Deep learning methods for *de novo* peptide sequencing

Wout Bittremieux[1], Varun Ananth[2], William E. Fondrie[3], Carlo Melendez[4], Marina Pominova[1], Justin Sanders[2], Bo Wen[4], Melih Yilmaz[2], and William S. Noble[*4,2]

[1]Department of Computer Science, University of Antwerp
[2]Paul G. Allen School of Computer Science and Engineering, University of Washington
[3]Talus Bioscience
[4]Department of Genome Sciences, University of Washington

Abstract

Protein tandem mass spectrometry data is most often interpreted by matching observed mass spectra to a protein database derived from the reference genome of the sample being analyzed. In many application domains, however, a relevant protein database is unavailable or incomplete, and in such settings *de novo* sequencing is required. Since the introduction of the DeepNovo algorithm in 2017, the field of *de novo* sequencing has been dominated by deep learning methods, which use large amounts of labeled mass spectrometry data to train multi-layer neural networks to translate from observed mass spectra to corresponding peptide sequences. Here, we describe these deep learning methods, outline procedures for evaluating their performance, and discuss the challenges in the field, both in terms of methods development and evaluation protocols.

1 Introduction

In mass spectrometry bottom-up proteomics, proteins in complex biological samples are enzymatically digested into peptides, which undergo at least two rounds of mass spectrometry analysis [27]. First, populations of intact peptides are analyzed (the MS1 scan), and then, populations of peptide fragments are examined (the MS2 scan). This process results in a collection of mass spectra, and the immediate analysis challenge posed by each MS2 spectrum is to identify the amino acid sequence of the peptide responsible for generating the spectrum.

Historically, this question was initially answered via laborious, manual annotation of individual MS2 spectra. In this approach, a human expert would examine an MS2 spectrum, searching for pairs of peaks whose mass difference corresponds to that of an amino acid. If the peptide fragmented fairly uniformly along its backbone, it might be possible to identify a series of such peak pairs and thereby reconstruct the full peptide sequence. Furthermore, experts leveraged their own set of additional, internalized rules—such as the suppression of a b-ion's intensity when followed by proline—to resolve ambiguities and enhance confidence in their sequence assignments [28, 29]. Early algorithms for *de novo* sequencing codified this process into algorithmic procedures [30, 31].

In 1994, however, an alternative approach was proposed: reconstructing the peptide sequence by comparing the observed MS2 spectrum to peptides from a given database. In this procedure, a database of proteins is digested *in silico* into peptides by using enzymatic cleavage rules. For each observed spectrum, all database peptides with masses close to the precursor mass associated with the spectrum are considered "candidate peptides." Here, the threshold for "close" masses depends on the precision with which the mass spectrometer measures the precursor mass. The key to the database search is a score function that evaluates how well the observed spectrum matches the peptide. This score function typically involves creating a theoretical spectrum on the basis of the peptide's amino acid sequence. This approach, pioneered by

*Correspondence: william-noble@uw.edu

Method	Year	Data type		Model architecture			Post-processor	Citations
		DDA	DIA	CNN	Transformer	Other		
DeepNovo [1]	2017	✓		✓				417
DeepNovo-DIA [2]	2019		✓	✓				324
SMSNet [3]	2019	✓		✓				60
RANovo [4]	2020	✓		✓				1
PepNet [5]	2023	✓	✓	✓				22
Casanovo [6]	2022	✓			✓			62
DPST [7]	2022*	✓			✓			7
BiATNovo [8]	2023*	✓	✓	✓				1
Contranovo [9]	2023*	✓			✓			8
GraphNovo [10]	2023	✓			✓			17
InstaNovo [11]	2024*	✓			✓			8
π-HelixNovo [12]	2024	✓			✓			13
NovoB [13]	2024	✓			✓			6
AdaNovo [14]	2024*	✓			✓			1
Transformer-DIA [15]	2024*		✓		✓			4
Cascadia [16]	2024*		✓		✓			1
π-PrimeNovo [17]	2024*							0
PowerNovo [18]	2024*							0
π-xNovo [19]	2024*							0
PointNovo [20]	2021	✓				✓		67
DEPS [21]	2022*	✓				✓		8
PGPointNovo [22]	2023	✓				✓		3
Denovo-GCN [23]	2023	✓				✓		2
SeqNovo [24]	2023	✓				✓		3
pNovo 3 [25]	2019	✓					✓	83
Spectralis [26]	2024	✓					✓	10

Table 1: **Deep learning methods for *de novo* peptide sequencing.** Methods marked with asterisks have not yet been subjected to peer review. Citation counts are from Google Scholar on October 4, 2024.

SEQUEST [32], quickly became the *de facto* standard in the field, facilitated by the development of dozens of search engines (reviewed in [33]).

Despite the popularity of database search for assigning peptides to MS2 spectra, the development of *de novo* sequencing continued in parallel, primarily because database searching does not work well in some application domains. By definition, a database search can only correctly assign peptides to *native* spectra, i.e., MS2 spectra for which the generating peptide is present in the database [34]. *Foreign* spectra, whose generating peptides are not in the database, cannot be correctly identified through this method. In the analysis of a typical human sample—or indeed, any organism with a well-characterized genome—most observed MS2 spectra are assumed to be native, with only a small fraction representing unexpected contaminants or peptides harboring genetic variants. On the other hand, in metaproteomics, where peptides are extracted from, e.g., environmental samples or a gut microbiome, constructing a relevant peptide database is challenging [35]. Even with a database derived from DNA sequencing of the same sample, many foreign spectra will remain. Other settings that require *de novo* sequencing include immunopeptidomics [36], antibody sequencing [37], and paleoproteomics [38]; the first two due to the vast search space of possible native peptides and the latter due to the presence of foreign spectra.

Over the past several decades, *de novo* peptide sequencing algorithms have evolved considerably. Early *de novo* peptide sequencing algorithms employed heuristic search [39] or dynamic programming procedures [40, 41]. The PepNovo algorithm [42] combined dynamic programming with a probabilistic score function that instantiated a set of rules governing peptide fragmentation, and Fisher et al. [43] proposed a closely related hidden Markov model approach. Notably, in 2015, the Novor algorithm [44] improved the state of the art by using a machine learning algorithm (a combination of two decision trees) as the score function for its dynamic programming algorithm.

More recently, as in many other fields, *de novo* sequencing has seen rapid progress due to the introduction of deep learning methods. "Deep learning" refers to any machine learning algorithm that uses a multi-layer neural network [45]. These methods typically possess a large number of trainable parameters and require a correspondingly large amount of training data. Deep learning has been successfully applied in various domains of mass spectrometry proteomics, including predicting fragment ion intensities [46–48], identifying peptide features in MS1 data [49, 50], performing large-scale embedding and clustering of MS2 spectra [51], and predicting peptide properties [47, 52–54]. The first deep learning method for *de novo* sequencing, introduced in 2017, is DeepNovo [1], and it has since been followed by at least 25 additional deep learning methods (Table 1). In addition to their superior performance, the rise of deep learning methods in mass spectrometry analysis can be attributed to three factors: the emergence of neural network architectures that are well suited for mass spectra and peptides, the development of hardware—including graphics processing units (GPUs)—that accelerate the parallel computations that comprise neural networks, and the availability of large-scale public data needed to train these models [55–58].

The purpose of this review is to describe these deep learning methods for *de novo* sequencing, discuss their relative merits, and outline some of the major challenges in this field. We focus solely on deep learning methods, as previous reviews have adequately covered earlier work in the field [59, 60].

2 Deep learning methods

Analogous to advances in deep learning, various neural network architectures have been used for *de novo* sequencing. Although categorizing these methods based on their model architecture is challenging due to the multi-component nature of many models, in what follows, we have identified two broad categories that capture many existing model architectures and have grouped the remaining methods into an "other" category. Additionally, we include two methods that use deep learning to post-process the results from an existing *de novo* sequencer.

The reproducibility of published results requires publicly available (and, ideally, open source) software implementations of these algorithms. Among the 23 tools, 17 have made their source code available, although only eight provide explicit open-source licenses (Table 2). Many tools continue to be under active development, as evidenced by ongoing updates on GitHub.

Method	URL	License	Most recent commit
DeepNovo	https://github.com/nh2tran/DeepNovo	Non-commercial	26 Nov 2020
DeepNovo-DIA	https://github.com/nh2tran/DeepNovo-DIA	Non-commercial	25 Nov 2020
SMSNet	https://github.com/cmb-chula/SMSNet	Apache 2.0	13 Jun 2024
PepNet	https://github.com/lkytal/pepnet	LGPL	21 Jun 2024
Casanovo	https://github.com/Noble-Lab/casanovo	Apache 2.0	19 Sep 2024
DPST	https://github.com/Yan98/DPST	N/A	17 Aug 2022
BiATNovo	Unavailable		
Contranovo	https://github.com/BEAM-Labs/ContraNovo	N/A	14 Mar 2024
GraphNovo	https://github.com/AmadeusloveIris/GraphNovo	Non-commercial	30 Jun 2023
InstaNovo	https://github.com/instadeepai/InstaNovo	Apache 2.0	22 Aug 2024
π-HelixNovo	https://github.com/PHOENIXcenter/pi-HelixNovo	GPL 3.0	23 Sep 2024
NovoB	https://github.com/ProteomeTeam/NovoB	N/A	9 May 2024
AdaNovo	Unavailable		
Transformer-DIA	https://github.com/Biocomputing-Research-Group/Transformer-DIA	GPL 3.0	24 Apr 2024
Cascadia	https://github.com/Noble-Lab/cascadia	Apache 2.0	5 Aug 2024
π-PrimeNovo	Unavailable		
PowerNovo	https://github.com/protdb/PowerNovo	MIT License	1 Jul 2024
π-xNovo	Unavailable		
PointNovo	https://zenodo.org/records/3960823	N/A	Static
DEPS	Unavailable		
PGPointNovo	https://github.com/shallFun4Learning/PGPointNovo	Apache 2.0	26 Jan 2023
Denovo-GCN	Unavailable		
SeqNovo	Unavailable		
pNovo 3	http://pfind.org/software/pNovo/index.html	N/A	6 Dec 2023
Spectralis	https://github.com/gagneurlab/spectralis	N/A	1 Oct 2024

Table 2: **Source code availability.** The most recent commit is as of 2 Oct 2024.

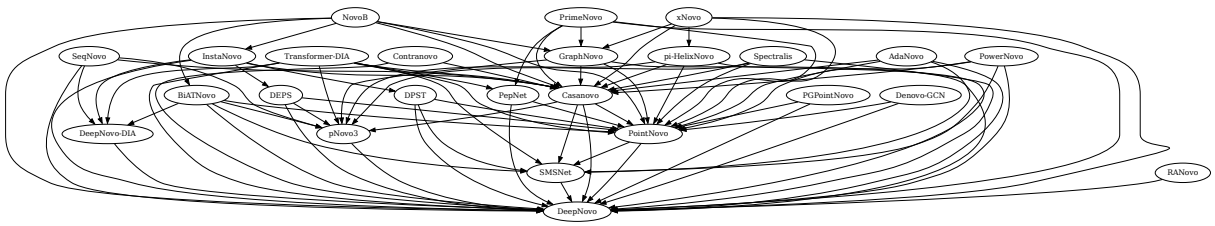

Figure 1: **Citation graph for *de novo* sequencing methods.**

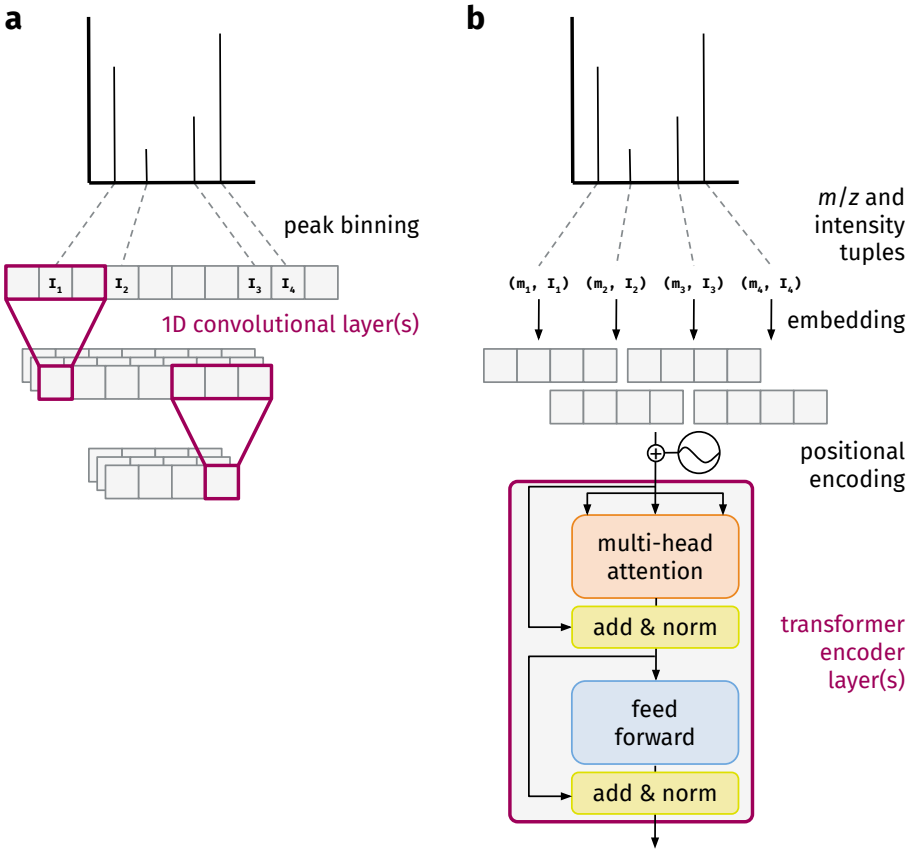

Figure 2: **Neural network architectures to encode mass spectra.** (a) Convolutional neural networks operate on sparse vectors by dividing the m/z range into equal bins and assigning the peak intensities to the corresponding bins. The spectrum vectors are processed through one or more convolutional layers, which can have a varying number of input and output channels, kernel size, etc. (b) Transformer neural networks operate on tuples of m/z and intensity values that are processed through one or more transformer blocks. Key aspects of transformers operating on mass spectra are the positional encodings to capture fine-grained information about m/z values and (optionally) peak intensities, and the attention mechanism to learn interactions between the input peaks.

2.1 Convolutional neural network models

Many *de novo* sequencing models employ convolutional neural networks (CNNs) [45]. A CNN processes vector inputs using a set of sliding windows, where each sliding window (a "filter") learns to recognize specific patterns within the data (Figure 2a). CNNs were instrumental in the advent of deep learning methods, partly because they provide powerful and general pattern recognition capabilities and partly because their computations can be extremely efficiently carried out on a GPU. Furthermore, CNNs provide a strong inductive bias; the learned convolutional filters emphasize relationships between neighboring elements in the input vector. An important factor here, however, is that in standard CNNs, the receptive field is relatively small, particularly when using conventional convolutional layers combined with max-pooling operations. This limited receptive field may not adequately capture patterns between peaks that are far apart in the m/z axis, such as complementary b- and y-ions or peaks separated by multiple amino acids. One approach to address this limitation is the use of dilated convolutions, as implemented in models like PepNet [5]. Dilated convolutions allow the network to expand its receptive field without losing resolution, enabling it to detect patterns across a broader range of m/z values while maintaining computational efficiency.

Another challenge that CNN models of mass spectra face is that, because the CNN operates on vectors, the m/z axis of the MS2 spectra must be discretized prior to input. Choosing the bin size for this discretization is difficult. Large bins sacrifice precision in the m/z measurements, whereas small bins lead to very large inputs that are extremely sparse and require a significant amount of GPU memory to process. Small bins also yield larger edge effects, where two peaks in different spectra may have nearly identical m/z values but end up being separated by a bin boundary.

The first deep learning model for *de novo* peptide sequencing, DeepNovo [1], uses an iterative decoding process that employs two parallel models. Each spectrum in the training set is converted to vector format by discretizing the m/z axis, using bins of size 0.1 or 0.01 m/z depending on the resolution of the training data. For the first of the two models, ion-CNN, these vectors are processed along with the predicted prefix to produce a tensor of dimension $128 \times 26 \times 8 \times 10$, where 128 is the batch size, 26 is the amino acid alphabet size (including post-translational modifications [PTMs]), 8 is the number of ion types (including b- and y-ions and various neutral losses), and 10 is the number of m/z bins extracted around each target ion. This tensor is then processed by ion-CNN, a two-layer CNN that takes the spectrum and a representation of a predicted peptide prefix as input, aiming to predict the subsequent amino acid. The second model, a type of recurrent neural network (RNN) known as a "long short-term memory" (LSTM) network [61], is preceded by a two-layer CNN, spectrum-CNN, to detect signals indicative of which amino acids are present in the spectrum. Using the CNN outputs, the LSTM then iteratively predicts amino acids in a manner similar to the ion-CNN model. During decoding, the penultimate layers of these two models, ion-CNN and the LSTM, are concatenated and fed through a single, fully-connected neural network layer, which outputs a 26-dimensional vector of unscaled log probabilities (logits), serving as the per-amino-acid prediction. DeepNovo also employs a dynamic programming post-processor that uses the predicted logits and the knapsack algorithm to ensure that the predicted peptide sequence has a mass within the specified tolerance of the measured precursor mass. As the first deep learning method in the field, the DeepNovo paper is cited by all subsequent *de novo* sequencing papers (Figure 1).

DeepNovo-DIA [2] generalizes the DeepNovo model to work with data generated using data-independent acquisition (DIA). Unlike traditional data-dependent acquisition (DDA) mass spectrometry data, where the precursors selected for fragmentation and MS2 characterization are selected in a data-driven way, DIA selects precursors according to a predetermined schedule. Consequently, DIA spectra are more complex than DDA spectra. On the other hand, in DDA each peptide species is typically observed once, whereas the DIA precursor schedule is designed to measure each peptide multiple times. *De novo* methods for analysis of DIA data must take into account these properties of DIA data. The core of the DeepNovo-DIA model is similar to DeepNovo, including the ion-CNN, spectrum-CNN, and LSTM components. The primary difference is that, because DIA data can be organized along a temporal axis, with multiple adjacent scans containing information about a given analyte, the preprocessing steps in DeepNovo-DIA involve detecting 3D fragment ion features and 2D precursor ion features. In practice, training or applying DeepNovo-DIA requires first processing the DIA MS1 data to detect precursor features using an external tool, after which the model makes a prediction for each such feature. Subsequent work by a different research group described an alternative, active learning strategy for training the DeepNovo-DIA model [62].

SMSNet [3] employs a model architecture similar to that of DeepNovo but applies it to a variant of the *de novo* search setting, in which a peptide database is used to fill in ambiguous predictions. The deep neural network consists of an encoder, similar to the ion-CNN network in DeepNovo, which encodes an overview of the input spectrum vector in a length-1024 feature vector. This vector is used to initialize the decoder, which uses an RNN architecture to iteratively predict each amino acid, given the predicted prefix. As in DeepNovo, the knapsack algorithm is used to ensure that the predicted peptide mass matches the observed precursor mass within a specified tolerance. During inference, SMSNet uses beam search decoding with a beam size of 20. The primary novelty of SMSNet lies in its rescoring and database search strategy. A neural network with two fully-connected layers is trained to re-score the amino acids predicted by the primary model, assigning a confidence score to each. SMSNet then uses these calibrated scores to mask out low-confidence predictions, thereby improving the model's accuracy while sacrificing some coverage. To mitigate this loss in coverage, SMSNet identifies "sequence tags"—contiguous blocks of high-confidence amino acid predictions—which are used to match to the peptide database.

RANovo [4] employs a CNN with a deep residual network architecture [63]. Thus, in addition to passing information directly from layer to layer, RANovo contains residual connections that pass information from shallow layers to much deeper layers in the network. The model also incorporates the squeeze-and-excitation mechanism [64], which aims to automatically identify important pairwise relationships between input features. The model consists of two convolutional layers and two fully-connected layers.

PepNet [5] is a fully convolutional neural network that directly outputs a predicted peptide sequence. The input spectrum is represented as a 20,480 $\times$ 4 input matrix, where the first dimension is a discretized m/z axis with a bin size of 0.1 m/z, the second dimension represents the peak intensities and corresponding m/z values, and these dimensions are repeated in both forward and reverse order. The core of the model is a series of five temporal convolutional blocks [65], each of which operates at a different resolution. The overall model thus aims to capture both local and global structures, which are then integrated in a final merging branch. The output of the model is produced via a final softmax layer, yielding a probability matrix of size 32 $\times$ 23, where 32 is the maximum peptide length and 23 is the alphabet size. Unlike DeepNovo and related methods, no post-processing is employed to force the predicted peptide to match the observed precursor mass.

2.2 Transformer models

A second category of *de novo* sequencing models employs the transformer architecture (Figure 2b). Transformers were first developed for natural language processing, such as translating from English to German and English to French [66], but have since then found success in a broad range of application domains, including modeling DNA and protein sequences [67, 68]. The transformer can handle variable-length inputs and the model architecture is indifferent to the order of input tokens. Hence, it is often necessary to explicitly encode the position of each input token and provide these encoded positions along with the tokens themselves. Notably, using positional embeddings removes the need to discretize the m/z axis of mass spectra, avoiding the corresponding issues mentioned before.

Furthermore, a key feature of the transformer is the "attention" mechanism, which enables the model to automatically learn important semantic relationships between pairs of input features. A key advantage of the attention mechanism is that it provides a global receptive field. This allows transformers to naturally model relationships between all peaks in the spectrum regardless of their distance on the m/z axis, in contrast to CNNs, which are constrained by a localized receptive field. This capacity for global context is one reason transformers have become the architecture of choice in recent *de novo* sequencing models.

Casanovo [6] uses a transformer architecture to treat *de novo* sequencing as a sequence-to-sequence translation task, translating from the series of peaks in an MS2 spectrum to a series of amino acids. The model comprises an encoder and a decoder, both transformers. The encoder learns an in-context representation of the input MS2 spectrum, while the decoder predicts the next amino acid in the peptide sequence given the spectrum representation and previously predicted amino acids. The encoder and decoder each consist of nine layers with eight heads per layer, a hidden dimension of 512, and a feedforward dimension of 1024. Like other deep learning models, Casanovo predicts a peptide sequence one amino acid at a time, using beam search decoding to find the predicted peptide sequence with the highest score [69]. No post-processing step is included to force the predicted peptide mass to match the observed precursor mass, but an optional filter

can penalize predictions that lie outside the precursor mass tolerance by assigning them negative scores.

DPST [7] also proposes a transformer-based model but introduces a set of inductive biases to constrain its search space. First, it reframes the *de novo* sequencing task in a Bayesian setting, where the amino acid posterior probabilities are predicted from spectrum information and amino acid priors. Higher prior probabilities are given to amino acids that minimize the difference between the precursor mass and the expected peptide mass calculated with dynamic programming. Second, the DPST encoder assigns a confidence value to each peak based on its consistency with neighboring peaks, prioritizing those separated by amino acid masses in the encoded spectrum representation. Finally, the DPST decoder comprises two branches: the global branch makes predictions from the entire spectrum representation, while the local branch focuses on the spectrum part corresponding to the currently predicted peptide prefix.

BiATNovo [8] combines a CNN spectrum encoder, directly modeled after the ion-CNN in DeepNovo, with two attention-based decoder networks to sequence a given spectrum. The key contribution is the introduction of two new approaches for bi-directional peptide decoding, called "independent" and "synchronous" decoding. During independent decoding, two separate decoders predict the peptide from N-terminal to C-terminal and from C-terminal to N-terminal. The two predictions are then combined into a single peptide sequence by retaining the top-scoring amino acid at each position, subject to the constraint that the final prediction matches the overall precursor mass. During synchronous decoding, a single transformer decoder predicts the peptide sequence simultaneously from both directions, with the prefix and suffix meeting in the middle when a stop token is predicted or the precursor mass is exceeded. A re-ranking step is then applied to the outputs from both the independent and synchronous decoding strategies to select the single top scoring peptide for each spectrum. The proposed decoding strategy is also compatible with the spectrum encoder in DeepNovo-DIA, allowing BiATNovo to be applied to both DDA and DIA data.

ContraNovo [9] is a reimplementation of the Casanovo model with two primary modifications. First, ContraNovo provides additional inputs to the decoder, including the masses of the previously predicted amino acids, the total mass of the remaining uninferred peptide, and the masses of individual amino acids. Second, ContraNovo introduces a contrastive loss term to the loss function, in addition to the cross-entropy loss employed by Casanovo, aiming to enhance the similarity between the encoded spectrum and encoded peptide.

GraphNovo [10] implements a three-stage procedure. Initially, the observed spectrum is converted into a spectrum graph representation, where nodes correspond to peaks and edges represent mass relationships between pairs of peaks. This spectrum graph is then processed successively by two networks: GraphNovo-PathSearcher produces the optimal node sequence corresponding to a partial peptide prediction and unresolved mass tags based on the mass difference encoded in the edges, and then GraphNovo-SeqFiller outputs the full amino acid sequence. Both networks adopt a six-layer Graphormer [70] encoder architecture, which combines transformers and graph neural networks by introducing centrality encoding, spatial encoding, and edge encoding in the attention mechanism. These embeddings are input into a 12-layer transformer decoder in both PathSearcher and SeqFiller to predict the optimal node sequence and full amino acid sequence, respectively.

InstaNovo [11] uses a transformer architecture very similar to Casanovo's but introduces a novel decoding strategy. Like Casanovo, the model is a 9-layer transformer; however, it differs in some dimensions: InstaNovo uses 16 heads per layer instead of eight, and a hidden dimension of 768 instead of 512. InstaNovo employs beam search decoding with a knapsack constraint to ensure that the prediction matches the observed precursor mass and introduces a diffusion-based post-processing strategy to refine predicted sequences through 20 rounds of diffusion using a new 12-layer fixed-length decoder.

π-HelixNovo [12] also employs a transformer architecture similar to Casanovo's. However, π-HelixNovo introduces the notion of a "complementary spectrum," in which, following a denoising step, each peak in the observed spectrum is reflected relative to the observed precursor m/z value. Thus, a b-ion in the observed spectrum is transformed into its corresponding y-ion in the complementary spectrum. In principle, this complementary spectrum can reconstruct some b- and y-ions when some b/y pairs are incomplete. This complementary spectrum is input to the encoder alongside the observed spectrum. Additionally, π-HelixNovo provides the precursor m/z value as an input to the encoder, rather than only providing it at the decoding step as in Casanovo.

NovoB [13] uses a transformer and introduces three changes relative to Casanovo. First, the encoder takes several additional inputs, including the precursor mass, charge, m/z, and intensity. Second, the decoder

receives an additional input, which is an explicit representation of the residual mass. Third, the model employs two decoders, operating from N-terminal to C-terminal and from C-terminal to N-terminal, and selects the prediction with the higher score, similar to the independent bi-directional decoder in BiATNovo.

AdaNovo [14] employs a transformer encoder–decoder architecture similar to that of Casanovo but introduces an adaptive training strategy based on the conditional mutual information (CMI) between each amino acid in a peptide sequence and the observed MS2 spectrum. During training, AdaNovo uses two decoders: one predicts the next amino acid based solely on the previously predicted subsequence, while the other also considers the mass spectrum information. By comparing their outputs, AdaNovo calculates CMI scores to adaptively re-weight both individual amino acids and training examples. This approach focuses the training on challenging amino acids, such as those with PTMs, and enhances the model's robustness to noisy data by emphasizing highly informative training examples. During inference, AdaNovo uses only the spectrum-aware decoder and filters out predictions that do not match the observed precursor mass.

Transformer-DIA [15] is an extension of the DeepNovo-DIA architecture that replaces the convolutional layers in the spectrum encoder with transformer self-attention layers. After extracting the same precursor profile and theoretical product ion array for each precursor feature as DeepNovo-DIA, the model encodes the temporal information of successive MS2 scans using a positional embedding, allowing the LSTM decoding to be replaced by a standard transformer decoder layer. Additionally, Transformer-DIA includes a beam search decoding procedure similar to the one employed by Casanovo.

Cascadia [16] extends the Casanovo architecture to work with DIA data. Unlike DeepNovo-DIA and Transformer-DIA, Cascadia does not rely on an initial precursor detection step. Instead, the model takes as input "augmented spectra," which include a series of temporally adjacent MS2 scans and all MS1 peaks within the corresponding isolation window. This approach enables the model to extract information from the MS1 data about the precursor isotope distribution, without excluding peptides with a weak MS1 signal but clear fragmentation pattern in the MS2. Cascadia encodes each input peak with a two-dimensional retention time-by-m/z embedding. Additionally, each peak is summed with a learned embedding that indicates whether it is derived from an MS1 or MS2 scan. This sequence of peak representations is then processed by a standard transformer encoder–decoder architecture akin to that of Casanovo. Furthermore, Cascadia employs an auxiliary fragment ion loss term that, in addition to predicting the amino acid sequence for a peptide, also predicts which peaks correspond to b/y fragment ions. This task helps stabilize model training on the much noisier DIA data, especially early in training when the model predicts few peptide sequences correctly.

PowerNovo [18] employs a two-stage *de novo* sequencing process that combines a transformer encoder–decoder for peptide prediction with a BERT [71] encoder for further refinement and evaluation of candidate sequences. The first model autoregressively generates the amino acid sequence from an input MS2 spectrum and precursor information, using beam search to identify the most likely sequences. Next, the BERT model, trained with a masked language modeling objective to correct noisy peptide sequences, adjusts amino acids in peptide hypotheses generated by beam search. This model also classifies each peptide by detectability as "decoy," "poorly detectable," or "highly detectable," filtering out decoys and rescoring the remaining sequences to prioritize those with high detectability. The sequence with the highest score is then selected. PowerNovo also integrates protein assembly and protein inference modules to provide a complete workflow for mass spectrometry data analysis.

π-PrimeNovo [17] introduces the first non-autoregressive transformer model for *de novo* peptide sequencing. Unlike most transformer-based methods, π-PrimeNovo predicts all amino acids in the sequence simultaneously, avoiding the error accumulation typical for next-token prediction strategies. The model is trained with a connectionist temporal classification loss [72] to enhance the global coherence of the peptide sequence and avoid the local ambiguities that can arise in non-autoregressive sequence generation. This loss function operates at the sequence level rather than at the token level. Finally, π-PrimeNovo includes a novel precise mass control unit that implements a CUDA-accelerated knapsack-type dynamic programming algorithm to ensure that the predicted sequence matches the precursor mass during decoding.

The π-xNovo [19] transformer architecture includes two primary innovations. First, the model employs a joint masking strategy during training. In this approach, the input peptide sequence is subjected to a random boolean mask, and the encoded spectrum representation is subjected to two "soft" (i.e., real-valued) masks. These latter masks are produced by a perceptron that is itself learned during the training procedure. The use of two masks represents the notion that they may learn to capture complementary b- and y-ion

series. Second, the decoder employs a multi-head attention mechanism that is designed to allow attention to accrue between each amino acid and all positions within the spectrum. This approach allows the model to explicitly indicate which peaks are responsible for each predicted amino acid, facilitating interpretability. In particular, the product of the peptide attention matrix and the spectrum attention matrix yields an interpretation matrix in which rows are predicted amino acids, columns are peaks in the spectrum, and values indicate high relevance of a given peak to a given predicted amino acid. The interpretation is a key component of an associated post-processing system, called "π-xNovo-QC," that combines the interpretation matrix with a set of rules to provide a confidence score for each prediction from π-xNovo.

2.3 Alternative architectures

PointNovo [20] is inspired by DeepNovo and produced by many of the same authors. PointNovo's primary innovation involves eliminating the need to discretize the spectrum m/z axis, thereby enabling the model to make use of high mass accuracy data without requiring a large memory footprint. Whereas DeepNovo uses an input vector of length 150,000 to represent a spectrum, PointNovo instead represents each spectrum as a set of (m/z, intensity) pairs. The model employs a novel architecture that is designed to handle a set of such tuples in an order-agnostic way, using ideas from the PointNet architecture [73]. Unlike DeepNovo, the LSTM component of PointNovo is optional, though empirical results suggest that including the LSTM tends to provide higher quality predictions.

PGPointNovo [22] is an improved implementation of PointNovo that enables parallel processing on multiple GPUs. This version includes several model enhancements, such as the rectified Adam learning rate scheduler [74], a lookahead strategy that uses two sets of model weights to reduce the need for extensive hyperparameter tuning [75], and a technique called gradient centralization [76] that regularizes the learned weight vectors to improve generalization performance.

DEPS [21] uses an architecture similar to PointNovo, but with several enhancements. Like PointNovo, each spectrum is represented as a set of (m/z, intensity) pairs, processed using a PointNet feature extractor. The extracted features are then processed by a CNN and an LSTM, and the concatenated outputs of those two networks are passed through an output layer to iteratively produce a predicted peptide sequence. However, the CNN incorporates an attention mechanism, inspired by deep residual shrinkage networks [77], which is designed to more robustly handle noise peaks in the observed spectrum.

Denovo-GCN [23] generates undirected graph representations of its input spectra, encoding each peak as a node, along with four additional "virtual peaks": the proton mass, water mass, peptide mass, and the mass of the peptide minus water loss. These virtual peaks aid in predicting the terminal amino acids of the peptide sequence. The graph's edges are constructed based on the differences between peak m/z values and a matrix of amino acid masses, compared against an error threshold, similar to the method employed by GraphNovo. Node features are derived by comparing peak m/z values with a matrix of theoretical m/z values for each unique amino acid and fragment ion type. The model's backbone comprises three blocks: (i) a three-layer CNN made up of 1D convolutional filters that learn local features of the graph, (ii) a two-layer graph convolutional network (GCN) that extracts higher-level relationships among the graph's nodes using features learned by the CNN block, and (iii) a series of linear layers and a softmax layer. The model's predictions are then passed to a knapsack algorithm, which selects a physically plausible amino acid candidate from the predictions, similar to DeepNovo.

SeqNovo [24] uses an RNN architecture composed of an encoder and decoder block, each consisting of gated recurrent units (GRUs) [78]. The model comes in two versions. In the first version, the primary input unit to the encoder is a peak, represented by m/z and intensity values, which is passed through a three-layer GRU block. The output from this block is concatenated with the output from three fully-connected layers that are fed the entire spectrum, incorporating higher-level features of the spectrum that complement the RNN's output. The model's decoder consists of a GRU-based RNN that predicts the next amino acid at each time step using the previous time step's amino acid prediction and hidden state, along with the output from the encoder. The second version of the model operates on the same inputs and employs the same three-layer GRU block but omits the fully-connected layers in the encoder. Instead, it implements an attention mechanism in the decoder, with a query vector generated by the previous time step's hidden state, while the key and value vectors for each time step are generated by the output from the encoder.

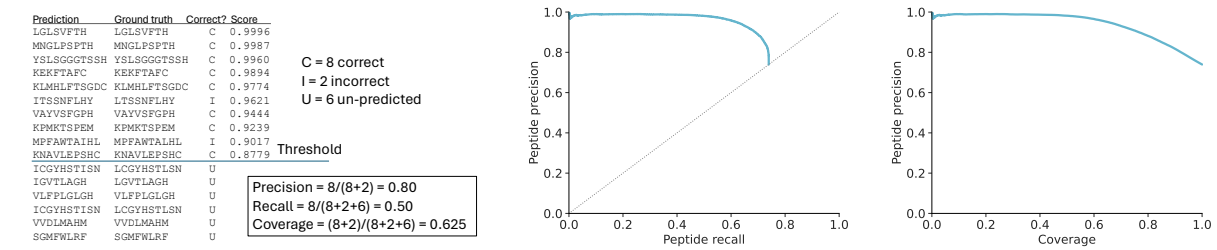

Figure 3: **Precision–recall versus precision–coverage.** (a) Sample computation of precision, recall, and coverage at a given threshold. (b) A sample precision–recall curve, calculated by applying Casanovo to the human dataset from the nine-species benchmark. For reference, the plot includes the line $x = y$. The area under this curve is 0.717. (c) The same curve as in panel (a), but plotted as precision–coverage. The area under this curve is 0.940.

2.4 Post-processing methods

Finally, we discuss two deep learning methods that address the *de novo* sequencing task by post-processing the results of an existing *de novo* sequencing method.

The pNovo 3 algorithm [25] re-ranks a given set of *de novo* predictions by using a deep learning model. The method builds upon pNovo+ [79], which uses a spectrum graph-based algorithm for *de novo* sequencing. In pNovo 3, the top 10 predicted candidate peptides are retained and provided as input to the pDeep deep learning model, which predicts fragment ion intensities [46]. A set of features, based on the pDeep output, are computed, and these feature vectors are used to train a ranking support vector machine (SVM) [80]. The final output of the trained model is the top-scoring candidate peptide, as determined by the SVM.

The Spectralis [26] model aims to improve a given set of *de novo* predictions through an auxiliary task of "bin classification." Operating on a mass axis discretized into 1 Da bins, the task involves predicting whether a given bin contains a singly charged b- or y-ion. An accurate solution to this task will, in principle, allow for a direct read-out of the corresponding peptide sequence. This prediction task is performed using a CNN that employs a novel type of convolutional layer, an "amino-acid gapped" layer, consisting of convolutional filters with gaps corresponding to the masses of the amino acids. The Spectralis model leverages predictions made by existing *de novo* prediction methods (Casanovo and Novor) to transform them into more accurate predictions. The paper also proposes a method, Spectralis-score, for recalibrating scores from Novor and Casanovo using a machine learning post-processor. Finally, the paper describes a genetic algorithm method, Spectralis-EA, that takes a candidate peptide predicted by Casanovo, produces a population of peptide variants, and then employs a genetic algorithm to optimize the Spectralis-score.

3 Performance measures

For performance evaluation, the output of a *de novo* sequencing algorithm is typically represented in one of two forms: a ranked list of predicted peptides or a ranked list of amino acids. Each entry in this list can then be marked as "correct" or "incorrect," based on a comparison to a ground truth annotation. At the peptide level, this annotation is straightforward: the predicted peptide sequence must match the ground truth peptide exactly, perhaps allowing for isomeric (Leu ↔ Ile) or isobaric (deamidated Asn ↔ Asp) substitutions. At the amino acid level, the annotation is somewhat more complex because we must also consider the masses of the preceding and following amino acids. In practice, the most common procedure is to follow the procedure from the DeepNovo paper [1], requiring that a predicted amino acid have either a prefix or suffix that differs by no more than 0.5 Da in mass from the corresponding amino acid sequence in the ground truth peptide.

From the annotated ranked lists, various performance measures can be calculated. Similar to the binary classification setting, we can apply a threshold to these ranked lists to separately consider predictions with confidence scores that fall above and below the threshold. In a binary classification setting, the precision

and recall performance metrics are defined as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned}$$

where TP, FP, and FN correspond to the number of true positives, false positives, and false negatives, respectively, defined with respect to a particular score threshold. Thus, "precision" is the proportion of correct predictions among all predictions with scores greater than the specified threshold, and "recall" is the proportion of examples with positive labels that receive scores greater than the specified threshold.

Many *de novo* sequencing studies borrow the terms "precision" and "recall" but employ somewhat idiosyncratic definitions. In particular, because *de novo* sequencing is not a binary classification task, the traditional categories of true positive, false positive, true negative, and false negative do not apply. Instead, there are only three categories: predictions above the threshold are marked "correct" or "incorrect," and predictions below the threshold are marked "unpredicted" (Figure 3A). Using these categories, we can define:

$$\begin{aligned} \text{precision} &= \frac{C}{C + I} \\ \text{recall} &= \frac{C}{C + I + U}, \end{aligned}$$

where C is the number of correct predictions, I is the number of incorrect predictions, and U is the number of peptides or amino acids with no predictions [44]. This alternative definition of "precision" aligns with the traditional definition from the binary classification setting, which is the proportion of predictions with scores greater than the specified score threshold that are correct. However, the alternative definition of "recall" is unusual. In the binary classification setting, "recall" is the proportion of examples with positive labels that are correctly predicted to be positive. The alternative definition, on the other hand, is the proportion of the full set of examples that are predicted correctly. As a result, a precision–recall curve that uses the alternative definition differs qualitatively from a traditional precision–recall curve. In particular, when the threshold is moved to the very end of the ranked list, the value of U goes to zero and, hence, the precision and recall are equal. Therefore, a precision–recall curve that employs the alternative definitions above terminates on the line $x = y$, whereas a traditional precision–recall curve terminates at $x = 1$, with y being equal to the proportion of positive examples in the dataset (Figure 3B).

To avoid this terminological confusion and produce a plot that spans the entire range from 0 to 1, some *de novo* sequencing studies instead employ a precision–coverage curve, where "precision" is defined as above but "coverage" is defined as the proportion of predictions with scores greater than the threshold, irrespective of whether the predictions are correct:

$$\text{coverage} = \frac{C + I}{C + I + U}$$

The resulting curve always terminates at $x = 1$ (Figure 3C).

One additional complication is that, when evaluating recall at the amino acid level, several papers [1, 2, 12, 44] use an alternate definition of recall:

$$\text{recall} = \frac{C}{N},$$

where N is the total number of amino acids in the ground truth dataset, rather than the total number of predicted amino acids. Thus, if the *de novo* prediction method occasionally predicts peptides that are too long or too short, then the total number of predicted amino acids ($C + I + U$) may not equal N. In this setting, the precision–recall curve will not terminate exactly on the line $x = y$, and the precision at the end of the list will not equal the recall.

Given a precision–recall or precision–coverage curve, three types of performance measures can be calculated. Some studies report the overall precision of the full list, either at the peptide or amino acid level. At the peptide level, this value is alternatively referred to as "peptide recall" [1] or "peptide-level accuracy." At the amino acid level, "positional accuracy" [5] or "amino acid-level accuracy" [13] are sometimes used. In

addition, some studies report the area under either the precision–recall curve [1] or the precision–coverage curve [6]. This approach has the advantage of giving a higher score to methods that successfully rank correct predictions above incorrect ones. Finally, at least one study measures the number of spectra that are correctly predicted subject to a specified precision threshold (of 95% or 99%) [5]. This approach captures the intuition that, in practice, many end users are interested in obtaining a large set of predictions with a high estimated precision.

4 Preventing leakage of information from training to test set

An important component of performance evaluation involves ensuring a proper separation between training and testing data. One simple way to accomplish this train/test split is to randomly split a large collection of annotated spectra, reserving some to train the model and some for testing. The nine-species benchmark dataset used in the original DeepNovo paper adopts this approach [1]. This benchmark has been used extensively in subsequent studies [4, 6–9, 11–15, 20, 21, 25, 26, 69]. Unfortunately, this simple spectrum-level splitting procedure does not ensure that peptide sequences seen during training do not also appear in the test set. Consequently, if a machine learning algorithm does a good job of "memorizing" the training set sequences, then using a spectrum-level split may give the algorithm an unfair advantage when presented with new spectra generated by the same peptides. To avoid this problem, some studies go a step further and carry out peptide-level splitting, thereby preventing leakage of sequence information from the training set to the test set [69]. Because spectra originating from peptides that only differ by one or more PTMs are often highly similar to one another, this peptide-level splitting should be performed on the base peptide sequence irrespective of PTMs, i.e., a peptide and all its modified variants should be present in only a single split.

However, even with peptide-level splitting, some additional information can potentially leak through if the training and test sets both contain spectra generated from the same experiment. In principle, a machine learning algorithm may be capable of identifying distinct properties of spectra from a given run, for example, related to calibration of the m/z axis. If the model can effectively say "This spectrum resembles these other spectra that I saw during training," then it can again achieve an unfair advantage, because in practice true "test" spectra will never come from the experiments used during training. Hence, to avoid batch effects and peptide-level leakage, a proper train/test setup should ensure that the training set and test set do not overlap in either sense. The revised nine-species benchmark dataset addresses this problem by splitting at the species level and eliminating overlaps at the peptide level [69, 81].

5 Performance comparisons

Perhaps the most obvious question to ask about the 26 deep learning *de novo* sequencing methods listed in Table 1 is, "Which method works the best?" Unfortunately, answering this question conclusively is challenging for several reasons. First, each study uses an idiosyncratic subset of the performance measures outlined above, and sometimes different measures identify different top-performing methods even within the same study. Second, each study uses different datasets, both for training their models and for evaluating them. These datasets may contain different levels and types of noise in the observed spectra, as well as differing rates and types of errors in the peptide assignments, making direct comparisons of performance measures across datasets challenging. Furthermore, this variety in datasets leads to a third problem: the initial question is not adequately formulated. A more precise framing of the question might be, "Which method works best when all methods are trained using exactly the same training data?" And as this reframing suggests, the answer may vary depending on the training data used. For example, a model with many millions of parameters may perform best when trained on many millions of peptide–spectrum matches (PSMs) and may perform poorly when trained on a relatively small dataset. Additionally, some methods, such as AdaNovo [14], which focuses on improving PTM prediction, may require corresponding data sets for successful training and representative evaluation. On the other hand, if we insist on asking, "Which method works best?" in an absolute sense, without controlling for the training data, we immediately encounter the difficulty of ensuring that the train and test sets do not overlap at the peptide level.

Not surprisingly, most studies claim that their method performs as well or better than the best-performing competing methods. In practice, each study typically compares to a handful of other methods. These

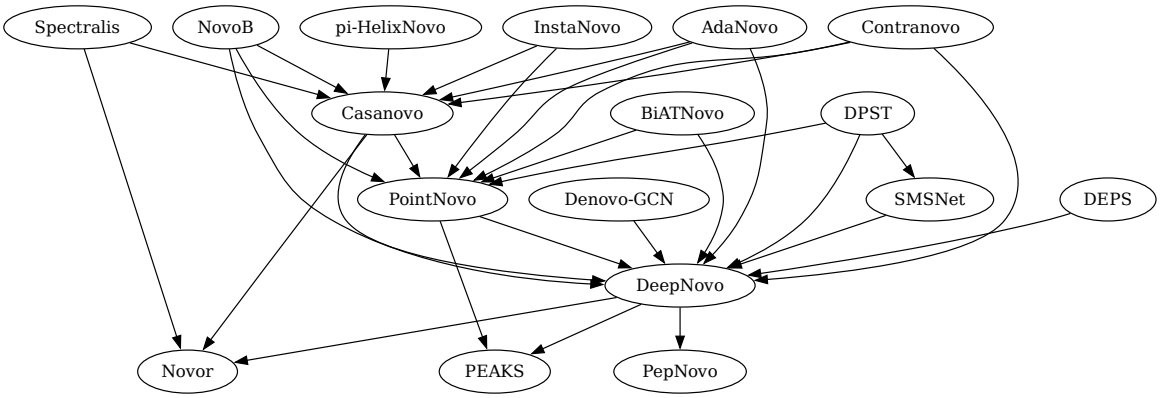

Figure 4: **Performance comparisons using the nine-species benchmark dataset.** In the graph, each arrow represents a performance comparison carried out using the high-resolution version of the nine-species benchmark dataset [1]. An arrow from method A to B implies that A outperformed B. In each case, the comparison was reported in the paper describing method A. The comparisons in the graph represent a variety of different performance measures, including the area under the precision–recall curve, area under the precision–coverage curve, peptide-level precision at 100% coverage (i.e., peptide-level accuracy), or the Position-BLEU score [8]. Note that some comparisons are excluded from the graph because they use the low-resolution version of the nine-species benchmark [4] or they only employed a subset of the species in the benchmark [10].

methods can be identified via the citation graph in Figure 1, although it is not uncommon for a study to cite more methods than it compares against.

If we focus only on papers reporting results using the nine-species benchmark described in the DeepNovo paper, then we can create a smaller graph in which an edge indicates that one method outperforms the other on this benchmark dataset (Figure 4). The field would clearly benefit from a systematic benchmarking study in which all models are trained on the same data and evaluated on independent test data with clearly defined metrics.

We are aware of two studies that have evaluated *de novo* sequencing tools on external data. First, Beslic *et al.* [82] compared the performance of Novor, pNovo3, DeepNovo, SMSNet, PointNovo, and Casanovo on the task of *de novo* sequencing for antibody discovery. To avoid biases due to different training datasets used, they first retrained all six tools on the MassIVE-KB human spectral library [56]. Evaluating the tools on human and mouse antibody data, the authors concluded that Casanovo and PointNovo show improved peptide recall across different enzymes and datasets compared with competing methods. Second, Tran *et al.* [83] have evaluated PEAKS, PointNovo, Casanovo, and GraphNovo on five datasets: human tryptic data, human non-tryptic data, *Arabidopsis thaliana* data, HLA class I data, and a simulated dataset generated using Prosit [47]. In contrast to the previous benchmarking effort, the models were not consistently retrained but rather used directly. Because all tools have originally been trained on human data, they achieved the strongest performance on the human test data as well. When evaluating on the *A. thaliana* data, however, the performances dropped substantially, suggesting some lack of generalizability to data that are dissimilar from the training data. Overall, Casanovo and GraphNovo achieved the best peptide-level performance across all evaluation datasets.

6 Post-translational modifications

An important challenge in both *de novo* and database search analysis of MS/MS data arises in the detection and localization of PTMs. Many of the methods in Table 1 can be configured to detect a variety of types

Training requirements	
PepNet	80 h using 8 cards of an NVIDIA A6000 GPU
Casanovo	4 RTX 2080 Ti GPUs
ContraNovo	8 A100 GPUs
GraphNovo	4 A100 GPUs
NovoB	24 h on one A100 GPU
AdaNovo	68 h on one A100 GPU
Transformer-DIA	8 GeForce RTX 2080 GPUs
Cascadia	100,000 spectra / hour on one A100 GPU
PointNovo	0.4 seconds for 16 spectra on one RTX 2080 Ti GPU
DEPS	one V100 GPU
Spectralis	36 h for 7,902,759 spectra on four A40 GPUs
Sequencing requirements	
InstaNovo+	2.4 h for 50,000 spectra on one RTX 3090 Ti GPU
pi-HelixNovo	40 m for 164,412 spectra on one V100 GPU
NovoB	400 spectra/s on one A100 GPU
AdaNovo	6 h on one A100 GPU
PointNovo	20 spectra/s on one RTX 2080 Ti GPU

Table 3: **Reported speed, memory, and hardware requirements.** Many methods do not provide details about hardware or running times and so are omitted from this table.

of PTMs; however, an important caveat is that these methods typically require labeled training data corresponding to each type of PTM to be recognized. In practice, many of the tools have been trained to detect a relatively small set of types of PTMs, as follows:

- oxidation of M: GraphNovo, Spectralis
- oxidation of M, phosphorylation of S: SMSNet
- oxidation of M, deamidation of N/Q: InstaNovo, NovoB, AdaNovo, DeepNovo, Denovo-GCN, DEPS, RANovo
- oxidation of M, deamidation of N/Q, phosporylation of S/T/Y: PointNovo
- oxidation of M, deamidation of N/Q, N-term carbamylation, N-term NH_3 loss, combination of N-term carbamylation and NH_3 loss: Casanovo, π-HelixNovo, Contranovo, Cascadia

The remaining methods either do not include any PTMs in their model or do not describe those PTMs in the associated paper.

7 Resource requirements

Resource requirements, including time, memory, and hardware, are critical characteristics of any *de novo* sequencing method, but these features are not always explicitly described in published papers. In general, we can distinguish between the resources required to train a model and resources required to use that trained model to carry out *de novo* sequencing (Table 3). All of these deep learning methods rely on GPUs for model training, with training times ranging from 24 to 80 hours, depending on the size of the model, the amount of training data, and available hardware. Some methods explicitly indicate that training or inference can be carried out using only CPUs [1, 6], though in practice this is likely to be slow.

8 Applications of deep learning *de novo* sequencing methods

Many *de novo* sequencing tools are too new to have been extensively utilized; however, we were able to identify applications of seven methods listed in Table 1. Among these, the oldest method, DeepNovo, is

Method	Year	Application	Ind
DeepNovo	2020	Detection of neoantigens [84]	
	2021	*De novo* sequencing in metaproteomics [85]	✓
	2021	Detection of neoantigens [86]	✓
	2021	Detection of junction peptides [87]	✓
	2021	Detection of shell proteins [88]	✓
	2023	Detection of neuropeptides [89]	✓
	2023	Detection of proteasome-generated spliced and non-spliced peptides [90]	✓
	2023	Antibody sequencing [91]	✓
	2023	Detection of short peptides [92]	✓
	2023	Noncanonical antigen detection [93]	✓
	2023	Detection of neoantigens [94]	
	2024	Antibody sequencing [95]	✓
	2024	Detection of neoantigens [96]	
	2024	Detection of venom proteins [97]	✓
DeepNovo DeepNovo-DIA	2022	Detection of antigen peptides in immunopeptidomics [98]	
DeepNovo pNovo 3	2021	Detection of bioactive peptides [99]	✓
DeepNovo-DIA	2023	Detection of small open reading frame-encoded peptides [100]	✓
SMSNet	2022	Detection of novel venom peptides [101]	
	2022	Detection of venom proteins [102]	
	2022	Detection of new candidate HLA ligands [103]	
	2024	*De novo* sequencing in metaproteomics [104]	✓
Casanovo	2023	Detection of giant genes in bacteria from metaproteomics data [105]	✓
Casanovo InstaNovo PointNovo	2024	Full antibody protein sequencing [106]	✓
PointNovo	2023	Noncanonical antigen detection [107]	✓
pNovo 3	2020	Detection of short peptides in urine samples [108]	✓
	2020	Detection of of novel sulfopeptides [109]	✓
	2022	Detection of neoantigens [110]	✓

Table 4: **Applications of deep learning *de novo* sequencing methods.** The final column ("Ind") indicates whether the application was published independently of the original authors of the work.

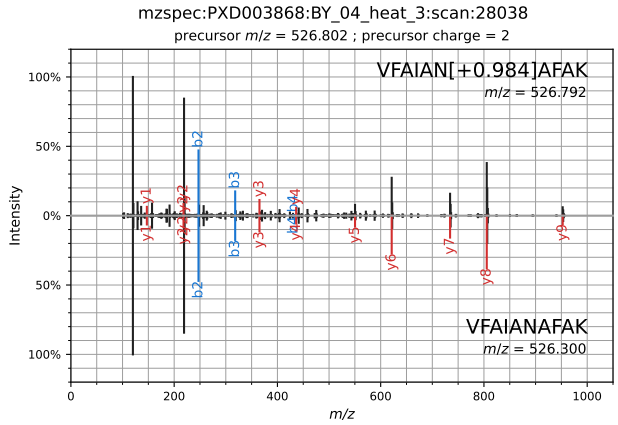

Figure 5: **Incorrect peptide labels can lead to a bias in the performance of *de novo* tools**. Peptide assignments for the MS/MS spectrum from the nine-species benchmark data with universal spectrum identifier [111] `mzspec:PXD003868:BY_04_heat_3:scan:28038`. This spectrum is annotated as peptide "VFAIAN[+0.984]AFAK" in the original version of this dataset [1] but was reannotated as "VFAIANAFAK" recently [69]. The difference in peptide assignments originates from the fact that the precursor *m/z* corresponds to the first isotopic peak rather than the monoisotopic peak. Because isotope errors were not accounted for during the original labeling process, an erroneous deamidation was applied (a mass difference of 1.003 Da for the first isotopic peak versus 0.984 Da for deamidation). The mirror plot was generated using spectrum_utils [112].

also the most widely used. Overall, we identified 27 studies that applied deep learning *de novo* sequencing methods, of which 16 employ DeepNovo (Table 4). The DeepNovo method, along with its successor method, PointNovo, has been incorporated into the commercial software PEAKS, and most of these 16 applications make use of that software. Among all 27 studies, the most common application is detection of neoantigens and noncanonical antigens [84, 86, 93, 94, 96, 98, 107, 110], followed by antibody sequencing [91, 95, 106], venomics [97, 101, 102], and metaproteomics [85, 104]. A variety of other studies use *de novo* sequencing tools to detect various classes of short or unexpected peptide sequences [87, 89, 90, 92, 99, 100, 103, 108, 109]. We also note that all seven of the tools have been used at least once in a study published independently of the original authors, suggesting that the software can be successfully used by others. Moving forward, as the quality of software tools in this domain continues to improve, the applications of *de novo* sequencing are likely to expand to additional domains.

9 Outstanding challenges

As alluded to above, a key challenge in this field is the proper evaluation of existing methods. An ideal performance evaluation protocol would involve comparing predictions from a *de novo* sequencing algorithm with a ground truth that accurately represents the peptide sequence generating every observed spectrum. In practice, of course, such an exhaustive ground truth labeling of mass spectrometry data is impossible to obtain. However, at least three possible alternative evaluation strategies exist.

The first solution is to use spectra that were generated from synthesized peptide sequences, such as those contained in the ProteomeTools database [58]. This type of data is ideal because the generating peptide for each observed spectrum can be confidently identified. However, the data itself are somewhat unrealistic, as the spectra do not come from a complex mixture and hence exhibit lower noise levels than those typical for natural biological samples. Nonetheless, data from synthetic peptides have been used to train multiple *de novo* sequencing methods [3, 5, 11] and have also been used for performance evaluation [83].

The second solution, which is the most widely used, involves using database search procedures to assign peptides to observed spectra and then treating those assignments as ground truth. Critical to the success of this approach is the availability of statistically rigorous methods for controlling the false discovery rate

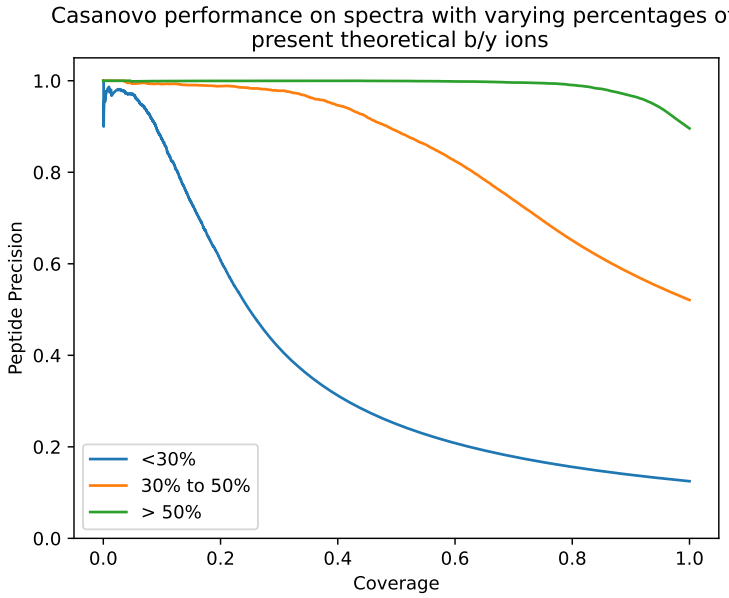

Figure 6: **Prediction performance improves when selecting for high-quality PSMs.** The figure plots peptide-level precision as a function of coverage for Casanovo (v4.1.0) applied to a series of test sets, with varying numbers of matched b- and y-ions.

(FDR) among a set of peptides detected via database search [113, 114]. Typically, datasets for training and validating *de novo* methods are generated using a 1% FDR threshold, applied at the PSM level. However, suboptimal settings during database searches can still lead to inaccurately labeled peptides. For instance, the nine-species benchmark data initially did not account for missassigned isotopic peaks [1], resulting in the incorrect identification of deamidated peptides from spectra for which the first isotopic peak rather than the mono-isotopic peak was used as the precursor m/z (Figure 5). This error was only identified during the development of the Casanovo tool six years later [69], while other *de novo* tools had already been trained on this partially incorrect data in the meantime. Such inaccuracies in training data propagate the limitations of the database search tools used to generate these labels, emphasizing the importance of using the latest advances in spectrum annotation to produce the highest quality training data possible.

Moreover, as *de novo* tools learn to approximate predictions of the database search tool used to generate the peptide labels, they inherit any limitations inherent to these tools. As discussed previously, a correct train/test split is crucial to prevent these tools from simply memorizing peptide sequences rather than learning to predict them accurately. Evaluating these tools on mass spectrometry data from heterogeneous sources, which have limited to no overlap in their proteomes, is one strategy to ensure robustness and generalizability.

Another potential drawback to using database search results to evaluate *de novo* methods is that the resulting performance estimates are unrealistically good. In practice, typically only around 25–50% of the given spectra in a dataset can be confidently identified using database search procedures [115]. *A priori*, we expect these identified spectra to have a greater proportion of diagnostic b- and y-ion peaks and an overall lower noise level than the unidentified spectra. Hence, when a *de novo* sequencing paper reports that a method correctly identifies, say, 60% of the spectra, this number typically represents the correct identification of only ∼30% of the full dataset. The rate of correct identifications that the method can achieve on the remaining 50% of the data is almost certainly much lower than 60%.

To demonstrate this effect, we carried out a simulation experiment to illustrate how the measured performance of a *de novo* sequencing algorithm can strongly depend on the selection of validation data. We evaluated a pre-trained Casanovo model on a series of datasets of varying quality, each containing 20,000 spectra. Each dataset was randomly sampled from the disjoint test set derived from MassIVE-KB [56] prior to Casanovo training in two steps: first, by selecting the subset of all PSMs with the desired proportion

of matched b- and y-ions; second, by randomly downsampling these PSMs to obtain sets of 20,000 spectra each. The results of this experiment (Figure 6) clearly demonstrate how a model's apparent performance is contingent on the quality of the data used for evaluation: the average precision changes from 0.99 on spectra with over 50% of the total spectrum intensity explained by b- and y-ions, to 0.84 when the percentage ranges from 30–50%, and to 0.37 when it is below 30%. This effect would likely be even more pronounced with models trained on datasets of varying quality.

The third method for evaluating *de novo* sequencing tools is to use statistical control of the FDR. This is the standard approach for evaluating database search algorithms: method A is deemed more powerful than method B if, at a fixed FDR threshold of, say, 1%, A detects more peptides than B from the same set of spectra. This approach avoids the need to focus solely on the "identifiable" subset of the observed spectra. However, a significant problem with this approach is that, currently, no one knows how to do it (although at least one recent study claims to have a solution [83]). Developing methods for *de novo* FDR control is one of the key outstanding challenges in the field.

Indeed, the availability of *de novo* FDR control procedures would have significant implications beyond the comparative evaluation of different sequencing methods. In practice, users of any *de novo* sequencing algorithm need to establish a threshold in the ranked list of predictions produced by the method. This can be achieved by selecting a score threshold corresponding to a target precision, as measured using a gold standard from the database search [5]. However, such thresholds are likely to be overly optimistic, for the reasons outlined above.

An additional complication in evaluating *de novo* sequencing is the presence of *chimeric* spectra, i.e., those generated by two or more peptides that co-fragment in the mass spectrometer. Even for current evaluation schemes, a chimeric spectrum generated by two peptides may be assigned the first peptide by the database search procedure and the other peptide by the *de novo* sequencing algorithm. This correct identification would be deemed incorrect according to the gold standard. Predicting chimeras in a *de novo* manner is challenging, and evaluating such predictions is even more complex.

Another significant complication arises from the fact that most tools only support a limited set of the most common PTMs, which are typically introduced during sample processing and of artificial origin [116]. To include novel PTMs and expand the amino acid alphabet, most *de novo* sequencing tools must be entirely retrained, incorporating additional data that includes these new PTMs. This is problematic because many biologically relevant PTMs occur infrequently, making it difficult to gather sufficient training data. Identifying peptides that contain multiple types of PTMs remains a formidable challenge for deep learning *de novo* sequencing tools.

Currently, most deep learning tools generate peptides in an autoregressive manner, predicting each amino acid sequentially. This approach poses several challenges: it limits the ability to correct early mistakes in the amino acid sequence, shows diverging behavior when discriminative peaks are absent (often for middle amino acids in longer peptides), requires heuristics to ensure that the predicted peptide matches the observed precursor m/z, and suffers from computational inefficiencies because autoregressive decoding cannot be parallelized. Although some tools like PepNet [5] and π-PrimeNovo [17] attempt to predict the entire peptide sequence at once, they can face limitations such as an inflexible maximum peptide length. Similarly, while the diffusion-based decoder in InstaNovo is an interesting conceptual contribution [11], it is only used as a postprocessor after an initial peptide sequence has been obtained via autoregressive decoding, and this postprocessor does not significantly improve the predictive performance. These approaches highlight the need for innovative decoding strategies that could significantly enhance the performance of *de novo* sequencing by better leveraging the information available in specific mass spectra.

Another challenge the field faces is deciding when to use database searching versus *de novo* sequencing methods, or perhaps some combination thereof. Principled methods for estimating what proportion of spectra in a given dataset are foreign would be helpful. We also need to better characterize the tradeoff inherent in increasing the size of the protein database, between potentially reducing the proportion of foreign spectra versus losing some detections due to (implicit) multiple testing correction. More generally, a principled method for FDR control that could take into account database search and *de novo* analysis of a single dataset would be very beneficial to the field.

A challenge faced by the entire field of deep learning, including *de novo* sequencing, lies in interpreting these models once they have been trained. Traditional *de novo* sequencing methods employed algorithms, such as dynamic programming with respect to a spectrum graph [41], that associate amino acids with pairs

of peaks with corresponding mass differences. Presumably, deep learning models are capable of learning these types of rules on the basis of the provided training data. However, explicitly revealing the logic behind any given prediction is non-trivial. A potentially significant step in this direction is the new π-xNovo model [19], which uses a multi-head attention mechanism that provides an after-the-fact explanation, in the form of an attention matrix, linking predicted amino acids to specific peaks in the spectrum.

Finally, an often overlooked aspect in the proper evaluation of *de novo* tools is the practical implementation of benchmarking, particularly when it involves retraining methods on the same data. To ensure optimal training conditions for each model, its training procedure may need to be adjusted for this particular data set, normally by re-running the hyperparameter search. Otherwise, the default hyperparameters proposed by the authors might turn out to be suboptimal, potentially leading to reduced performance and compromising the benchmarking results.

Although the field faces numerous challenges, none of them is obviously insurmountable. On the contrary, the rapid progress over the past seven years since the publication of the DeepNovo paper illustrates how quickly this field is moving forward. With novel machine learning strategies, growing collections of publicly available data, and improving mass spectrometry instrumentation, we can expect the use of *de novo* sequencing tools to become more common, enabling many types of analyses that have previously been challenging or impossible to carry out.

References

[1] Tran, N. H., Zhang, X., Xin, L., Shan, B., and Li, M. "De novo peptide sequencing by deep learning". *Proceedings of the National Academy of Sciences of the United States of America* 31 (2017), pp. 8247–8252.

[2] Tran, N. H., Qiao, R., Xin, L., Chen, X., Liu, C., Zhang, X., Shan, B., Ghodsi, A., and Li, M. "Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry". *Nature Methods* 16 (2019), pp. 63–66.

[3] Karunratanakul, K., Tang, H.-Y., Speicher, D. W., Chuangsuwanich, E., and Sriswasdi, S. "Uncovering Thousands of New Peptides with Sequence-Mask-Search Hybrid De Novo Peptide Sequencing Framework". *Molecular and Cellular Proteomics* 18 (2019), pp. 2478–2491.

[4] Liu, Z. and Zhao, C. "A residual network for de novo peptide sequencing with attention mechanism". *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE. 2020, pp. 1165–1170.

[5] Liu, K., Ye, Y., Li, S., and Tang, H. "Accurate de novo peptide sequencing using fully convolutional neural networks". *Nature Communications* 14.1 (2023), p. 7974.

[6] Yilmaz, M., Fondrie, W. E., Bittremieux, W., Oh, S., and Noble, W. S. "*De novo* mass spectrometry peptide sequencing with a transformer model". *Proceedings of the International Conference on Machine Learning*. 2022, pp. 25514–25522.

[7] Yang, Y., Hossain, Z., Asif, K., Pan, L., Rahman, S., and Stone, E. "DPST: de novo peptide sequencing with amino-acid-aware transformers". *arXiv preprint arXiv:2203.13132* (2022).

[8] Wu, S., Luan, Z., Fu, Z., Wang, Q., and Guo, T. "BiATNovo: A Self-Attention based Bidirectional Peptide Sequencing Method". *bioRxiv* (2023), pp. 2023–05.

[9] Jin, Z., Xu, S., Zhang, X., Ling, T., Dong, N., Ouyang, W., Gao, Z., Chang, C., and Sun, S. "ContraNovo: A Contrastive Learning Approach to Enhance De Novo Peptide Sequencing". *arXiv preprint arXiv:2312.11584* (2023).

[10] Mao, Z., Zhang, R., Xin, L., and Li, M. "Mitigating the missing fragmentation problem in de novo peptide sequencing with a two stage graph-based deep learning model". *Nature Machine Intelligence* 5 (2023).

[11] Eloff, K., Kalogeropoulos, K., Morell, O., Mabona, A., Jespersen, J. B., Williams, W., van Beljouw, S. P. B., Skwark, M., Laustsen, A. H., Brouns, S. J. J., et al. "De novo peptide sequencing with InstaNovo: Accurate, database-free peptide identification for large scale proteomics experiments". *bioRxiv* (2023), pp. 2023–08.

[12] Yang, T., Ling, T., Sun, B., Liang, Z., Xu, F., Huang, X., Xie, L., He, Y., Li, L., He, F., et al. "Introducing π-HelixNovo for practical large-scale de novo peptide sequencing". *Briefings in Bioinformatics* 25.2 (2024), bbae021.

[13] Lee, S. and Kim, H. "Bidirectional de novo peptide sequencing using a transformer model". *PLOS Computational Biology* 20.2 (2024), e1011892.

[14] Xia, J., Chen, S., Zhou, J., Lin, T., Du, W., and Li, S. Z. "AdaNovo: Adaptive De Novo Peptide Sequencing with Conditional Mutual Information". *arXiv:2043.07013v1* (2024).

[15] Ebrahimi, S. and Guo, X. "Transformer-based de novo peptide sequencing for data-independent acquisition mass spectrometry". *arXiv preprint arXiv:2402.11363* (2024).

[16] Sanders, J., Oh, S., and Noble, W. S. "A transformer model for *de novo* sequencing of data-independent acquisition mass spectrometry data". Manuscript in preparation.

[17] Zhang, X., Ling, T., Jin, Z., Xu, S., Gao, Z., Sun, B., Qiu, Z., Dong, N., Wang, G., Wang, G., et al. "π-PrimeNovo: An Accurate and Efficient Non-Autoregressive Deep Learning Model for De Novo Peptide Sequencing". *bioRxiv* (2024), pp. 2024–05.

[18] Petrovskiy, D. V., Nikolsky, K. S., Kulikova, L. I., Rudnev, V. R., Butkova, T. V., Malsagova, K. A., Kopylov, A. T., and Kaysheva, A. L. "PowerNovo: de novo peptide sequencing via tandem mass spectrometry using an ensemble of transformer and BERT models". *Scientific Reports* 14.1 (2024), p. 15000.

[19] Wang, Y., Liang, Z., Ling, T., Chang, C., Yang, T., Xie, L., and He, Y. "Transforming de novo peptide sequencing by explainable AI". *ResearchSquare* (2024). https://www.researchsquare.com/article/rs-4716013/v1.

[20] Qiao, R., Tran, N. H., Xin, L., Chen, X., Li, M., Shan, B., and Ghodsi, A. "Computationally instrument-resolution-independent de novo peptide sequencing for high-resolution devices". *Nature Machine Intelligence* 3 (2021), pp. 420–425.

[21] Ge, C., Lu, Y., Qu, J., Xie, L., Wang, F., Zhang, H., Kong, R., and Chang, S. "DePS: an improved deep learning model for de novo peptide sequencing". *arXiv preprint arXiv:2203.08820* (2022).

[22] Xu, X., Yang, C., He, Q., Shu, K., Xinpu, Y., Chen, Z., Zhu, Y., and Chen, T. "PGPointNovo: an efficient neural network-based tool for parallel de novo peptide sequencing". *Bioinformatics Advances* 3.1 (2023).

[23] Wu, R., Zhang, X., Wang, R., and Wang, H. "Denovo-GCN: De Novo Peptide Sequencing by Graph Convolutional Neural Networks". *Applied Sciences* 13.7 (2023).

[24] Wang, K., Zhu, M., Boulila, W., Driss, M., Gadekallu, T. R., Chen, C.-M., Wang, L., Kumari, S., and Yiu, S.-M. "SeqNovo: De Novo Peptide Sequencing Prediction in IoMT via Seq2Seq". *IEEE Journal of Biomedical and Health Informatics* (2023).

[25] Yang, H., Chi, H., Zeng, W., Zhou, W., and He, S. "pNovo 3: precise de novo peptide sequencing using a learning-to-rank framework". *Bioinformatics* 35.14 (2019), pp. i83–i90.

[26] Klaproth-Andrade, D., Hingerl, J., Bruns, Y., Smith, N. H., Träuble, J., Wilhelm, M., and Gagneur, J. "Deep learning-driven fragment ion series classification enables highly precise and sensitive de novo peptide sequencing". *Nature Communications* 15.1 (2024), p. 151.

[27] Jiang, Y., Rex, D. A. B., Schuster, D., Neely, B. A., Rosano, G. L., Volkmar, N., Momenzadeh, A., Peters-Clarke, T. M., Egbert, S. B., Kreimer, S., et al. "Comprehensive overview of bottom-up proteomics using mass spectrometry". *ACS Measurement Science Au* (2024).

[28] Hunt, D. F., Yates, III, J. R., Shabanowitz, J., Winston, S., and Hauer, C. R. "Protein sequencing by tandem mass spectrometry". *Proceedings of the National Academy of Sciences of the United States of America* 83 (1986), pp. 6233–6237.

[29] Steen, H. and Mann, M. "The ABC's (and XYZ's) of peptide sequencing". *Nature Reviews Molecular Cell Biology* 5 (2004), pp. 699–711.

[30] Sakurai, T., Matsuo, T., Matsuda, H., and Katakuse, I. "Paas 3: A computer program to determine probable sequence of peptides from mass spectrometric data". *Biomedical Mass Spectrometry* 11.8 (1984), pp. 396–399.

[31] Bartels, C. "Fast algorithm for peptide sequencing by mass spectroscopy". *Biomed. Environmental Mass Spectrometry* 19 (1990), pp. 363–368.

[32] Eng, J. K., McCormack, A. L., and Yates, III, J. R. "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database". *Journal of the American Society for Mass Spectrometry* 5 (1994), pp. 976–989.

[33] Nesvizhskii, A. I. "A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics". *Journal of Proteomics* 73.11 (2010), pp. 2092–2123.

[34] Keich, U., Kertesz-Farkas, A., and Noble, W. S. "Improved false discovery rate estimation procedure for shotgun proteomics". *Journal of Proteome Research* 14.8 (2015), pp. 3148–3161.

[35] Timmins-Schiffman, E., May, D. H., Mikan, M., Riffle, M., Frazar, C., Harvey, H., Noble, W. S., and Nunn, B. L. "Critical decisions in metaproteomics: achieving high confidence protein annotations in a sea of unknowns". *The ISME journal* 11.2 (2017), pp. 309–314.

[36] Shapiro, I. E. and Bassani-Sternberg, M. "The Impact of Immunopeptidomics: From Basic Research to Clinical Implementation". *Seminars in Immunology* 66 (2023), p. 101727.

[37] Georgiou, G., Ippolito, G. C., Beausang, J., Busse, C. E., Wardemann, H., and Quake, S. R. "The Promise and Challenge of High-Throughput Sequencing of the Antibody Repertoire". *Nature Biotechnology* 32.2 (2014), pp. 158–168.

[38] Warinner, C., Korzow Richter, K., and Collins, M. J. "Paleoproteomics". *Chemical Reviews* 122.16 (2022), pp. 13401–13446.

[39] Taylor, J. A. and Johnson, R. S. "Sequence database searches via *de novo* peptide sequencing by tandem mass spectrometry". *Rapid Communications in Mass Spectrometry* 11 (1997), pp. 1067–1075.

[40] Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., and Lajoie, G. "PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry". *Rapid Communications in Mass Spectrometry* 17.13 (2003), pp. 2337–2342.

[41] Dancik, V., Addona, T., Clauser, K., Vath, J., and Pevzner, P. "*De novo* peptide sequencing via tandem mass spectrometry". *Journal of Computational Biology* 6.3-4 (1999), pp. 327–342.

[42] Frank, A. and Pevzner, P. "PepNovo: de novo peptide sequencing via probabilistic network modeling". *Analytical Chemistry* 77 (2005), pp. 964–973.

[43] Fischer, B., Roth, V., Buhmann, J. M., Grossmann, J., Baginsky, S., Gruissem, W., Roos, F., and Widmayer, P. "A hidden Markov model for de novo peptide sequencing". *Advances in Neural Information Processing Systems* 17 (2005), pp. 457–464.

[44] Ma, B. "Novor: Real-Time Peptide de Novo Sequencing Software". *Journal of the American Society for Mass Spectrometry* 26 (2015), pp. 1885–1894.

[45] LeCun, Y., Bengio, Y., and Hinton, G. "Deep learning". *Nature* 521 (2015), pp. 436–444.

[46] Zhou, X., Zeng, W., Chi, H., Luo, C., Liu, C., Zhan, J., He, S. M., and Zhang, Z. "pDeep: predicting MS/MS spectra of peptides with deep learning". *Analytical Chemistry* 89.23 (2017), pp. 12690–12697.

[47] Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghed, B., Huhmer, A., Reimer, U., Ehrlich, H., Aiche, S., Kuster, B., and Wilhelm, M. "Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning". *Nature Methods* 16.6 (2019), p. 509.

[48] Tiwary, S., Levy, R., Gutenbrunner, P., Soto, F. S., Palaniappan, K. K., Deming, L., Berndl, M., Brant, A., Cimermancic, P., and Cox, J. "High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis". *Nature Methods* 16.6 (2019), pp. 519–525.

[49] Zohora, F. T., Rahman, M. Z., Tran, N. H., Xin, L., Shan, B., and Li, M. "DeepIso: a deep learning model for peptide feature detection from LC-MS map". *Scientific Reports* 9.1 (2019), p. 17168.

[50] Zohora, F. T., Rahman, M. Z., Tran, N. H., Xin, L., Shan, B., and Li, M. "Deep neural network for detecting arbitrary precision peptide features through attention based segmentation". *Scientific Reports* 11.1 (2021), p. 18249.

[51] Bittremieux, W., May, D. H., Bilmes, J., and Noble, W. S. "A learned embedding for efficient joint analysis of millions of mass spectra". *Nature Methods* 19.6 (2022), pp. 675–678.

[52] Bouwmeester, R., Martens, L., and Degroeve, S. "Comprehensive and empirical evaluation of machine learning algorithms for LC retention time prediction". *bioRxiv* (2018), p. 259168.

[53] Plante, P.-L., Francovic-Fontaine, É., May, J. C., McLean, J. A., Baker, E. S., Laviolette, F., Marchand, M., and Corbeil, J. "Predicting Ion Mobility Collision Cross-Sections Using a Deep Neural Network: DeepCCS". *Analytical Chemistry* 91.8 (2019), pp. 5191–5199.

[54] Meier, F., Köhler, N. D., Brunner, A.-D., Wanka, J.-M. H., Voytik, E., Strauss, M. T., Theis, F. J., and Mann, M. "Deep Learning the Collisional Cross Sections of the Peptide Universe from a Million Experimental Values". *Nature Communications* 12.1 (2021), p. 1185.

[55] Perez-Riverol, Y., Csordas, A., Bai, J., Bernal-Llinares, M., Hewapathirana, S., Kundu, D. J., Inuganti, A., Griss, J., Mayer, G., Eisenacher, M., Pérez, E., Uszkoreit, J., Pfeuffer, J., Sachsenberg, T., Yilmaz, S., Tiwary, S., Cox, J., Audain, E., Walzer, M., Jarnuczak, A. F., Ternent, T., Brazma, A., and Vizcaíno, J. A. "The PRIDE database and related tools and resources in 2019: improving support for quantification data". *Nucleic Acids Research* 47.D1 (2019), pp. D442–D450.

[56] Wang, M., Wang, J., Carver, J., Pullman, B. S., Cha, S. W., and Bandeira, N. "Assembling the Community-Scale Discoverable Human Proteome". *Cell Systems* 7 (4 2018), 412–421.e5.

[57] Deutsch, E. W., Bandeira, N., Sharma, V., Perez-Riverol, Y., Carver, J. J., Kundu, D. J., García-Seisdedos, D., Jarnuczak, A. F., Hewapathirana, S., Pullman, B. S., Wertz, J., Sun, Z., Kawano, S., Okuda, S., Watanabe, Y., Hermjakob, H., MacLean, B., MacCoss, M. J., Zhu, Y., Ishihama, Y., and Vizcaíno, J. A. "The ProteomeXchange Consortium in 2020: Enabling 'Big Data' Approaches in Proteomics". *Nucleic Acids Research* 48.D1 (2019), pp. D1145–D1152.

[58] Zolg, D. P., Wilhelm, M., Schnatbaum, K., Zerweck, J., Knaute, T., Delanghe, B., Bailey, D. J., Gessulat, S., Ehrlich, H., Weininger, M., Yu, P., Schlegl, J., Kramer, K., Schmidt, T., Kusebauch, U., Deutsch, E. W., Aebersold, R., Moritz, R. L., Wenschuh, H., Moehring, T., Aiche, S., Huhmer, A., Reimer, U., and Kuster, B. "Building ProteomeTools Based on a Complete Synthetic Human Proteome". *Nature Methods* 14.3 (2017), pp. 259–262.

[59] Vitorino, R., Guedes, S., Trindade, F., Correia, I., Moura, G., Carvalho, P., Santos, M. A., and Amado, F. "De novo sequencing of proteins by mass spectrometry". *Expert Review of Proteomics* 17.7-8 (2020), pp. 595–607.

[60] Ng, C. C. A., Zhou, Y., and Yao, Z.-P. "Algorithms for de-novo sequencing of peptides by tandem mass spectrometry: A review". *Analytica Chimica Acta* (2023), p. 341330.

[61] Hochreiter, S. and Schmidhuber, J. "Long short-term memory". *Neural computation* (1997).

[62] Ebrahimi, S. and Guo, X. "Deep Active Learning for De Novo Peptide Sequencing from Data-Independent-Acquisition Mass Spectrometry". *Proceedings of the International Conference on Machine Learning*. 2022.

[63] He, K., Zhang, X., Ren, S., and Sun, J. "Deep residual learning for image recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[64] Hu, J., Shen, L., and Sun, G. "Squeeze-and-excitation networks". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[65] Bai, S., Kolter, J. Z., and Koltun, V. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". *International Conference on Learning Representations*. 2018.

[66] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. "Attention Is All You Need". en. *Advances in Neural Information Processing Systems* 30 (2017).

[67] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". *Proceedings of the National Academy of Sciences of the United States of America* 118.15 (2021), e2016239118.

[68] Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. "Effective gene expression prediction from sequence by integrating long-range interactions". *Nature methods* 18.10 (2021), pp. 1196–1203.

[69] Yilmaz, M., Fondrie, W. E., Bittremieux, W., Nelson, R., Ananth, V., Oh, S., and Noble, W. S. "Sequence-to-sequence translation from mass spectra to peptides with a transformer model". *Nature Communications* (2024). In press.

[70] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T. "Do Transformers Really Perform Bad for Graph Representation?" *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 28877–28888.

[71] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. 2019, pp. 4171–4186.

[72] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 369–376.

[73] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. "PointNet: deep learning on point sets for 3D classification and segmentation". *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition*. 2016, pp. 652–660.

[74] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. "On the Variance of the Adaptive Learning Rate and Beyond". *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*. 2020.

[75] Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. "Lookahead optimizer: k steps forward, 1 step back". *Advances in neural information processing systems* 32 (2019).

[76] Yong, H., Huang, J., Hua, X., and Zhang, L. "Gradient centralization: A new optimization technique for deep neural networks". *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 635–652.

[77] Zhao, M., Zhong, S., Fu, X., Tang, B., and Pecht, M. "Deep residual shrinkage networks for fault diagnosis". *IEEE Transactions on Industrial Informatics* 16.7 (2019), pp. 4681–4690.

[78] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". *arXiv preprint arXiv:1406.1078* (2014).

[79] Chi, H., Chen, H., He, K., Wu, L., Yang, B., Sun, R.-X., Liu, J., Zeng, W.-F., Song, C.-Q., He, S.-M., et al. "pNovo+: de novo peptide sequencing using complementary HCD and ETD tandem mass spectra". *Journal of Proteome Research* 12.2 (2013), pp. 615–625.

[80] Joachims, T., Finley, T., and Yu, C.-N. J. "Cutting-plane training of structural SVMs". *Machine learning* 77 (2009), pp. 27–59.

[81] Wen, B. and Noble, W. S. "A multi-species benchmark for training and validating mass spectrometry proteomics machine learning models". *Scientific Data* (2024). 10.26434/chemrxiv-2024-z5b8m.

[82] Beslic, D., Tscheuschner, G., Renard, B. Y., Weller, M. G., and Muth, T. "Comprehensive evaluation of peptide de novo sequencing tools for monoclonal antibody assembly". *Briefings in Bioinoformatics* (2022). Advance online access.

[83] Tran, N. H., Qiao, R., Mao, Z., Pan, S., Zhang, Q., Li, W., Xin, L., Li, M., and Shan, B. "NovoBoard: a comprehensive framework for evaluating the false discovery rate and accuracy of de novo peptide sequencing". *bioRxiv* (2024), pp. 2024–04.

[84] Tran, N. H., Qiao, R., Xin, L., Chen, X., Shan, B., and Li, M. "Personalized deep learning of individual immunopeptidomes to identify neoantigens for cancer vaccines". *Nature Machine Intelligence* 2.12 (2020), pp. 764–771.

[85] Kleikamp, H. B., Pronk, M., Tugui, C., Silva, L. G. da, Abbas, B., Lin, Y. M., Loosdrecht, M. C. van, and Pabst, M. "Database-independent de novo metaproteomics of complex microbial communities". *Cell Systems* 12.5 (2021), pp. 375–383.

[86] Qi, Y. A., Maity, T. K., Cultraro, C. M., Misra, V., Zhang, X., Ade, C., Gao, S., Milewski, D., Nguyen, K. D., Ebrahimabadi, M. H., et al. "Proteogenomic analysis unveils the HLA class I-presented immunopeptidome in melanoma and EGFR-mutant lung adenocarcinoma". *Molecular & Cellular Proteomics* 20 (2021).

[87] He, C., Guo, J., Tian, W., and Wong, C. C. "Proteogenomics Integrating Novel Junction Peptide Identification Strategy Discovers Three Novel Protein Isoforms of Human NHSL1 and EEF1B2". *Journal of Proteome Research* 20.12 (2021), pp. 5294–5303.

[88] Sakalauskaite, J., Mackie, M., Taurozzi, A. J., Collins, M. J., Marin, F., and Demarchi, B. "The degradation of intracrystalline mollusc shell proteins: A proteomics study of Spondylus gaederopus". *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* 1869.12 (2021), p. 140718.

[89] Vu, N. Q., Yen, H.-C., Fields, L., Cao, W., and Li, L. "HyPep: An open-source software for identification and discovery of neuropeptides using sequence homology search". *Journal of proteome research* 22.2 (2023), pp. 420–431.

[90] Roetschke, H. P., Rodriguez-Hernandez, G., Cormican, J. A., Yang, X., Lynham, S., Mishto, M., and Liepe, J. "InvitroSPI and a large database of proteasome-generated spliced and non-spliced peptides". *Scientific Data* 10.1 (2023), p. 18.

[91] Peng, W., Boer, M. A. den, Tamara, S., Mokiem, N. J., Lans, S. P. van der, Bondt, A., Schulte, D., Haas, P.-J., Minnema, M. C., Rooijakkers, S. H., et al. "Direct Mass Spectrometry-Based Detection and Antibody Sequencing of Monoclonal Gammopathy of Undetermined Significance from Patient Serum: A Case Study". *Journal of Proteome Research* 22.9 (2023), pp. 3022–3028.

[92] Hollebrands, B., Hageman, J. A., Sande, J. W. van de, Albada, B., and Janssen, H.-G. "Improved LC–MS identification of short homologous peptides using sequence-specific retention time predictors". *Analytical and Bioanalytical Chemistry* 415.14 (2023), pp. 2715–2726.

[93] Bedran, G., Wang, T., Pankanin, D., Weke, K., Laird, A., Battail, C., Zanzotto, F. M., Pesquita, C., Axelson, H., Rajan, A., et al. "The immunopeptidome from a genomic perspective: Establishing immune-relevant regions for cancer vaccine design". *bioRxiv* (2022), pp. 2022–01.

[94] Li, M., Tran, N. H., Peng, C., Lei, Q., Xin, L., Lang, J., Zhang, Q., Li, W., Qiao, R., Qin, H., et al. "A complete mass spectrometry-based immunopeptidomics pipeline for neoantigen identification and validation" (2023).

[95] Peng, W., Giesbers, K. C., Šiborová, M., Beugelink, J. W., Pronker, M. F., Schulte, D., Hilkens, J., Janssen, B. J., Strijbis, K., and Snijder, J. "Reverse-engineering the anti-MUC1 antibody 139H2 by mass spectrometry–based de novo sequencing". *Life Science Alliance* 7.6 (2024).

[96] Tran, N. H. and Li, M. "Predicting immunogenicity by modeling the positive and negative selection of CD8+ T cells in individual patients". *bioRxiv* (2022), pp. 2022–07.

[97] Nolasco, M., Mariano, D. O., Pimenta, D. C., Freitas, H. F. de, Rocha Pita, S. S. da, and Branco, A. "Oligopeptides analysis in spiderhawk's venom (Pepsis decorata Perty, 1833, Hymenoptera: Pompilidae)". *Peptide Science* (2024), e24347.

[98] Xin, L., Qiao, R., Chen, X., Tran, H., Pan, S., Rabinoviz, S., Bian, H., He, X., Morse, B., Shan, B., et al. "A streamlined platform for analyzing tera-scale DDA and DIA mass spectrometry data enables highly sensitive immunopeptidomics". *Nature Communications* 13.1 (2022), p. 3108.

[99] Cerrato, A., Aita, S. E., Cavaliere, C., Laganà, A., Montone, C. M., Piovesana, S., Chiozzi, R. Z., and Capriotti, A. L. "Comprehensive identification of native medium-sized and short bioactive peptides in sea bass muscle". *Food Chemistry* 343 (2021), p. 128443.

[100] Fan, S.-M., Li, Z.-Q., Zhang, S.-Z., Chen, L.-Y., Wei, X.-Y., Liang, J., Zhao, X.-Q., and Su, C. "Multi-integrated approach for unraveling small open reading frames potentially associated with secondary metabolism in Streptomyces". *Msystems* 8.5 (2023), e00245–23.

[101] Choksawangkarn, W., Sriswasdi, S., Kalpongnukul, N., Wongkongkathep, P., Saethang, T., Chanhome, L., Laoungbua, P., Khow, O., Sumontha, M., Chaiyabutr, N., et al. "Combined proteomic strategies for in-depth venomic analysis of the beaked sea snake (Hydrophis schistosus) from Songkhla Lake, Thailand". *Journal of Proteomics* 259 (2022), p. 104559.

[102] Saethang, T., Somparn, P., Payungporn, S., Sriswasdi, S., Yee, K. T., Hodge, K., Knepper, M. A., Chanhome, L., Khow, O., Chaiyabutr, N., et al. "Identification of Daboia siamensis venome using integrated multi-omics data". *Scientific reports* 12.1 (2022), p. 13140.

[103] Sricharoensuk, C., Boonchalermvichien, T., Muanwien, P., Somparn, P., Pisitkun, T., and Sriswasdi, S. "Unsupervised mining of HLA-I peptidomes reveals new binding motifs and potential false positives in the community database". *Frontiers in Immunology* 13 (2022), p. 847756.

[104] Kleikamp, H. B., Palacios, P. A., Kofoed, M. V., Papacharalampos, G., Bentien, A., and Nielsen, J. L. "The Selenoproteome as a Dynamic Response Mechanism to Oxidative Stress in Hydrogenotrophic Methanogenic Communities". *Environmental Science & Technology* (2024).

[105] West-Roberts, J. A., Valentin Alvarado, L. E., Mullen, S., Sachdeva, R., Smith, J., Hug, L. A., Gregoire, D., Liu, W., Lin, T.-Y., Husain, G., et al. "Giant genes are rare but implicated in cell wall degradation by predatory bacteria". *bioRxiv* (2023), pp. 2023–11.

[106] Tang, D., Gueto-Tettay, C., Hjortswang, E., Ströbaek, J., Ekström, S., Happonen, L., Malmström, L., and Malmström, J. "Multimodal Mass Spectrometry Identifies a Conserved Protective Epitope in S. pyogenes Streptolysin O". *Analytical Chemistry* (2023).

[107] Bedran, G., Gasser, H.-C., Weke, K., Wang, T., Bedran, D., Laird, A., Battail, C., Zanzotto, F. M., Pesquita, C., Axelson, H., et al. "The Immunopeptidome from a Genomic Perspective: Establishing the Noncanonical Landscape of MHC Class I–Associated Peptides". *Cancer immunology research* 11.6 (2023), pp. 747–762.

[108] Cerrato, A., Aita, S. E., Capriotti, A. L., Cavaliere, C., Montone, C. M., Laganà, A., and Piovesana, S. "A new opening for the tricky untargeted investigation of natural and modified short peptides". *Talanta* 219 (2020), p. 121262.

[109] Capriotti, A. L., Cerrato, A., Laganà, A., Montone, C. M., Piovesana, S., Zenezini Chiozzi, R., and Cavaliere, C. "Development of a sample-preparation workflow for Sulfopeptide enrichment: from target analysis to challenges in shotgun Sulfoproteomics". *Analytical Chemistry* 92.11 (2020), pp. 7964–7971.

[110] Xiang, H., Zhang, L., Bu, F., Guan, X., Chen, L., Zhang, H., Zhao, Y., Chen, H., Zhang, W., Li, Y., et al. "A novel proteogenomic integration strategy expands the breadth of neo-epitope sources". *Cancers* 14.12 (2022), p. 3016.

[111] Deutsch, E. W., Perez-Riverol, Y., Carver, J., Kawano, S., Mendoza, L., Van Den Bossche, T., Gabriels, R., Binz, P.-A., Pullman, B., Sun, Z., Shofstahl, J., Bittremieux, W., Mak, T., Klein, J., Zhu, Y., Lam, H., Vizcaino, J. A., and Bandeira, N. "Universal Spectrum Identifier for Mass Spectra". *Nature Methods* 18 (2021), pp. 768–770.

[112] Bittremieux, W., Levitsky, L., Pilz, M., Sachsenberg, T., Huber, F., Wang, M., and Dorrestein, P. C. "Unified and Standardized Mass Spectrometry Data Processing in Python Using spectrum_utils". *Journal of Proteome Research* 22.2 (2023), pp. 625–631.

[113] Elias, J. E. and Gygi, S. P. "Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry". *Nature Methods* 4.3 (2007), pp. 207–214.

[114] Lin, A., See, D., Fondrie, W. E., Keich, U., and Noble, W. S. "Target-decoy false discovery rate estimation using Crema". *Proteomics* (2023), p. 2300084.

[115] Griss, J., Perez-Riverol, Y., Lewis, S., Tabb, D. L., Dianes, J. A., Del-Toro, N., Rurik, M., Walzer, M., Kohlbacher, O., Hermjakob, H., Wang, R., and Vizcaíno, J. A. "Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets". *Nature Methods* 13.8 (2016), pp. 651–656.

[116] Bittremieux, W., Tabb, D. L., Impens, F., Staes, A., Timmerman, E., Martens, L., and Laukens, K. "Quality control in mass spectrometry-based proteomics". *Mass Spectrometry Reviews* 37.5 (2018), pp. 697–711.

Conflicts of interest The authors declare that they have no conflicts of interest.