Identification of variant peptides using mass spectrometry

Author Paper of Dissertation thesis

Miroslav Hruška

Department of Computer Science
Faculty of Science
Palacký University Olomouc

Olomouc, 2022

Kandidát
Ing. Miroslav Hruška
hruska.miro@gmail.com

Školiteľ
doc. Ing. Petr Sosík, Dr.

Oponenti

Miesto a termín obhajoby

S dizertačnou prácou a s posudkami sa bude možné oboznámiť na katedre informatiky, PřF UP, 17. listopadu 12, Olomouc.

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.

Keywords: peptide detection, prior probability, mass spectrometry, variant peptides

Preface

The paper 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 paper 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. Finally, several results from the article are presented in the section 4.1.

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 4.1.

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, and 3.2. Finally, several results of the work are presented in section 4.3.

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.

The paper is organized as follows. First, in section 1, we introduce the research problem and specify our research aims. Afterward, in section 2, we review the literature relevant to peptide detection, including how researchers utilized peptide prior probabilities. In section 3, we develop a theoretical framework for probabilistic analysis of causes given their agreement with the data and translate the approach into computational proteomics. Finally, in section 4, we show the application of the developed methods to detection of peptides, and present several downstream applications of detected variant peptides.

1 Introduction

The paper 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

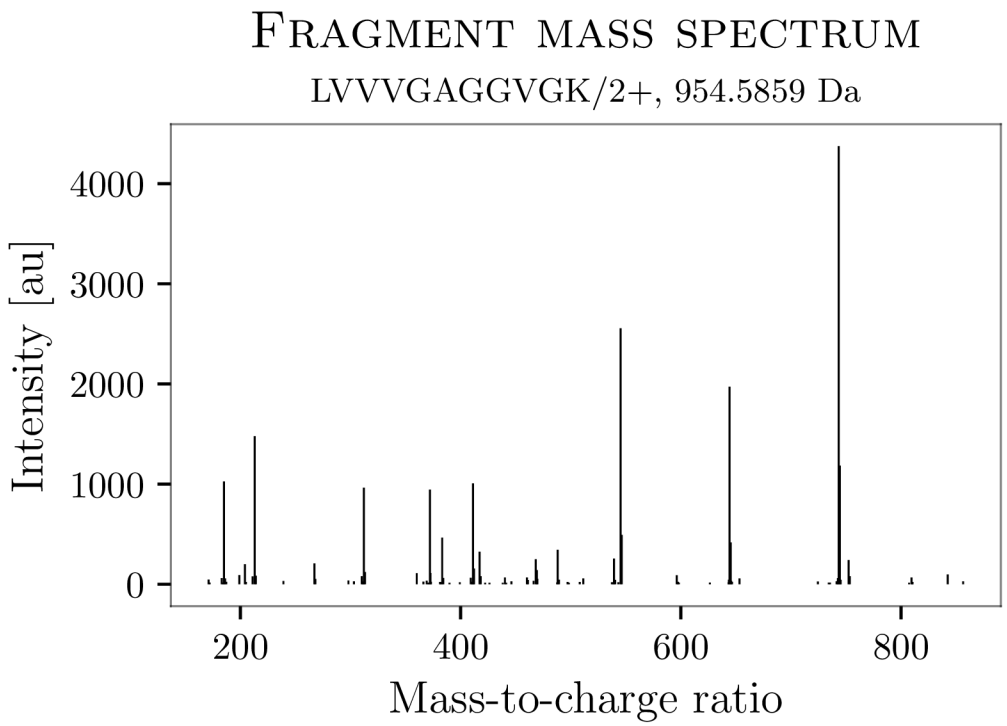


Figure 1.1: An example of a mass spectrum.

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

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 paper 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.

1.1 Research problem

Let us now introduce the research problem that we aim to address in more detail. 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. 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].

2 Literature review

The detection of peptides from fragment spectra lies at the core of computational proteomics [5, 10]. 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 [11–13], 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 [14, 15], 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 [16, 17]. Even though *de novo* sequencing is fast [18], and allows large-scale modification search [19], it achieves only around 35 % agreement with a database search, making it impractical for routine analyses [16]. Hybrid approaches typically utilize partial *de novo* sequencing to extract *sequence tags* [20, 21], 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 [22], with more than 30 search engines available in 2017 [23]. 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 [11–13]. 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 [24, 25], 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].

2.1 Detection of variant peptides

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, 26], 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 [27–29]. 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, 30].

To allow DNA/mRNA-independent detection, one can perform the database searches against a peptide database constructed from a globally observed DNA or mRNA variants [3, 31, 32], 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 [33], 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 paper.

Another option for detecting variant peptides is to use some of the database-guided and hybrid detection methods [11, 12, 34, 35]. The most straightforward possibility is to use the exhaustive substitution of amino acids per peptide [11, 12]. For instance, the *point mutation search* in X!Tandem [11] or the *error-tolerant search* in MASCOT [12] 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 [36] or TagGraph [37], 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 [38]. 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 [39] implemented in the fast MSFragger algorithm [40], utilizes a standard database search against very wide precursor mass window (e.g., 500 Da instead of typical range on the

order of ≈ 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 [41], or open searches against spectral libraries [34]. 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 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 [16]. 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 [42] 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 [43] 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 [36] 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 [44] and ExAC [45] 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.

3 Theoretical framework

The section deals with the theoretical core of the paper 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.

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:

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

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.

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).

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

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

Note that the set $\mathbb{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$ is especially relevant in our data 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. Due to space limitations, we refer the reader to the prediction of theoretical spectra in [46], and will only consider matching of existing fragment spectra. As an example of a theoretical spectrum, see **Fig. 3.1**.

For simplicity, we will calculate the *the number of matching fragments* (NMF) between the spectra. 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,

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

Then, 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

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

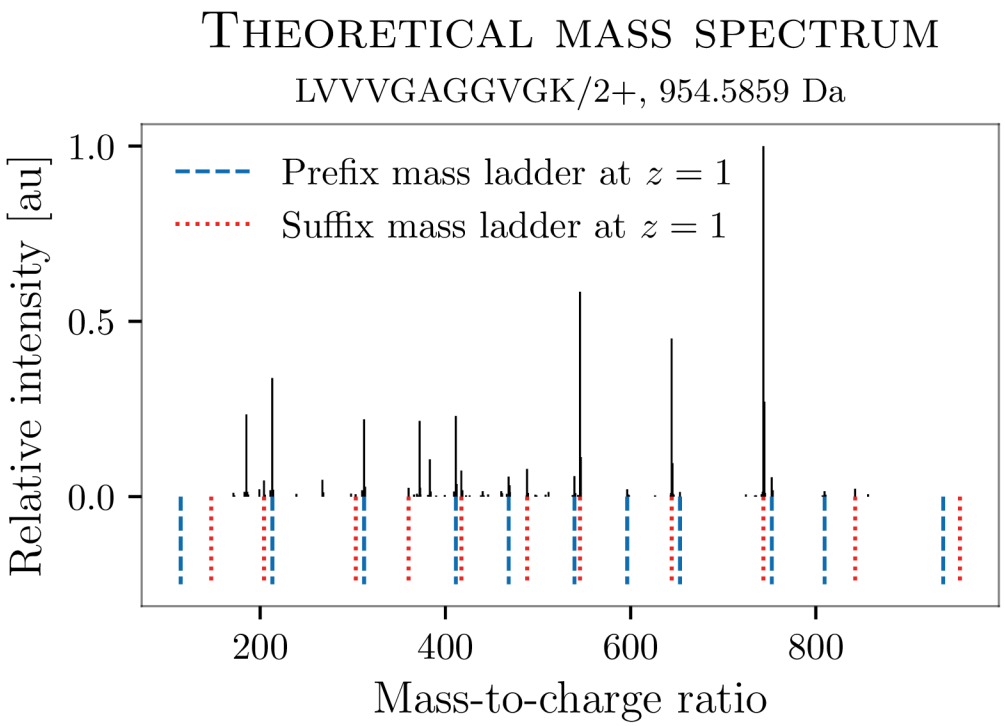


Figure 3.1: Correspondence between experimental and theoretical mass spectra.

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

3.2.3 Simple peptide prior probability models

The section introduces several 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). Although the prior models are simple, they 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 6 (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 7 (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 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.

3.2.3.3 Other prior models

For other simple prior models, we refer the reader to the dissertation thesis.

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 8 (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 9 (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 ($\texttt{M} \oplus \texttt{Oxidation}$). For simplicity, suppose we would expect to see both forms in equal proportions. Then, $\mathcal{M}(\texttt{M}) = \{\texttt{M}, \texttt{M} \oplus \texttt{Oxidation}\}$, and $\mathcal{M}_{\texttt{M}}(\texttt{M}) = 0.5 = \mathcal{M}_{\texttt{M}}(\texttt{M} \oplus \texttt{Oxidation})$.

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 $r = \texttt{M}$. 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 $\mathcal{S}_{\texttt{M}}(\texttt{I}) = 10^{-4}$.

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 1 (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 2 (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 Parental sequence 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

Herein, we specify a particular model of cutting-after-residue. For the rationale and assumptions underlying the selection of the model, we refer the reader to the dissertation thesis.

In this model, the relative prior probability of a cut $f \overset{s}{\leftrightarrow} t$ from a sequence s, is then

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

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 $\Pr^{\star}(s \underset{i}{\dashrightarrow} p)$.

Definition 10 (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 $\Pr^{\star}(s \underset{i}{\dashrightarrow} p)$, is defined as follows:

$$\begin{aligned}\Pr^{\star}(s \underset{i}{\dashrightarrow} p) =& \Pr^{\star}_{\alpha}(s \dashrightarrow i \overset{s}{\leftrightarrow} i + n - 1) \\ & \cdot \Pr(\langle s_i, \ldots, s_{i+n-1} \rangle \rightarrow \langle p_1, \ldots, p_n \rangle) \\ & \cdot \mathcal{M}_{\vdash}(p_{\vdash}) \\ & \cdot \mathcal{M}_{\dashv}(p_{\dashv}).\end{aligned}$$

Clarification In other words, the $\Pr^{\star}(s \underset{i}{\dashrightarrow} p)$ 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 3 (Maximal expected proportion of a sequence form p originating from a sequence s)**.**

$$\Pr^{\star}_{\max}(s \dashrightarrow p) = \max_{1 \leq i \leq m-n+1} \Pr^{\star}(s \underset{i}{\dashrightarrow} p).$$

Proof. The only indices over which $\Pr^{\star}(s \underset{i}{\dashrightarrow} p)$ 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

$$\Pr^{\star}(p) = \Pr^{\star}_{\max}(S \dashrightarrow p). \tag{3.5}$$

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), and then illustrate its behavior for simplified parameters of the prior model (section 3.2.5.2).

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.

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

```
/* 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

/* 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
        Q ← ⟨⟩
        /* For each starting position excluding N- and C-termini */
        foreach i ∈ ⟨1, ..., k⟩ do
            /* Cleavage required before the previous residue */
            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)

```
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 peptide is extended
         Expected proportion p_cleaved if the peptide is cleaved
         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 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 or m ≥ m_max then
   |  return
   end

   /* [INCORPORATION OF A NEW RESIDUE] */
   e ← s_i /* Store the original residue */
   foreach r ∈ F(e) do /* For each form of e */
   |  r_p ← F_e(r) /* Obtain the expected proportion */
   |  if i < k − 1 then /* If still not at the C-term */
   |  |  /* Expected proportion if cleaved after the residue */
   |  |  p_cleaved* ← p_extended · r_p · α(r)
   |  |  /* Expected proportion if not cleaved after the 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
```

3.2.5.2 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,

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

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 [40]. 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 11 (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.

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. 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 $\epsilon > 0$.

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
```

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 $\textsc{nmf}_\epsilon(T_a, E)$ 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 $\textsc{nmf}_\epsilon(T_a, E)$ 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* $M_a = \textsc{nmf}_\epsilon(T_a, E)$*.*

Proof. We prove the theorem for a particular a so that $M_a = \textsc{nmf}_\epsilon(T_a, E)$. 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

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
```

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 $\Omega(n)$.

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 partition such a dataset $\mathbf{Q}$ into mass-binned datasets $\mathbf{Q}_b$ containing only peptides whose masses overlap with $M_b = \langle b \cdot w, (b+1) \cdot w)$, 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 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. 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.

Definition 12 (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, herein referring to such an algorithm as Load-Fragment-Ion-Subindex (details in the dissertation). 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 (for a visual overview, see **Fig. 3.2**). Thus, suppose a fragment spectrum E, its measured precursor mass $\hat{m}_\mathrm{p}$, 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 $\textsc{Match-Against-Database}(E, \epsilon, \hat{m}_\mathrm{p}, \epsilon_\mathrm{p}, \mathbf{Q})$ 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 $\mathrm{Pr}^\star(p)$ 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). 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 $\mathbf{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$. As a result, we have all the necessary ingredients to calculate the $\mathrm{Pr}_{\max}$ of each peptide $p \in \mathbf{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$.

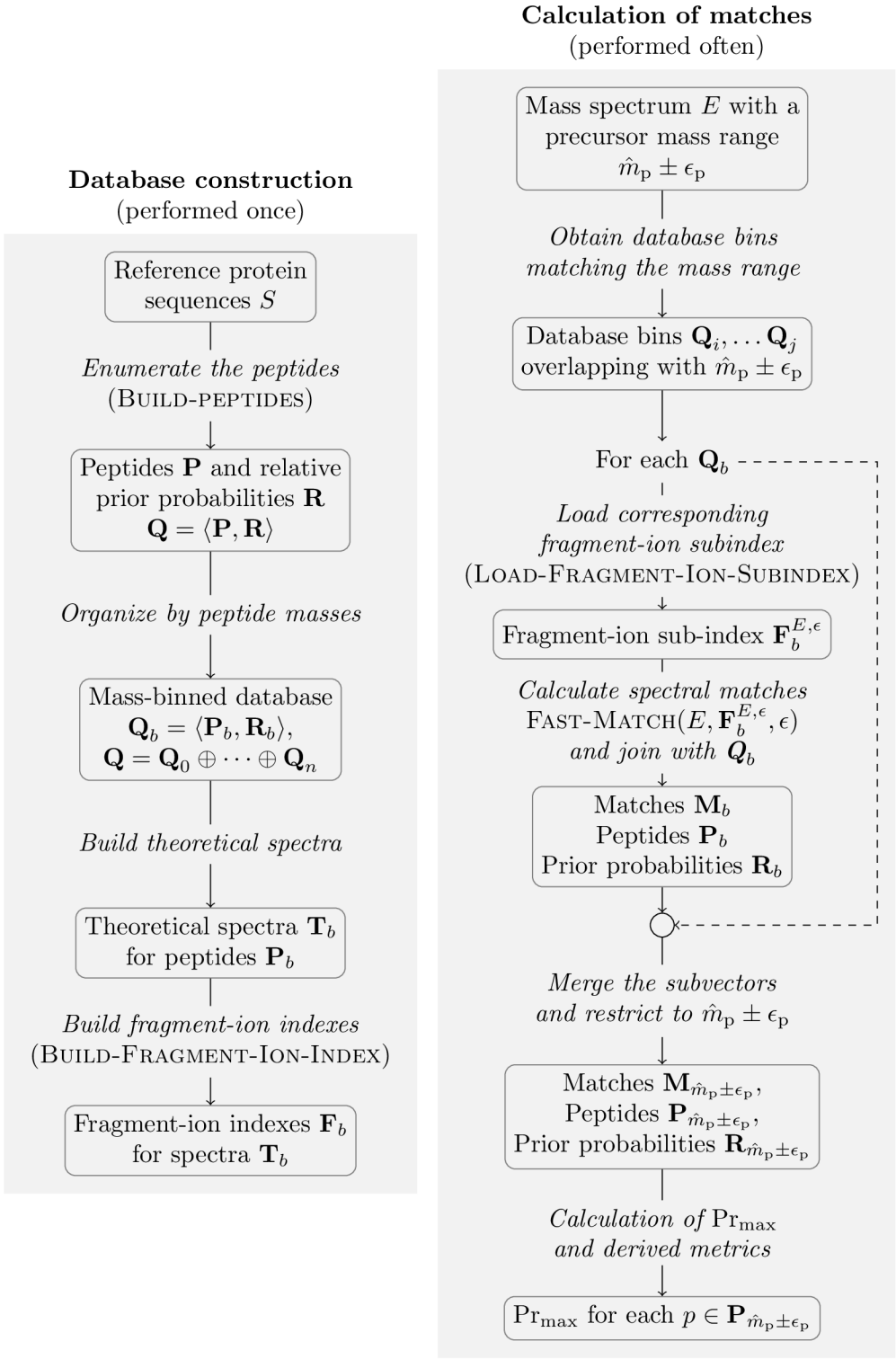


Figure 3.2: Data processing overview

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.

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

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

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, $\Pr_\mathrm{max}$ has limitations when multiple candidates are of similar fragment match and prior probabilities. To improve the situation, we introduce a relaxation of $\Pr_\mathrm{max}$, denoted $\tilde{\Pr}^k_\mathrm{max}$, which assigns a trade-off k between the importance of the spectral match and prior probabilities. The $\tilde{\Pr}^k_\mathrm{max}$ calculated for a given peptide and spectrum then gives a value in the $\langle 0, 1 \rangle$ interval, related to the posterior probability (details in the dissertation thesis).

Definition 13 ($\tilde{\Pr}^k_\mathrm{max}$)**.** 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 $\Pr_\mathrm{max}$ of a peptide $p \in \mathbf{P}_{\hat{m}_\mathrm{p} \pm \epsilon_\mathrm{p}}$ at trade-off k, denoted $\tilde{\Pr}^k_\mathrm{max}(p, m)$, is

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

4 Results

The section deals with applications of the methods presented in the paper. First, in section 4.1, 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 4.2, 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 four 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 4.3, we illustrate downstream applications of the developed methods in cancer research, research reproducibility, and forensics.

4.1 Peptide detection in the combinatorial peptide library

Herein, we evaluate the Bayesian cause-detection model from section 3.1.3 on a combinatorial peptide library [1] dataset while utilizing multiple simple models of peptide prior probabilities (section 3.2.3). First, we show that the numerical values of posterior probabilities tended towards their expected long-term behavior (section 4.1.1). Afterward, 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 4.1.2).

4.1.1 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 two extreme cases of prior distributions. For additional prior distributions, we refer the interested reader to our articles [1, 2] and to the dissertation thesis, which also contain detailed treatment of the dataset.

Herein, we consider 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

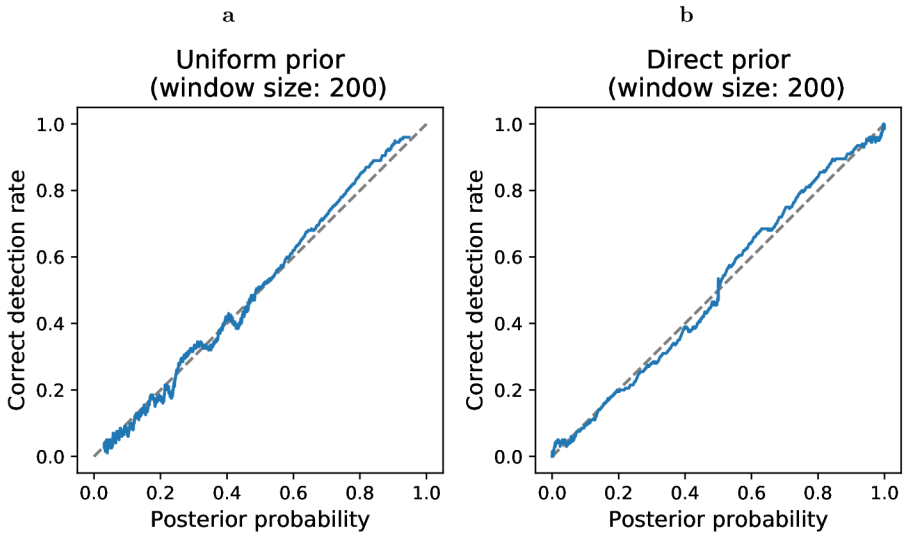


Figure 4.1: Behavior of posterior probabilities for uniform and direct prior models.

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.

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 $\mathbb{P}_L = \{\mathtt{LVVVGA}xy\mathtt{VGK} \,|\, x, y \in \mathbb{A}_\wedge\}$, and thus represents a near-completely informed prior model. Formally, the direct prior for a particular spectrum m thus behaves as follows:

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

As is evident from the figure **Fig. 4.1**, 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.

4.1.2 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.

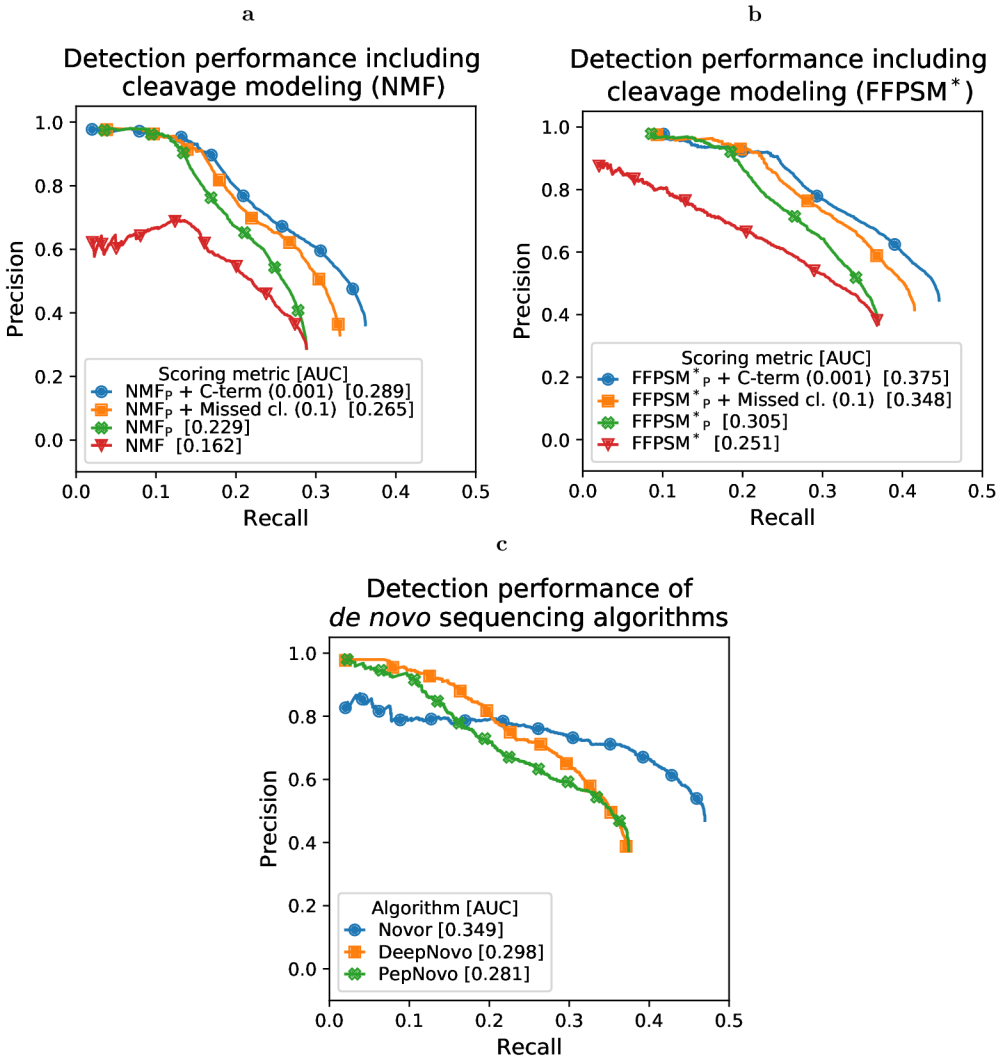


Figure 4.2: Comparison of peptide detection with state-of-the-art de novo sequencing algorithms

(**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**).

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. 4.2a** 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.

4.2 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 [47] 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 [48]). For a detailed treatment of the sequencing-based validation and configuration of the software, we refer the reader to the dissertation thesis.

Let us now provide a brief overview of the main results. In section 4.2.1, 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, we introduce CLAIRE—our system for detection of peptide variants (section 4.2.2).

Finally, in section 4.2.3, we show that CLAIRE 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.

4.2.1 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$_{\mathrm{ES}}$) [12], a Bayesian approach BICEPS for detecting variably-mutated sequences [36], an open-search approach MSFragger [40], and a global peptide-variant database search using X!Tandem (X!Tandem$_{\mathrm{GPV}}$). 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 DNA/mRNA variants.

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 3.2.7), 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 $\tilde{\mathrm{Pr}}^{k}_{\mathrm{max}}$. 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). Further, we also consider an adjustment of the prior probability of a variant peptide p by replacing the general probability of amino acid substitution with a sequence-specific one, based on population frequency of corresponding DNA variant (details in the dissertation). When we utilize such adjustment of peptide prior probabilities, we include the symbol † in the superscript (e.g., $\tilde{\mathrm{Pr}}^{k,\dagger}_{\mathrm{max}}$). As mass spectrometers sometimes incorrectly measure the parental mass of a molecule, we also consider measurements shifted by masses of up to ±2 neutrons. 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 (multiplication by 0.1 with each shift in either direction, see the dissertation for details). Altogether, this brings us to the metric $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ that utilizes all these extensions

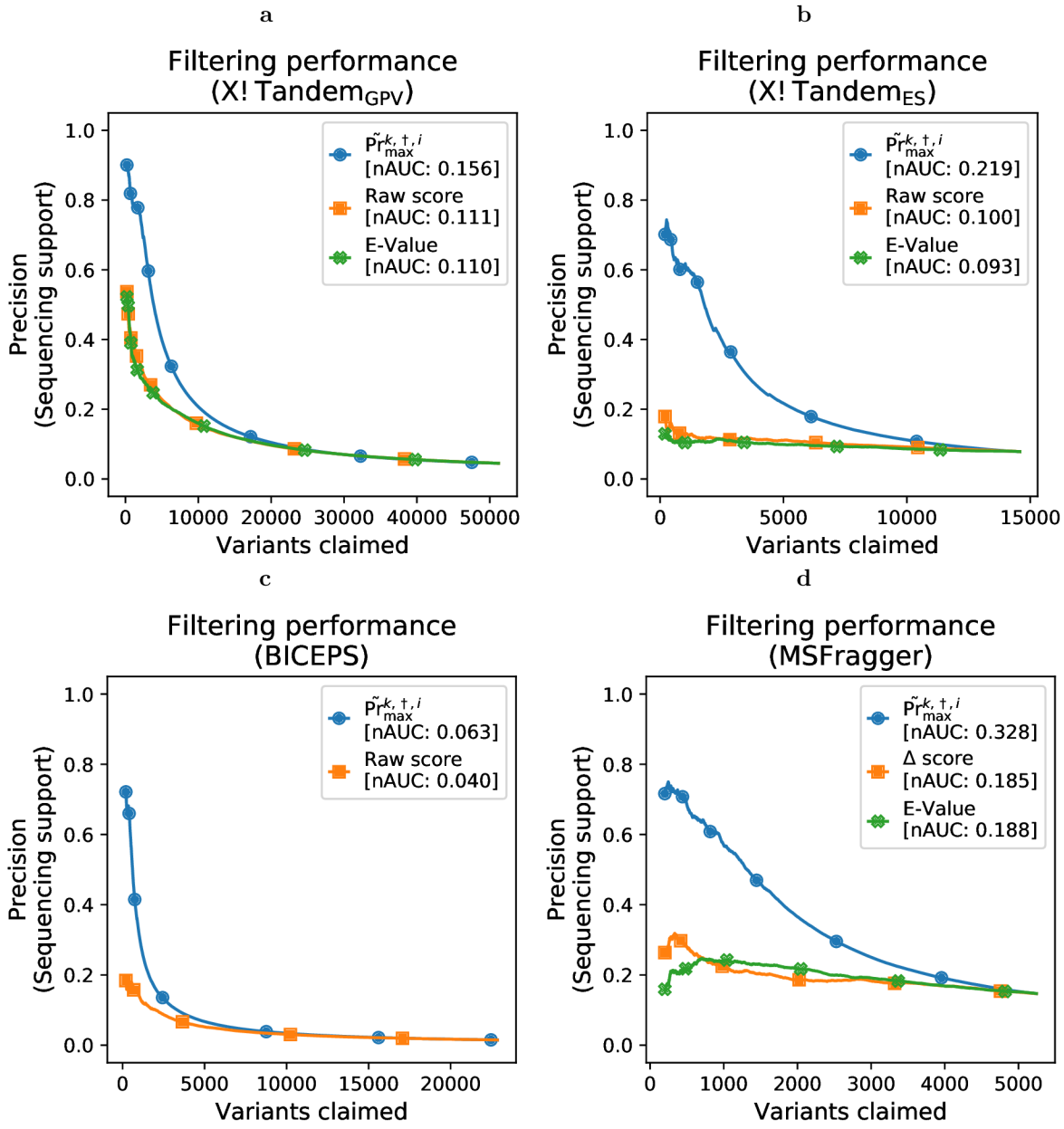


Figure 4.3: Filtering efficiency using native scores and the deep search score $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$.

(**a**–**d**) The plots show the post-search filtering efficiency of claimed variant peptides both by their native scores and using the $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ score derived from our probabilistic deep search method. In the analysis, all claimed variant peptides were subjected to the deep search, and the corresponding $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ of the claimed variant peptide was calculated. The behavior shows that individual approaches highly benefited from filtering using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\mathrm{max}}$ 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.

over $\mathrm{Pr}_{\max}$, 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, and visualized them on **Fig. 4.3**. Overall, the figures show the filtering of peptide variants using their native scores compared to the probabilistic deep search score $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$. As is evident from the figures, filtering results using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ 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. 4.3b**). On the other hand, filtering using $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ 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 $\tilde{\mathrm{Pr}}^{k,\dagger,i}_{\max}$ allowed substantially more sensitive detection of candidate variant peptides compared to the native scoring metrics.

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. 4.4**). 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. 4.4a** 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. 4.4b** 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$_{\mathrm{GPV}}$ 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	$\text{NMF}_\epsilon(p,m)$	$\text{Pr}^\star_\dagger(p)$	$\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$
$\rightarrow$	LGEHNI$^{\text{I}\rightarrow\text{V}}$EVLEGNEQFINAAK	20	5.68×10^{-5}	2.50×10^{-3}
$\odot$	LGEHNIE$^{\text{E}\rightarrow\text{D}}$VLEGNEQFINAAK	22	5.68×10^{-5}	0.9975
	LGEHNIEVL$^{\text{L}\rightarrow\text{V}}$EGNEQFINAAK	18	5.71×10^{-5}	6.27×10^{-6}
	L$^{\text{L}\rightarrow\text{V}}$GEHNIEVLEGNEQFINAAK	17	5.71×10^{-5}	3.13×10^{-7}
	LGE$^{\text{E}\rightarrow\text{D}}$HNIEVLEGNEQFINAAK	17	5.68×10^{-5}	3.12×10^{-7}
	LGEHN$^{\text{N}\rightarrow\text{T}}$IEVLEGNEQFINAAK	17	5.57×10^{-6}	3.06×10^{-8}
	LGEHNIEVLE$^{\text{E}\rightarrow\text{D}}$GNEQFINAAK	16	5.68×10^{-5}	1.56×10^{-8}
	LGEHNIEVLEGN$^{\text{N}\rightarrow\text{T}}$EQFINAAK	14	5.57×10^{-6}	3.82×10^{-12}
	QD$^{\text{D}\rightarrow\text{A}}GM^{\text{Ox}}$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	$\text{NMF}_\epsilon(p,m)$	$\text{Pr}^\star_\dagger(p)$	$\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$
$\rightarrow\odot$	SS$^{\text{S}\rightarrow\text{A}}$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$^{\text{S}\rightarrow\text{A}}$SLFAQINQGESITHALK	9	5.40×10^{-5}	2×10^{-4}
	SPFSLPQKSL$^{\text{L}\rightarrow\text{Q}}$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$^{\text{Lys}\rightarrow\text{AminoadipicAcid}}$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}
	$\vdash^{\text{Acetyl:2H(3)}}$PEFALALPPEPPPGPEVK	8	3.36×10^{-6}	6.21×10^{-7}
	AAEEAERQRQIQLAQK$^{\text{Carb}}$	9	1.60×10^{-7}	5.92×10^{-7}

Legend

$\odot$	The peptide with the highest $\tilde{\text{Pr}}^{k,\dagger,i}_{\max}$ in the deep search.
$\rightarrow$	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 4.4: 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 ($\rightarrow$) 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 ($\odot$) of similar prior probability, drawing the claimed variant peptide $\rightarrow$ unlikely. In **b**, the X!Tandem global peptide-variant search claimed variant peptide ($\rightarrow$) 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.

4.2.2 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 author paper and the dissertation 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.

4.2.2.1 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 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.

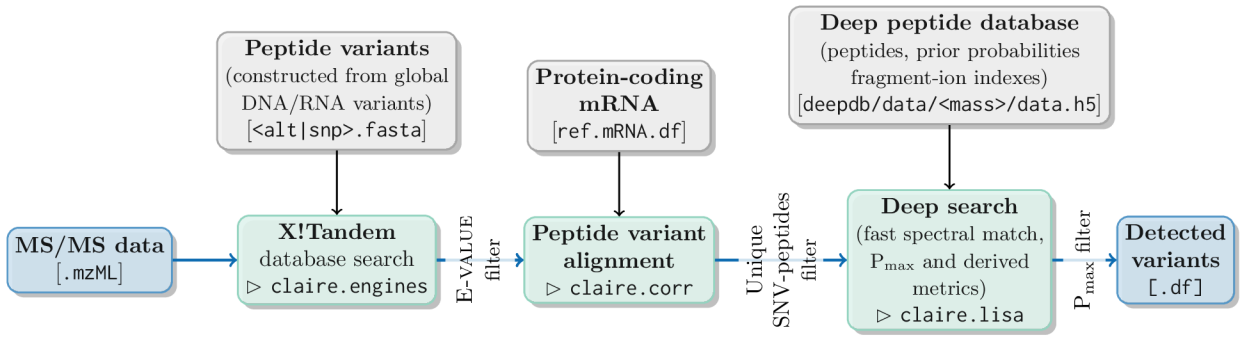


Figure 4.5: Flow of data in CLAIRE's detection of peptide variants

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 (X!Tandem$_{\text{GPV}}$, module `claire.engines`), and candidate peptides are filtered using the significance of their match (E-VALUE $\leq$ 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.

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. 4.5**). 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 [49] (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.

4.2.2.2 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. 4.6**)—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 [50], 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.

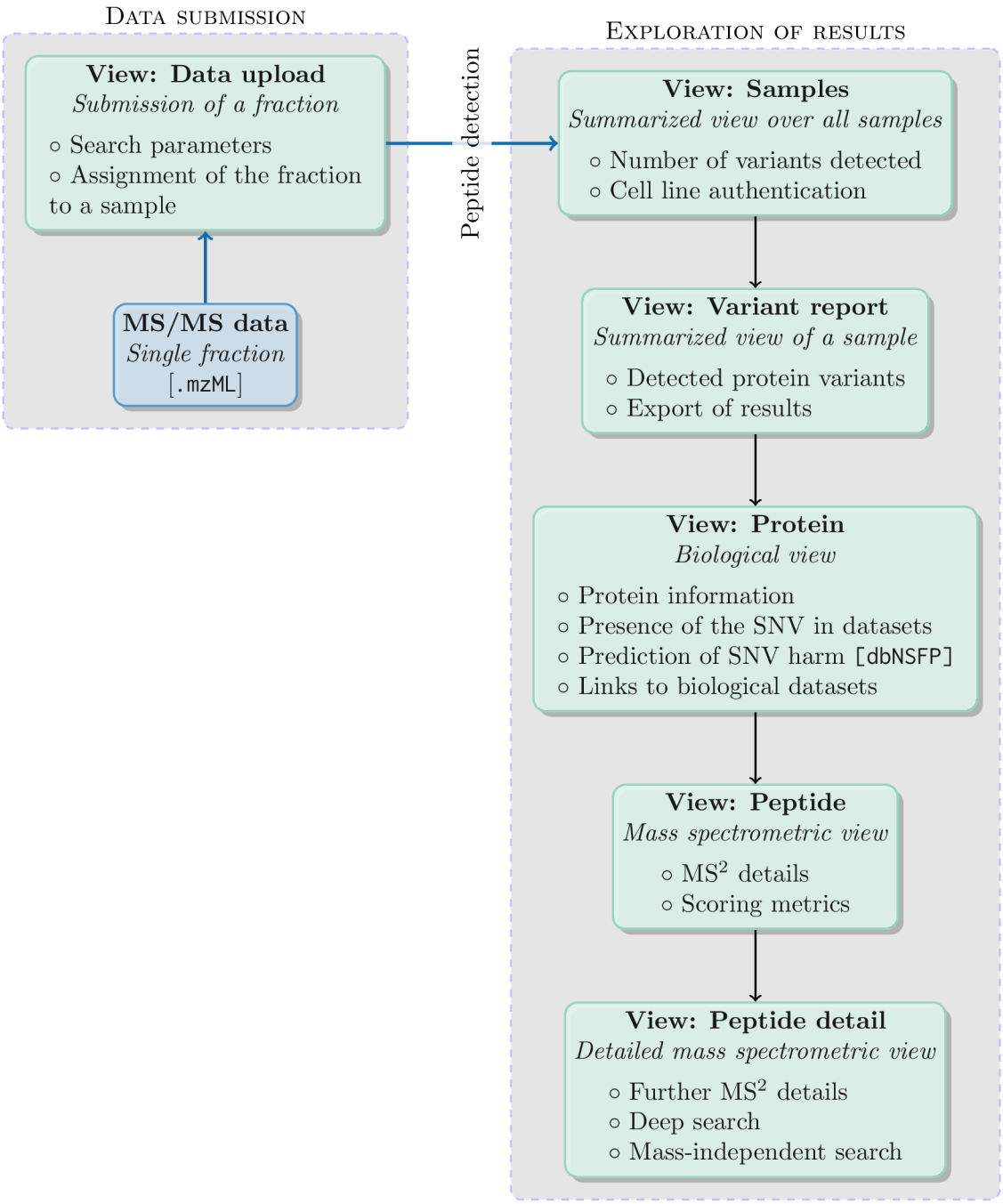


Figure 4.6: Organization of CLAIRE's web interface

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.

4.2.2.3 Note

Before the publication of [3], `http://claire.imtm.cz` is under restricted access. Please contact the author for access credentials.

4.2.3 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 the number of variants claimed on the **Fig. 4.7a**. 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$_{\mathrm{ES}}$, 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 score 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. 4.7b** shows such correlations for the raw X!Tandem$_{\mathrm{GPV}}$ scores, i.e., HyperScore and E-Value [12], in comparison to $\mathrm{Pr}_{\mathrm{max}}$ and its extensions. As is clear from the figure, filtering using metrics derived from $\mathrm{Pr}_{\mathrm{max}}$ exhibited substantially higher correlations with the sequencing support (e.g., Spearman's $\rho = 0.457$ for $\tilde{\mathrm{Pr}}_{\mathrm{max}}^{k,\dagger,i}$ 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 $\mathrm{Pr}_{\mathrm{max}}$ has a substantial impact in this re-

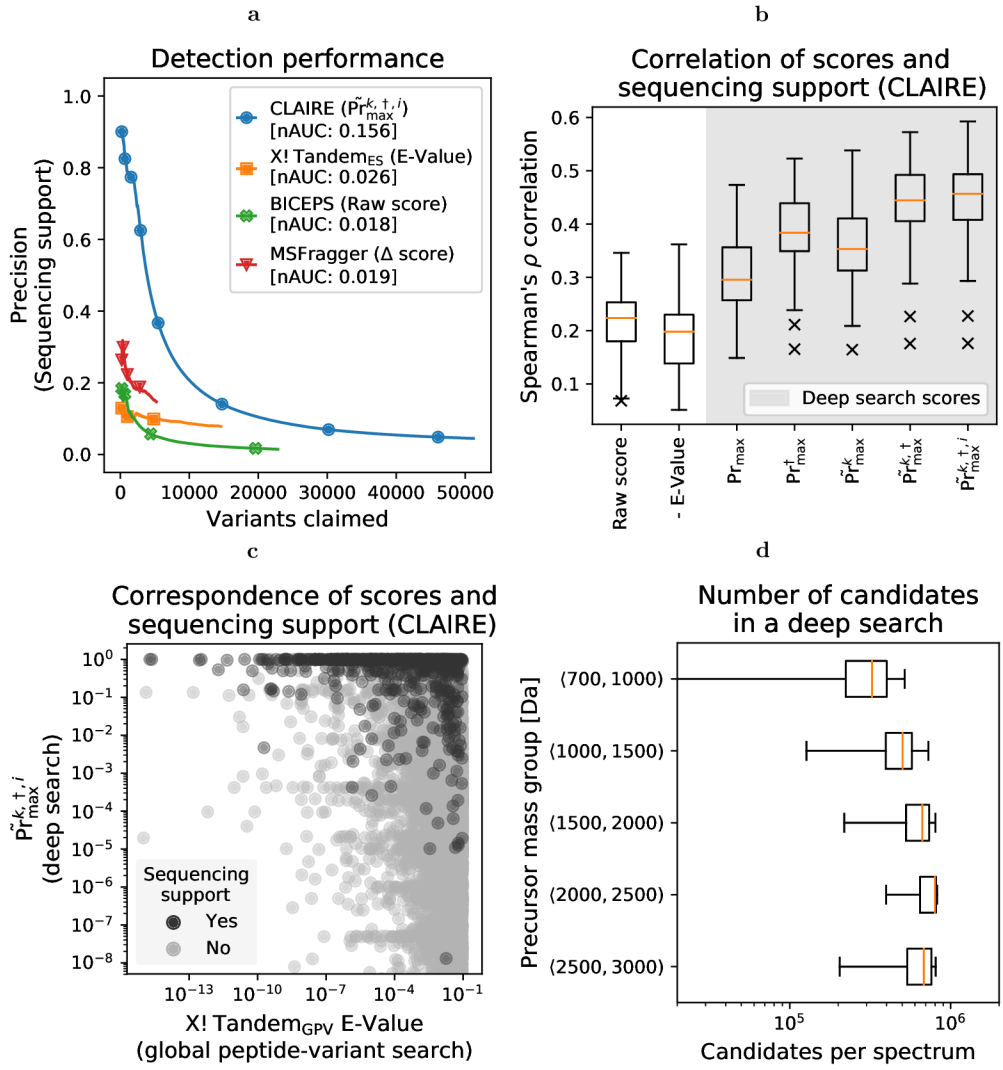


Figure 4.7: Overall view of CLAIRE's behavior.

(**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 $\tilde{\text{Pr}}^{k,\dagger,i}_{\text{max}}$ 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.

spect (Spearman's $\rho = 0.353$ for $\tilde{\text{Pr}}^{k}_{\text{max}}$ vs. $\rho = 0.296$ for 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 $\tilde{\text{Pr}}^{k,\dagger}_{\text{max}}$ vs. $\rho = 0.353$ for $\tilde{\text{Pr}}^{k}_{\text{max}}$; 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 $\tilde{\text{Pr}}^{k,\dagger,i}_{\text{max}}$ vs. $\rho = 0.444$ for $\tilde{\text{Pr}}^{k,\dagger}_{\text{max}}$; 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$_{\text{GPV}}$'s E-Values of variant peptides and their respective $\tilde{\text{Pr}}^{k,\dagger,i}_{\text{max}}$ (**Fig. 4.7c**). The figure shows that most of the sequencing-supported variant peptides had high $\tilde{\text{Pr}}^{k,\dagger,i}_{\text{max}}$, and thus the metric is a better indicator of correctness than the X!Tandem$_{\text{GPV}}$'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. 4.7d**). Given the depth of our peptide database $p_{\text{min}} = 4 \cdot 10^{-6}$, 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.

4.3 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 4.3.1, showing the evidence that CLAIRE can detect *hypermutation status* of tumors—a relevant clinical parameter. Afterward, in section 4.3.2, 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 4.3.3 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, and the details of the methods

involved are presented in the dissertation thesis.

4.3.1 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 [51]. Colorectal cancer (CRC) is the third most common cancer worldwide, expected to result in more than 2.2 million cases annually by 2030 [52]. 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 [53–55]. 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 [56]. 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. 4.8d**). 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.

4.3.2 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* [57]. Mislabeling of a cell line refers to

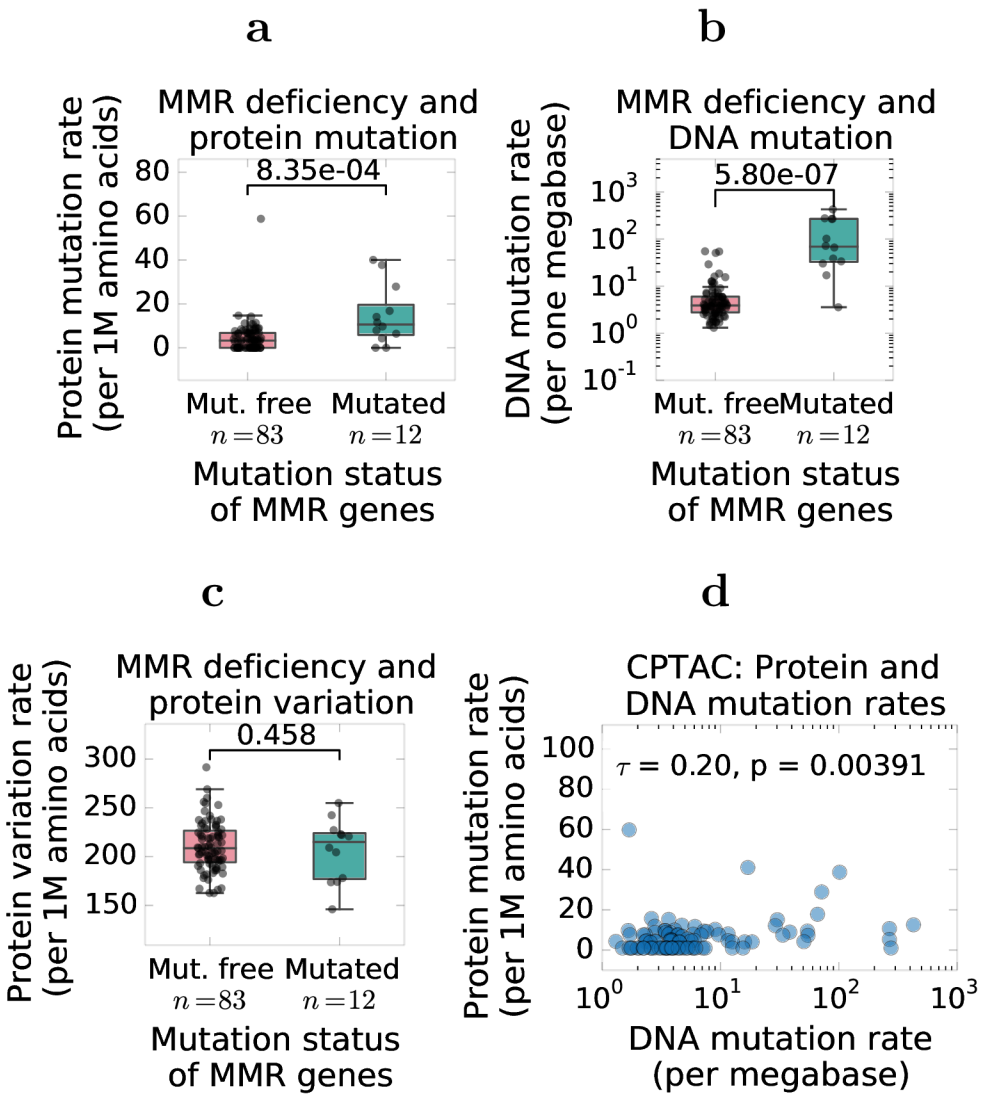


Figure 4.8: Gene and protein variation rates in patients' samples.

(**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.

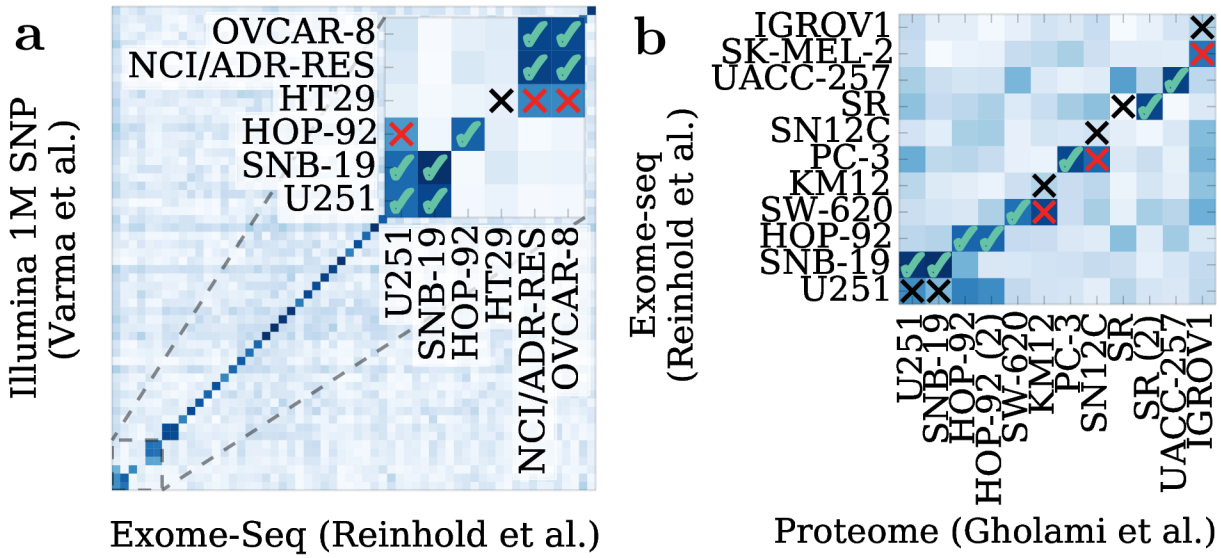


Figure 4.9: Pairwise matches between three NCI_{60} datasets.

(**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.

a situation when researchers unknowingly work on another 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 [58]. Unlike in proteomics, genomic data allow simple authentication of cell lines [59]. 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 [47, 48, 60].

4.3.2.1 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 our methods from [3]. 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:

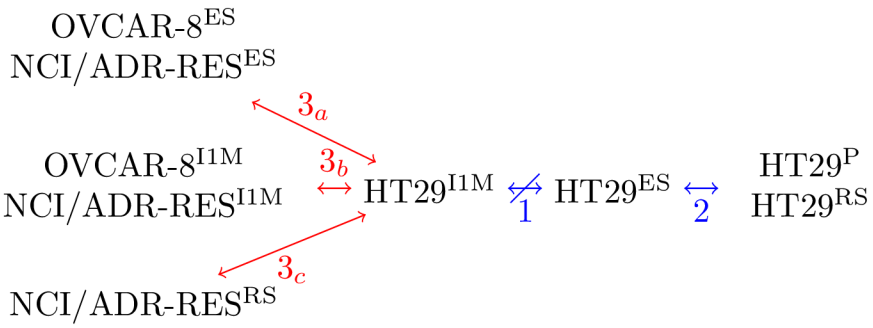
(a) OVCAR-8 and NCI/ADR-RES;
(b) ME-14 and MDA-MB-435; and
(c) SNB-19 and U251.

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) [60], Exome-Seq (ES) [48], RNA-Seq (RS) [48], and the proteomics dataset (P) [47] analyzed using CLAIRE [3]. As an example, **Fig. 4.9** 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. 4.9a** 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:



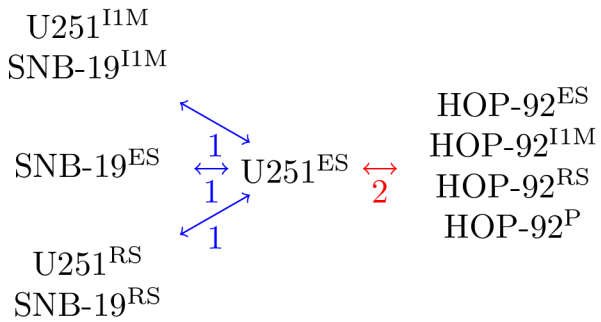
Arrow	Meaning
$\longleftrightarrow$	Expected relationship
$\not\longleftrightarrow$	Lack of expected relationship
$\longleftrightarrow$	Unexpected relationship

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$^{\text{I1M}}$ is indeed one of them, we would expect it to match both cell lines. Based on this, we conclude that HT29$^{\text{ES}}$ is likely HT29 and HT29$^{\text{I1M}}$ 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

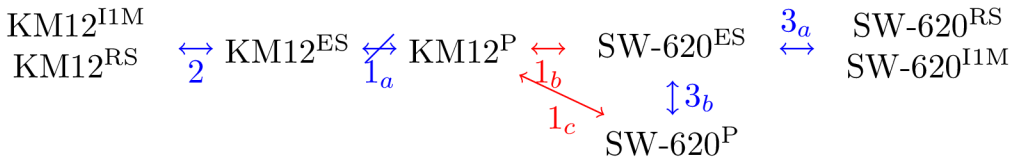
Now, we provide an analysis in the similar spirit of the previous one, showing that U251$^{\text{ES}}$ was likely contaminated by HOP-92. To simplify the exposition, we again provide the diagram of the relevant matches:



Foremost, the data indicates that U251$^{\text{ES}}$ 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$^{\text{ES}}$ also matched HOP-92 in all four analyzed datasets (arrow 2). As a result, we conclude that U251$^{\text{ES}}$ 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. 4.9b**). The diagram of the relevant matches is as follows:



Overall, the data indicate that KM12$^{\text{P}}$ was actually SW-620. Foremost, KM12$^{\text{P}}$ did not match KM12$^{\text{ES}}$ (arrow 1_a) but did match SW-620$^{\text{ES}}$ and SW-620$^{\text{P}}$ instead (arrows 1_b and 1_c, respectively). Furthermore, the KM12$^{\text{ES}}$ was indeed likely KM12, as indicated by its significant matches in other datasets (arrow 2). Similarly, the SW-620$^{\text{ES}}$ was likely SW-620 as indicated by its

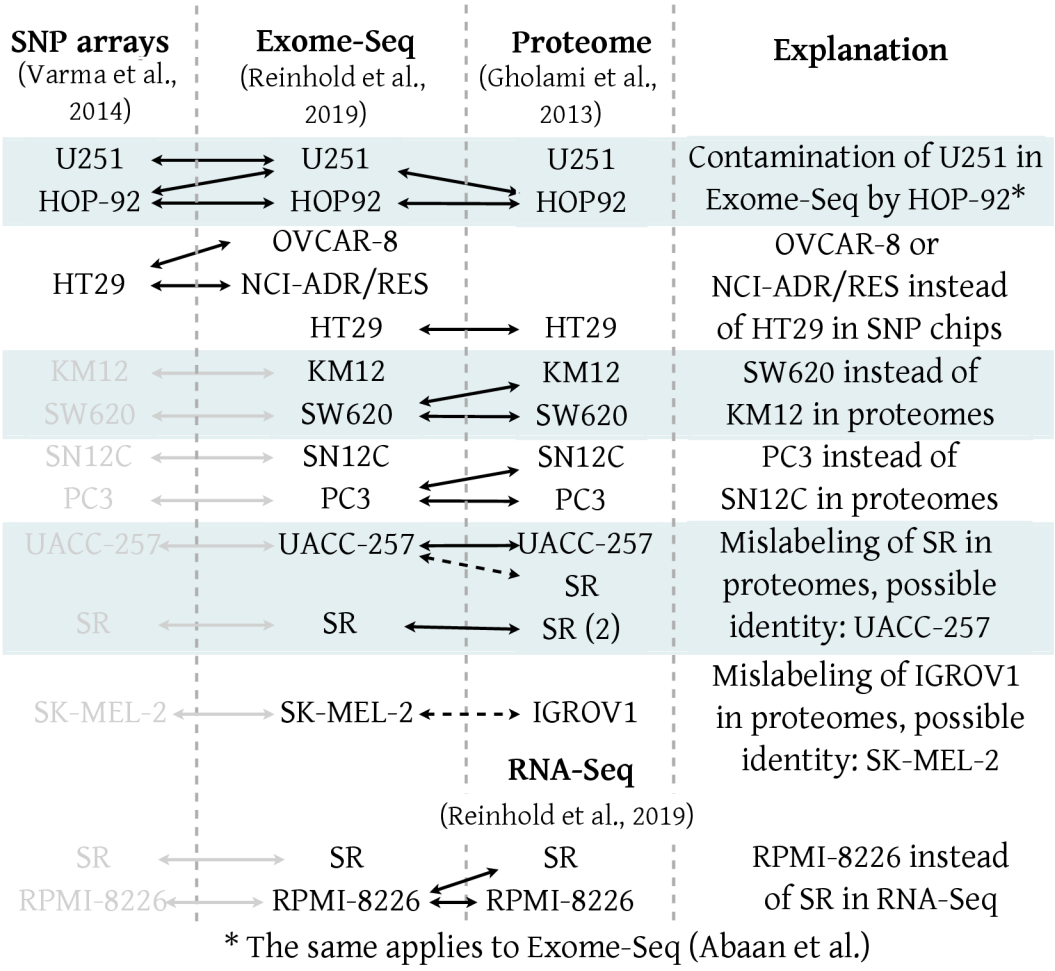


Figure 4.10: Summary of the analysis of NCI_{60} datasets using germline variants.

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. 4.10**.

4.3.3 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—

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 4.11: Identification of individuals against DNA database.

The table shows the results of applying our methods in [3] 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.

by matching against the corresponding DNA dataset. For this purpose, we use our population-frequency method to calculate probability of DNA origin [3] and analyze the data of a seven-member family [3]. The probabilities of individual DNA origins for each family member are visualized on the **Fig. 4.11**. 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.

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 ex-

periments, 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, 61]. The computational proteomics community focused primarily on the second point—improving the capacity to discriminate peptides by predicting more accurate spectra [62–64] or by utilizing additional detection models [64–67]. 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. 4.3**), 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 findings 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. 4.8d**) 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 vysoko účinná 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, 61]. Komunita výpočtovej proteomiky sa sústredila primárne na druhý bod—zvyšovanie kapacity rozlišovania peptidov pomocou predikcie viac presných fragmentačných spektier [62–64], alebo za použitia doplňujúcich detekčných modelov [64–67]. 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 (sekcia 4.3), 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. 4.8d**) 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. *et al.* 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. *et al.* 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. Käll, L. *et al.* Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *Journal of Proteome Research* **7,** 29–34. ISSN: 15353893 (2008).
11. 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).
12. Craig, R. & Beavis, R. C. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics* **20,** 1466–1467. ISSN: 1367-4803 (June 2004).
13. 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).
14. 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).
15. Craig, R. *et al.* Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research* **5,** 1843–1849. ISSN: 15353893 (2006).
16. 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).
17. Muth, T. *et al.* 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).

18. Ma, B. Novor: Real-Time Peptide de Novo Sequencing Software. *Journal of the American Society for Mass Spectrometry* **26,** 1885–1894. ISSN: 18791123 (2015).

19. 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).

20. Tabb, D. L. *et al.* DirecTag: Accurate sequence tags from peptide MS/MS through statistical scoring. *Journal of Proteome Research* **7,** 3838–3846. ISSN: 15353893 (2008).

21. 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).

22. 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).

23. 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).

24. Keller, A. *et al.* 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).

25. Käll, L. *et al.* Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods* **4,** 923–925. ISSN: 15487091 (2007).

26. Sheynkman, G. M. *et al.* Proteogenomics: Integrating Next-Generation Sequencing and Mass Spectrometry to Characterize Human Proteomic Variation. *Annual Review of Analytical Chemistry* **9,** 521–545. ISSN: 19361335 (2016).

27. Zhang, B. *et al.* Proteogenomic characterization of human colon and rectal cancer. *Nature* **513,** 382–387. ISSN: 0028-0836 (Sept. 2014).

28. 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).

29. 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).

30. Zhang, B. *et al.* Proteogenomic characterization of human colon and rectal cancer. *Nature* **513,** 382–387. ISSN: 0028-0836 (2014).

31. 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).

32. Huang, P.-J. *et al.* CMPD: cancer mutant proteome database. *Nucleic Acids Research* **43,** D849–D855. ISSN: 1362-4962 (Jan. 2015).

33. 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).

34. Ahrné, E. *et al.* QuickMod: A tool for open modification spectrum library searches. *Journal of Proteome Research* **10,** 2913–2921. ISSN: 15353893 (2011).

35. 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).

36. 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).

37. Devabhaktuni, A. *et al.* TagGraph reveals vast protein modification landscapes from large tandem mass spectrometry datasets. *Nature Biotechnology* **37,** 469–479. ISSN: 15461696 (2019).

38. Tanner, S. *et al.* InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra. *Analytical Chemistry* **77,** 4626–4639. ISSN: 0003-2700 (July 2005).

39. 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).

40. Kong, A. T. *et al.* MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature Methods* **14,** 513–520. ISSN: 1548-7091 (2017).

41. 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).

42. 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).

43. 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).

44. Sherry, S. T. *et al.* dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research* **29,** 308–311. ISSN: 1362-4962 (Oct. 2000).

45. Lek, M. *et al.* Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536,** 285–291. ISSN: 14764687 (2016).

46. 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).

47. Gholami, A. M. *et al.* Global Proteome Analysis of the NCI-60 Cell Line Panel. *Cell Reports* **4,** 609–620. ISSN: 22111247 (Aug. 2013).

48. 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).

49. Kessner, D. *et al.* ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* **24,** 2534–2536. ISSN: 13674803 (2008).

50. Liu, X., Jian, X. & Boerwinkle, E. dbNSFP: A lightweight database of human non-synonymous SNPs and their functional predictions. *Human Mutation* **32,** 894–899. ISSN: 10597794 (Aug. 2011).

51. 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).

52. Arnold, M. *et al.* Global patterns and trends in colorectal cancer incidence and mortality. *Gut* **66,** 683–691. ISSN: 0017-5749 (Apr. 2017).

53. 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).

54. Yuza, K. *et al.* Hypermutation and microsatellite instability in gastrointestinal cancers. *Oncotarget* **8,** 112103–112115. ISSN: 19492553 (2017).

55. Zhao, P. *et al.* 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).

56. 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).

57. Masters, J. R. W. Cell line misidentification: the beginning of the end. *Nature Reviews Cancer* **10,** 441–448. ISSN: 1474-175X (June 2010).

58. Freedman, L. P. *et al.* Reproducibility: changing the policies and culture of cell line authentication. *Nature Methods* **12,** 493–497. ISSN: 1548-7091 (2015).

59. Reid, Y. *et al.* Authentication of Human Cell Lines by STR DNA Profiling Analysis. *Assay Guidance Manual,* 435–452 (2004).

60. Varma, S. *et al.* 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).

61. 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).

62. 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).

63. Liu, K. *et al.* Full-Spectrum Prediction of Peptides Tandem Mass Spectra using Deep Neural Network. *Analytical Chemistry* **92,** 4275–4283. ISSN: 15206882 (2020).

64. Gessulat, S. *et al.* Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods* **16,** 509–518. ISSN: 15487105 (2019).

65. Klammer, A. A. *et al.* Improving tandem mass spectrum identification using peptide retention time prediction across diverse chromatography conditions. *Analytical Chemistry* **79,** 6111–6118. ISSN: 00032700 (2007).

66. 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).

67. 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).