Note that in the results section, we will refer to c as a tag non-matching factor. For additional information on the sequence tag prior, we refer the interested reader to our article [2].

4.4.1.2 Calculation of posterior probabilities

Herein, we describe calculation of posterior probabilities of candidate peptides using the Bayesian method from section 3.1.3. In doing so, we introduce a transformation of the agreement function such that it provides spectrum-specific agreement instead of a raw agreement—such a transformation then improves the separation of true and random matches [2]. Afterward, we specify how to obtain fixed distributions of true and random matches and finally introduce a method for predicting the true match distribution based on the characteristics of a particular fragment spectrum.

Overall objective Let us now reiterate our overall objective. Altogether, we aim to specify the distribution of true matches $\Pr\big(\Theta(p, m) = x \,|\, p\big)$ and the distribution of random matches $\Pr\big(\Theta(p, m) = x\big)$ to calculate the posterior probability $\Pr\big(p \,|\, \Theta(p, m) = x\big)$. In line with the section 3.1.3, we do so from a training dataset M of mass spectra $M = \langle m_1, \ldots, m_n \rangle$ and the corresponding correct peptides $P = \langle p_1, \ldots, p_n \rangle$. In what follows, let us denote the corresponding precursor mass of a fragment spectrum m as $\hat{m}_{\mathrm{p}}$. Because we assume that we can always measure the precursor mass up to a tolerance ϵ_{p}, the correct peptide $\Gamma(m)$ for a spectrum m is always in $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$ and thus the set $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$ is complete. In turn, this allows us to normalize the posterior probabilities to sum to one.

Transformation of the agreement As indicated before, we transform the agreement into its relative form because such a transformation improves the separation of correct and random peptide matches [2]. Thus, suppose an agreement function $\Theta \colon \mathbb{P} \times \mathbb{M} \mapsto \mathbb{R}$. We transform Θ into its relative-maximum form $\Theta^{\star}$ as follows,

$$\Theta^{\star}(p, m) = \max_{q \in \mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}} \Theta(q, m) - \Theta(p, m).$$

Note that after this transformation, the best matching peptide has an agreement of zero.

Fixed distribution models

We now turn to the specification of the distributions of true and random matches. Note that in applying the methods from the section 3.1.3, a slight complication arises because there, we assume a fixed finite set $\mathbb{C}$ of causes. Herein, for each precursor mass $\hat{m}_{\mathrm{p}}$, we have a potentially

different finite set of peptides $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. Nevertheless, we assume that such distributions are the same for each $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, allowing us to train the model directly.

True matches Thus, expanding on the equations in the section 3.1.3, and utilizing our transformed agreement, we have the following distribution of true matches:

$$\Pr\big(\Theta^\star(p,m) = x \,|\, p\big) = \frac{|\{i \in \mathbb{I} \,|\, \Theta^\star(p_i, m_i) = x\}|}{|\mathbb{I}|},$$

where $\mathbb{I} = \{1, \ldots, n\}$, n being the number of mass spectra in the training dataset.

Random matches Let us denote $\Theta^\star_M(x)$ the total number of peptides having a relative match x in M, thus

$$\Theta^\star_M(x) = \sum_{m \in M} \big|\{p \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \,|\, \Theta^\star(p,m) = x\}\big|.$$

Then, we have the distribution of random matches,

$$\Pr\big(\Theta^\star(p,m) = x\big) = \frac{\Theta^\star_M(x)}{\sum\limits_x \Theta^\star_M(x)}.$$

Prediction of true match distribution

A fixed true match distribution built directly from the matches of correct peptides derived already reasonably accurate posterior probabilities in the analysis of peptide library [2]. Nevertheless, we further considered a parametric true match distribution—predicted for a particular fragment spectrum based on its characteristics. For instance, if the precursor of a fragment spectrum was of high intensity, it was more likely that the correct peptide had a maximal match in it, and we wanted to take such and other dependence into account.

In general, let us represent the prediction as a function ζ, giving a true match distribution for a spectrum $m \in \mathbb{M}$. For simplicity, let us assume that the relative-maximum matches are natural numbers. Then ζ is of the following functional form, $\zeta\colon \mathbb{M} \mapsto (\mathbb{N} \mapsto \langle 0, 1\rangle)$. Further, it must hold that for each $m \in \mathbb{M}$,

$$\sum_{k \in \mathbb{N}} \zeta(m)(k) = 1.$$

Shape of the true match distribution Motivated by the shape of the fixed true match distribution [1], we assume that the true matches for a spectrum m follow a geometric distribution. Let us denote a particular geometric distribution as a function $\mathcal{G}_p\colon \mathbb{N} \mapsto \langle 0, 1\rangle$, where p is its only parameter. For completeness, let us specify the probability mass function of $\mathcal{G}_p$, thus

$$\mathcal{G}_p(k) = (1-p)^k \cdot p.$$

Finally, the parameter p aims to represent the probability that the true peptide will have a maximal match (note that $\mathcal{G}_p(0) = p$).

Predicting the parameter p Briefly, we built a logistic regression predictor on the training dataset $\langle M, P\rangle$, and used it to predict the probability that the correct peptide will have a maximal

match. For the prediction, we used the precursor intensity, the precursor mass, and the total number of peptides in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ as independent variables (using natural logarithms of each). The only dependent variable was a Boolean variable indicating whether the correct peptide had a maximal match.

The parametric model For a particular spectrum m, we predicted the parameter p for the geometric distribution $\mathcal{G}_p$, and set

$$\Pr\big(\Theta^\star(p, m) = k \,|\, p\big) = \mathcal{G}_p(k).$$

For further details, including the modeling of continuous distributions, we refer the interested reader to our article [1].

4.4.1.3 Comparison with other approaches

For the details of comparison we refer the reader into our articles [1] and [2].

4.4.2 Adjustment of peptide detection for typical experiments

The analysis of typical proteomics datasets brings its own set of challenges compared to the idealized conditions of the peptide library. First, we discuss technical errors in precursor mass measurement and describe our approaches to counteract them (section 4.4.2.1). Afterward, we specify an adjustment to prior probabilities of unique SNV-peptides that replaces the general probability of amino acid substitution with a sequence-specific one (section 4.4.2.2). Finally, we introduce a relaxation of Pr_max, which assigns a trade-off k between the importance of peptide prior probabilities and spectral match (section 4.4.2.3).

4.4.2.1 Errors in precursor mass

Measurement of mass spectra is sometimes affected by undesirable technical errors. Herein, we focus on errors when deriving the *precursor mass* of the parental molecule. First, we discuss the relatively common event of selecting a non-monoisotopic peak of a molecule by a mass spectrometer. Afterward, we describe a more general approach for dealing with less common errors in precursor-mass determination.

Selection of non-monoisotopic precursor peak

Operating systems of mass spectrometers generally aim to select the *monoisotopic* peak of a particular precursor for its fragmentation. However, because the determination of such a peak is not always straightforward, the mass spectrometer may select a non-monoisotopic peak instead. As a result, it can happen that the true monoisotopic mass m_p is not within the tolerance ϵ_p of the measured mass $\hat{m}_\mathrm{p}$, and thus the correct peptide $\Gamma(m)$ for the fragment spectrum m is not within $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. To cover this type of measurement errors, we need to consider the possibility that the mass spectrometer selected a non-monoisotopic molecule—a molecule that has a different number of neutrons than the monoisotopic one. Let us now expand our approach to incorporate such situations.

In general, we assume, that the mass spectrometer always selects precursors that can differ from the monoisotopic mass only by masses of

$$k \in K = \{k_{\min}, \ldots, k_{\max}\} \subset \mathbb{Z}$$

neutrons. Denoting n the mass of a neutron, the true monoisotopic mass m_{p} of the precursor is then within

$$(\hat{m}_{\mathrm{p}} - k \cdot n) \pm \epsilon_{\mathrm{p}}, \tag{4.4}$$

for some k in K. To implement the approach in practice, we thus calculate the spectral matches against additional database bins that overlap with all the precursor mass ranges specified in the equation 4.4.

Note Because of the probabilistic nature of our approach, we can also assign different prior probabilities to peptides corresponding to individual neutron mass shifts. In our analysis of typical computational proteomics experiments (section 5.2), we considered $K = \{-2, -1, 0, 1, 2\}$, and a factor $= 0.1^{|k|}$ by which we multiply the prior probabilities corresponding to the candidate peptides with mass shift of k neutrons. When we utilize such an adjustment, we include the superscript i in the deep search metrics derived from $\mathrm{Pr}_{\max}$ (e.g., $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ as in the section 5.2.1).

Generic mass misdeterminition

Mass spectrometers can also derive the precursor mass incorrectly for other reasons. For instance, the mass spectrometer can wrongly determine the charge of the parental molecule, which then results in incorrect precursor mass. As a result, the correct peptide $\Gamma(m)$ for a fragment spectrum m does not have to be in $\mathbb{P}_{\hat{m}_{\mathrm{p}} \pm \epsilon_{\mathrm{p}}}$. In what follows, we describe an approach that aims to account for such and other situations.

In particular, we implement a *precursor-mass-independent search* of mass spectra against a set of reasonably likely peptides. Thus, suppose a set R of protein-coding mRNA from which we create a database $\mathbf{Q} = \langle \mathbf{P}, \mathbf{R} \rangle$ of peptides $\mathbf{P}$ and relative prior probabilities $\mathbf{R}$ using the same parameters of the more realistic prior model as for the deep database (section 4.4.3.2). However, for practical reasons, such a database should be substantially less deep, and we used the limiting relative prior probability of $p_{\min} = 0.001$ in its construction. Then, for each fragment spectrum of candidate variant peptide, we match the spectrum against this smaller database—independent of its precursor mass. Finally, we merge the search results with peptides from the standard analysis, and the data processing continues as usual.

Note Similarly, as in the previous case, the probabilistic nature of our approach allows us to adjust the probabilities of peptides in such a search (e.g., by scaling them down by the probability of the mass-misdetermination event). Note that we did not utilize the precursor-mass-independent search in the comparisons in the section 5.2.

4.4.2.2 Adjustment of prior probabilities

We now describe an adjustment of relative prior probabilities for unique SNV-peptides whose corresponding DNA variants are of known frequency in the population. Overall, for a peptide p

with amino acid substitution $a \to b$, the adjustment replaces the probability of the substitution with the population frequency $\lambda(v)$ of the actual DNA variant v responsible for the substitution in the peptide p. Foremost, we note that such an adjustment would be properly handled in the peptide enumeration algorithm (section 3.2.5); that, however, would also require the algorithm to work on the level of DNA/RNA nucleotides (as opposed to the level of amino acids). Instead, we implemented the adjustment in a specific way applicable to re-analysis of variant peptides claimed using other approaches. In particular, we adjust the prior probability of the claimed variant peptide *only*, assuming that no other peptide within the corresponding set $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ needs such an adjustment.

Thus, suppose a unique SNV-peptide p containing a single amino acid substitution $a \to b$. In the more realistic prior model (section 3.2.4), we specified the expected proportion of such substitution as $\mathcal{S}_a(b)$. Now, let us have the relative prior probability $\Pr^\star(p)$, as defined in the prior probability model. Then, we let the frequency-adjusted relative prior probability of p, denoted $\Pr^\star_\dagger(p)$, equal

$$\Pr^\star_\dagger(p) = \Pr^\star(p) \cdot \frac{\lambda(v)}{\mathcal{S}_a(b)}, \tag{4.5}$$

where $\lambda(v)$ is the population frequency of the nucleotide variant corresponding to peptide p. Finally, if the population frequency of the corresponding variant v is not known, we let $\Pr^\star_\dagger(p) = \Pr^\star(p)$.

In section 3.2.7, we described in detail the calculation of $\Pr_\mathrm{max}$ for deep search of fragment spectra using relative prior probabilities $\Pr^\star(p)$. Therein, we noted that the important aspect of $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ is that the database is closed in the following sense: for each $p \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, if $q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ and $\Pr^\star(q) \geq \Pr^\star(p)$ then $q \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. We now focus on the analogous situation with the relative prior probabilities $\Pr^\star_\dagger$ instead of $\Pr^\star$.

Lemma 5. *Let us have a vector $\boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ of peptides, such that for each $r \in \boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, if $q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ and $\Pr^\star(q) \geq \Pr^\star(r)$, then $q \in \boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. Now, suppose that a particular $p \in \boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ is the only unique SNV-peptide in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ with known population frequency $\lambda(v)$ of the corresponding variant v. Finally, suppose that $\lambda(v) \geq \mathcal{S}_a(b)$, for the amino acid substitution $a \to b$ that corresponds to p. Then, for each $r \in \boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, if $q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ and $\Pr^\star_\dagger(q) \geq \Pr^\star_\dagger(r)$, then $q \in \boldsymbol{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$.*

Proof. Foremost, p is the only unique SNV-peptide with known population frequency of the corresponding variant within $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, and thus only its relative prior probability $\Pr^\star(p)$ is affected. Further, $\Pr^\star_\dagger(p) \geq \Pr^\star(p)$ by our assumption. As a result, for each peptide $r \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, if $q \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ and $\Pr^\star_\dagger(q) \geq \Pr^\star_\dagger(r)$, then $q \in \mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. □

The lemma thus says that as long as we *increase* the prior probability of the single variant peptide p, the dataset of peptides remains closed—there will still be all peptides in $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ that are at least as likely *a priori* as any peptide in $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. When we utilize such adjustment in our analyses, we will include the superscript † in the deep search scoring metrics derived from $\Pr_\mathrm{max}$ (e.g., $\Pr^\dagger_\mathrm{max}$).

4.4.2.3 Relaxation of $\Pr_\mathrm{max}$

The $\Pr_\mathrm{max}$ is useful for removing unlikely peptides by means of existence of other at-least-as-good candidates for a given spectrum—both in terms of their prior probability and their spectral

match. However, $\mathrm{Pr}_{\max}$ has limitations when multiple candidates are of similar fragment match and prior probabilities. To improve the situation, we introduce a relaxation of $\mathrm{Pr}_{\max}$, denoted $\tilde{\mathrm{Pr}}^{k}_{\max}$, which assigns a trade-off k between the importance of the spectral match and prior probabilities. The $\tilde{\mathrm{Pr}}^{k}_{\max}$ calculated for a given peptide and spectrum then gives a value in the $\langle 0, 1\rangle$ interval, related to the posterior probability. We now introduce the notion of the $\tilde{\mathrm{Pr}}^{k}_{\max}$ for a given spectrum m and each candidate peptide $p \in \mathbf{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$.

Definition 27 ($\tilde{\mathrm{Pr}}^{k}_{\max}$)**.** Suppose a fragment spectrum m, with its precursor mass $\hat{m}_{\mathrm{p}}$ measured up to tolerance ϵ_{p}. Now, let us have a database of peptides $\mathbf{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$ of the appropriate mass range as obtained from the peptide enumeration algorithm (section 3.2.5). Further, for each $p \in \mathbf{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$, let us have its spectral match $\Theta(p, m)$. Then, the relaxed $\mathrm{Pr}_{\max}$ of a peptide $p \in \mathbf{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$ at trade-off k, denoted $\tilde{\mathrm{Pr}}^{k}_{\max}(p, m)$, is

$$\tilde{\mathrm{Pr}}^{k}_{\max}(p, m) = \frac{\mathrm{Pr}(p) \cdot k^{\Theta(p,m)}}{\sum\limits_{q \in \mathbf{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}} \mathrm{Pr}(q) \cdot k^{\Theta(q,m)}}.$$

We now establish the circumstances under which $\tilde{\mathrm{Pr}}^{k}_{\max}(p, m)$ equals the posterior probability of a peptide p, given its match $\Theta(p, m)$.

Theorem 5 (Correspondence of $\tilde{\mathrm{Pr}}^{k}_{\max}$ and posterior probability)**.** *Suppose a fragment spectrum m, its precursor mass $\hat{m}_{\mathrm{p}}$ measured up to tolerance ϵ_{p}. Further, suppose that the probability of observing a particular agreement a for a true cause p is the same for all agreements,*

$$\mathrm{Pr}\big(\Theta(p, m) = a \,|\, p\big) = x,$$

and that Θ is true-cause normalized. Further, suppose that the probability of a particular agreement a at random is k times more likely than than the probability of an agreement $a + 1$,

$$\mathrm{Pr}\big(\Theta(p, m) = a + 1\big) = k \cdot \mathrm{Pr}\big(\Theta(p, m) = a\big),$$

and that Θ is random-cause normalized. Finally, suppose that the vector $\boldsymbol{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$ contains all the peptides for the corresponding mass, thus $\boldsymbol{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}} = \mathbb{P}_{\hat{m}_{\mathrm{p}}\pm\epsilon_{\mathrm{p}}}$. Then,

$$\tilde{\mathrm{Pr}}^{k}_{\max}(p, m) = \mathrm{Pr}\big(p \,|\, \Theta(p, m) = a\big).$$

Proof. From Bayes theorem, we have

$$\mathrm{Pr}\big(p \,|\, \Theta(p, m) = a\big) = \frac{\mathrm{Pr}\big(\Theta(p, m) = a \,|\, p\big) \cdot \mathrm{Pr}(p)}{\mathrm{Pr}\big(\Theta(p, m) = a\big)}.$$

Because we consider all peptides, let us focus only on the relative differences in the posterior probabilities, knowing that we can normalize them afterward. In the relative comparisons, we can disregard the term

$$\mathrm{Pr}\big(\Theta(p, m) = a \,|\, p\big)$$

because we assume that the probability is equal for all agreements and Θ is true-cause normalized.

The relative difference in probabilities of observing a particular agreement at random follows

from our assumptions,

$$\frac{\Pr\big(\Theta(p,m)=a\big)}{\Pr\big(\Theta(q,m)=b\big)} = k^{a-b}$$

and because Θ is random-cause normalized. As $\mathbf{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ equals $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$, the sum of posterior probabilities over all peptides equals one. It follows that

$$\Pr\big(p \,|\, \Theta(p,m)=a\big) = \frac{\Pr(p)\cdot k^{\Theta(p,m)}}{\sum\limits_q \Pr(q)\cdot k^{\Theta(q,m)}} = \tilde{\Pr}{}^{k}_{\max}(p,m).$$

□

4.4.3 Comparison of detection performance in typical experiments

Herein, we describe the details of comparisons used in evaluating the performance of variant peptide detection approaches. First, we describe the configuration of each software employed in the comparison, along with the processing of its detection results so that the output contains only the detected variant peptides (section 4.4.3.1). Afterward, we describe the parameters used in our more realistic prior model, which we then utilized in the deep search of fragment spectra (section 4.4.3.2). Finally, we describe how we applied our deep search method to analyze peptides detected using other approaches (section 4.4.3.3).

4.4.3.1 Configuration of the analyzed software

First, we start with the common configuration of each approach. We run each software with the relative precursor mass tolerance of 5 parts-per-million (*ppm*) and an absolute fragment tolerance of 0.3 Da. As a reference proteome, we used proteins resulting from the translation of reference protein-coding mRNA (`Ensembl`, `GRCh37` genome), and restricted them to the longest protein isoform per gene, giving, 20 704 proteins. Note that we used such protein-coding mRNA instead of other reference proteomes (e.g., those from UniProt [18]) to allow establishing correspondence between proteins and DNA/mRNA in order to evaluate the sequencing support of detected variant peptides (section 4.2.2). We now turn to the description of the individual approaches.

X!Tandem$_{\mathbf{ES}}$

X!Tandem (v. 2017.2.1.4, `Alanine`) in its exhaustive substitution (ES) form was run with the default parameters (`default-input.xml`) updated by the appropriate mass tolerances and with the detection of amino acid substitutions enabled. For the latter, we set the parameters as follows: refinement of peptide detection results (`refine` set to `yes`), and search for amino acid substitutions (`point mutations` set to `yes`). The peptides with the detected amino acid substitutions then constituted the variant peptides for further analyses.

X!Tandem$_{\mathbf{GPV}}$

X!Tandem (v. 2017.2.1.4, `Alanine`) in its global peptide-variant (GPV) form was run against three databases. The first database consisted of the translated reference-protein coding mRNA as described previously (`Ensembl`, `GRCh37`). The other two databases consisted of variant peptides

built from a globally-observed nucleotide variation incorporated into the human reference protein-coding mRNA (`Ensembl`, `GRCh37`). Each nucleotide variant was introduced independently into the single protein-coding mRNA it affected, the whole protein-coding mRNA was translated, digested using trypsin (one missed cleavage allowed), and non-reference peptides kept. The first peptide variant database consisted of peptides translated from nucleotide variants from dbSNP (v. 147), while the second consisted of peptides translated from COSMIC (v. 77), ICGC (v. 20), and TCGA (accessed on May 12, 2016) combined. The mass spectra were then searched independently against each of these databases, and the peptide detection results were then combined for each database search. Afterward, we prefiltered the candidate detected variant peptides as follows. If a reference peptide with an equal or higher score for a particular spectrum was found in the search results, we removed the variant peptide because reference peptides are more likely *a priori* (score: `HyperScore`). The remaining peptides then constituted the detected variant peptides for further analyses.

MSFragger

MSFragger (v. 3.2) was run with the following adjustment to the default parameters of open search (`open_fragger.params`). We specified the maximum digest mass to 3 kDa to correspond to the mass range of our deep peptide database (`digest_mass_range = 500.0 3000.0`). Note that this step was done to improve the computational efficiency because we filter out variant peptides above 3 kDa for use with the deep search anyway. To further improve the computational performance of the method, we lowered the maximal number of variable modifications per peptide from 3 to 2 (`max_variable_mods_per_peptide = 2`). Once the peptides were detected, we first split the results into reference peptides and mass-modified peptides. Afterward, we filtered the mass-modified peptides as follows. For each mass-modified peptide, and for each its potential localization (`best_locs` field), we considered all applicable modifications and substitutions to explain the mass (`UniMod`, 963 modifications and substitutions). If the peptide mass could *only* be explained by a single amino acid substitution, we introduced the amino acid substitution into the sequence, and we retained the peptide as a detected variant peptide.

BICEPS

BICEPS (v 1.0) was run with the following parameters: `--tol 5` for precursor mass tolerance of 5 ppm, `--penaltyvector 2` for a maximum of one amino acid substitution per peptide, and `--tool 2` to use both DirecTag [56] and PepNovo [94] for deriving sequence tags. Peptides that were detected with an amino acid substitution were then retained as variant peptides.

4.4.3.2 Parameters of the more realistic prior model

Herein, we describe the parameters of the more realistic prior model (section 3.2.4) used for enumeration of peptides and their relative prior probabilities (section 3.2.5). The model was used to model the prior knowledge, and we utilized it to calculate $\mathrm{Pr}_{\mathrm{max}}$ and derived metrics in the analysis of the typical proteomics experiments (sections 5.2 and 5.3).

Modifications

To specify the modified forms of residues, we utilized the Unimod database [77], which at the time of our accession contained 963 modifications (accessed: Feb 9, 2015). We first set the expected proportion of these modified forms: $\mathcal{M}_{\mathtt{M}}(\mathtt{M} \oplus \mathtt{Oxidation}) = 0.3$; $\mathcal{M}_{\mathtt{S}}(\mathtt{S} \oplus \mathtt{Methylation}) = 0.01$; and $\mathcal{M}_{\mathtt{T}}(\mathtt{T} \oplus \mathtt{Methylation}) = 0.01$. Afterward, to allow for fixed modification of cysteines (C), we set $\mathcal{M}_{\mathtt{C}}(\mathtt{C}) = 0.001$, expressing that the non-modified form is unexpected. Then, except for $\mathtt{C} \oplus \mathtt{Carbamidomethylation}$ and the rest of reference amino acids and terminals, we set the expected proportions to 0.001.

Once these were set, we set the expected proportion of all reference residues (except C) and terminals as follows. For each $a \in \mathbb{A}_{\wedge}^{\vdash\dashv} \setminus \{\mathtt{C}\}$, we set the expected proportion of the non-modified form of a, $\mathcal{M}_a(a)$, equal to the remaining proportion, thus

$$\mathcal{M}_a(a) = 1 - \sum_{b \in \mathcal{M}(a) \setminus \{a\}} \mathcal{M}_a(b).$$

Analogously, we set the expected proportion of $\mathtt{C} \oplus \mathtt{Carbamidomethylation}$ as follows:

$$1 - \sum_{b \in \mathcal{M}(\mathtt{C}) \setminus \{\mathtt{C} \oplus \mathtt{Carbamidomethylation}\}} \mathcal{M}_{\mathtt{C}}(b).$$

Substitutions

We set up the expected proportion of substituted form $a \to b$ in terms of the minimal Hamming distance between the RNA codons coding for a and b. Thus, suppose that amino acid a is coded by RNA codons $A = \Phi^{-1}(a)$, and analogously $B = \Phi^{-1}(b)$ for amino acid b. Now, let us denote the minimal Hamming distance between A and B as d, thus

$$d = \min_{x \in A, y \in B} \Delta(x, y).$$

Then, we set $\mathcal{S}_a(b) = c^d$, for a constant $c = 2 \times 10^{-4}$.

Cleavage

We set the expected proportion $\alpha(r)$ of cuts after residue $r \in \mathbb{A}$ as follows: $\alpha(\mathtt{K}) = 0.7$, $\alpha(\mathtt{R}) = 0.85$, and 0.002 for the remaining residues. Note that the configuration aimed to mimic the behavior of trypsin.

Peptide enumeration

We generated peptides up to minimal relative prior probability $p_{\min} = 4 \times 10^{-6}$, and within a mass range $m_{\min} = 700$ Da, and $m_{\max} = 3\,000$ Da.

4.4.3.3 Deep probabilistic re-analysis of candidate variant peptides

The variant peptides detected using any of the previous approaches were then subjected to our probabilistic deep search method. Because we were interested only in peptides that can originate from a single nucleotide variant (SNV-peptides, section 4.2.2.3), we aligned the variant

peptides against the human reference protein-coding mRNA (4.2.2.1). Further, because we aimed to calculate the sequencing support of such peptides, we retained only SNV-peptides that can originate only from a single genomic location, hence retaining unique SNV-peptides. We further restricted the variant peptides only to those that are within the mass range of our deep human peptide database (700–3 000 Da). Afterward, we performed deep searches of all candidate peptides and calculated $\mathrm{Pr}_{\mathrm{max}}$ and its various extensions (sections 3.2.7 and 4.4.2). For calculation of $\tilde{\mathrm{Pr}}^{k}_{\mathrm{max}}$ and derived metrics, we always used the trade-off $k = 20$. Finally, in the deep search, we utilized neutron mass shifts corresponding to -2, -1, 0, 1, and 2 neutrons (section 4.4.2.1).

4.5 CLAIRE—a system for detecting peptide variants

Herein, we briefly describe CLAIRE, our software system for detecting peptide variants, which implements the mathematical and computational methods presented in the thesis. CLAIRE is available in two forms: in a standalone form and an online form—both can be accessed at `https://claire.imtm.cz`. For the standalone form, we briefly describe its functionality, organization of code, the user interface, the documentation, and the software testing. For the online form, we provide an overview of its functionality, along with the description of the views by which the researchers can inspect the data after detecting peptide variants.

4.5.1 Standalone, cross-platform version

The standalone CLAIRE (v. 0.2.0) is an open-source, cross-platform system implemented in Python (v. 2.7) and consists of around 20 000 lines of code. CLAIRE was developed initially on Rocks 6.0 Linux distribution, but runs with the help of `Anaconda` environment system on all three major operating systems (Linux, Mac OS, and Windows). Internally, CLAIRE relies heavily on `pandas` and `numpy` data-scientific libraries, and its time-critical algorithms are implemented using `Cython`—a library for interfacing Python with C. CLAIRE can be used directly for detecting peptide variants by using its command-line interface or its modules imported within the Python programming language.

Code organization CLAIRE was developed using a functional programming paradigm. In essence, CLAIRE is an organized collection of functions that map one data structure into another—without resorting to any hidden state. Overall, we organized these functions into around 40 modules, and each such module aims to provide particular functionality. Although detailed descriptions are present in the software's documentation, let us provide some examples of the available modules. For instance, the high-level module `claire.lisa` deals with all aspects of the deep search, including peptide enumeration, the building of fragment-ion indexes, and the calculation of $\tilde{\mathrm{Pr}}_{\mathrm{max}}$. Another higher-level module, `claire.corr` implements the functionality for establishing the correspondence between peptides and DNA/mRNA. As an example of a low-level module, `claire.tolerance` contains routines for transforming between absolute and relative tolerances, expressing them as intervals, or calculating their overlaps. Besides the functionality related to mass spectrometry, CLAIRE also includes more general modules, e.g., for the analysis of tabular data (`claire.pandas_utils`), `NumPy` arrays (`claire.numpy_utils`), or for downloading

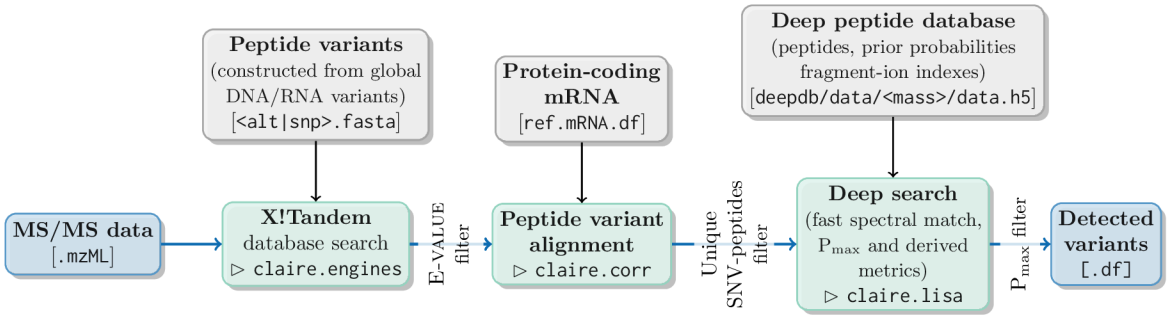


Figure 4.2: Flow of data in CLAIRE's detection of peptide variants

The figure depicts the flow of data in executing the fraction-level command `claire-detect-snvs`. The MS/MS data are first searched against global peptide variants ($\text{X!Tandem}_{\text{GPV}}$, module `claire.engines`), and candidate peptides are filtered using the significance of their match (E-VALUE ≤ 0.1). Afterward, the candidate variant peptides are aligned against protein-coding mRNA to establish their candidate origins (module `claire.corr`), and unique SNV-peptides are retained. Unique SNV-peptides are then subjected to deep search against a deep mass-partitioned peptide database while calculating Pr_{max} and derived metrics (module `claire.lisa` and submodules). Afterward, the variant peptides are filtered according to the deep search metrics, and their output is stored in a native `pandas`' `DataFrame` format. Note that to obtain the results in `CSV` format, one then runs the sample-level command `claire-variant-report` that aggregates detected variants from individual fractions.

the required databases (`claire.download`). To better understand how these functions interact, we refer the reader to the documentation and to the source code of executable scripts within CLAIRE.

User interface The user runs the individual analyses using a command-line interface. Overall, these analyses operate on three levels: a fraction (single .mzML file), a sample (collection of fractions), and an experiment (collection of samples). The actual detection of peptide variants is performed using the fraction-level command `claire-detect-snvs`, which detects peptide variants from a single mzML file, while calculating scoring metrics derived from Pr_{max} (**Fig. 4.2**). Once all fractions from a sample are analyzed, the sample-level command `claire-variant-report` collates the data from individual fractions, creating a detailed variant report for the analyzed sample (`CSV` format). If variant reports are created from multiple samples, one can utilize an experiment-level command `claire-mutation-rate-report`, which then calculates the protein variation rates for all samples within the experiment. Detailed descriptions of the commands are available in the software's documentation, and by using either `-h` or `--help` switch.

Documentation CLAIRE (v. 0.2.0) contains extensive documentation written in reStructuredText, and compiled into HTML using Sphinx. The reference documentation of individual functions and modules contains around 130 A4 pages, and the root of the documentation is available at `https://claire.imtm.cz/repo/doc/`.

Software testing CLAIRE's extensive documentation also contains executable tests (doctests) of the expected behavior of individual functions; altogether, this amounts to 186 doctests. Further, CLAIRE contains several unit tests, and integration tests with X!Tandem, and with the ProteoWizard suite [95] (in total, 13). One can run both sets of tests using the `pytest` package (details in the documentation). CLAIRE has also a full post-installation test of peptide variant

detection (command `claire-test-detection`). The test first downloads a small mzML file and a deep database for a narrow precursor mass range (1341–1344 Da). Afterward, the test invokes the commands for the detection of peptide variants, the construction of a variant report, and the calculation of protein variation rate.

Installation The installation of CLAIRE proceeds using an automatic installation script (`https://claire.imtm.cz/repo/install/`), which first initializes the `Anaconda` environment, and then downloads and installs CLAIRE. The automatic installation of CLAIRE was tested on the following operating systems: Linux (Ubuntu: v. 16.04, v. 18.04; and CentOS: v. 6.0), Windows (v. 10), and Mac OS (High Sierra, v. 10.13; and Catalina, v. 10.15.5). The software's documentation also describes a manual installation of CLAIRE if the automatic one fails.

Notes on the implementational differences In the calculation of the comparisons with other approaches (section 5.2), we adapted the script `claire-detect-snvs` to calculate several additional metrics derived from $\mathrm{Pr}_{\mathrm{max}}$. Further, CLAIRE in the version 0.2.0 does not implement the memory-load optimization (section 3.2.6.4). Although we utilized the optimization in our analyses, it also requires different technical optimizations of the deep database, and it is not yet included in the current version. Similarly, the peptide enumeration algorithm in the version 0.2.0 uses a different handling of amino acid substitutions than the one presented in section 3.2.5.1; both, however, derive highly similar relative prior probabilities.

4.5.2 Online version

CLAIRE also has an online form, which wraps the detection functionality into an easily-accessible web interface. In essence, the online form allows users without bioinformatics expertise to submit samples for variant analysis, and export or interpret the peptide detection results. The results within the interface can be viewed on different levels of abstraction (**Fig. 4.3**)—from a very general overview up to details of the deep search for a particular spectrum. Besides the mass spectrometric output of the analysis, the web interface aims to provide a partial biological view of the results, most notably in the *Protein view*. Therein, the view provides an estimate of the harm of the detected variant [96], details about the presence of SNV in other datasets, or by providing summaries and cross-references to other relevant databases. Technically, the user interface is implemented in Python using the `Flask` web development framework, and submits individual tasks to the Sun Grid Engine job management system deployed on our supercomputing infrastructure. To summarize, the user interface thus allows utilizing our peptide variant detection methods to interpret MS/MS spectra without the need to install CLAIRE locally.

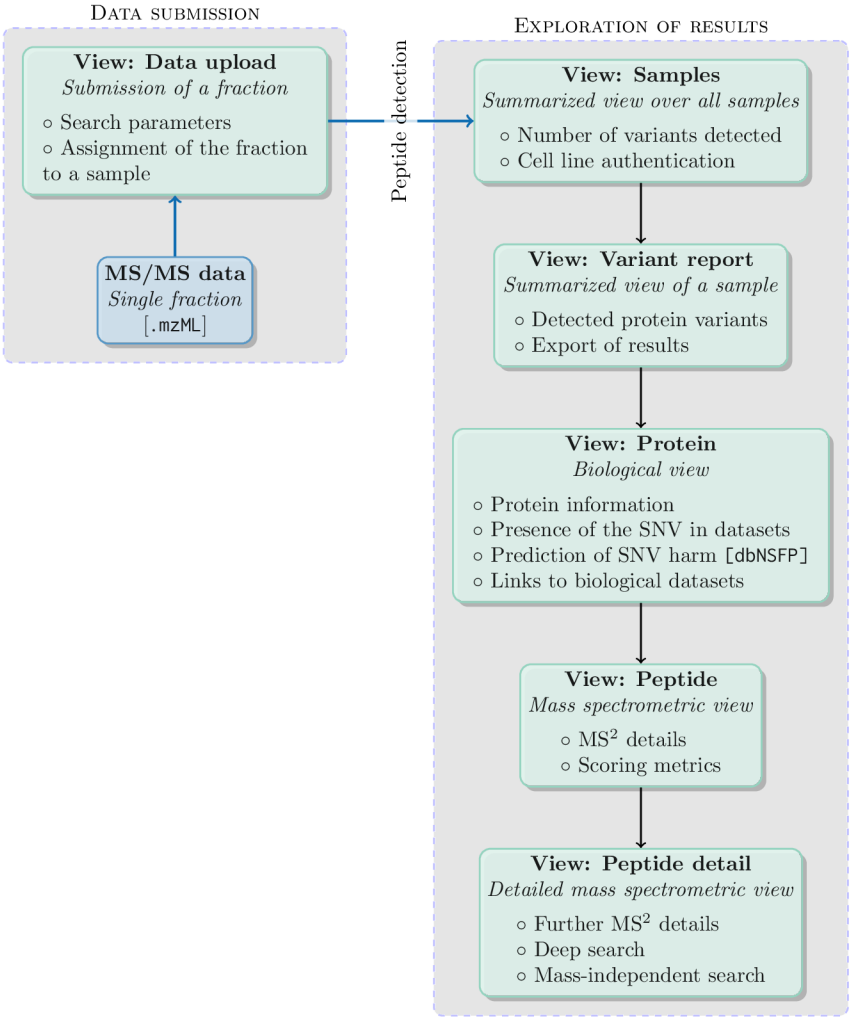


Figure 4.3: Organization of CLAIRE's web interface

The interface consists of two major parts—the submission of MS/MS spectra and the exploration of the results of the analyses. To perform an analysis, the user uploads an mzML file, specifies which sample the fraction belongs to, and submits the task. Once the fraction is analyzed, the user can explore the data based on multiple levels of detail—starting from summarized overviews to an in-depth look at individual peptides including the deep search results.

Chapter 5

Results

The chapter deals with direct and downstream applications of the methods presented in the thesis. First, in section 5.1, we focus on the analysis of peptide detection in the idealized conditions of the combinatorial peptide library. Therein, we show that the posterior probabilities calculated using our Bayesian model behaved desirably in several circumstances and that the use of simple prior models outperformed state-of-the-art *de novo* sequencing algorithms. Then, in section 5.2, we shift our focus to typical experiments and investigate the relevance of the maximal posterior probability ($\mathrm{Pr}_{\mathrm{max}}$) for the re-analysis of variant peptides detected using four popular approaches. Our results show that all these approaches substantially benefited from our deep probabilistic search of fragment spectra—especially when using extended deep search score metrics derived from $\mathrm{Pr}_{\mathrm{max}}$. Finally, in section 5.3, we illustrate downstream applications of the developed methods in cancer research, research reproducibility, and forensics.

5.1 Peptide detection in the combinatorial peptide library

Herein, we evaluate the Bayesian peptide detection model from section 4.4.1.2 on the combinatorial peptide library dataset while utilizing multiple simple models of peptide prior probabilities (section 4.4.1.1). First, we show that the numerical values of posterior probabilities tended towards their expected long-term behavior and that their departures followed from a lack of correspondence between the prior models and the analyzed dataset (section 5.1.1). Then, we evaluate the impact of the prior models on the posterior probabilities assigned to the correct peptides, showing that we can detect correct peptides with high probabilities when using adequately powerful prior models (section 5.1.2). Finally, we compare our approach with the state-of-the-art *de novo* sequencing algorithms, showing that even a simple scoring metric combined with a weak prior model can attain surprisingly high detection performance (section 5.1.3).

5.1.1 Posterior probabilities of peptides tended towards the desired behavior

The posterior probabilities of peptides are most useful in practice if they follow a particular behavior—capturing the correctness of peptides in the long run. For instance, if we select a large collection of peptides with posterior probabilities r, it is desirable that a corresponding proportion r of peptides was detected correctly. We will now investigate the behavior of the posterior probabilities calculated using our Bayesian model, and we do so for multiple prior

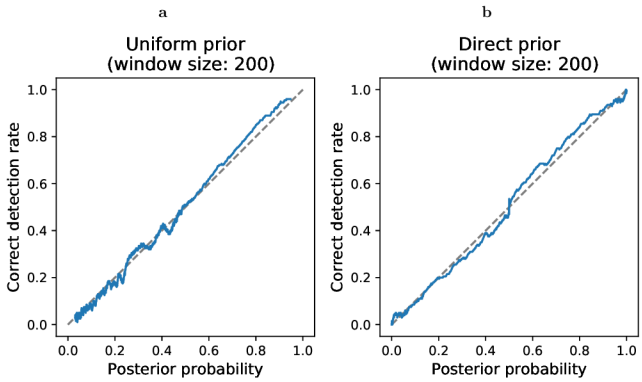


Figure 5.1: Behavior of posterior probabilities for uniform and direct prior models.

The figure shows the relationship of posterior probabilities of the best candidates per spectrum and the correct detection rates. The close correspondence of the desired and the observed behavior indicates that the proposed Bayesian model worked well on the dataset.

distributions.

5.1.1.1 Extreme cases of prior distributions

First, we start with two extreme cases of prior distributions: the uniform prior and the direct prior. In what follows, suppose a spectrum m, its precursor mass $\hat{m}_\mathrm{p}$ and the corresponding set of all candidate peptides $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ for the given precursor mass range. The uniform prior assigns each peptide in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ equal relative prior probability, and thus represents the lack of any information about the sample (section 3.2.3.1). The direct prior, on the other hand, assigns constant non-zero prior probabilities only to the 400 peptides in the peptide library $\mathbb{P}_L$, and thus represents a near-completely informed prior model. Formally, the direct prior for a particular spectrum m thus behaves as follows:

$$\mathrm{Pr}^\star_\mathrm{direct} = \begin{cases} 1 & \text{if } p \in \mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}} \cap \mathbb{P}_L, \\ 0 & \text{otherwise.} \end{cases}$$

As is evident from the figure **Fig. 5.1**, the behavior of posterior probabilities was close to the ideal one, showing that our Bayesian model behaved desirably on this dataset for both prior models.

5.1.1.2 Correct pattern prior

The peptides in the combinatorial library $\mathbb{P}_L$ are all of the same pattern LVVVGA**XX**VGK, allowing us to study the posterior probabilities for a prior model based on the distance to this pattern (section 4.4.1.1). Overall, we analyzed the behavior for a varying *distance factor* (DF)—a number in the $\langle 0, 1\rangle$ interval, which specifies the multiplicative decrease in the relative prior probability with a unit increase in the distance to the pattern. Intuitively, the distance factor specifies the importance of the distance and would be roughly set to correspond to the probability of amino acid substitution in practice.

The **Fig. 5.2a** shows that the posterior probabilities generally tended towards the desired behavior but have also exhibited oscillations around it. Note that these oscillations were present

mostly at higher distance factors (DF = 0.5 and DF = 0.9) yet again disappeared at DF = 1.0 (**Fig. 5.1a**). In general, the oscillations resulted from situations when *multiple high-scoring peptides had the same agreement* with the spectrum, and the prior probabilities of these peptides did not correspond to the distribution of peptides in the analyzed dataset. In these circumstances, the prior distribution is of high relevance in our Bayesian model—if the agreement of peptides is the same, the ratios of their posterior probabilities are equal to the ratios of their prior probabilities.

To illustrate this on a simplified example, suppose a correct peptide p and an incorrect peptide q that both have a maximal match x among all candidate peptides in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$. Now, all correct peptides in $\mathbb{P}_L$ are of distance 2 to the pattern LVVVGA**XX**VGK, thus $\Pr^\star(p) = \mathrm{DF}^2$. Suppose that the incorrect peptide q was of a distance 3, so $\Pr^\star(q) = \mathrm{DF}^3$. The ratio of their posterior probabilities is then $\frac{\mathrm{DF}^2}{\mathrm{DF}^3} = \mathrm{DF}^{-1}$. For further simplicity, suppose that these are the only candidate peptides per spectrum. Then, at DF = 0.9, the posterior probability of q will be 0.9× that of p, thus only a little less (i.e., ≈ 0.526 for p and ≈ 0.474 for q). Under such circumstances, the posterior probabilities for correct peptides thus aggregate slightly above 0.5, while those for the incorrect ones aggregate slightly below 0.5. The behavior can also be seen in the figure **Fig. 5.2a** even though we note that the situations are usually more complicated in practice. To summarize, if peptides can not be distinguished by their agreement with the fragment spectrum, their prior probabilities become important. If, in turn, these prior probabilities are inadequate, the posterior probabilities locally depart from the desired behavior. Finally, note that this phenomenon would not happen if both peptides p and q had the *same* prior probabilities—i.e., their posterior probabilities would be 0.5, and the averaging would cancel out the oscillation. For a more detailed treatment of the situation, we refer the interested reader to our articles [1, 2].

5.1.1.3 Reference proteome prior

The peptide library is motivated by a particular human peptide LVVVGA**GG**VGK, allowing us to study a prior model based on the minimal distance to peptides derived from human reference proteins (section 4.4.1.1). As indicated on the **Fig. 5.2b–c**, the behavior of posterior probabilities has shown similar tendencies as for the correct pattern prior. In particular, the underestimation of posterior probabilities at high distance factors resulted from a similar phenomenon as for the correct pattern prior [2]. However, the behavior had also shown an *overestimation* of posterior probabilities for low-enough distance factors (i.e., DF = 0.01 and 0.001, **Fig. 5.2b**), and we now focus on its origins.

Overall, the reason was that the reference proteome prior *corresponded only partially* to the peptide library dataset. According to the prior model, we would expect some non-library peptides more often than the library peptides—yet the dataset contains only library peptides. In other words, some candidate peptides in $\mathbb{P}_{\hat{m}_\mathrm{p}\pm\epsilon_\mathrm{p}}$ are closer to a reference peptide than are the actual peptides from the library. With increased relevance of the distance (DF < 0.01), these peptides start to have increasingly higher posterior probabilities but they are incorrect. The behavior thus again shows the relative importance of adequately capturing the prior probabilities if we are interested in calculating accurate posterior probabilities.

Nevertheless, note that the level of the correspondence between the dataset and the prior

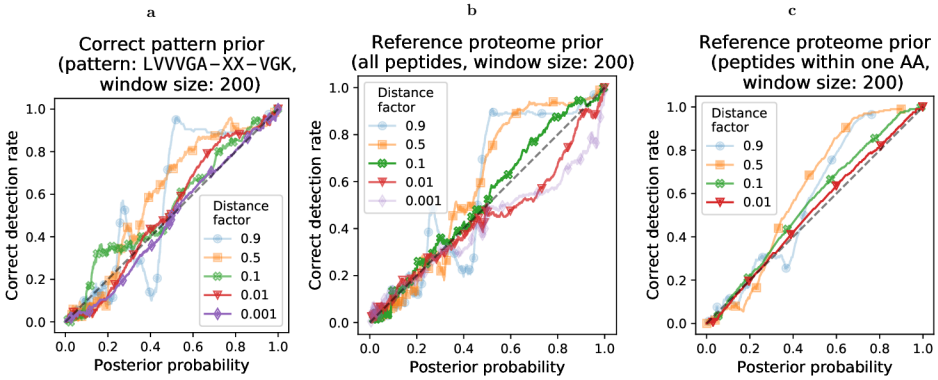


Figure 5.2: Behavior of posterior probabilities for prior models based on the distance to a single sequence **a**, and minimal distance to multiple sequences **b**, and **c**.

(**a**) The posterior probabilities for the correct pattern prior generally tended towards the desired behavior but showed oscillations for higher distance factors (i.e., DF = 0.5 and DF = 0.9). The oscillations resulted from situations when multiple high-scoring peptides had the same agreement with the spectrum—in our Bayesian model, the ratios of posterior probabilities of such peptides are equal to the ratios of their prior probabilities. The prior model, however, did not sufficiently correspond to the distribution of peptides in the analyzed dataset at these distance factors (see the main text for more detail). (**b**) The prior distribution based on the minimal distance to reference human peptide sequences behaved similarly as in **a**. However, it had also shown an overestimation of posterior probabilities for low distance factors (i.e., DF = 0.01 and DF = 0.001). In general, this was because the prior model only partially corresponded to the peptide library dataset. For instance, according to the prior model, we would expect some non-library peptides $q \notin \mathbb{P}_L$ more often than the actual library peptides $p \in \mathbb{P}_L$. In consequence, putting too much relevance on the distance resulted in preferring some reference-close peptides in $\mathbb{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$ over the library peptides in $\mathbb{P}_L$, and these were necessarily incorrect—resulting in overestimating their posterior probabilities. (**c**) Restricting the analyses of **b** to library peptides that were at most one amino acid from the sequence pattern LVVVGA**XX**VGK resulted in a reasonably desirable behavior of posterior probabilities—even though the prior had only a limited correspondence to the dataset.

model does not have to be necessarily very high. To illustrate this, we restricted the peptide library only to peptides that were at most one amino acid from the library pattern (**Fig. 5.2c**). For such a restricted dataset, the behavior of posterior probabilities was improved substantially, and the posterior probabilities were not overestimated anymore. Altogether, this indicates that the prior model was good enough to detect peptides with one amino acid substitution and to obtain reasonably accurate posterior probabilities—even though the prior model only partially corresponded to the analyzed dataset.

5.1.1.4 Additional analyses

In our former research [1], we have also analyzed prior models based on less complete library patterns while considering different criteria for evaluating the accuracy of posterior probabilities. In [2], we investigated the posterior probabilities for increasingly incorrect prior models, for the combination of agreement models (e.g., with the retention time model), and for combination of prior probability models. In the latter, we also compared the approach to PeptideProphet [42] and Percolator [43], two popular methods for assigning posterior probabilities, showing that our method derived substantially more accurate posterior probabilities.

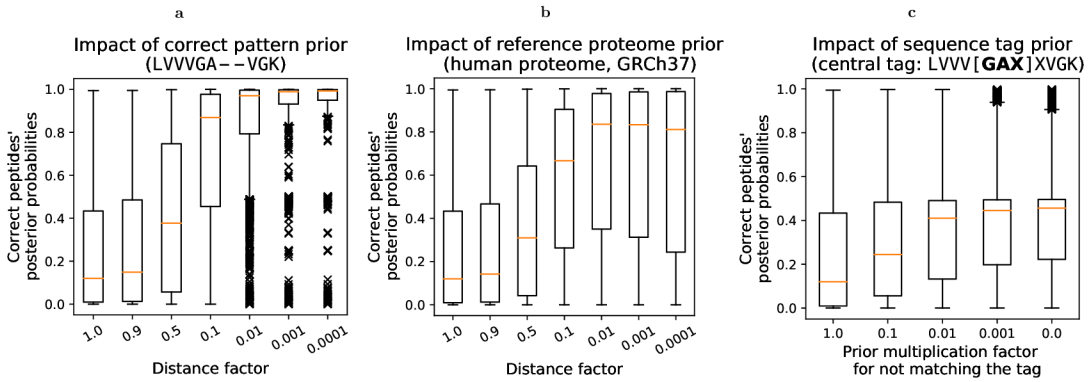


Figure 5.3: Posterior probabilities assigned to the correct peptides

(**a**) The use of the correct pattern prior increased the posterior probabilities for the correct peptides. Note that unlike in **b**, the probabilities of correct peptides increased *monotonically* with lower distance factors. (**b**) Similarly, as in **a**, the reference proteome prior increased the probabilities assigned to the correct peptides. Note that the probabilities did *not* increase monotonically with stronger importance of distance (e.g., see DF = 0.001 or 0.0001). The reason is that the reference proteome prior has only partial correspondence to the P_L dataset, and putting too much relevance on the prior model eventually starts forcing the incorrect behavior. (**c**) The plot illustrates the effect of a correct 3-length central sequence tag on posterior probabilities for correct peptides. Although the posterior probabilities increased with the certainty of the sequence tag, they did not allow confident detection for most spectra.

5.1.2 Peptide prior models improved the detection of correct peptides

Intuitively, the use of peptide prior probabilities should improve the detection of correct peptides as long as the prior distribution corresponds well to the analyzed dataset. Herein, we show that such an intuition is correct—by showing an increase in posterior probabilities of correct peptides depending on the strength and the adequacy of the prior model. The analysis thus evaluates, in a certain probabilistic sense, the increase in sensitivity of peptide detection.

Overall, we depicted the posterior probabilities of correct peptides for multiple prior models on **Fig. 5.3**. Note that each figure also contains the behavior for the uniform prior, allowing us to directly assess the relative improvements. We now briefly interpret the observed behavior.

The correct pattern prior monotonically increased the posterior probabilities with the increased importance of the distance (**Fig. 5.3a**). For low distance factors, the prior model tightly restrained the expected peptide sequences and thus essentially allowed only peptides that fit the sequence pattern LVVVGA**XX**VGK. For instance, at DF = 0.01, the sum of posterior probabilities of correct peptides represented around 94.6 % of the direct prior, and the rate further rose to 99.0 % for DF = 0.001. The posterior probabilities of correct peptides thus increased with stronger relevance of the distance, and with low-enough distance factors basically reached the performance of the direct prior.

The behavior for the reference proteome prior was similar—it also substantially increased the posterior probabilities of correct peptides (**Fig. 5.3b**). The growth, however, stopped at around DF = 0.01 because the dataset only partially corresponded to the prior model—forcing the prior thus did not further increase their posterior probabilities. Nevertheless, at DF = 0.01, the sum of posterior probabilities represented around 70.6 % of the direct prior yet required only the knowledge of reference proteins of the organism instead of the actual sequence pattern—making it generally applicable to peptide detection.

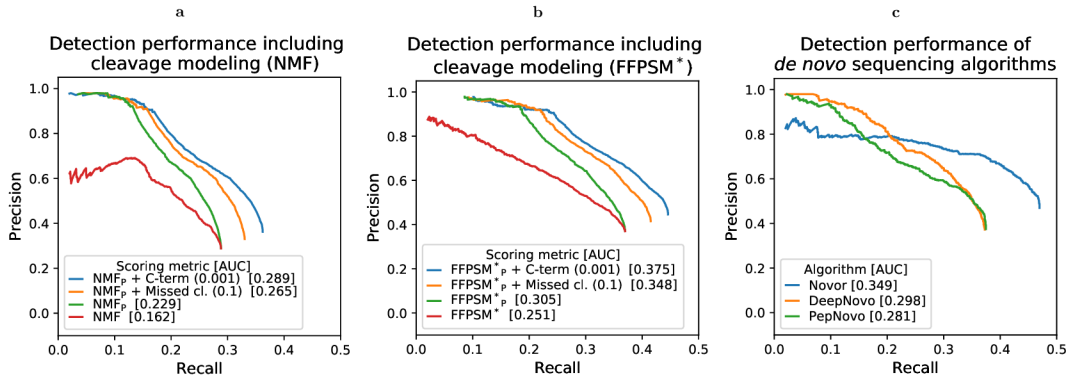


Figure 5.4: Comparison of peptide detection with state-of-the-art de novo sequencing algorithms

(**a**) Reformulating the NMF metric into its probabilistic version NMF_p improved the detection performance by allowing to select peptides at much higher precision. The utilization of probabilistic modeling of expected cleavage further improved the performance. Note that the numbers in parentheses signify the decrease in prior probabilities of peptides (0.001 in C-term for non-specific cleavage and multiplication by 0.1 for each missed cleavage within the peptide). (**b**) Similarly, as in **a**, the probabilistic version of FFPSM* outperformed its non-probabilistic counterpart. Further improvements followed with the probabilistic modeling of cleavage behavior. To read more about FFPSM*, we refer the reader to our article [1]. (**c**) The performance of the probabilistic version of simple scoring metrics was on par with the state-of-the-art *de novo* sequencing algorithms when used with probabilistic modeling of enzymatic cleavage (see **a** and **b**).

Finally, we also illustrate the relevance of a 3-length central sequence tag LVVV[**GAX**]XVGK on posterior probabilities (section 4.4.1.1). The knowledge of such a peptide substructure substantially improved the detection but was not, in general, convincing evidence for detecting correct peptides (**Fig. 5.3c**). For instance, even knowing the correct substructure with certainty (non-matching factor of 0), the medians of posterior probabilities of peptides remained below 0.5, showing an inability to uniquely detect the correct peptide. The 3-length central sequence tag had thus only limited ability to uniquely detect peptides, and we note that the same was true also for longer tags [2].

Overall, the results thus illustrate that peptide prior probabilities have a substantial impact on posterior probabilities assigned to the correct peptides. This is another way of saying that the agreement of peptides and spectra is often not powerful enough to overcome these *a priori* differences. Furthermore, in biological samples, the prior probabilities of peptides range over several orders of magnitude, further elevating their impact (section 2.2.2). In summary, due to the limited power of peptide-spectrum agreement and the large variability of peptide prior probabilities, the prior probabilities of peptides play a substantial role in peptide detection.

5.1.3 The use of prior models outperformed state-of-the-art *de novo* sequencing algorithms

We now study the detection performance of two simple scoring metrics combined with prior models of enzymatic cleavage and compare it with the performance of popular *de novo* sequencing algorithms. Overall, we show that the use of such prior models substantially improved peptide detection, up to the point of outperforming state-of-the-art *de novo* sequencing algorithms. Note that *de novo* algorithms, in contrast, typically use highly complex scoring metrics and, in essence,

model the fragmentation process of a peptide. The results thus illustrate that even weak prior models have a positive and substantial impact on peptide detection.

The **Fig. 5.4a** shows the behavior of *number of matching peaks* scoring metric (NMF) in its raw form, its probabilistic form NMF_p, and when employed with prior models based on the expected behavior of enzymatic cleavage. Interestingly, although NMF is an extremely simple metric, its performance, when combined with cleavage-derived prior models, was just slightly less than the one obtained using DeepNovo—a system utilizing deep neural networks for prediction of fragment spectra (AUC: 0.289 vs 0.298). Afterward, we considered a more advanced scoring metric, called FFPSM*, which utilizes *a priori* distribution of expected fragments to suppress noise peaks (to read more on FFPSM*, we refer the reader to our article [1]). The combination of FFPSM* with the prior model of cleavage outperformed other approaches on the analyzed dataset (e.g., AUC: 0.375 vs. 0.349 for the best performing *de novo* sequencing algorithm Novor). Note that to make the comparisons appropriate, we ran the individual *de novo* algorithms with trypsin set as an enzyme, hence allowing them to also benefit from the expected enzymatic behavior. In summary, the results thus illustrate that use of prior models based on cleavage behavior largely improved peptide detection and outperformed complex *de novo* scoring algorithms on this dataset.

Further comparisons For comparisons of the detection performance for prior models based on the distance to the library sequence pattern, we refer the reader to our article [1]. For comparisons to database search engines that mimic both the correct pattern prior and the reference proteome prior, we refer the reader to the article [2].

5.2 Detection of peptide variants in typical experiments

We now investigate the detection of peptide variants in samples that are more representative of typical experiments in computational proteomics. In particular, we analyze 61 samples of NCI_{60} proteomes [11] using four approaches for detecting peptide variants and post-process them using our deep search method that calculates scoring metrics based on Pr_max. Because we do not directly know which peptides are detected correctly, we utilize the presence of DNA sequencing support of detected peptide variants as an indicator of their correctness (NCI_{60} exomes [81], section 4.2.2).

Let us now provide a brief overview of the main results. In section 5.2.1, we show that the filtering of peptide variants using deep search scoring metrics substantially improved the detection performance for all four analyzed approaches—showing broad applicability of the method. Afterward, in section 5.2.2, we show that CLAIRE—our approach for detecting peptide variants—detected substantially more variants at much higher precision compared to the other analyzed approaches. Altogether, the results show that the use of peptide prior probabilities in conjunction with a deep search of fragment mass spectra allows substantial improvements for the detection of peptide variants.

5.2.1 Deep probabilistic search substantially improved the performance of variant peptide detection approaches

We now show that our probabilistic deep search method is generally applicable for the post-analysis of peptide detection results. In doing so, we evaluate four approaches: an exhaustive substitution of amino acids using X!Tandem (X!Tandem$_{\mathrm{ES}}$) [30], a Bayesian approach BICEPS for detecting variably-mutated sequences [54], an open-search approach MSFragger [10], and a global peptide-variant database search using X!Tandem (X!Tandem$_{\mathrm{GPV}}$). X!Tandem$_{\mathrm{ES}}$ is a database search approach that considers all amino acid substitutions of peptides constructed from a reference protein database. BICEPS works similarly as X!Tandem$_{\mathrm{ES}}$ but further utilizes sequence tags and prior probabilities of peptides to justify an increase in search space exploration—a method designed to improve precision and computational efficiency of the detection. MSFragger is an efficient implementation of the open-search detection approach [57], allowing detection of peptides with modifications of unknown masses, making it also applicable for detecting variant peptides. Finally, X!Tandem$_{\mathrm{GPV}}$ performs a database search of peptide variants constructed from globally observed DNA/mRNA variants. Altogether, we are interested in the ability of both the raw scoring metrics and those derived from the deep search to discriminate between likely correct and likely incorrect peptides—as determined by the sequencing support of the corresponding nucleotide variants (section 4.2.2).

In what follows, we will illustrate the filtering performance using multiple deep search scores derived from $\mathrm{Pr}_{\mathrm{max}}$. Let us recall that $\mathrm{Pr}_{\mathrm{max}}$ is the maximal posterior probability of a candidate peptide (section 3.1.2), and thus if $\mathrm{Pr}_{\mathrm{max}}$ is low, the candidate peptide is unlikely. However, to better handle the situations when $\mathrm{Pr}_{\mathrm{max}}$ is still high yet the peptide might be incorrect, we introduced the relaxation of $\mathrm{Pr}_{\mathrm{max}}$ at a trade-off k, denoting the metric as $\tilde{\mathrm{Pr}}^{k}_{\mathrm{max}}$ (section 4.4.2.3). The parameter k relates the importance of prior probabilities with the importance of the spectral match and serves us to circumvent the potentially complicated modeling of true and random match distributions ($k = 20$ in all our analyses). When we utilize the adjustment of peptide prior probabilities by the population frequencies of corresponding variants, we include the symbol † in the superscript (e.g., $\tilde{\mathrm{Pr}}^{k,\dagger}_{\mathrm{max}}$, section 4.4.2.2). When we assign lower prior probabilities to candidate peptides whose parental mass does not correspond to the non-monoisotopic mass, we include the letter i in the superscript (section 4.4.2.1). Altogether, this brings us to the metric $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ that utilizes all these extensions over $\mathrm{Pr}_{\mathrm{max}}$, and its behavior is of our primary interest.

For the performance comparisons, we constructed curves that relate the number of variants claimed with the precision of detection (section 4.2.2.6), and visualized them on **Fig. 5.5**. The figures show the filtering of peptide variants using their native scores compared to the probabilistic deep search score $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$. As is evident from the figure, filtering results using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ allowed selecting much more sequencing supported—and thus likely correct—variant peptides. For instance, the exhaustive substitution approach of X!Tandem$_{\mathrm{ES}}$ resulted, even for the most strict native criteria, in just around 20 % of sequencing support for variant peptides (**Fig. 5.5b**). On the other hand, filtering using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ improved the sequencing support above 70 %, and generally resulted in a much higher number of variants detected at any level of precision. In general, all analyzed approaches behaved similarly in this respect, thus showing universal applicability of the deep search approach. In conclusion, the deep search metric $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ allowed substantially more sensitive detection of candidate variant peptides compared to the native scoring metrics.

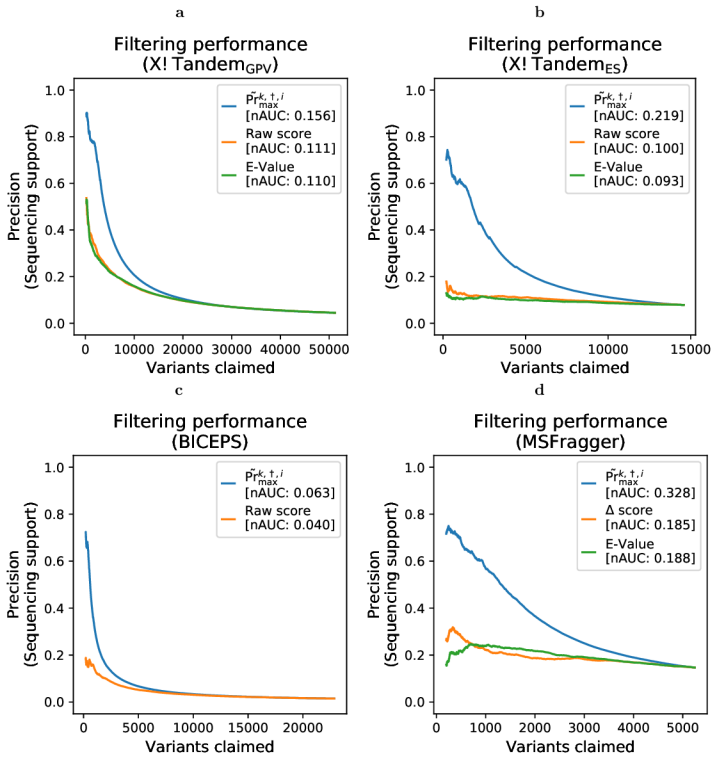


Figure 5.5: Filtering efficiency using native scores and the deep search score $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$.

(**a**–**d**) The plots show the post-search filtering efficiency of claimed variant peptides both by their native scores and using the $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ score derived from our probabilistic deep search method. In the analysis, all claimed variant peptides were subjected to the deep search, and the corresponding $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ of the claimed variant peptide was calculated. The behavior shows that individual approaches highly benefited from filtering using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ score as opposed to their native scores. Note that the normalized area under the curve (nAUC) refers to the area wherein the maximal number of claimed variants is normalized to one.

To get a better idea of where the capability of $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ comes from, we now illustrate its behavior on the deep search results of two fragment spectra (**Fig. 5.6**). For the first spectrum, we show the ability to remove variant peptides that are unlikely even though their match is highly significant. In particular, the table on **Fig. 5.6a** shows an example of a claimed variant peptide with a highly significant match as suggested by X!Tandem's global peptide-variant database search approach (E-VALUE $= 1.1 \times 10^{-7}$). Nonetheless, the claimed peptide was without sequencing support and thus was likely incorrect. In accordance, the deep search revealed another candidate peptide that was of a higher score and similar prior probability—drawing, in essence, the claimed peptide unlikely ($\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max} = 0.002496$). On the second spectrum, we illustrate the capacity to detect likely correct peptides even though their match is only mildly significant. The table on **Fig. 5.6b** shows an example of a deep search where the claimed variant peptide has an agreement shared with other peptides and is of a mediocre significance (X!Tandem$_{GPV}$ E-VALUE $= 0.029$). The table shows that the variant peptide is of a high frequency in the population, and thus its relative prior probability is correspondingly high ($\mathrm{Pr}^{\star}_{\dagger} = 0.2702$). In consequence, $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ remains high, hence preserving the claimed variant peptide ($\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max} = 0.9975$). The results thus illustrate that the probabilistic deep search approach allows both specific and sensitive detection of variant peptides based on detailed spectrum-specific circumstances.

a HIGHLY SIGNIFICANT MATCH BUT LIKELY INCORRECT DETECTION

	Candidate peptide p	$\text{NMF}_\epsilon(p, m)$	$\text{Pr}^\star_\dagger(p)$	$\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$
→	LGEHNI$^{\text{I}\to\text{V}}$EVLEGNEQFINAAK	20	5.68×10^{-5}	2.50×10^{-3}
⊙	LGEHNIE$^{\text{E}\to\text{D}}$VLEGNEQFINAAK	22	5.68×10^{-5}	0.9975
	LGEHNIEVL$^{\text{L}\to\text{V}}$EGNEQFINAAK	18	5.71×10^{-5}	6.27×10^{-6}
	L$^{\text{L}\to\text{V}}$GEHNIEVLEGNEQFINAAK	17	5.71×10^{-5}	3.13×10^{-7}
	LGE$^{\text{E}\to\text{D}}$HNIEVLEGNEQFINAAK	17	5.68×10^{-5}	3.12×10^{-7}
	LGEHN$^{\text{N}\to\text{T}}$IEVLEGNEQFINAAK	17	5.57×10^{-6}	3.06×10^{-8}
	LGEHNIEVLE$^{\text{E}\to\text{D}}$GNEQFINAAK	16	5.68×10^{-5}	1.56×10^{-8}
	LGEHNIEVLEGN$^{\text{N}\to\text{T}}$EQFINAAK	14	5.57×10^{-6}	3.82×10^{-12}
	QD$^{\text{D}\to\text{A}}$GM$^{\text{Ox}}$FDLVANGGASLTLVFER	14	2.16×10^{-6}	1.48×10^{-12}
	SVSQSSSQSLASLATT$^{\text{Methyl}}$FLQEK	14	4.74×10^{-8}	3.25×10^{-14}

b MILDLY SIGNIFICANT MATCH BUT LIKELY CORRECT DETECTION

	Candidate peptide p	$\text{NMF}_\epsilon(p, m)$	$\text{Pr}^\star_\dagger(p)$	$\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$
→⊙	SS$^{\text{S}\to\text{A}}$LFAQINQGESITHALK	9	0.2702	0.9978
	SS$^{\text{Deoxy}}$LFAQINQGESITHALK	9	2.71×10^{-4}	10^{-3}
	S$^{\text{Deoxy}}$SLFAQINQGESITHALK	9	2.71×10^{-4}	10^{-3}
	S$^{\text{S}\to\text{A}}$SLFAQINQGESITHALK	9	5.40×10^{-5}	2×10^{-4}
	SPFSLPQKSL$^{\text{L}\to\text{Q}}$PVSLTANK	9	9.08×10^{-7}	3.35×10^{-6}
	E$^{\text{Glu}}$C$^{\text{Carb}}$AHLLLAHNAPVKVK	8	5.67×10^{-6}	1.05×10^{-6}
	SPFSLPQK$^{\text{Lys}\to\text{AminoadipicAcid}}$SLPVSLTANK	8	5.56×10^{-6}	1.03×10^{-6}
	IIIQRD$^{\text{Label:15N(1)}}$SEQQMINIAR	8	5.13×10^{-6}	9.48×10^{-7}
	⊢$^{\text{Acetyl:2H(3)}}$PEFALALPPEPPPGPEVK	8	3.36×10^{-6}	6.21×10^{-7}
	AAEEAERQRQIQLAQK$^{\text{Carb}}$	9	1.60×10^{-7}	5.92×10^{-7}

Legend

⊙	The peptide with the highest $\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$ in the deep search.
→	The variant peptide claimed using X!Tandem in global peptide-variant database search.
$\text{NMF}_\epsilon(p, m)$	The number theoretical fragments of p matching a fragment in m at tolerance ϵ.
$\text{Pr}^\star_\dagger(p)$	The population-frequency adjusted relative prior probability of p.

Figure 5.6: Examples of deep search results.

The tables illustrate the discriminative power of $\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$ metric. In **a**, the variant peptide claimed using X!Tandem global peptide-variant database search (→) was of a high statistical significance but without sequencing support, indicating it is an incorrect peptide. In accordance, the deep search found a better candidate peptide (⊙) of similar prior probability, drawing the claimed variant peptide → unlikely. In **b**, the X!Tandem global peptide-variant search claimed variant peptide (→) of a mild statistical significance, but the peptide had sequencing support, indicating it is a correct peptide. Although the deep search found multiple candidates of a similar match, all were much less likely a priori, assigning high $\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$ of the variant peptide even though its spectral match was only mildly significant.

5.2.2 CLAIRE outperformed other approaches on detection of SNV-peptides

We now turn to the comparison of our peptide variant detection system CLAIRE with the other detection approaches introduced in the previous section. First, we show that CLAIRE substantially outperformed other approaches in terms of detected variant peptides. Afterward, we show that the deep search metrics had generally much higher correlations with sequencing support—allowing, in essence, to better separate between likely correct and likely incorrect detections. Finally, we look at the search depth of our method, showing that it evaluates up to one million candidates per fragment spectrum.

We visualized the comparison in terms of precision and number of variants claimed on the **Fig. 5.7a**. As is clear from the figure, CLAIRE substantially outperformed other analyzed approaches on this dataset. For instance, utilizing the normalized area under the curve (nAUC) metric, the corresponding nAUC for CLAIRE was high relative to other approaches (nAUC = 0.156 for CLAIRE vs. nAUC = 0.026 for BICEPS, nAUC = 0.018 for X!Tandem$_{\text{ES}}$,

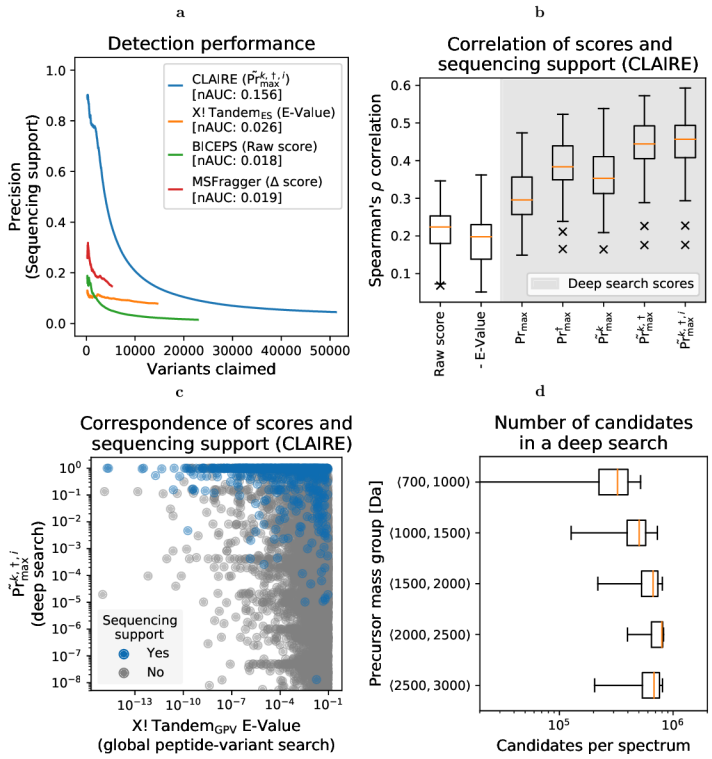


Figure 5.7: Overall view of CLAIRE's behavior.

(**a**) CLAIRE substantially outperformed other analyzed detection approaches in detecting sequencing-supported variant peptides. (**b**) The boxplot shows the correlation of scores and sequencing support of claimed variant peptides aggregated over individual samples. Note that the higher the correlation, the more likely we are to retain sequencing supported—and thus likely correct—variant peptides when filtering using a more strict criterion. As the plot indicates, the deep search score metrics were generally of higher correlations, showing that these metrics were better at determining likely correct peptides. (**c**) The plot shows that high $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ was a much better indicator of the correctness of variant peptide than the statistical significance of claimed variant peptide using X!Tandem's global peptide-variant search. (**d**) The plot shows the number of candidate peptides considered in the deep search per mass spectrum. The numbers of candidate peptides slightly increased with the precursor mass of peptides but were generally less than one million. Note that in our analyses, we considered precursor mass tolerance of 10 parts-per-million and mass shifts corresponding to one of $\{-2, -1, 0, 1, 2\}$ neutrons.

and nAUC = 0.019 for MSFragger; nAUC refers to the area under the curve when the maximal number of claimed variants is normalized to one). One reason for CLAIRE's performance is the initial use of X!Tandem$_{\mathrm{GPV}}$ which considers peptides built from variants already observed on a global level, and such peptides are more likely *a priori*. In line with this, CLAIRE retains such candidate variant peptides even if they are of a mild significance (i.e., E-VALUE ≤ 0.1). Afterward, CLAIRE performs deep searches to allow highly sensitive filtering based on scoring metrics derived from $\mathrm{Pr}_{\mathrm{max}}$. In consequence, this allows CLAIRE to retain a high number of variant peptides.

We now turn to an alternative evaluation of the filtering performance by evaluating the correlations between sequencing support of claimed variant peptides and their scores. The figure **Fig. 5.7b** shows such correlations for the raw X!Tandem$_{\mathrm{GPV}}$ scores, i.e., HyperScore and E-Value [30], in comparison to $\mathrm{Pr}_{\mathrm{max}}$ and its extensions. As is clear from the figure, filtering using metrics

derived from $\mathrm{Pr}_{\max}$ exhibited substantially higher correlations with the sequencing support (e.g., Spearman's $\rho = 0.457$ for $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ vs. $\rho = 0.224$ for HyperScore; medians over all samples). In other words, by choosing a more strict criterion using deep search metrics, we are more likely to retain peptides that are sequencing-supported and thus likely correct. Further, the figure shows that the relaxation of $\mathrm{Pr}_{\max}$ has a substantial impact in this respect (Spearman's $\rho = 0.353$ for $\tilde{\mathrm{Pr}}^{k}_{\max}$ vs. $\rho = 0.296$ for $\mathrm{Pr}_{\max}$; medians over all samples). Similarly, the figure shows that adjusting the prior probabilities by population-frequency of corresponding nucleotide variants substantially elevates the correlation (Spearman's $\rho = 0.444$ for $\tilde{\mathrm{Pr}}^{k,\dagger}_{\max}$ vs. $\rho = 0.353$ for $\tilde{\mathrm{Pr}}^{k}_{\max}$; medians over all samples). As a result, the population frequency of individual variants plays a significant role in detection, and thus some variant peptides are easier to detect than others. Finally, we note that the utilization of lower prior probabilities based on neutron shifts resulted in a minor improvement (Spearman's $\rho = 0.457$ for $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ vs. $\rho = 0.444$ for $\tilde{\mathrm{Pr}}^{k,\dagger}_{\max}$; median over all samples). In summary, the scores derived from the deep search had shown a substantially higher capacity to discriminate between likely correct and likely incorrect variant peptides.

Finally, we focus on a more peripheral aspect of peptide detection using CLAIRE. First, we directly visualized the relationship between X!Tandem$_{\mathrm{GPV}}$'s E-Values of variant peptides and their respective $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ (**Fig. 5.7c**). The figure shows that most of the sequencing-supported variant peptides had high $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$, and thus the metric is a better indicator of correctness than the X!Tandem$_{\mathrm{GPV}}$'s E-Value of the spectral match. From a computational perspective, we visualized the number of candidate peptides tested by the deep search approach (**Fig. 5.7d**). Given the depth of our peptide database $p_{\min} = 4 \cdot 10^{-6}$, a precursor tolerance of 10 parts per million and five allowed neutron shifts, the deep search generally considered less than one million candidates per spectrum. Note that because the fast spectral match algorithm runs in linear time (section 3.2.6), this does not translate into substantial computational problems. Our deep search method thus allowed testing against a large number of candidate peptides, and the use of the more realistic prior probability model enabled efficient discrimination between likely correct and likely incorrect variant peptides.

5.3 Downstream applications

Herein, we provide several downstream applications of CLAIRE in typical shotgun proteomics experiments. First, we focus on the detection of protein somatic variants in section 5.3.1, showing the evidence that CLAIRE can detect *hypermutation status* of tumors—a relevant clinical parameter. Afterward, in section 5.3.2, we present a large-scale analysis of germline variants within NCI60 datasets, revealing several mislabeled and contaminated cell lines in public datasets—showing an application in research reproducibility. Finally, in section 5.3.3 we provide an application in forensics by identifying family members against DNA dataset. The content of the section is adapted from our article [3], which contains additional analyses.

5.3.1 CLAIRE recognized tumors suitable for immunotherapy

We now investigate the protein and gene variation rates of patients with colorectal cancer using data from the Clinical Proteomic Tumor Analysis Consortium [85]. Colorectal cancer (CRC) is the third most common cancer worldwide, expected to result in more than 2.2 million cases

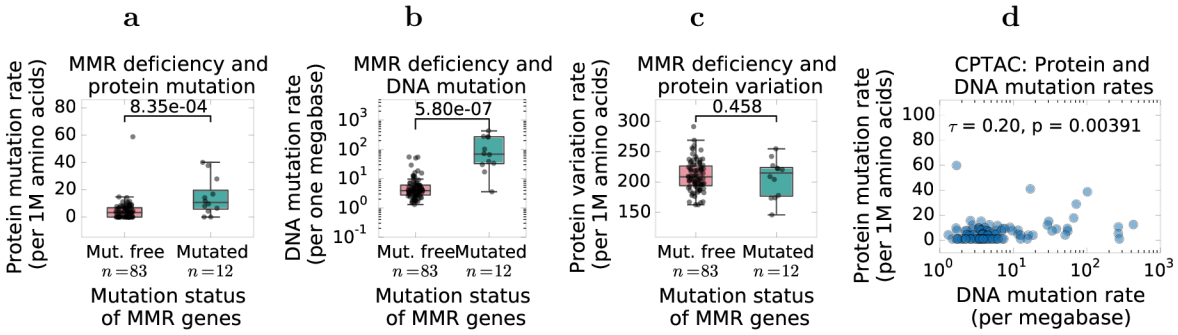


Figure 5.8: Gene and protein variation rates in patients' samples.

(**a-b**) The plot **a** shows that the rates of somatic protein variants were elevated in samples with deficient DNA mismatch-repair mechanisms (MMR). A similar but much more pronounced difference in DNA variation rates can be seen for the corresponding gene variation rates **b**. (**c**) The plot shows that the MMR deficiencies did not affect the rates of inherited protein variation, thus serving as additional control of the method. (**d**) The plot shows that although the somatic variation rates corresponded to a certain degree on the protein and gene level, some samples had also shown rather large disparities. As a result, it would be interesting to know which rates better predict the efficacy of immunotherapeutic cancer treatment—leaving room for future investigations.

annually by 2030 [97]. Around 14% of CRCs have so-called *MSI/hypermutation status*, which makes these tumors more likely to elicit an immune response and thus more suitable for immunotherapy [83, 84, 92]. The MSI/hypermutation status in these cancers is mostly a result of deficiencies in mismatch repair mechanisms (MMR), evidenced commonly in MLH1, MSH2, MSH6, and PMS2 genes [98]. The categorization of patients based on the MSI/hypermutation status is thus of clinical importance and allows oncologists to select preferable therapies.

To assess the ability of CLAIRE to detect the MSI/hypermutation status, we analyzed protein variation rates in the colorectal cancer patients cohort, depending on the presence of MMR deficiencies. We found that tumors with somatic variation in any of the four common MMR genes had shown a significantly higher rate of protein somatic variation than did the non-deficient ones (median 10.6 vs. 3.3 somatic variants per 1M amino acids, Mann-Whitney $U = 224.0, p \approx 8.35 \times 10^{-4}, n_1 = 12, n_2 = 83$). A similar but more striking difference can also be seen in the data of somatic variants detected by the exome sequencing (median 66.1 vs. 3.9 somatic variants per megabase, Mann-Whitney $U = 60.5, p \approx 2.1 \times 10^{-6}, n_1 = 11, n_2 = 79$). Note that the deficiencies in MMR genes did not affect the rates of protein germline variation, thus serving as additional control of the method (median 215.0 vs. 208.6 germline variants per 1M amino acids, Mann-Whitney $U = 488.0, p \approx 0.458, n_1 = 12, n_2 = 83$). Interestingly, some patients exhibited discordance between protein and DNA rates of somatic variation, leaving room to investigate further the implications of this difference in terms of clinical relevance (**Fig. 5.8d**). CLAIRE thus detected a higher protein somatic variant rate in tumors with deficient mismatch repair mechanisms, showing the potential to identify MSI/hypermutated tumors and thus to select patients suitable for immunotherapy.

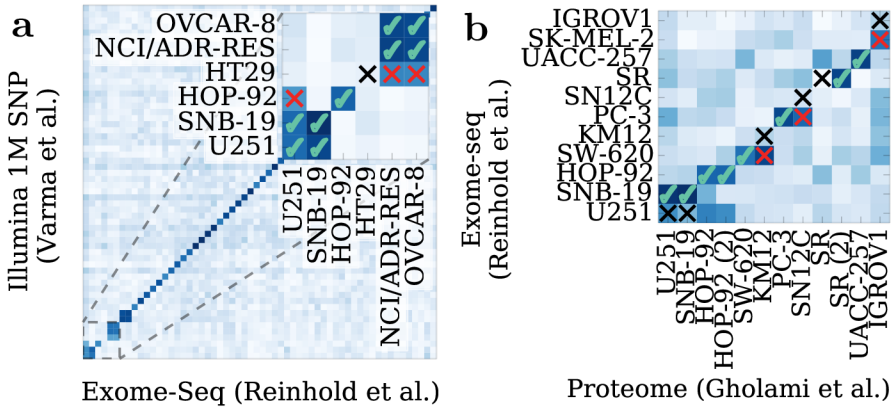


Figure 5.9: Pairwise matches between three NCI_{60} datasets.

(**a**) The heatmap shows the variant matches between Illumina 1M SNP dataset and Exome-Seq dataset. The inconsistencies are depicted using the cross symbol—the lack of an expected relationship in black and the presence of an unexpected relationship in red. (**b**) The heatmap shows the inconsistent relationships between the NCI_{60} Exome-Seq dataset and the NCI_{60} proteome dataset.

5.3.2 Large-scale variant analysis revealed inconsistencies in public datasets

Reproducibility is a significant issue in biomedical research, which is often worsened by *mislabeling of cell lines* [99]. Mislabeling of a cell line refers to a situation when researchers unknowingly work on other than the claimed cell line. The extent of the problem is rather large—analyses of major cell repositories have shown that, in some cases, as many as 20 % of all deposited cell lines were mislabeled during submission [100]. To alleviate this issue, researchers are required to deposit raw data into publicly available datasets to allow re-analysis of results by other groups (e.g., Sequence Read Archive [101] in genomics, and ProteomeXchange [102] in proteomics). Unlike in proteomics, genomic data allow simple authentication of cell lines [103]. However, the ability to detect protein variants allows shotgun proteomics to fulfill this function as well, and we illustrate this on the analysis of samples from NCI_{60} cell lines [11, 81, 82].

5.3.2.1 Analysis of significant relationships among NCI_{60} datasets

Herein, we investigate the utility of detected germline variants to establish significant relationships between NCI_{60} samples using the methods from section 4.3.2. A significant match between a pair of samples then indicates that they are genetically related. As the situation with cell lines in NCI_{60} datasets is quite entangled, we illustrate the analysis on a few examples and refer the reader to the full study in our article [3].

Let us first point out that three pairs of samples within NCI_{60} are genetically related, and we would thus expect to see significant relationships between them. The three pairs of genetically related cell lines within NCI_{60} are as follows:

(a) OVCAR-8 and NCI/ADR-RES;
(b) ME-14 and MDA-MB-435; and
(c) SNB-19 and U251.

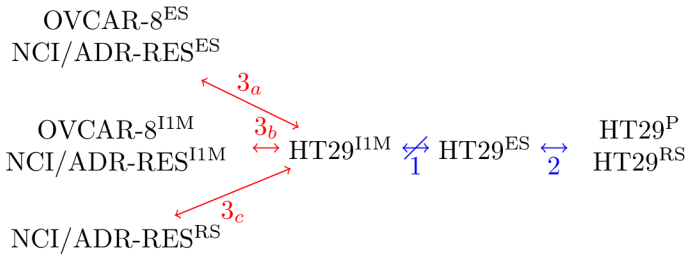
With these prerequisites, we now turn to the analysis. In what follows, we will restrict the analysis to genetic datasets measured using Illumina 1M SNP (I1M) [82], Exome-Seq (ES) [81],

RNA-Seq (RS) [81], and the proteomics dataset (P) [11] analyzed using CLAIRE [3]. As an example, **Fig. 5.9** shows raw pair-wise matches between data of I1M and ES, and ES and P. Overall, the figure shows that some unexpected relationships did show up, while some expected relationships were missing. In turn, we interpreted the observed relationships as mislabeling and contamination of cell lines, and we now provide a more detailed study of a few such discrepancies.

Notation We will use the label of a sample and the superscript of the corresponding dataset to refer to the sample of interest. Thus, for instance, HT29^{ES} refers to a sample labeled as HT29 in the Exome-Seq (ES) dataset.

Mislabeling of HT29 in Illumina 1M SNP dataset

The **Fig. 5.9a** showed a lack of expected correspondence between HT29^{ES} and HT29^{I1M}. Such a lack of correspondence was of importance because other expected matches were highly statistically significant (median of p-values: 2.016×10^{-52}). To simplify the explanation, we visualized the situation on a diagram that summarizes the status of matches between the relevant samples:

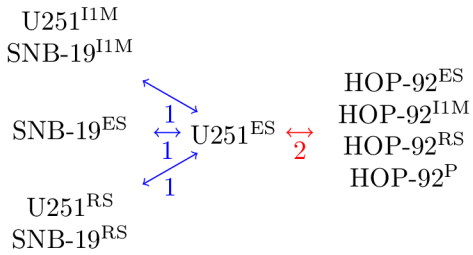


Arrow	Meaning
⟷	Expected relationship
⟷ (crossed)	Lack of expected relationship
⟷	Unexpected relationship

The lack of expected match of interest is the one between HT29^{I1M} and HT29^{ES} depicted by the arrow 1. Overall, the data indicate that HT29^{I1M} is mislabeled. In particular, we have evidence that HT29^{ES} is indeed HT29 because HT29^{ES} also matched HT29^{P} and HT29^{RS} but no other samples (arrow 2). On the other hand, we have evidence that HT29^{I1M} is not HT29 because it matched cell lines OVCAR-8 and NCI/ADR-RES but no other samples (arrows 3_a, 3_b, and 3_c). Note that OVCAR-8 and NCI/ADR-RES are genetically related cell lines; thus, if HT29^{I1M} is indeed one of them, we would expect it matches both cell lines. Based on this, we conclude that HT29^{ES} is likely HT29 and HT29^{I1M} is either OVCAR-8 or NCI/ADR-RES, with more evidence for the latter (arrow 3_c). In summary, the data thus suggests that HT29 in Illumina 1M SNP dataset is likely mislabeled.

Contamination of U251 by HOP-92 in Exome-Seq

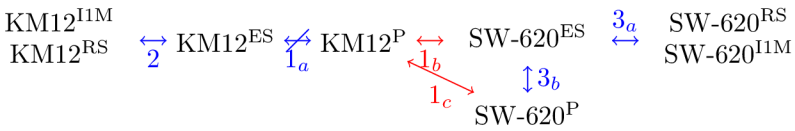
Now, we provide an analysis in the similar spirit of the previous one, showing that U251^{ES} was likely contaminated by HOP-92. To simplify the exposition, we again provide the diagram of the relevant matches:



Foremost, the data indicates that U251^{ES} was likely U251 because it also matched the relevant samples in other datasets (arrows 1). Again, because U251 is genetically related to SNB-19, we expect to see significant matches to SNB-19 too. However, U251^{ES} also matched HOP-92 in all four analyzed datasets (arrow 2). As a result, we conclude that U251^{ES} was likely contaminated by HOP-92.

Mislabeling of KM12 in NCI_{60} proteomes

We now show a last example of a mislabeled cell line—in NCI_{60} proteome data (**Fig. 5.9b**). The diagram of the relevant matches is as follows:



Overall, the data indicate that KM12^{P} was actually SW-620. Foremost, KM12^{P} did not match KM12^{ES} (arrow 1_a) but did match SW-620ES and SW-620^{P} instead (arrows 1_b and 1_c, respectively). Furthermore, the KM12^{ES} was indeed likely KM12, as indicated by its significant matches in other datasets (arrow 2). Similarly, the SW-620ES was likely SW-620 as indicated by its matches in other datasets (arrows 3_a and 3_b). As a result, we conclude that KM12^{P} was indeed SW-620.

Other mislabeled and potentially contaminated cell lines

The previous analyses presented an interpretation of three issues within the public NCI60 datasets. However, as there were more discrepancies, we refer the interested reader to our article for further details [3]. Therein, we also consider additional datasets and additional criteria for evaluating the correspondence between samples. Finally, we provide an overall summarization of the discrepancies on **Fig. 5.10**.

5.3.2.2 Authentication of cell lines

The NCI_{60} panel contains a rather large number of samples, allowing us to build the null models of the variant match rather easily and use statistical methods to discover genetic relationships. Herein, we provide an alternative approach to authenticate cell lines based on the population frequencies of individual variants (section 4.3.1). Further, we note that this approach is also suitable for use when the number of samples is small or the individual samples are genetically related (section 5.3.3).

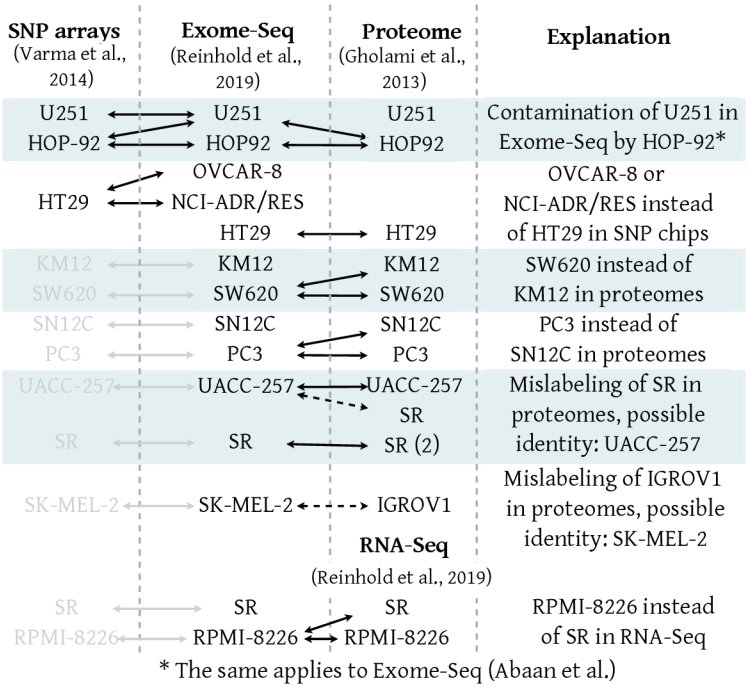


Figure 5.10: Summary of the analysis of NCI_{60} datasets using germline variants.

The table on **Fig. 5.11** shows the results of cell line authentication of NCI_{60} proteomes [11], which are in line with the analysis from the previous section. For instance, the method again derived that $KM12^{P}$ is actually SW-620, or that $SN12C^{P}$ is PC-3 (we did not present the latter in the previous section, but the same applies). The approach further identified less clear-cut discrepancies, and we again refer the interested reader to [3] for their detailed analysis. In summary, the method suggested the possibility to routinely authenticate the origin of cell lines based on protein variants, which has applications in the costly problem of research reproducibility.

5.3.3 Peptide variants identified individuals against DNA database

We now present an application of CLAIRE in forensics, wherein we show the ability to identify genetically-related individuals from their protein variants—by matching against the corresponding DNA dataset. For this purpose, we use the population-frequency method to calculate probability of DNA origin (section 4.3.1) and analyze the data of a seven-member family (section 4.1.2).

The probabilities of individual DNA origins for each family member are visualized on the **Fig. 5.12**. The table shows that except for one of the monozygotic twins, the identities of all individuals were resolved correctly (Pr_{err} ranged from 5.34×10^{-13} to 1.01×10^{-2}). Further, the probabilities of error for both twins were substantially elevated ($Pr \approx 0.45$ and $Pr \approx 0.44$), showing that the approach also correctly captured the impossibility to resolve their identities based on genetic variation. CLAIRE has thus shown a potential to identify individuals from protein samples and may be useful in forensic medicine, e.g., when DNA samples are unavailable, or other analyses are inconclusive.

Sample	Claimed cell line	Best cell line	$\Pr_{\text{err}}$	
P001891	UACC–62	UACC–62	9.39×10^{-17}	
P001565	HT29	HT29	5.84×10^{-18}	
P003198	NCI–H23	NCI–H23	7.52×10^{-18}	
P003207	ACHN	ACHN	3.07×10^{-15}	
P003381	NCI–H460	NCI–H460	9.90×10^{-15}	
P003199	MALME–3M	MALME–3M	1.04×10^{-11}	
P001906	A549	A549	1.53×10^{-10}	
P003362	MCF7	MCF7	3.36×10^{-10}	
P0001751	SN12C	PC–3	3.17×10^{-9}	✗
P001897	UACC–257	UACC–257	3.42×10^{-9}	
P003208	HCT–15	HCT–15	3.50×10^{-9}	
P001570	SK–MEL–5	SK–MEL–5	1.40×10^{-8}	
P003820	KM12	SW–620	2.85×10^{-8}	✗
P001389	MDA–MB–231	MDA–MB–231	3.17×10^{-8}	
P003196	A498	A498	1.11×10^{-7}	
P001888	OVCAR–4	OVCAR–4	1.24×10^{-7}	
P003487	PC–3	PC–3	3.27×10^{-7}	
P003203	TK–10	TK–10	1.84×10^{-6}	
P001567	RXF–393	RXF–393	1.93×10^{-6}	
P003201	UO–31	UO–31	2.31×10^{-6}	

Figure 5.11: Results of cell line authentication.

The table shows an excerpt of cell line authentication using the population frequency method developed in 4.3.1 for probabilistic assignment of DNA origin to variant peptides. The table is ordered by the lowest probability of error of cell line authentication.

Other analyses For further analysis of the data of family members, we refer the interested reader to our article [3] which also includes the reconstruction of variant clusters between family members—both on protein and gene level.

DNA sample	Best matching protein sample	Error prob. (P_{err})	
Father	Father	$1.01 \cdot 10^{-2}$	✓
Mother	Mother	$2.80 \cdot 10^{-8}$	✓
Daughter 1	Daughter 1	$6.63 \cdot 10^{-6}$	✓
Daughter 2	Daughter 2	$4.66 \cdot 10^{-11}$	✓
Daughter 3	Daughter 3	$5.34 \cdot 10^{-13}$	✓
Son 1 (twin)	Son 1 (twin)	0.45	✓
Son 2 (twin)	Son 1 (twin)	0.44	

Figure 5.12: Identification of individuals against DNA database.

The table shows the results of applying the methods in 4.3.1 to detect genetically-related individuals against a DNA database. Note that the only misidentification was that of a monozygous twin, which was, however, also indicated by a higher probability of error.

Chapter 6

Discussion

This chapter discusses several aspects of the methods proposed in our research, their other potential downstream applications, and summarizes answers to our research questions. In section 6.1, we discuss the option of employing the deep search approach as a standalone system for detecting peptides—instead of post-processing detection results of other approaches. Peptide prior probability models lie at the core of our research, and in section 6.2, we focus on limitations and improvements of the more realistic prior model. In section 6.3, we discuss the utility of large peptide databases and their value for storing precomputed data relevant in peptide detection. We discuss other potential applications of our methods in section 6.4, and we conclude the chapter by answering the research questions we formulated in the introductory chapter.

6.1 Deep search for standalone peptide detection

In our research, we applied the methods based on the maximal posterior probability ($\mathrm{Pr}_{\mathrm{max}}$) only to post-process detection results of other peptide detection approaches (section 5.2). As the performance of the post-processing was fairly high (**Fig. 5.5**), it is natural to ask whether a standalone deep search system would allow effective detection of unlikely peptides. Ideally, such a system should also derive posterior probabilities that accurately capture the correctness of the detected peptides in the long run. Note that in the analyses of the combinatorial peptide library (section 5.1), our Bayesian model derived accurate posterior probabilities even for homologous peptides under multiple circumstances [1, 2]. However, this analysis investigated search strategies related to a complete search—a strategy that considers all candidate peptides for a given precursor mass. Now, we discuss whether it is reasonable to expect that a deep yet incomplete database search can do the same.

Deriving accurate posterior probabilities in incomplete searches is problematic. As we have shown previously [2], incomplete searches are prone to calculating inadequate posterior probabilities, and this also affects other measures of confidence [1]. The reason is simple—in an incomplete search, we can miss a high-scoring peptide because it was not present in the search database. Missing a high-scoring peptide is of potential concern because the highest posterior probabilities are typically assigned to the highest-scoring peptides; the situation depends, of course, on their prior probabilities [1]. Note that in practice, we are usually interested in peptides detected with high posterior probabilities (say, ≥ 0.9) as these are most useful in the follow-up analyses. As posterior probabilities for all candidate peptides sum to one, missing a high-scoring peptide of a

relatively high prior probability in an incomplete search is thus likely to result in an inadequate estimation of posterior probabilities—especially in the probabilistic range of interest [1].

Nevertheless, we can quickly obtain all high-scoring peptides for additive scoring metrics, such as NMF, using score-histogram methods [65, 69]—employing a search strategy that we called *tail-complete* search [2]. As a result, integrating deep search with the tail-complete search then allows us to essentially extend the deep search into a complete search. Then, for each new high-scoring peptide discovered using the tail-complete search, we assign its prior probability, and the analysis continues as usual. One option for the assignment of prior probabilities to newly discovered peptides is to set them just below $p_{\min}$—the minimal relative prior probability used in the construction of the deep database (otherwise, the peptides would be in the deep database). Once assigned, we can then directly utilize our Bayesian model to calculate the posterior probabilities (section 4.4.1.2), or calculate $\tilde{\Pr}^{k}_{\max}$ if we want to avoid modeling the true and random distributions of peptide-spectrum matches.

Notably, in the analysis of the combinatorial peptide library, we have shown that as long as the tail-complete search is sufficiently complete, it derives posterior probabilities that are highly similar to those derived from the complete search [2]. For instance, 3-tail-complete search—search containing all peptides with at most 3 matching peaks less than the highest-scoring peptide among all theoretical peptides—resulted in very high correlations of posterior probabilities compared to the complete search (Spearman's $\rho = 0.9993$). To summarize, a standalone deep search system is likely to allow efficient detection and calculation of accurate posterior probabilities; it is, however, necessary to make it more complete, and one such option is to integrate it with the tail-complete search strategy.

6.2 Improvements of the peptide prior probability model

The more realistic prior probability model that we developed in section 3.2.4 allows assigning distinct probabilities to individual peptide-producing events, which can result in highly diverse prior probabilities for individual peptides. Nevertheless, we kept the parameters of the model simple—assigning just rough estimates of probabilities to a few classes of peptide-producing events (section 4.4.3.2). As illustrated by the filtering efficiency on **Fig. 5.5**, the approach worked already reasonably well for detecting variant peptides, a problem which typically results in high rates of false positives [7–9, 104]. Still, more precise information about the peptide-producing events—e.g., exact probabilistic behavior of enzymatic cleavage—is likely to further improve the detection just by changing the relevant parameters. As a result, our existing prior probability model can be adjusted depending on the available biological knowledge, and this will likely result in improved peptide detection performance.

We now discuss several possible extensions of the prior model. As shotgun proteomics experiments are biased towards more abundant proteins, one can utilize a protein abundance database, such as PaxDB [105], to adjust prior probabilities—peptides from more abundant proteins are more likely *a priori*. To improve the peptide cutting model, one can use a recent deep-learning system DeepDigest [106] to obtain accurate sequence-dependent cleavage probabilities—instead of fixed probabilities in our cleavage-after-residue model (section 3.2.4). As some peptides are more suitable for detection using mass spectrometry, one can employ the DeepMSPeptide system

for predicting peptide detectability and thus further adjust the prior probabilities of individual peptides [107]. From a direct evidence-based perspective, UniProt [18] database contains a large number of already detected post-translational modifications (PTMs) *on a particular residue of a particular protein* and such sequence-specific PTMs are thus more likely *a priori*—a situation similar to nucleotide variants of high population frequencies. One can also use the estimated rates of protein-synthesis errors [108] for detecting peptides that did not originate from DNA variation but by an error in translation of mRNA to proteins. Overall, multiple shotgun proteomics tools and databases can be employed to likely improve peptide detection by adequately modeling the peptide prior probabilities. Nonetheless, we note that all such improvements also need to consider the necessary adjustments to the peptide enumeration algorithm (section 3.2.5). In particular, our more realistic yet relatively simple model of prior probabilities (section 3.2.4) allowed us to obtain all peptides with a relative prior probability above some prespecified threshold $p_{\min}$—and thus allowing us to calculate $\mathrm{Pr}_{\max}$. However, even though the enumeration algorithm was rather straightforward in our case, it is likely to require care when incorporating some of the less clear-cut predictive models.

Finally, we discuss a conceptual change to the prior probability model by shifting it closer to the genomics data. Our current peptide enumeration algorithm works on the level of proteins, and an algorithm that builds peptides from DNA-level data would be preferable. Foremost, such an extension would allow direct utilization of the population frequency of individual DNA variants, which we now adjust using our sub-optimal procedure (section 4.4.2.2). The algorithm would also allow incorporating the behavior of particular mutagens and their preference for creating variation in DNA [109], further diversifying prior probabilities of individual variant peptides. In addition, as most amino acids are encoded using multiple RNA codons, this draws some amino acid substitutions more likely depending on the RNA codon that encodes them. For instance, if an amino acid `R` was coded using the `AGA` codon, it can result in the amino acid `S` by two different SNVs: either `AG[A→C]` or `AG[A→T]`. However, if `R` was coded by `CGC`, it can result in `S` only by one SNV: `[C→A]GC`. From this perspective, the substitution `R` → `S` is thus more likely *a priori* if the RNA codon behind `R` is `AGA`. Therefore, implementing the peptide enumeration algorithm on the DNA level would better capture prior probabilities of individual peptides, and this should again likely translate to better peptide detection performance.

6.3 Utility of deep peptide databases

For the analysis of typical shotgun proteomics experiments, we built a database of peptides with a minimal relative prior probability $p_{\min} = 4 \times 10^{-6}$ that had around 400 GB after multiple technical optimizations (mass range of 700–3 000 Da, section 5.2). Although the database is relatively large, the use of the fragment-ion index allows calculating the peptide-spectrum matches in linear time (section 3.2.6), and the precursor-mass indexation with the memory-load optimization allow loading just the portion of the database required for the analysis (sections 3.2.6.3 and 3.2.6.4). Because the database is built infrequently, one can also precompute further relevant data useful in peptide detection. For instance, we and others have shown that the use of retention time—the time it takes a peptide to enter the mass spectrometer using liquid chromatography—improves peptide detection [2, 110–113]. The retention time for each candidate peptide from the

database can be thus precomputed, possibly even using highly accurate methods [113]. When interpreting fragment spectra, the retention times can be easily aligned to a particular experiment under investigation, and the deviations from observed retention times used to update posterior probabilities of individual peptides [2]. Similarly, instead of using simple theoretical spectra of peptides, one can predict more accurate MS^2 spectra using some of the recently developed methods for the purpose [113–115]. Such MS^2 spectra exhibit high correspondence with the real fragment spectra, making the spectral matches more discriminative while still allowing to build fragment-ion indexes and thus perform fast matching. One could also precompute the expected isotopic distributions of peptides and use them for matching on the MS^1 level [2]. Although these were of limited importance in our analysis of the combinatorial peptide library due to its highly homologous nature, they might still be valuable for analyzing typical proteomics experiments. The ability to quickly match peptides against large peptide databases thus further invites for precomputing more data relevant for each candidate peptide, and the use of such predictive data is likely to improve peptide detection performance.

6.4 Further applications

Although the thesis presented applications of our methods in cancer research, research reproducibility, and forensics, let us further discuss some other applications that we described in more detail in our patent application [4] and our article [3]. For instance, the use of prior probabilities also allows rather natural detection of non-host peptides, e.g., bacteria peptides in human samples, by adequately scaling down their prior probabilities. In particular, we illustrated that such a method can detect mycoplasma [4], bacteria that commonly contaminate samples and largely affect their behavior, negatively impacting research reproducibility [116]. The method for detecting peptide variants might also have applications in personalized medicine for signalizing an early rejection of a transplanted organ [4]. In this case, the detection system aims to detect uniquely donor proteins—based on donor-specific SNV-peptides—that are circulating in the blood of the host if the host's immune system is attacking the donor's organ. In our article [3], we also investigated the possibility to derive tumor stage of colorectal cancer patients depending on the observed protein mutation rate, analogously as can be derived using DNA mutation rate. Although the protein mutation rates increased with the tumor stage, the growth was very mild (Kendall's $\tau = 0.12$, $p = 0.075$), indicating that deeper proteomics measurements are likely needed to implement such application in practice. In summary, the relative richness of potential applications of our peptide variant detection methods indicates their general utility.

6.5 Answers to the research questions

Herein, we provide summarized answers to the research questions we formulated in the introductory section 1.1.

[Q_1] How effective are the peptide detection methods in detecting variant peptides?

Our literature review (section 2.2) indicated that popular peptide detection methods result in a rather low number of detected variant peptides and potentially largely incorrectly estimated

error rates [1, 7–9, 104]. Note that the behavior was also evident in our analysis of typical proteomics experiments wherein we considered four popular approaches (**Fig. 5.5**, **Fig. 5.7**). An effective approach for detecting peptide variants is employed in proteogenomics—albeit at the substantial costs of additional biochemical analyses. Therein, the researchers first sequence DNA or mRNA of the sample, construct a sample-specific protein database and utilize standard database search—the results of such approach are generally reliable (section 2.2). Nevertheless, we have shown that once we adequately employ peptide prior probabilities in detection, we can detect peptides reliably even in very large database searches that otherwise end up with incorrect error rates [7–9, 104] (**Fig. 5.5**). Furthermore, we have shown that one can estimate accurate posterior probabilities in extensive searches, and we demonstrated this for search spaces up to 10^8 candidates in our combinatorial peptide library [1, 2]. Notably, prior probabilities also allow us to naturally interpret why the popular proteogenomics approach is reliable. In particular, once the DNA or mRNA of the sample is sequenced, the variant proteins from a sample-specific database become highly likely *a priori*—the search is then essentially equivalent to the well-established detection of reference peptides.

[Q_2] What factors do impact the precision and recall of the variant peptide detection?

Our works [1, 2] suggested that to obtain precise results, the detection method might need to consider all theoretical high-scoring peptides for a fragment mass spectrum. Particularly, we have found that the biggest obstacle for accurately estimating error rates is the absence of a high-scoring peptide in a search database—the relevance of the existence of such a peptide then depends on its prior probability [1, 2]. Note that this is in contrast to many popular approaches, which often consider just the best matching peptide per spectrum [5], hence opening possibilities for incorrect estimation of error rates [1, 2].

The recall of peptide detection directly depends on the presence of a variant peptide in the search database, and thus large databases naturally have more options to detect variant peptides. However, methods based on the statistical significance of spectral match quickly lose sensitivity in such searches due to an increase in search space [5], and similar problems affect other approaches [117]. To reduce the search space size but keep it relevant, hybrid methods use sequence tags to prefilter it, and this substantially improves detection [55, 118]. Nevertheless, we have found that even definite knowledge of correct sequence tags has only limited applicability for discriminating homologous peptides [2]. In contrast, the use of peptide prior probabilities allows adjusting the detection to the detailed spectrum-specific situation, resulting in a substantially improved recall of the method (**Fig. 5.6**).

[Q_3] What factors do impact the detection of the individual, i.e., sequence-specific, variant peptides?

Our analyses revealed that variant peptides that are more common in the human population are substantially easier to detect than infrequent variant peptides (section 5.2.2). Note that such a result is intuitive—peptides more likely *a priori* require less strict criteria for their correct detection at the same level of precision. In accordance, the adjustment of peptide probabilities of variant peptides by population frequency of the corresponding DNA variant substantially improved peptide detection (e.g., $\tilde{\mathrm{Pr}}^{k}_{\mathrm{max}}$ vs $\tilde{\mathrm{Pr}}^{k,\dagger}_{\mathrm{max}}$, **Fig. 5.7)**. Thus, the detection of sequence-

specific variant peptides also substantially depends on their frequency in the population. Finally, we note that because shotgun proteomics data are biased towards more abundant peptides, variant peptides from more abundant proteins are more likely to be detected (this, however, holds for peptides in general).

[Q_4] What are the ways to validate variant peptide detection methods?

In our research, we considered two strategies: direct validation (section 4.2.1) and sequencing-based validation (section 4.2.2). Direct validation allows one to investigate the peptide detection once we know the correct peptide for each spectrum; such circumstances are, however, atypical and usually much more idealized. Nevertheless, as we illustrated in our works [1, 2], data from such experiments are useful for the conceptual development of peptide detection methods. Sequencing-based validation, on the other hand, allows detecting peptides in samples of natural complexity and then independently validate them against DNA or mRNA sequencing data. In an adequately designed experiment (section 4.2.2.4), the probability of sequencing support of a detected variant peptide by chance is low. In turn, this allows one to interpret the presence of the DNA/mRNA sequencing support of a variant peptide as a sign of its correct detection, allowing to validate the behavior of peptide variant detection methods using an external criterion.

[Q_5] To what degree do peptide prior probabilities influence peptide detection?

As indicated by our former works [1–3] and multitude of results in the thesis (**Fig. 5.3**, **Fig. 5.4**, **Fig. 5.5**, **Fig. 5.6**, **Fig. 5.7**), peptide prior probabilities influence peptide detection to a substantial degree. Notably, we have also illustrated that just by using a different prior probability model, we can essentially turn a *de novo* sequencing into a reference-guided database search [2]. Similarly, we can probabilistically incorporate sequence tags to affect prior probabilities of peptides and then directly use our Bayesian model [2] in peptide detection. Several peptide detection strategies can be thus identified with particular prior models, and the framework of prior probabilities can be therefore also thought of as a generalization over various peptide detection approaches.

Conclusions

Herein, we conclude the main findings of our research. Our overall conclusion is as follows:

> The prior probabilities of peptides play a significant role in peptide detection, their utility is substantially underexplored in computational proteomics, and their integration into peptide detection largely improves its performance—especially when detecting unlikely peptides.

Let us briefly reiterate the reasons for this conclusion. First, in typical experiments, peptides result from complex biological events whose prevalence is highly variable. For instance, prior probabilities of the *most likely class of variant peptides* range at least over six orders of magnitude. Second, albeit powerful, mass spectrometry *has only limited ability* to discriminate between correct and incorrect peptides based purely on their match with the fragment spectrum. In consequence, the large variability of peptide prior probabilities plays a substantial role in peptide detection, evident especially when detecting *unlikely peptides*—such as variant peptides. Our approach provides evidence that the neglect of peptide prior probabilities is one of the reasons for the large rates of incorrect detections even at strict confidence criteria that affects detection of variant peptides [7–9, 104]. The computational proteomics community focused primarily on the second point—improving the capacity to discriminate peptides by predicting more accurate spectra [113–115] or by utilizing additional detection models [110–113]. Our research focused on the first point—by systematically modeling prior probabilities of peptides based on what is known about the analyzed sample in advance [1–3]. Importantly, both these approaches are *orthogonal*, and their integration is thus likely to offer substantial improvements in the field of computational proteomics in the future.

In our research, we developed mathematical and computational methods to utilize peptide prior probabilities in detection, allowing substantial improvements in detection performance (**Fig. 5.5**), and accurate estimation of posterior probabilities [1, 2]. Although we developed the methods primarily for detecting unlikely molecules, their general formulation allows further potential applications once suitably translated to the problem domain of interest. Therefore, besides the direct utility of the methods in computational proteomics and computational mass spectrometry, the methods are likely to have general value for the detection of unlikely causes (section 3.1).

Finally, we have shown that our methods have downstream applications in multiple fields, including cancer research, research reproducibility, and forensics, while describing further such applications in our patent application [4]. On the one hand, the successful application of these methods provides evidence of their correct implementation and affirms that our more realistic model of prior probabilities is already reasonably accurate. On the other hand, the actual find-

ings from such investigations are also of substantial practical value. For instance, the recognition of mislabeled and contaminated cell lines in public NCI_{60} datasets prevents researchers from inferring invalid conclusions once the fact that the corresponding samples are mislabeled is discovered. Similarly, the discrepancy between the observed DNA and protein mutation rates in tumor samples (**Fig. 5.8d**) allows investigating whether either rate is a better indicator of the suitability of cancer treatment using immunotherapy.

Altogether, we believe that we have provided compelling evidence for the importance of peptide prior probabilities in peptide detection and that our computational methods will find numerous direct and downstream applications in computational proteomics.

Zhrnutie v slovenskom jazyku

V nasledujúcich odstavcoch zhrnieme najpodstatnejšie závery nášho výskumu. Náš hlavný záver je nasledovný:

> A priori pravdepodobnosti peptidov zohrávajú zásadnú rolu v detekcii peptidov, ich využitie je nedostatočne preskúmané vo výpočtovej proteomike a ich integrácia do detekcie výrazne zlepšuje jej efektivitu—špeciálne v prípade detekcie nepravdepodobných peptidov.

Pripomeňme si v krátkosti dôvody uvedeného záveru. Za prvé, v typických proteomických experimentoch vznikajú peptidy z komplexných biologických udalostí, ktorých prevalencia je vysoko variabilná. Ako príklad, a priori pravdepodobnosti *najpravdepodobnejšej triedy variantných peptidov* majú rozsah minimálne šesť rádov. Za druhé, aj keď je hmotnostná spektrometria vysokoúčinná analytická metóda, má iba *limitovanú schopnosť* rozlíšiť medzi korektnými a nekorektnými peptidmi len na základe ich zhody s fragmentačným spektrom. Dôsledkom je, že vysoká variabilita a priori pravdepodobností peptidov zohráva zásadnú rolu v ich detekci a najvýraznejšie sa prejavuje pri detekcii *nepravdepodobných peptidov*—ako napríklad variantných peptidov. Náš výskum podáva evidenciu, že zanedbanie a priori pravdepodobnosti je jednou z príčin vysokej miery nesprávnych detekcií, ktorá postihuje detekciu variantných peptidov [7–9, 104]. Komunita výpočtovej proteomiky sa sústredila primárne na druhý bod—zvyšovanie kapacity rozlišovania peptidov pomocou predikcie viac presných fragmentačných spektier [113–115], alebo za použitia doplňujúcich detekčných modelov [110–113]. Náš výskum sa sústredil na prvý bod—na systematické modelovanie a priori pravdepodobností peptidov na základe toho, čo vieme o analyzovanej vzorke povedať pred samotnou analýzou pomocou hmotnostnej spektrometrie. Dôležité je, že oba prístupy sú na sebe nezávislé, a teda je vysoká šanca, že ich integrácia sa prenesie do zásadných vylepšení vo výpočtovej proteomike v budúcnosti.

V našom výskume sme vyvinuli matematické a algoritmické metódy, ktoré využívajú a priori pravdepodobnosti peptidov, poukazujúc na zásadne zlepšenie výkonnosti detekcie (**Fig. 5.5**), a na korektné odhady posteriórnych pravdepodobností za mnohých okolností [1, 2]. Aj keď sme uvedené metódy vyvinuli primárne pre detekciu nepravdepodobných molekúl, ich všeobecná formulácia dovoľuje ďalšie aplikácie za predpokladu, že sú vhodne adaptované do konkrétnej problémovej domény. Ako dôsledok, mimo priamej hodnoty našich metód vo výpočtovej proteomike a hmotnostnej spektrometrii, je vysoká šanca, že dané metódy sú celkovo užitočné pre detekciu nepravdepodobných príčin (sekcia 3.1).

V závere sme ukázali, že naše metódy majú využitie vo viacerých vedeckých oblastiach vrátane výskumu rakoviny, reprodukovateľnosti výskumu a forenznej vedy, pričom sme popísali ďalšie aplikácie v našej patentovej aplikácií. Na jednej strane, úspešné aplikovanie daných metód

podáva evidenciu o ich korektnej implementácii a potvrdzuje, že naše modely a priori pravdepodobností sú už v ich existujúcej forme dostatočne presné. Na druhej strane, samotné výsledky z daných štúdií majú významnú praktickú hodnotu. Ako príklad, rozpoznanie nesprávne označených a kontaminovaných vzoriek vo verejných NCI_{60} dátových zdrojoch zabraňuje vedcom vyvodiť neplatné závery v momente odhalenia faktu, že dané dáta boli vytvorené z iných než uvedených vzoriek. Podobne, nesúlad medzi mierou mutácií na úrovní DNA a proteínov v nádorových vzorkách (**Fig. 5.8d**) umožňuje študovať, ktorá miera je lepším indikátorom vhodnosti k liečbe rakoviny pomocou imunoterapie.

Veríme teda, že sa nám podarilo podať presvedčivú evidenciu o dôležitosti a priori pravdepodobností v detekcií peptidov a zároveň, že naše metódy nájdu početné priame a sprostredkované aplikácie vo výpočtovej proteomike a ďalších vedných oblastiach.

Bibliography

1. Hruska, M. & Holub, D. A complete search of combinatorial peptide library greatly benefited from probabilistic incorporation of prior knowledge. *International Journal of Mass Spectrometry* **471,** 116723. ISSN: 13873806 (Jan. 2022).
2. Hruska, M. & Holub, D. Evaluation of an integrative Bayesian peptide detection approach on a combinatorial peptide library. *European Journal of Mass Spectrometry,* 146906672110667. ISSN: 1469-0667 (Jan. 2022).
3. Hruska, M. *et al.* Deep probabilistic search detects protein variants in shotgun proteomics data independently of DNA/mRNA sequencing. *eLife* (Submitted).
4. Hruska, M., Hajduch, M. & Dzubak, P. *Method of identification of entities from mass spectra.* European Patent Application (EP 18184710.4), 2018.
5. Nesvizhskii, A. I. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics* **73,** 2092–2123. ISSN: 18743919 (2010).
6. Zhang, B. *et al.* Clinical potential of mass spectrometry-based proteogenomics. *Nature Reviews Clinical Oncology* **16,** 256–268. ISSN: 1759-4774 (Apr. 2019).
7. Sheynkman, G. M., Shortreed, M. R., Frey, B. L., Scalf, M. & Smith, L. M. Large-scale mass spectrometric detection of variant peptides resulting from nonsynonymous nucleotide differences. *Journal of Proteome Research* **13,** 228–240. ISSN: 15353893 (2014).
8. Cesnik, A. J., Shortreed, M. R., Sheynkman, G. M., Frey, B. L. & Smith, L. M. Human Proteomic Variation Revealed by Combining RNA-Seq Proteogenomics and Global Post-Translational Modification (G-PTM) Search Strategy. *Journal of Proteome Research* **15,** 800–808. ISSN: 15353907 (2016).
9. Nesvizhskii, A. I. Proteogenomics: concepts, applications and computational strategies. *Nature Methods* **11,** 1114–1125. ISSN: 1548-7091 (2014).
10. Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D. & Nesvizhskii, A. I. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature Methods* **14,** 513–520. ISSN: 1548-7091 (2017).
11. Gholami, A. M. *et al.* Global Proteome Analysis of the NCI-60 Cell Line Panel. *Cell Reports* **4,** 609–620. ISSN: 22111247 (Aug. 2013).
12. Aebersold, R. & Mann, M. Mass-spectrometric exploration of proteome structure and function. *Nature* **537,** 347–355. ISSN: 0028-0836 (2016).

13. Wilhelm, M. *et al.* Mass-spectrometry-based draft of the human proteome. *Nature* **509,** 582–587. ISSN: 0028-0836 (2014).

14. Alberts, B. *et al. Molecular Biology of the Cell* (eds Wilson, J. & Hunt, T.) **12,** 739–739. ISBN: 9781315735368. doi:10.1201/9781315735368 (W.W. Norton & Company, Aug. 2017).

15. Scheer, M. *et al.* BRENDA, the enzyme information system in 2011. *Nucleic Acids Research* **39,** 670–676. ISSN: 03051048 (2011).

16. Saier, M. H. *et al.* The transporter classification database (TCDB): 2021 update. *Nucleic Acids Research* **49,** D461–D467. ISSN: 13624962 (2021).

17. Liu, X. R., Zhang, M. M. & Gross, M. L. Mass Spectrometry-based protein footprinting for higher-order structure analysis: Fundamentals and applications. *Chemical Reviews* **120,** 4355–4454. ISSN: 15206890 (2020).

18. Bateman, A. UniProt: A worldwide hub of protein knowledge. *Nucleic Acids Research* **47,** D506–D515. ISSN: 13624962 (2019).

19. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422,** 198–207. ISSN: 0028-0836 (Mar. 2003).

20. Awad, H., Khamis, M. M. & El-Aneed, A. Mass spectrometry, review of the basics: Ionization. *Applied Spectroscopy Reviews* **50,** 158–175. ISSN: 1520569X (2015).

21. Gross, J. H. *Mass Spectrometry* ISBN: 9783642107092. doi:10.1007/978-3-319-54398-7 (Springer International Publishing, Cham, 2017).

22. Campana, J. E. Time-of-Flight Mass Spectrometry: a Historical Overview. *Instrumentation Science & Technology* **16,** 1–14. ISSN: 1073-9149 (Jan. 1987).

23. Chandramouli, K. & Qian, P.-Y. Proteomics: Challenges, Techniques and Possibilities to Overcome Biological Sample Complexity. *Human Genomics and Proteomics* **1.** ISSN: 17574242. doi:10.4061/2009/239204 (2009).

24. Mamyrin, B. A. Time-of-flight mass spectrometry (concepts, achievements, and prospects). *International Journal of Mass Spectrometry* **206,** 251–266. ISSN: 13873806 (2001).

25. Nesvizhskii, A. I. in *Mass Spectrometry Data Analysis in Proteomics* 87–120 (Humana Press, New Jersey). doi:10.1385/1-59745-275-0:87.

26. Michalski, A., Cox, J. & Mann, M. More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS. *Journal of Proteome Research* **10,** 1785–1793. ISSN: 15353893 (2011).

27. Li, Y. F. & Radivojac, P. Computational approaches to protein inference in shotgun proteomics. *BMC Bioinformatics* **13.** ISSN: 14712105. doi:10.1186/1471-2105-13-S16-S4 (2012).

28. Käll, L., Storey, J. D., MacCoss, M. J. & Noble, W. S. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *Journal of Proteome Research* **7,** 29–34. ISSN: 15353893 (2008).

29. Eng, J. K., Mccormack, A. L. & Yates, J. R. An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *American society for Mass Spectrometry* **5,** 976–989. ISSN: 1044-0305 (1994).

30. Craig, R. & Beavis, R. C. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics* **20,** 1466–1467. ISSN: 1367-4803 (June 2004).

31. Kim, S. & Pevzner, P. A. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nature Communications* **5.** ISSN: 20411723. doi:10.1038/ncomms6277 (2014).

32. Lam, H. *et al.* Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics* **7,** 655–667. ISSN: 16159853 (2007).

33. Craig, R., Cortens, J. C., Fenyo, D. & Beavis, R. C. Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research* **5,** 1843–1849. ISSN: 15353893 (2006).

34. Muth, T. & Renard, B. Y. Evaluating de novo sequencing in proteomics: already an accurate alternative to database-driven peptide identification? *Briefings in Bioinformatics* **19,** 954–970. ISSN: 1467-5463 (Sept. 2018).

35. Muth, T., Hartkopf, F., Vaudel, M. & Renard, B. Y. A Potential Golden Age to Come—Current Tools, Recent Use Cases, and Future Avenues for De Novo Sequencing in Proteomics. *Proteomics* **18.** ISSN: 16159861. doi:10.1002/pmic.201700150 (2018).

36. Ma, B. Novor: Real-Time Peptide de Novo Sequencing Software. *Journal of the American Society for Mass Spectrometry* **26,** 1885–1894. ISSN: 18791123 (2015).

37. Yang, H. *et al.* Open-pNovo: De Novo Peptide Sequencing with Thousands of Protein Modifications. *Journal of Proteome Research* **16,** 645–654. ISSN: 15353907 (2017).

38. Tabb, D. L., Ze-Qiang, M., Martin, D. B., Ham, A. J. L. & Chambers, M. C. DirecTag: Accurate sequence tags from peptide MS/MS through statistical scoring. *Journal of Proteome Research* **7,** 3838–3846. ISSN: 15353893 (2008).

39. Tabb, D. L., Saraf, A. & Yates, J. R. GutenTag: High-Throughput Sequence Tagging via an Empirically Derived Fragmentation Model. *Analytical Chemistry* **75,** 6415–6421. ISSN: 00032700 (2003).

40. Sinitcyn, P., Rudolph, J. D. & Cox, J. Computational Methods for Understanding Mass Spectrometry–Based Shotgun Proteomics Data. *Annual Review of Biomedical Data Science* **1,** 207–234. ISSN: 2574-3414 (2018).

41. Verheggen, K. *et al.* Anatomy and evolution of database search engines-a central component of mass spectrometry based proteomic workflows. *Mass Spectrometry Reviews,* 1–15. ISSN: 02777037 (2017).

42. Keller, A., Nesvizhskii, A. I., Kolker, E. & Aebersold, R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry* **74,** 5383–5392. ISSN: 00032700 (2002).

43. Käll, L., Canterbury, J. D., Weston, J., Noble, W. S. & MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods* **4,** 923–925. ISSN: 15487091 (2007).

44. Sheynkman, G. M., Shortreed, M. R., Cesnik, A. J. & Smith, L. M. Proteogenomics: Integrating Next-Generation Sequencing and Mass Spectrometry to Characterize Human Proteomic Variation. *Annual Review of Analytical Chemistry* **9,** 521–545. ISSN: 19361335 (2016).

45. Zhang, B. *et al.* Proteogenomic characterization of human colon and rectal cancer. *Nature* **513,** 382–387. ISSN: 0028-0836 (Sept. 2014).

46. Park, H. *et al.* Compact variant-rich customized sequence database and a fast and sensitive database search for efficient proteogenomic analyses. *Proteomics* **14,** 2742–2749. ISSN: 16159861 (2014).

47. Wang, X. *et al.* Protein identification using customized protein sequence databases derived from RNA-seq data. *Journal of Proteome Research* **11,** 1009–1017. ISSN: 15353907 (2012).

48. Zhang, B. *et al.* Proteogenomic characterization of human colon and rectal cancer. *Nature* **513,** 382–387. ISSN: 0028-0836 (2014).

49. Zhang, M. *et al.* CanProVar 2.0: An Updated Database of Human Cancer Proteome Variation. *Journal of Proteome Research* **16,** 421–432. ISSN: 15353907 (2017).

50. Huang, P.-J. *et al.* CMPD: cancer mutant proteome database. *Nucleic Acids Research* **43,** D849–D855. ISSN: 1362-4962 (Jan. 2015).

51. Li, J. *et al.* A Bioinformatics Workflow for Variant Peptide Detection in Shotgun Proteomics. *Molecular & Cellular Proteomics* **10,** M110.006536. ISSN: 1535-9476 (May 2011).

52. Ahrné, E., Nikitin, F., Lisacek, F. & Müller, M. QuickMod: A tool for open modification spectrum library searches. *Journal of Proteome Research* **10,** 2913–2921. ISSN: 15353893 (2011).

53. Tabb, D. L., Saraf, A. & Yates, J. R. GutenTag: High-Throughput Sequence Tagging via an Empirically Derived Fragmentation Model. *Analytical Chemistry* **75,** 6415–6421. ISSN: 00032700 (2003).

54. Renard, B. Y. *et al.* Overcoming Species Boundaries in Peptide Identification with Bayesian Information Criterion-driven Error-tolerant Peptide Search (BICEPS). *Molecular & Cellular Proteomics* **11.** ISSN: 15359476. doi:10.1074/mcp.M111.014167 (July 2012).

55. Devabhaktuni, A. *et al.* TagGraph reveals vast protein modification landscapes from large tandem mass spectrometry datasets. *Nature Biotechnology* **37,** 469–479. ISSN: 15461696 (2019).

56. Tanner, S. *et al.* InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra. *Analytical Chemistry* **77,** 4626–4639. ISSN: 0003-2700 (July 2005).

57. Chick, J. M. *et al.* A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nature Biotechnology* **33,** 743–749. ISSN: 1087-0156 (July 2015).

58. Mordret, E. *et al.* Systematic Detection of Amino Acid Substitutions in Proteomes Reveals Mechanistic Basis of Ribosome Errors and Selection for Translation Fidelity. *Molecular Cell* **75,** 427–441.e5. ISSN: 10974164 (2019).

59. Elias, J. E. & Gygi, S. P. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods* **4,** 207–214. ISSN: 1548-7091 (Mar. 2007).

60. Käll, L., Storey, J. D., MacCoss, M. J. & Noble, W. S. Posterior error probabilities and false discovery rates: Two sides of the same coin. *Journal of Proteome Research* **7,** 40–44. ISSN: 15353893 (2008).

61. Lee, S., Park, H. & Kim, H. Comparison of false-discovery rates of various decoy databases. *Proteome Science* **19,** 1–7. ISSN: 14775956 (2021).

62. The, M., MacCoss, M. J., Noble, W. S. & Käll, L. Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0. *Journal of the American Society for Mass Spectrometry* **27,** 1719–1727. ISSN: 18791123 (2016).

63. Chalkley, R. J. When Target-Decoy False Discovery Rate Estimations Are Inaccurate and How to Spot Instances. *Journal of Proteome Research* **12,** 1062–1064. ISSN: 15353893 (2013).

64. Gupta, N., Bandeira, N., Keich, U. & Pevzner, P. A. Target-Decoy Approach and False Discovery Rate: When Things May Go Wrong. *Journal of The American Society for Mass Spectrometry* **22,** 1111–1120. ISSN: 1044-0305 (July 2011).

65. Kim, S., Gupta, N. & Pevzner, P. A. Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases. *Journal of Proteome Research* **7,** 3354–3363. ISSN: 15353893 (2008).

66. Ivanov, M. V., Lobas, A. A., Karpov, D. S., Moshkovskii, S. A. & Gorshkov, M. V. Comparison of False Discovery Rate Control Strategies for Variant Peptide Identifications in Shotgun Proteogenomics. *Journal of Proteome Research* **16,** 1936–1943. ISSN: 1535-3893 (May 2017).

67. Keller, A., Eng, J., Zhang, N., jun Li, X. & Aebersold, R. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Molecular systems biology* **1.** ISSN: 17444292. doi:10.1038/msb4100024 (2005).

68. Shteynberg, D. *et al.* iProphet: Multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Molecular and Cellular Proteomics* **10,** 1–16. ISSN: 15359484 (2011).

69. Alves, G., Ogurtsov, A. Y. & Yu, Y. K. RAId_aPS: MS/MS analysis with multiple scoring functions and spectrum-specific statistics. *PLoS ONE* **5.** ISSN: 19326203. doi:10.1371/journal.pone.0015438. arXiv: 0806.2685 (2010).

70. Alves, G. *et al.* Calibrating E-values for MS2 database search methods. *Biology Direct* **2,** 1–14. ISSN: 17456150 (2007).

71. Geer, L. Y. *et al.* Open mass spectrometry search algorithm. *Journal of Proteome Research* **3,** 958–964. ISSN: 15353893 (2004).

72. Zhang, N., Aebersold, R. & Schwikowski, B. ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics* **2,** 1406–1412. ISSN: 16159853 (2002).

73. Shilov, I. V. *et al.* The Paragon Algorithm, a Next Generation Search Engine That Uses Sequence Temperature Values and Feature Probabilities to Identify Peptides from Tandem Mass Spectra. *Molecular & Cellular Proteomics* **6,** 1638–1655. ISSN: 1535-9476 (Sept. 2007).

74. Sherry, S. T. *et al.* dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research* **29,** 308–311. ISSN: 1362-4962 (Oct. 2000).

75. Lek, M. *et al.* Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536,** 285–291. ISSN: 14764687 (2016).

76. Chong, K. F. & Leong, H. W. *Tutorial on de novo peptide sequencing using MS/MS mass spectrometry* **6,** 1–38. ISBN: 0219720012310. doi:10.1142/S0219720012310026 (2012).

77. Creasy, D. M. & Cottrell, J. S. Unimod: Protein modifications for mass spectrometry. *Proteomics* **4,** 1534–1536. ISSN: 16159853 (2004).

78. Bryant, K. L., Mancias, J. D., Kimmelman, A. C. & Der, C. J. KRAS: Feeding pancreatic cancer proliferation. *Trends in Biochemical Sciences* **39,** 91–100. ISSN: 09680004 (2014).

79. Perez-Riverol, Y. *et al.* The PRIDE database and related tools and resources in 2019: Improving support for quantification data. *Nucleic Acids Research* **47,** D442–D450. ISSN: 13624962 (2019).

80. Reinhold, W. C. *et al.* CellMiner: A web-based suite of genomic and pharmacologic tools to explore transcript and drug patterns in the NCI-60 cell line set. *Cancer Research* **72,** 3499–3511. ISSN: 00085472 (2012).

81. Reinhold, W. C. *et al.* RNA Sequencing of the NCI-60: Integration into CellMiner and CellMiner CDB. *Cancer Research* **79,** 3514–3524. ISSN: 0008-5472 (July 2019).

82. Varma, S., Pommier, Y., Sunshine, M., Weinstein, J. N. & Reinhold, W. C. High resolution copy number variation data in the NCI-60 cancer cell lines from whole genome microarrays accessible through CellMiner. *PLoS ONE* **9.** ISSN: 19326203. doi:10.1371/journal.pone.0092047 (2014).

83. Bourdais, R. *et al.* Polymerase proofreading domain mutations: New opportunities for immunotherapy in hypermutated colorectal cancer beyond MMR deficiency. *Critical Reviews in Oncology/Hematology* **113,** 242–248. ISSN: 10408428 (May 2017).

84. Yuza, K., Nagahashi, M., Watanabe, S., Takabe, K. & Wakai, T. Hypermutation and microsatellite instability in gastrointestinal cancers. *Oncotarget* **8,** 112103–112115. ISSN: 19492553 (2017).

85. Edwards, N. J. *et al.* The CPTAC Data Portal: A Resource for Cancer Proteomics Research. *Journal of Proteome Research* **14,** 2707–2713. ISSN: 1535-3893 (June 2015).

86. Lappalainen, I. *et al.* The European Genome-phenome Archive of human data consented for biomedical research. *Nature Genetics* **47,** 692–695. ISSN: 15461718 (2015).

87. Allan Drummond, D. & Wilke, C. O. The evolutionary consequences of erroneous protein synthesis. *Nature Reviews Genetics* **10,** 715–724. ISSN: 14710056 (2009).

88. Hortin, G. L. & Sviridov, D. The dynamic range problem in the analysis of the plasma proteome. *Journal of Proteomics* **73,** 629–636. ISSN: 18743919 (2010).

89. Eisenberg, E. & Levanon, E. Y. A-to-I RNA editing - Immune protector and transcriptome diversifier. *Nature Reviews Genetics* **19,** 473–490. ISSN: 14710064 (2018).

90. Picardi, E., D'Erchia, A. M., Giudice, C. L. & Pesole, G. REDIportal: A comprehensive database of A-to-I RNA editing events in humans. *Nucleic Acids Research* **45,** D750–D757. ISSN: 13624962 (2017).

91. Wilson, D. J. Erratum: The harmonic mean p-value for combining dependent tests (Proceedings of the National Academy of Sciences of the United States of America (2019) 166 (1195-1200) DOI: 10.1073/pnas.1814092116). *Proceedings of the National Academy of Sciences of the United States of America* **116,** 21948. ISSN: 10916490 (2019).

92. Zhao, P., Li, L., Jiang, X. & Li, Q. Mismatch repair deficiency/microsatellite instability-high as a predictor for anti-PD-1/PD-L1 immunotherapy efficacy. *Journal of Hematology & Oncology* **12,** 54. ISSN: 1756-8722 (Dec. 2019).

93. Huang, T., Wang, J., Yu, W. & He, Z. Protein inference: A review. *Briefings in Bioinformatics* **13,** 586–614. ISSN: 14774054 (2012).

94. Frank, A. & Pevzner, P. PepNovo: De novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry* **77,** 964–973. ISSN: 00032700 (2005).

95. Kessner, D., Chambers, M., Burke, R., Agus, D. & Mallick, P. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* **24,** 2534–2536. ISSN: 13674803 (2008).

96. Liu, X., Jian, X. & Boerwinkle, E. dbNSFP: A lightweight database of human nonsynonymous SNPs and their functional predictions. *Human Mutation* **32,** 894–899. ISSN: 10597794 (Aug. 2011).

97. Arnold, M. *et al.* Global patterns and trends in colorectal cancer incidence and mortality. *Gut* **66,** 683–691. ISSN: 0017-5749 (Apr. 2017).

98. Baudrin, L. G., Deleuze, J.-F. & How-Kit, A. Molecular and Computational Methods for the Detection of Microsatellite Instability in Cancer. *Frontiers in Oncology* **8,** 1–11. ISSN: 2234-943X (Dec. 2018).

99. Masters, J. R. W. Cell line misidentification: the beginning of the end. *Nature Reviews Cancer* **10,** 441–448. ISSN: 1474-175X (June 2010).

100. Freedman, L. P. *et al.* Reproducibility: changing the policies and culture of cell line authentication. *Nature Methods* **12,** 493–497. ISSN: 1548-7091 (2015).

101. Leinonen, R., Sugawara, H. & Shumway, M. The sequence read archive. *Nucleic Acids Research* **39,** 2010–2012. ISSN: 03051048 (2011).

102. Deutsch, E. W. *et al.* The ProteomeXchange consortium in 2020: Enabling 'big data' approaches in proteomics. *Nucleic Acids Research* **48,** D1145–D1152. ISSN: 13624962 (2020).

103. Reid, Y., Storts, D., Riss, T. & Minor, L. Authentication of Human Cell Lines by STR DNA Profiling Analysis. *Assay Guidance Manual,* 435–452 (2004).

104. Zhang, F. *et al.* DeepFunc: A Deep Learning Framework for Accurate Prediction of Protein Functions from Protein Sequences and Interactions. *Proteomics* **19.** ISSN: 16159861. doi:10.1002/pmic.201900019 (2019).

105. Wang, M. *et al.* PaxDb, a Database of Protein Abundance Averages Across All Three Domains of Life. *Molecular & Cellular Proteomics* **11,** 492–500. ISSN: 15359476 (Aug. 2012).

106. Yang, J. *et al.* DeepDigest: Prediction of Protein Proteolytic Digestion with Deep Learning. *Analytical Chemistry* **93,** 6094–6103. ISSN: 15206882 (2021).

107. Serrano, G., Guruceaga, E. & Segura, V. DeepMSPeptide: Peptide detectability prediction using deep learning. *Bioinformatics* **36,** 1279–1280. ISSN: 14602059 (2020).

108. Mordret, E. *et al.* Systematic Detection of Amino Acid Substitutions in Proteomes Reveals Mechanistic Basis of Ribosome Errors and Selection for Translation Fidelity. *Molecular Cell* **75,** 427–441.e5. ISSN: 10974164 (2019).

109. Alexandrov, L. B. *et al.* The repertoire of mutational signatures in human cancer. *Nature* **578,** 94–101. ISSN: 14764687 (2020).

110. Klammer, A. A., Yi, X., MacCoss, M. J. & Noble, W. S. Improving tandem mass spectrum identification using peptide retention time prediction across diverse chromatography conditions. *Analytical Chemistry* **79,** 6111–6118. ISSN: 00032700 (2007).

111. Searle, B. C. *et al.* Chromatogram libraries improve peptide detection and quantification by data independent acquisition mass spectrometry. *Nature Communications* **9.** ISSN: 20411723. doi:10.1038/s41467-018-07454-w (2018).

112. Ivanov, M. V. *et al.* DirectMS1: MS/MS-Free Identification of 1000 Proteins of Cellular Proteomes in 5 Minutes. *Analytical Chemistry* **92,** 4326–4333. ISSN: 15206882 (2020).

113. Gessulat, S. *et al.* Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods* **16,** 509–518. ISSN: 15487105 (2019).

114. Zeng, W. F. *et al.* MS/MS Spectrum prediction for modified peptides using pDeep2 Trained by Transfer Learning. *Analytical Chemistry* **91,** 9724–9731. ISSN: 15206882 (2019).

115. Liu, K., Li, S., Wang, L., Ye, Y. & Tang, H. Full-Spectrum Prediction of Peptides Tandem Mass Spectra using Deep Neural Network. *Analytical Chemistry* **92,** 4275–4283. ISSN: 15206882 (2020).

116. Olarerin-George, A. O. & Hogenesch, J. B. Assessing the prevalence of mycoplasma contamination in cell culture via a survey of NCBI's RNA-seq Archive. *Nucleic Acids Research* **43,** 2535–2542. ISSN: 13624962 (2015).

117. Blakeley, P., Overton, I. M. & Hubbard, S. J. Addressing statistical biases in nucleotide-derived protein databases for proteogenomic search strategies. *Journal of Proteome Research* **11,** 5221–5234. ISSN: 15353893 (2012).

118. Tabb, D. L., Huang, Y., Wysocki, V. H. & Yates, J. R. Influence of Basic Residue Content on Fragment Ion Peak Intensities in Low-Energy Collision-Induced Dissociation Spectra of Peptides. *Analytical Chemistry* **76,** 1243–1248. ISSN: 00032700 (2004).