Predicting the Spectrum Quality and Digestive Enzyme for Shotgun Proteomics

by

Soroosh Gholamizoj

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In proteomics, database search programs are routinely used for peptide identification from tandem mass spectrometry data. However, many low-quality spectra cannot be interpreted by any programs. Meanwhile, certain high-quality spectra may not be identified due to incompleteness of the database, failure of the software, or sub-optimal search parameters. Thus, spectrum quality assessment tools are helpful programs that can eliminate poor-quality spectra before the database search and highlight the high-quality spectra that are not identified in the initial search. These spectra may be valuable candidates for further analyses.

We propose SPEQ: a spectrum quality assessment tool that uses a deep neural network to classify spectra into high-quality, which are worthy candidates for interpretation, and low-quality, which lack sufficient information for identification. SPEQ was compared with a few other prediction models and demonstrated improved prediction accuracy.

Furthermore, we propose a statistical model to automatically detect the enzyme used for digestion in a proteomics experiment, by analyzing the distribution of amino acids in peptides *de novo* sequenced with a nonspecific enzyme setting. Results demonstrate that this algorithm can accurately identify correct enzymes.

Acknowledgements

I would like to sincerely thank my supervisor, Professor Bin Ma, for his constant support and guidance throughout my time at the University of Waterloo.

Dedication

This is dedicated to my lovely parents, Narges and Shahryar,
and my dear brother, Kiavash.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Proteomics is the study of the whole proteome in a systematic large-scale manner. It involves analyzing the structure and functions of the entire set of proteins of an organism, tissue, or cell produced under a specific and defined set of conditions [40]. Proteins are functional elements of biomolecules and macromolecules in any living organism. These fundamental components comprise one or more long sequences of amino acids built based on templates of RNA and DNAs. Additional to identifying possible proteins [17], proteomics evaluates quantities [39], localization, post-translational modifications, isoforms, roles, and dynamics of all proteins present in a sample. Technologies and discoveries in proteomics are mainly directed towards systems biology approaches for disease diagnostics and cancer research [21], drug discovery [7], and identifying critical diagnostic and prognostic biomarkers [34].

One of the main analytical approaches in proteomics is mass spectrometry (MS) [2], which due to its extreme sensitivity, is mostly applied to qualitative and quantitative problems. This instrumental technique is based on the ionization and fragmentation of proteomic samples, by moving the charged particles through an electric or magnetic field to separate and quantify them according to their mass-to-charge ratio (m/z). Since each molecule has a unique fragmentation pattern, its structural information can be determined by studying the resulting ion fragmentation patterns.

Accordingly, in bottom-up (shotgun) proteomics studies, tandem mass spectrometry (MS/MS or MS2) is the main approach used for protein and peptide identification [23]. In this technique, high-performance liquid chromatography is coupled with tandem mass spectrometry (LC-MS/MS) to identify peptides from complex mixtures of proteins [3]. Here, different peptides are eluted at different retention times (RTs) from the LC column. Thereby, proteins are first digested into smaller peptides using proteases, such as Trypsin, and then the resulting complex mixture of peptides is subject to LC-MS/MS analysis. From each of these analyses, a mass spectrum is generated, which is a recording of the signal *intensity* of the ion at each mass-to-charge (m/z) ratio at a certain retention time (RT). Identifying peptides, and proteins eventually, is completed by either matching the MS/MS spectra to theoretical spectra generated from protein databases, or by deriving the peptide sequence directly from the experimental spectra (*de novo* sequencing). Figure 1 (The mass-spectrometry/proteomic experiment) in the work by Steen and Mann [50] properly illustrates this workflow in detail.

Each of these experiments may produce thousands of MS/MS spectra, each supposedly corresponding to a peptide. In the most common approach for protein identification, a database search engine, such as MS-GF+ [28], Comet [14], MaxQuant [10], PEAKS DB [55], Mascot [42], or XTandem [11] is used to identify the peptide for each spectrum. Here, experimental MS/MS spectra produced using the mass spectrometer are compared with theoretical spectra predicted from peptides in a protein database. Other approaches can be used to interpret the experimentally produced spectra, such as *de novo* sequencing, where identification is made without a database and instead by interpreting the peaks of an experimental spectrum. Several programs, such as PEAKS [31], Novor [30], PepNovo [16], and pNovo 3 [54] have been developed for *de novo* sequencing.

1.2 Protein Identification Problem

In a protein identification process, there is a trade-off between the quality of the search result (number of the true identifications) and the speed of the analysis. The use of some specific search parameters such as post-translational modifications, which can increase the number of identifications, is limited during search analyses in order to save time. In addition, other limitations such as the incompleteness of databases and search engines being not perfect may result in unidentified spectra which are in fact identifiable.

Due to these limitations as well as other issues such as low spectrum quality, the aforementioned database search analysis may cause false identifications, including false-positive and false-negative. With much research, the false positives can now be reliably controlled by the false-discovery rate (FDR). Usually, a target-decoy method [12, 35] is used to establish a score threshold. Only the peptide-spectrum matches (PSMs) with scores above the threshold are reported by the analysis. An FDR of 1% is often used to ensure that there are (on average) at most 1% false-positive identifications in the reported PSMs.

However, the false negatives of the database search are rarely studied. Typically, 50% or more of the MS/MS spectra cannot be confidently identified by database search. Many of these unidentified spectra are due to poor spectrum quality, e.g., the spectrum does not contain enough information for any meaningful interpretation. But an unknown portion of these unidentified spectra is actually high-quality spectra. Their missing is solely because of the limitation of the data analysis, such as inadequate software, sub-optimal search parameters, unspecified post-translational modifications (PTMs), and incomplete sequence databases [37]. These spectra can be regarded as the false negatives of the data analysis. Though, it is worth mentioning that labelling unidentified high-quality spectra as false negatives is due to the incompleteness of the current identification process. Therefore, since only a subset of all high-quality samples can be identified, we are forced to use labels that are assigned by imperfect identification tools.

1.3 Usefulness of Spectrum Quality Assessment

A spectrum quality score that labels the high-quality spectra would be useful for peptide identification analyses, in particular in dealing with the false-negative spectra.

Firstly, the amount of the unidentified high-quality spectra may potentially provide a proxy to estimate the level of false negatives. If proved to be true by future research, this approach can be attractive as there is no established method in proteomics to estimate the false-negative rate of a data analysis experiment. Certain statistical approaches such as PeptideProphet [26] may be used to estimate the false negatives caused by the correctly-assigned but low-scoring PSMs. However, the ones caused by the unassigned or wrongly-assigned spectra remain unknown.

Secondly, a small portion (such as 1%) of the uninterpreted spectra with the highest quality score can be analyzed by a more time-consuming method or by a human expert manually to troubleshoot for the reasons for their missing. The identified reasons can be used by a data analyst to adjust the search strategy, by a tool developed to improve the software, or by a lab scientist to improve the MS experiment. A similar idea called Preview was described earlier by Kil et al. [27], where a subset of spectra are searched to determine the best search parameters before the full search. The spectrum quality score would be helpful here to select the subset.

Thirdly, when the quality score function is sufficiently developed to have a nearly-perfect accuracy, one can safely discard the low-quality spectra at the very beginning of the data analysis. This should both improve the speed of the downstream data analysis, and reduce the false positives caused by the low-quality spectra. In Chapter 2, we discuss this issue and elaborate our proposed solution, SPEQ.

1.4 Automated Determination of the Digestive Enzyme

Another way to improve the protein identification process and prevent the wrong detection is to reduce the probability of conducting experiments with inappropriate search parameters. Currently, in order to perform an effective bottom-up proteomics data analysis, all proteomics software requires a human user to set several important parameters, including the digestive enzyme (proteases) used in sample preparation, post-translational modifications, and mass error tolerance. Very often, the user may not know the right parameters to use.

Automatically determining these parameters from the mass spectrometry data could reduce potential wrong parameters and facilitate automated and streamlined data analysis. Chapter 3 presents our effort to take one step towards this goal, by introducing an algorithm to automatically detect the digestive enzyme from the MS data.

1.5 Overview and Contributions

In this thesis, we present two works. In Chapter 2, we introduce SPEQ (Spectrum Quality), a new method to predict the spectrum quality by using a Deep Neural Network (DNN) model. This open-source tool is freely available at `https://github.com/sor8sh/SPEQ`. A manuscript of this research has been published in Bioinformatics journal [19]. Chapter 2 is organized as follows:

- Section 2.1 presents a literature review on related works in peptide quality assessment. Afterward, an overview of the tools used in this work is given.
- In Section 2.2, we provide the data and materials used in this work. Then we explain the data preprocessing steps.

- In Section 2.3, we present our proposed method, SPEQ, a quality assessment tool for peptide tandem mass spectra that uses deep learning for automatic feature detection.
- In Section 2.4, we discuss the method's accuracy and show the results. Then the applications of such tools are examined in real use case examples, and we briefly discuss the future work.

Then, in Chapter 3, we present a machine-learning algorithm to automatically detect the digestive enzyme for a proteomics experiment by studying amino acids distribution in *de novo* sequenced peptides. The proposed method, data used in our experiments, and the results are also described in this chapter. This work is presented as a poster at the 70th ASMS Conference on Mass Spectrometry and Allied Topics [18]. Finally, Chapter 4 concludes the thesis by reviewing what is achieved in this work, and what were the challenges.

Chapter 2

Quality Assessment of Peptide Tandem Mass Spectra with Deep Learning

2.1 Background

In this Section, first, we review existing software tools related to the topic discussed in this thesis. Then, we discuss the drawbacks of the previous methods, how to improve them, and where the initial idea of the proposed method came from. In the end, we give an overview of the *de novo* sequencing tool and MS/MS search engines used in this work.

2.1.1 Related Work

In recent years, several related quality assessment tools have been developed. Bern et al. [4] used a set of handcrafted features and a support vector machine (SVM) to conduct spectral quality assessments. Salmi et al. [45] combined previous work with more handcrafted features and used a decision tree and random forest to conduct classification. Meanwhile, Flikka et al. [15] utilized multiple machine-learning classifiers to classify spectra based on 17 manually extracted features. Similarly, Nesvizhskii et al. [37] used a linear discriminant function to combine 15 scoring features selected by a human expert into one discriminant score. Na and Paek [36] proposed a new score function based on Cumulative Intensity Normalization to filter spectra based on their score. To describe the quality of tandem mass spectra, Wu et al. [53] first proposed a method that mapped each tandem spectrum into a feature vector before using fisher linear discriminant analysis to construct the classifier. More recently, Ma et al.[32] evaluated spectral quality via sequence tagging.

Handcrafted features played a primary role in these studies, which subsequently combined them with other classification methods. Thus, the challenge was to find the most optimal set of spectral features that could separate as many spectra containing useful information from noisy spectra as possible. Moreover, none of these software tools have been actively maintained, causing them either not anymore available or fail to work on today's computer systems or for the data produced with today's mass spectrometers.

Figure 2.1 illustrates how the spectra of different qualities can be classified. These scans are manually selected as examples to highlight some of the differences between low and high-quality spectra but may not represent all the differences. In these examples, if a significant peak was defined as a peak with a relative intensity of around 5% or higher, as seen in Figure 2.1, a high-quality spectrum consists of many peaks, and the number of significant peaks in that spectrum was relatively high compared with a low-quality spectrum. Moreover, the m/z differences between significant peaks in a high-quality spectrum encode meaningful information in terms of the mass of different amino acid residues. In contrast, a low-quality spectrum has fewer peaks, and the significant peaks are sparse. These observable differences can be combined with other proteomic features to build a program for spectrum qualification. However, handcrafting all these features is tedious, particularly because different types of mass spectrometers may require different features.

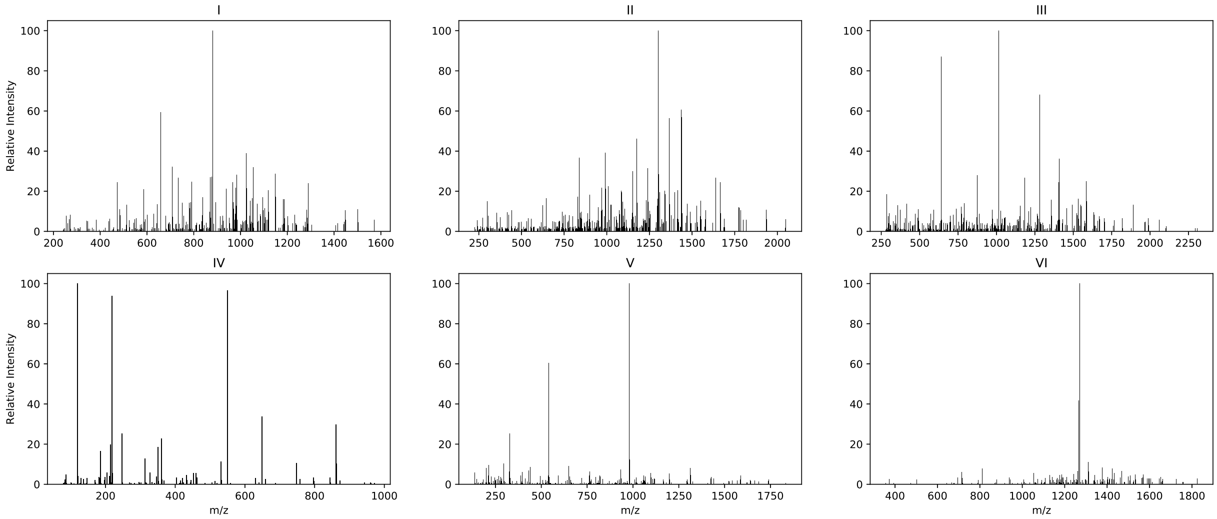


Figure 2.1: The first row (I, II, and III) shows three scans that were assigned confidently by MS-GF+ (≤1% FDR). In the second row (IV, V, and VI), three scans that were not assigned confidently by MS-GF+ are shown. Scans in the first row are considered high-quality spectra, while the bottom row contains low-quality spectra.

Compared to earlier machine learning-based methods, DNN does not require the features to be handcrafted. Instead, the input of the new model is just the whole spectrum, and the model's training will automatically extract beneficial features from the training data. Therefore, in contrast to previous related works, the extracted features differ for each type of data set and are related to specific characteristics, such as the instrument or experimental methods used to generate the data set. In addition, as one of the main characteristics of the deep learning method, SPEQ benefits from the availability of large proteome training data sets and offers better performance with relatively less development time compared with other methods. The proposed method's performance and usefulness were tested in different scenarios to demonstrate that:

1. SPEQ has a better prediction accuracy than the other models tested.
2. Most of the high-quality but unidentified spectra have a confident *de novo* sequence

tag, suggesting they are indeed spectra of peptides.

3. A small portion of the unidentified spectra with the highest quality score can be examined to troubleshoot a data analysis, leading to the identification of more spectra.

2.1.2 Peptide Identification Tools

Our prediction tool works on the spectra directly and does not require the peptides of these spectra to be identified. However, peptide identification tools were used for benchmarking purposes. In this work, we used two MS/MS search engines, MS-GF+ [28] and Comet [14], and one *de novo* sequencing tool, Novor [30], for protein identification.

First, we used MS-GF+ (also known as MSGF+ or MSGFPlus) to build the training data sets, by identifying peptides and assigning a label (high-quality/good or low-quality/bad) to each spectrum. The model was then trained based on the labels produced in this step. Next, Comet was used as an additional database search engine for identification validation. Spectra that were initially labelled as low-quality, meaning that in the first round of database search with MS-GF+, no peptides were confidently assigned to them, were searched again with Comet. In the end, Novor was used to *de novo* sequence spectra that were unidentified by both MS/MS search engines, where we were looking for spectra with a confident *de novo* sequence tag.

MS-GF+ is an MS/MS database search tool, freely available at `https://github.com/MSGFPlus/msgfplus/releases`. It performs peptide identification, on a spectrum input file (`*.mzML, *.mzXML, *.mgf, *.ms2, *.pkl, or *_dta.txt`), by scoring MS/MS spectra against peptides derived from a protein sequence database (`*.fasta, *.fa, or *.faa`). The result is a file (`*.mzid`) containing identified spectra with their information, including spectrum ID, scan number, fragmentation method, precursor ion, isotope error, precursor ion error (ppm), charge state, and identified peptide and protein(s). In addition, for every PSM, MS-GF+ reports multiple scores, including *de novo* score, spectral E-Value, database level E-Value, PSM-level Q-Value, and Peptide-level Q-Value. In this work, the database level E-value is used for FDR calculation. Detailed documentation on the input parameters and output fields is provided at https://msgfplus.github.io/msgfplus/MSGFPlus.html.

Comet is the other MS/MS database search tool used in this thesis, which can be accessed from https://uwpr.github.io/Comet/. It takes in one or more input spectral files (`*.mzXML, *.mzML, *.mgf, or *.ms2/cms2`) and a "`comet.params`" file, which includes the search parameters. A thorough definition of Comet search parameters can be found at https://uwpr.github.io/Comet/parameters/parameters_202102/.

Novor is a real-time *de novo* peptide sequencing tool, which derives peptide sequences purely from an MS/MS spectrum data file without prior knowledge of the peptide sequence. It is available upon request (please contact qliu@rapidnovor.com or bin.ma@uwaterloo.ca). To use Novor, a "`params.txt`" file is needed, which includes necessary parameters such as fragmentation method (CID or HCD), mass analyzer (Trap, TOF, or FT), precursor and fragment ion error tolerance, and a list of fixed and variable modifications. A newer version of Novor is freely available to use at https://novor.cloud. It is a free cloud-based *de novo* sequencing and protein identification web application.

2.2 Data and Materials

In this Section, we describe the data sets that are used in our experiments, including the data preprocessing steps and the testing procedure. Subsequently, we explain the steps needed to take in order to create a labelled training data set from a raw MS/MS file.

2.2.1 Testing Data

We used four data sets generated using different high-resolution instruments in this thesis. The primary source of MS data is ProteomeXchange [52], which is a globally coordinated proteomics data submission and dissemination. The mentioned data sets (2.2.1

and 2.2.1) can be retrieved by searching their project identifier on ProteomeXchange (http://proteomecentral.proteomexchange.org/cgi/GetDataset). Furthermore, the entire protein sequence information of the species examined in this thesis can be accessed from UniProt (https://www.uniprot.org/)

Q-TOF Data Set

This data set was previously used by Flikka et al. [15] for the development of their SpectrumQuality tool and was downloaded from http://services.cbu.uib.no/software/spectrumquality. This data set (Q-TOF N-terminal in the original paper) contains 10,055 MS/MS spectra from extreme amino-terminal peptides of proteins, measured with a quadrupole time-of-flight (Q-TOF) mass spectrometer. Mascot [42] was used to search the human proteins IPI database and IPI-derived N-terminally truncated databases for the peptide identification. Spectra with a score equal to or above the Mascot identity threshold, when the confidence level was set to 95%, were considered as “good”, while all other spectra were labelled “bad”. The same spectrum quality labels made by Flikka et al. [15] were kept and used in the present thesis. This gives 1,683 positive and 8,372 negative spectra in this data set.

Orbitrap Human Data Set

This proteome data set was first provided by Bruderer et al. [6] and is available with identifier PXD005573 (`Fig4_HeLa-1m_DDA_R01_T0.raw`) at the ProteomeXchange repository. It is a data-dependent acquisition data set obtained on a Q Exactive HF instrument using a HeLa lysate. ProteoWizard [8] was used to centroid each profile spectrum before converting the data set from a raw mass spectrometer output file to an XML file.

Two different database search programs, MS-GF+ and Comet, were used to search the UniProt Homo sapiens proteome (UP000005640) for the peptide identification. The search parameters are the following:

- Precursor tolerance = 20ppm
- Fixed PTM = Carbamidomethyl on C
- Variable PTM = Oxidation on M, Acetylation at protein N-term, and Deamidation on N and Q
- Enzyme = Semi-tryptic

A decoy database generated with the de-Bruijn method [35] was also searched together to determine the FDR. A spectrum is considered high-quality if a database peptide can be identified with less than 1% FDR by either MS-GF+ or Comet. This gives 95,646 positive and 123,702 negative spectra in this data set.

NIST Data Set

This data set consists of a Homo sapiens hair peptide spectral library [51] and was downloaded from https://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:cdownload. It is created from data generated by an Orbitrap Fusion Lumos instrument. A total of 6,280 spectra with 2,240 unique peptide sequences are available in this data set. Using MS-GF+, these spectra were then searched against the FASTA file provided along with the spectral library (with 20,183 proteins in it). The search parameters are the same as used in the Orbitrap human dataset. Spectra with a peptide identification with less than 1% FDR were labelled as high-quality. This gives 3,932 positive and 2,348 negative spectra in this data set.

Orbitrap Mouse Data Set

The last data set used in this study was first published by McDonagh et al. [33] and is available with identifier PXD001054 `(BMD_2013_07_31_Gastro_Aged_2.mgf)` at the ProteomeXchange repository. Similar to the second data set, this data set is also obtained on a Q Exactive instrument. However, the proteins in this data set are from the Mus musculus (Mouse) species. The data set has 22,686 spectra, which were searched against the UniProt Mus musculus proteome (UP000000589) with MS-GF+. The search parameters are the same as used in the Orbitrap human data, except that the precursor tolerance is set to 10ppm to match the original publication [33]. Spectra with a peptide identification with less than 1% FDR were labelled as high-quality. This gives 3,648 positive and 19,038 negative spectra in this data set.

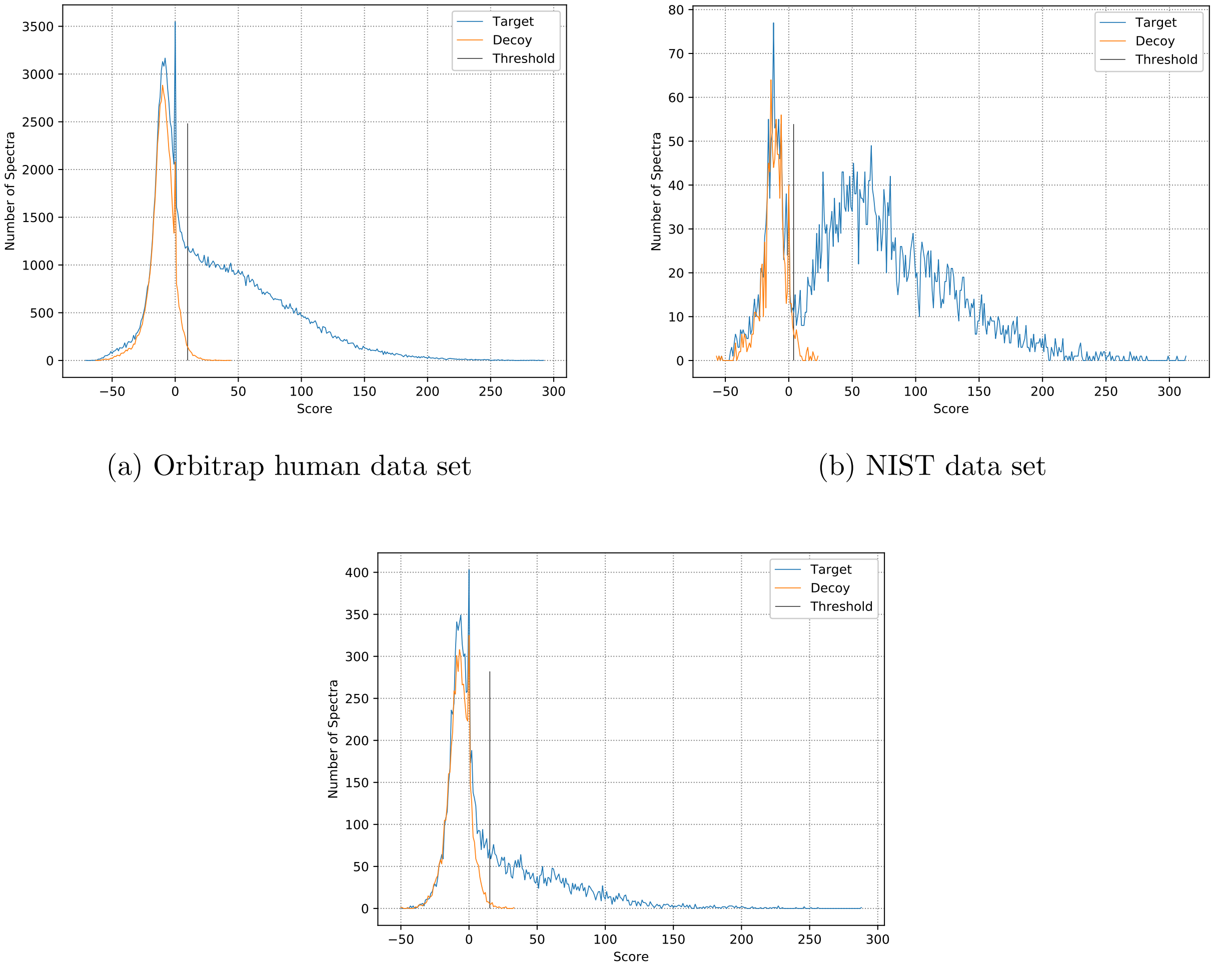
2.2.2 Data Processing

In this section, we explain the necessary steps for searching an MS file with a database search tool. Many MS data sets that are available in ProteomeXchange are only published in "`.raw`" format, which is not supported by most MS/MS database search tools, such as MS-GF+ and Comet (see 2.1.2). In addition, in a "`.raw`" file, the signal intensity of each ion in a mass spectrum may be represented as a Gaussian curve over the m/z, which is not proper for the protein identification search tools. To change this representation, we need to centroid each signal intensity, which means to select the peak of each Gaussian curve as the intensity of that ion. In this matter, MSConvert from ProteoWizard [8] is used to centroid MS/MS spectra (`--filter "peakPicking true [1,2]"`) and to convert "`.raw`" files into Mascot generic format files (`--mgf`). To do so, the following command is used:

```
msconvert data.raw --mgf --zlib --filter "peakPicking true [1,2]"
```

As mentioned in 2.2.1, for FDR contorl, we used the de-Bruijn method [35] (available at https://github.com/johramoosa/deBruijn) to generate a decoy database. To do so, the following command is used:

```
java -jar "deBruijn.jar --input proteome.fasta --output
proteome_debruijn.fasta"
```



(a) Orbitrap human data set

(b) NIST data set

(c) Orbitrap mouse data set

Figure 2.2: FDR curves of the Orbitrap human, NIST, and Orbitrap mouse data sets based on MS-GF+ search results. In each plot, the "Threshold" line is calculated based on the 1% FDR. Since the Q-TOF data set is labelled in the original study, no FDR curve is produced for this data set.

The resulted ".mgf" file is then passed to MS-GF+ for database searching. According to the parameters mentioned in 2.2.1, the following command is used to run MS-GF+:

```
java -Xmx2000M -jar MSGFPlus.jar -s data.mgf -d proteome_debruijn.fasta
-t 20ppm -m 3 -inst 3 -ntt 1 -conf params.txt
```

where params.txt contains the list of fixed and variable modifications:

```
# Fixed modifications:
C2H3N1O1, C, fix, any, Carbamidomethyl  # Fixed Carbamidomethyl C
# Variable modifications:
O1, M, opt, any, Oxidation   # Oxidation M
C2H2O, *, opt, Prot-N-term, Acetyl # Acetylation Protein N-term
H-1N-1O1, NQ, opt, any, Deamidated # Deamidation on N and Q
```

Next, the output of MS-GF+, which is a ".mzid" file, is converted to a ".tsv" file with the following command:

```
java -Xmx2000M -cp MSGFPlus.jar edu.ucsd.msjava.ui.MzIDToTsv
-I filename.mzid -o filename.tsv
```

For Comet, as described in 2.1.2, the search parameters are included in a file named "comet.params". After setting the search parameters, the following command is used to run Comet:

```
./comet.exe filename.mgf
```

After the MS/MS database searches are done, for FDR calculation, every scan that is matched to a protein with "debruijn" in its name is considered as a decoy. After calculating the 1% FDR threshold according to the PSM's E-value, every scan above the threshold is labelled as "good", while all other scans are labelled as "bad".

Lastly, to run Novor for *de novo* sequencing, for using the java version as described in 2.1.2, the search parameters are set in the "`params.txt`" file and the following command is used:

```
./novor.sh -p params.txt filename.mgf -f
```

Otherwise, for using the novor.cloud version, the MS files (`*.RAW, *.mzML, or *.MGF`) and the search parameters can be set in the `https://novor.cloud/request`. In the results file (or webpage if using novor.cloud), every scan with a *de novo* sequence that contains at least five amino acids with a high confidence score (>70) is considered as a confident de novo sequence tag.

2.3 Methods

In this section, we present the details about the implemented neural network, the data representation, and the data preprocessing steps used in SPEQ. Additionally, we introduce other methods used in this work for result comparison, and the testing procedure.

2.3.1 Vector Representation

As an input, SPEQ takes a spectrum in Mascot Generic Format (MGF). In an MGF file, for each spectrum, the peak list is provided as tuples, where a peak is an (m/z value, raw intensity) tuple. The intensity is first converted to a value between 0 and 100 by normalizing against the intensity of the most abundant peak in the spectrum. Then we transform each peak list into a vector by binning the m/z range using a 1.000507 m/z as the bin width and a 0.4 m/z as the offset.

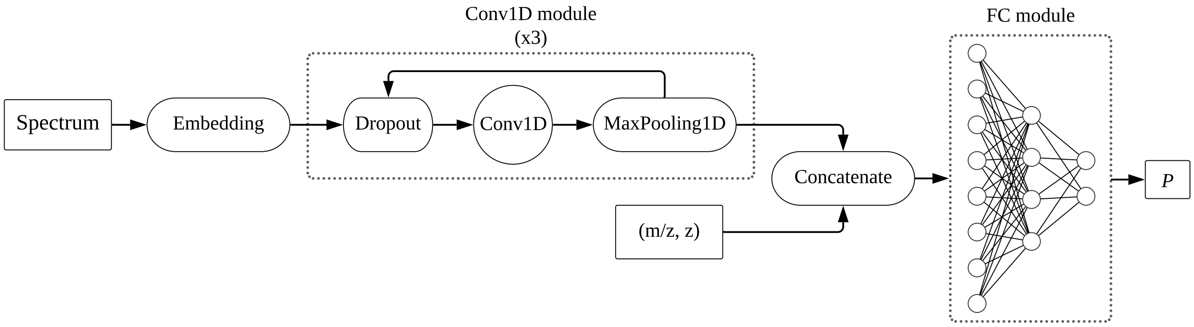


Figure 2.3: The architecture of the DNN model. A feature extraction part using 3 Conv1D modules is connected to a three-layer FC module. A Conv1D is a 1-dimensional convolutional layer, and an FC is a fully connected network. The inputs of the model are the vector representation of the MS/MS spectrum, along with the mass to charge ratio (m/z) and charge state (z) of the precursor ion.

Each dimension of the vector corresponds to the maximum relative peak intensity of all the peaks in the same m/z bin. If a bin has no peak in it, then the dimension of the vector is set to 0. A similar procedure was used in Comet [13] to convert the spectrum to a vector representation. By choosing this data representation, we could induce the m/z distance between peaks of a spectrum to the model, which is related to the quality of the spectrum. In addition, we also include the precursor's charge state and m/z as input of the model.

2.3.2 Neural Network Components

A deep neural network was used to predict the quality of a spectrum from its vector representation. The implementation was conducted in Python 3.7 (`https://www.python.org`). We used the Pyteomics toolkit [20] to read the MGF files and TensorFlow [1] to build the SPEQ model. As shown in Figure 2.3, the model used in SPEQ consisted of a one-dimensional convolutional module for automatic feature extraction, which is connected to a fully connected neural network that was used to conduct the classification.

One-dimensional Convolutional Layer

A convolutional neural network (CNN) [38] usually refers to a two-dimensional CNN, where the input data is a two-dimensional matrix (for example, an image), and there is a kernel that slides through the matrix in both directions, x-axis and y-axis. In one-dimensional CNN (1D-CNN), the kernel only moves in one direction. 1D-CNN is mostly used in problems with time-series data, a sequence of data taken at successive equally spaced points in time, such as the temperature of a room over time. One can see an MS/MS spectrum as time-series data, a sequence of intensities (data) over equally spaced m/z values (time). A 1D-CNN has two hyper-parameters, the size of the kernel and stride, which is the amount of movement between each time the kernel is applied to the input data. These hyper-parameters, along with other details of our model, are provided in Section 2.3.3.

Fully Connected Network

The smallest unit in a neural network is called a neuron. A neuron gets input values $(x_1, \cdots, x_n)$ and weights $(w_1, \cdots, w_n)$, and produces an output as a result (y), which is a function of the sum of input values multiplied by corresponding weights plus a constant value, bias (b):

$$y = f(\sum_{i=1}^{n} w_i \cdot x_i + b) \tag{2.1}$$

Inspired by the human brain, the simplest neural network is Perceptron [43], which is a single neuron. Accordingly, a multi-layer Perceptron [44] (MLP) is built when more than one layer of multiple neurons are sequentially connected to each other. Since each neuron in layer i is connected to all of the neurons on layer $i-1$ (inputs) and layer $i+1$ (output), it is called a fully connected neural network (sequential network). An illustration of the connections is shown in “FC module” in Figure 2.3. There are three types of layers in an MLP:

- **Input layer:** The input data determines the number of neurons on this layer. In our problem, as illustrated in Figure 2.3 the fully connected network is connected to a feature extraction module, thus the number of neurons is based on the number of the extracted features, which is 128.
- **Hidden layer(s):** We used one hidden layer, which contains 64 neurons. Each one of these 64 neurons is connected to the 128 neurons on the input layer, and passes its output to the next layer.
- **Output layer:** The number of neurons on this layer is based on the problem. For example, if the problem is a classification task with ten classes, the number of neurons is equal to 10. In our problem, we only have two classes, "low-quality" and "high-quality" spectra. Therefore, there is one neuron on the last layer of the network, P, which can be interpreted as the probability of a spectrum being high-quality.

In Figure 2.3, the FC module is a representation of a fully connected neural network with three layers, 8 neurons on the input layer, 4 neurons on the hidden layer, and 2 neurons on the output layer.

Other Components

In our proposed neural network, four other components are used: Embedding, Dropout, MaxPooling, and Concatenate.

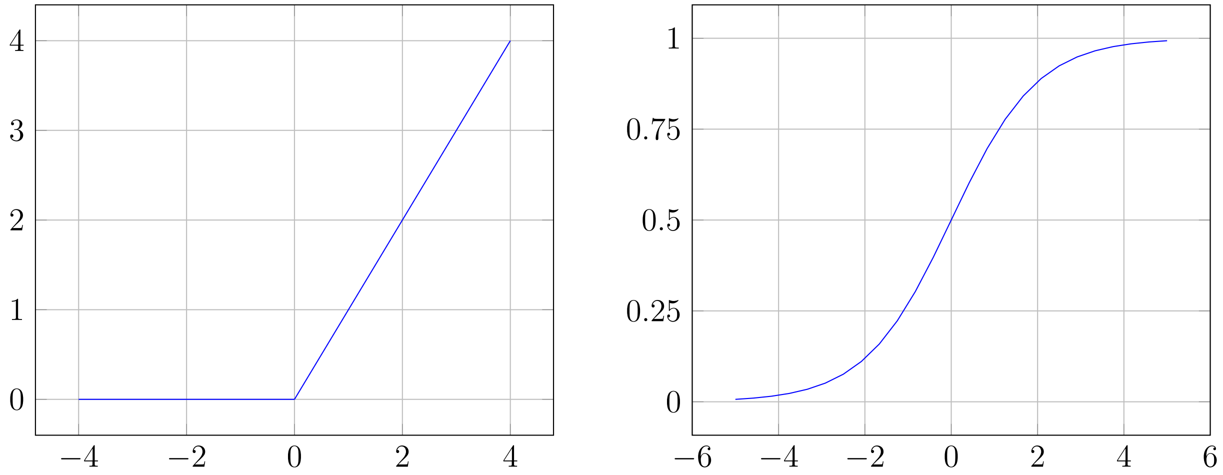
- **Embedding layer:** An embedding layer converts an input vector into a compressed feature space with a fixed size. This layer is mainly used in problems with sparse data. In our problem, an MS/MS spectrum is sparse data when it is represented as a large vector with very few peaks, leading to many elements being zero. Therefore, instead of feeding large sparse input vectors with variable lengths into a model, the input data can be transformed into a fixed vector with reduced dimensions without losing any information, by using an embedding layer as the first layer of a neural network. Since we normalize each peak into a value between 0 and 100, the input dimension of the embedding layer is 101. The output dimension is also set to 64.

- **Dropout:** In a neural network, a dropout layer is used to avoid overfitting [49]. In this technique, a predefined proportion of neurons in each layer are randomly selected to be ignored during a learning step. In our model, we used a dropout layer with a probability of 0.3 before each one-dimensional convolutional layer.
- **MaxPooling:** Similar to a convolutional layer, a MaxPooling layer has a window that slides through a matrix. However, instead of doing a convolutional operation, in each iteration, only the maximum value within the window is selected. Therefore, a MaxPooling layer has no parameters to train and is used to reduce the size of the input matrix.
- **Concatenate:** A concatenate layer is a simple component that merges multiple inputs into a larger vector. In our model, we concatenate the extracted features into the two proteomic features, the precursor's charge state (z) and mass over charge ratio (m/z).

Activation Functions

An activation function is used to add non-linearity to deep neural networks. It decides whether a neuron should be activated or not. In other words, an activation function determines how the input value should be transformed into output [47]. In this work, we used two different activation functions: Rectified linear unit (ReLU), and sigmoid. Plots of both functions are illustrated in Figure 2.4.

ReLU is a piecewise linear function that outputs the input directly if it is positive and returns zero otherwise. Because there is no upper bound in this activation function, it helps the model to prevent vanishing gradient. In addition, since ReLU is computationally



(a) ReLU

(b) Sigmoid

Figure 2.4: Plots of the two activation functions used in this work.

efficient, it is commonly used in the middle layers of a network.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Sigmoid is a logistic function that has an “S”-shaped curve and is bounded between 0 and 1. Thus, we use sigmoid as the activation function for the last layer of the model, taking the output value as the probability of the input MS/MS spectrum being high-quality.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Loss Function and Optimizer

A loss function is used to measure the performance of a classification model. It calculates a loss value based on what label the model predicts and what the actual label is, which eventually helps the model to learn from its mistakes and make better predictions. Since we are solving a binary classification problem, with the output being a probability value between 0 and 1, we used binary cross-entropy as the model's loss function. Consider p being the probability of the true label and q the probability of the predicted label. Then in a binary classification problem, with labels y being 0 or 1, $p \in \{y, 1-y\}$ and $q \in \{\hat{y}, 1-\hat{y}\}$. Hence, cross-entropy for two classes is defined as:

$$H(p, q) = -\sum_i p_i \log q_i = -(y \cdot \log \hat{y} + (1-y) \cdot \log(1-\hat{y})) \tag{2.4}$$

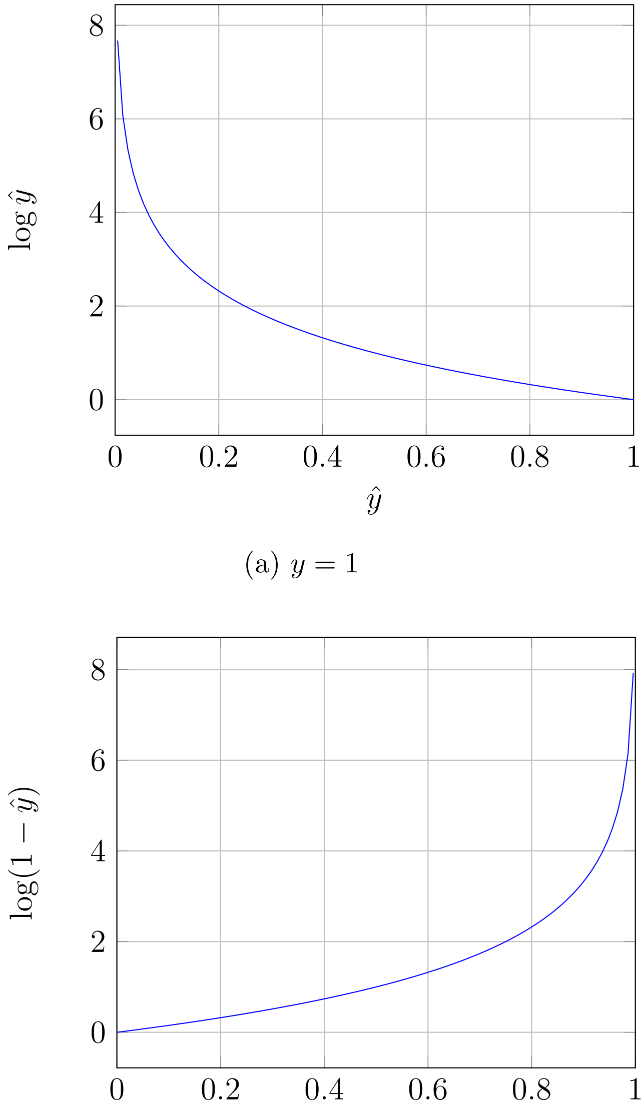
As a result, when the predicted label is correct ($\hat{y} = y = 0$ or $\hat{y} = y = 1$), the loss value is 0, and it grows up as the dissimilarity between p and q increases. The cross-entropy loss value is plotted in Figure 2.5.

An optimization function, or optimizer, is an algorithm that helps the neural network to minimize the loss value, and consequently to improve the accuracy of the model. This is done by modifying the parameters of the neural network, such as weights and learning rate. For our model's optimizer, we used Adam [29]. It is a variant of the gradient descent optimization algorithm which is suitable for problems with very noisy or sparse gradients.

2.3.3 SPEQ's Neural Network Architecture

For predicting the quality of a spectrum, in contrast to previous works, no handcrafted features are used in our model. Instead, we want the neural network to automatically extract beneficial features from an input spectrum. For this reason, the model is designed in two parts: one part for the automatic feature extraction from the input spectrum (Conv1D modules), and one part for using those extracted features for the classification (FC module).

The reason for using one-dimensional convolutional layers for extracting features comes from the differences between high-quality and low-quality spectra, which are shown and discussed in Figure 2.1. By using one-dimensional convolutional layers with large kernel sizes, it can be possible to disregard small noisy peaks and induce the distance between significant peaks of an input spectrum to the neural network model. Once the useful



(a) $y = 1$

(b) $y = 0$

Figure 2.5: Cross-entropy loss value plots for when (a) $y = 1$, and (b) $y = 0$.

features are extracted, a fully connected network can be used to exploit these features for the classification.

The first part contained an embedding layer on top of three convolutional blocks (Conv1D module), where each block had a dropout layer connected to a one-dimensional convolutional layer and a one-dimensional MaxPooling layer at the end. This part was designed to process the vector representation and extract features from the peaks of the input spectrum. The parameters of each convolutional layer (first layer: kernel size 11 with strides 5, second layer: kernel size 51 with strides 10, third layer: kernel size 3 with strides 1) were selected so that the important information within a spectrum was best induced to the model.

The kernel size of the first layer is set to 11 in order to filter out noisy peaks that are close to a significant peak, and only take significant peaks which have the most important role in peptide identification. Next, when noisy peaks are filtered out and significant peaks are selected, for the next Conv1D layer the kernel size is set to 51 to cover the distance between significant peaks. The last Conv1D layer is used to reduce the dimension of the extracted feature vector while maintaining all the information.

In the next part, the output of the first part was concatenated to the two additional features: the precursor's charge state and m/z. The obtained vector was fed to a three-layer fully connected neural network (FC module). ReLU is the activation function for the first and the second layer. After applying a sigmoid activation function on the last layer, we obtained P, which is the probability of a spectrum being high-quality; it can also be interpreted as the quality score for the input spectrum. After applying a threshold on this probability, the output was assigned a label of “good” or “bad” for the input spectrum. The model was then compiled with the Adam optimizer and was trained using a binary cross-entropy loss function.

Prior to this model, two other different neural networks were designed and tested for the feature extraction part, which failed to confidently separate high-quality spectra from low-quality spectra. First, a deep two-dimensional CNN connected to a fully connected was designed. For this model, the input data was a two-dimensional plot of relative intensity to m/z for each spectrum. However, for any CNN model, it is an extremely hard task to assess the quality of an MS/MS spectrum from this way of data representation. Many different architectures were tested, and all of them failed to achieve an AUC above 60% on the Orbitrap human data set 2.2.1. In addition, we used top pre-trained CNN models such as ResNet-50 [22] and VGG-19 [48] with proper changes to match our problems requirements. Although there was an improvement in the achieved AUC, using CNN models with such data representation could not solve this task.

As a consequence of the CNN model failure, the data representation was changed to the current vector representation 2.3.1. Accordingly, we designed and tested various recurrent neural networks (RNN), including bidirectional RNN and long short-term memory (LSTM) networks. The best performance was achieved from a model with 25 LSTM cells. However, the main problem with using RNN was that in an MS/MS spectrum the peaks are usually far from each other, which would cause vanishing gradient [24], resulting in poor classification performance. By using one-dimensional CNN, we were able to properly extract effective features from the vector data representation, while avoiding the vanishing gradient.

2.3.4 Other Methods Tested

Three other methods, a baseline model, SpectrumQuality, and Bern's model were tested together with the SPEQ method. The baseline model simply uses the number of peaks in the spectrum as the quality score. SpectrumQuality is the method published by Flikka et al. [15] and the software was downloaded from `http://services.cbu.uib.no/software/spectrumquality`.

The Bern's model was previously published by Bern et al. [4]. In this model, each input is a histogram of m/z differences from a spectrum. The histogram is a 187-length vector, where an element i is the bin of m/z difference of $[i-0.5, i+0.5]$ (187 is the maximum mass of an amino acid residue). As mentioned by Bern et al. [4], due to the time complexity of the algorithm used in the proposed data representation, this method requires significant training and testing time. The software that uses the Bern's model has not been published by the author and is not available for use. Therefore, the results presented here are derived from our implementation based on the model described by Bern et al. [4].

To obtain a score for each sample, we used the Epsilon-Support Vector Regression (SVR) package available in scikit-learn [41], with radial basis functions as the kernel. In the SVR model, the width parameter, γ, must be set. This value was set to 500 in the study by Bern et al. [4]. However, this led to very low classification performance in our experiments (area under the ROC curve around 50%). Instead, γ was set to "auto" first. A linear search was also performed to find the best γ. These two approaches gave very similar performances. The better performance of the two was used in each comparison.

2.3.5 Testing Procedure

SPEQ's performance was evaluated in three different aspects.

Comparing to Other Methods

Firstly, the prediction accuracy of SPEQ and other models was compared. The five-fold cross-validation method was used to measure the prediction accuracy. More specifically, the data set was first divided into five parts $\{D_1, \cdots, D_5\}$. Following this, each part (D_i) was used once as the test set, while the four other partitions ($D_j : i \neq j$) were combined and used as the training set. This process was conducted five times, and the final reported performance of the model was the average of these five processes.

The dividing procedure ensured that spectra with the same m/z and z were in the same part. This way, repeated scans of the same precursor were not simultaneously present in the training and testing set. Moreover, a cross-species validation was also performed, where the models were trained on the Orbitrap human data set and tested on the independent Orbitrap mouse data set. The results of this accuracy test are provided in Section 2.4.1.

Identification Evaluation

Secondly, the unidentified spectra from a database search analysis were subject to additional analyses to check whether they can be interpreted by other analytical methods. In this test, the Orbitrap human dataset was first searched with MS-GF+ with the parameters provided in Section 2.2.1. Using the labels generated from the MS-GF+ search, we trained the SPEQ model and used the model to assign a quality score (SPEQ score) to all the spectra.

The unidentified spectra were further searched with the Comet software with the same parameters. The spectra unidentified by the first two analyses were *de novo* sequenced with the Novor software [30]. A *de novo* sequence that contains at least five amino acids with a high confidence score (>70) is regarded as a confident *de novo* tag. It is expected that a higher SPEQ score in the spectra unidentified by the first search is associated with a higher percentage of spectra assigned by either Comet or *de novo* sequencing. The results of this test are provided in Section 2.4.2.

Analysis Troubleshooting

The third test is to demonstrate the usefulness of the SPEQ score in "troubleshooting" a data analysis. In this test, the Orbitrap human dataset was first searched with MS-GF+ with the parameters provided in Section 2.2.1. It was suspected that many of the unidentified spectra were because they contain a PTM unspecified in the search parameter. However, searching with too many variable PTMs on the whole data set is prohibitively slow.

To troubleshoot, SPEQ was used to score all the unidentified spectra. The top 1% of these unidentified spectra according to the SPEQ score were selected to conduct a search with many additional variable PTMs. From the identified peptides, a few most common PTMs were selected. Then a third-round search was conducted to identify more peptides by using all of the unidentified spectra and these few additionally selected variable PTMs. The results of this "troubleshooting" test are provided in Section 2.4.3.

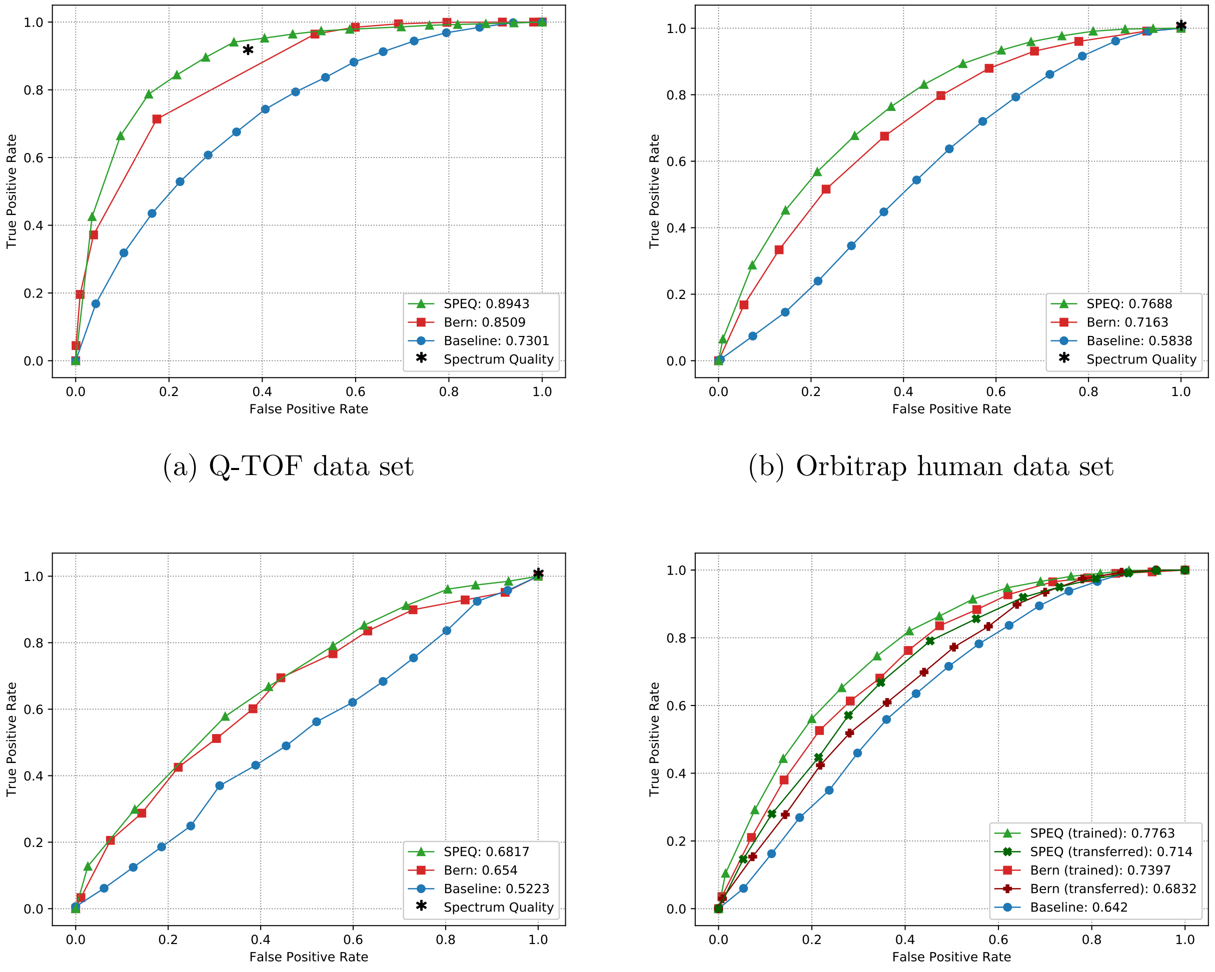
2.4 Results, Discussion, and Future Work

In this section, first, we evaluate the proposed model's performance against other methods on different data sets, by comparing the achieved area under the receiver operating characteristic curves. Then we assess the unidentified high-quality spectra recognized by the model, and how SPEQ-directed troubleshooting works. In the end, we discuss how such a tool can benefit the MS/MS analyses.

2.4.1 Prediction Accuracy

Figures 2.6a, 2.6b, and 2.6c show the receiver operating characteristic (ROC) curves of the predictions made by each model on the Q-TOF, Orbitrap human, and NIST data sets, respectively. The area under curve (AUC) of the ROC curve for each method is also provided in the figures. The figures clearly show that SPEQ's prediction accuracy outperforms all other tools in all data sets. Note that SpectrumQuality's curve has only one data point. This is because it does not output a quality score, but only classifies the spectrum into two classes. Also, SpectrumQuality failed to make any valid prediction on the Orbitrap human dataset (Figure 2.6b) and the NIST data set (Figure 2.6c). Instead, it assigned high quality to every spectrum. This is likely because both data sets were obtained from Orbitrap instruments, while the model was developed based on data from the Q-TOF instruments.

All models' performances were low for the NIST data set. This is likely because the spectra in the NIST data set have already been selected when they were collected in the



(a) Q-TOF data set

(b) Orbitrap human data set

(c) NIST data set

(d) Orbitrap mouse data set

Figure 2.6: A, B, C: ROC curves and areas under curve (AUC) of different tools on the Q-TOF, Orbitrap human, and NIST data sets, respectively. D: ROC curves and AUC of different methods on the Orbitrap mouse data set. The SPEQ (trained) and Bern (trained) curves were obtained when the models were trained with the same Orbitrap mouse data set and tested with 5-fold cross-validation. The SPEQ (transferred) and Bern (transferred) curves were obtained when the models were trained with the Orbitrap human data set and tested with the Orbitrap mouse data set.

spectrum library, and lack extremely low-quality spectra. It is a harder task to distinguish between the high and medium-quality spectra than between the high and low-quality spectra.

Figure 2.6d shows the ROC curves and their AUC of different methods on the Orbitrap mouse data set. The SpectrumQuality tool failed to make any valid prediction here and is not included in the figure. To produce the curves, the SPEQ and Bern's methods were either trained on the same data set and tested with a 5-fold cross-validation or trained on the Orbitrap human data set. Not surprisingly, for both models, the cross-validation test on the same data set produced better accuracy than training on a different data set. Nevertheless, SPEQ achieved a decent prediction accuracy when training and testing were performed on different data sets. Also, SPEQ's performance is better than both the Bern's method and the baseline method in both testing scenarios.

2.4.2 Unidentified High-quality Spectra

Figure 2.7 shows the results of the test for the unidentified spectra (as described in the testing procedure section). After searching the 52,285 spectra in the Orbitrap human dataset using MS-GF+, 20,885 spectra failed to be confidently identified. These unidentified spectra were subject to a second database search with Comet and *de novo* sequencing with Novor. The histogram of Figure 2.7 (A) shows the number of spectra in each SPEQ score interval, and the distribution of the four categories of spectra:

- identified in the first search by MS-GF+,
- not identified in the first search but identified by Comet in the second search,
- not identified by the first two searches but containing a confident *de novo* sequence tag, and
- not interpreted by any of these analyses.

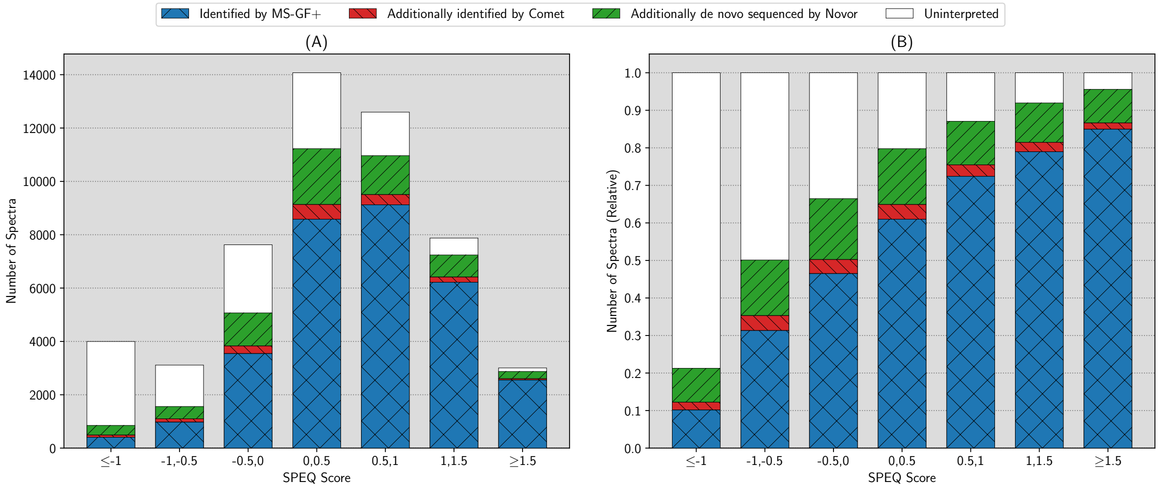


Figure 2.7: (A) The number of identified and unidentified spectra in different intervals of the quality score assigned by SPEQ. (B) The relative ratio of identified and unidentified spectra in different intervals of the SPEQ score. The SPEQ score used in this figure is the logit of the probability predicted by SPEQ. The blue, red, and green bars represent the spectra identified by MS-GF+, Comet but not MS-GF+, and Novor but not the other two tools, respectively. The white bars represent the spectra unidentified by any of the three tools.

Figure 2.7(B) is the same as Figure 2.7(A) except that the y-axis becomes the percentage in each SPEQ score interval. To plot these figures, the logit of the probability predicted by SPEQ was used as the SPEQ score for a more proper division of the score intervals.

As can be seen, the percentage of spectra confidently identified by at least one engine grew as the quality score provided by SPEQ increased. This observation suggests that high-scoring spectra not identified by the first search engine can often be further interpreted in other analyses, while low-score spectra are usually not interpretable. It is more likely for a spectrum containing valuable information to obtain a higher score compared with an uninterpretable scan.

Moreover, when the score is high enough (e.g. ≥ 1.5), the majority of the spectra not identified by any of two database search tools contain confident *de novo* sequencing tags. This strongly suggests that these spectra are indeed produced by peptides, but unidentified because of the inadequate data analysis.

2.4.3 SPEQ-directed Troubleshooting

For the troubleshooting test (as described in the testing procedure section), using the Orbitrap human data set, the first search with MS-GF+ used the following variable PTMs.

- Oxidation on M,
- Acetyl at protein N-term, and
- Deamidated on N and Q.

Note that the original publication [6] of the data set used only the first two variable-PTMs in this list. But adding the third PTM resulted in more identifications in the first search, which identified 31,400 of the 52,285 spectra. Among the 20,885 unidentified spectra, the top 1% spectra (according to their SPEQ score) were searched again with the following variable PTMs:

- Oxidation on M,
- Deamidated on N and Q,
- Carbamidomethyl at peptide N-term,
- Pyro-Glu at peptide N-term on Q and E,
- Acetyl at protein N-term,
- Acetyl on K,

- Methyl on K, and
- Phospho on S, T and Y.

The search took 3.75 minutes and identified 24 confident peptides with PTMs listed above. Three of the most common PTMs in these peptides were:

- Pyro-Glu at peptide N-term on Q,
- Carbamidomethyl at peptide N-term, and
- Deamidated on N and Q.

Because deamidation was used in the first search already, for the 24 peptides identified here, deamidation always appeared together with another PTM in the same peptide.

A third search was conducted to search with these three most common PTMs and the 20,885 unidentified spectra. The search finished in 12.06 minutes and confidently identified 829 spectra that contain at least one of these three PTMs. To compare, searching all the 20,885 unidentified spectra with the longer list of PTMs took 78.44 minutes and identified 863 spectra.

The results here demonstrate that the SPEQ score can indeed be used to select a small portion (1%) of the unidentified spectra for troubleshooting, and the factors identified by the troubleshooting can be used to adjust the search strategy in additional searches to identify more peptides.

2.4.4 Discussion

The use of deep learning may be an important reason for the improvement of SPEQ. With sufficiently large training data, deep learning can automatically discover the features important for the prediction. This is in contrast to the traditional machine learning that requires the tool developers to hand-craft features. Handing off the feature extraction to the learning algorithm not only saves the developers' time, but also allows the learning algorithm to discover new features that the tool developers may not be able to. This is particularly interesting in a cross-disciplinary area such as bioinformatics, where sometimes the tool developer may not be the most knowledgeable domain expert to handcraft the features. Also, this makes the model more adaptive to different types of MS instruments and experimental methods.

Experiments were also carried out to demonstrate the potential usefulness of SPEQ in proteomics data analysis. In general, a quality assessment tool helps the proteomics data analysis in two ways:

Firstly, it provides some hope in dealing with the false negatives. The proteomics research community has established a standard way (the FDR) to control false positives of a data analysis experiment. However, there is no established way to know the level of false negatives in a data analysis experiment. The results of Figure 2.7 show a strong correlation between the SPEQ score and the percentage of the false-negative spectra (i.e., the spectra that were not identified by the initial database search but identifiable with additional efforts). This study suggests that the quality score assigned by SPEQ (or by another tool) can potentially be used to estimate the level of false negatives. However, more research is needed to prove this can indeed provide an accurate estimation.

Secondly, the quality score can be used to direct the allocation of resources (either computing power or human experts' time) to focus on the high-quality spectra, which have the best chance to be interpreted by the data analysis effort. This is demonstrated in the thesis with the "troubleshooting" experiment, where the top 1% of the unidentified spectra were analyzed with a much more extensive and costly search by selecting a long list of variable PTMs. This revealed that some peptides contain PTMs that had not been specified in the original database search. By adding back the most frequent PTMs found by the troubleshooting, more spectra were identified. This troubleshooting practice is in line with the Preview idea proposed by Kil et al. [27], where only a subset of spectra was searched first in order to determine the best search parameters for the full search. The quality score can be also used here to select the best subset of spectra for the pre-search.

2.4.5 Future Work

While SPEQ improved the prediction accuracy relative to other tools, its accuracy is still not ideal. For example, there is still a big gap between 1 and the AUC of SPEQ (0.7688) on the Orbitrap human data set. Since the spectra were labelled according to whether they are confidently identified by database search, the false negatives of the database search would have created mislabels in the training and testing data. The mislabels in the testing data could have an adversary effect on the AUC. In another word, the actual performance of SPEQ (and other tools) may be better than indicated here. Meanwhile, if the number of mislabeled spectra in the training data can be reduced, the machine learning algorithm will learn a better model too. Thus, one way to improve this work is to provide training data sets with very high-confident labels. This can be achieved by labelling the samples of each data set with the results of more than one MS/MS search engine.

Other possible ways to enhance the model include different neural network structures, larger training data, and other learning strategies such as transferred learning. The current architecture of the SPEQ's neural network is fairly simple, compared to the state-of-the-art neural networks of other machine learning tasks. One main issue encountered in this work was that the MS/MS data is too sparse, which makes it highly challenging to design very deep neural networks, due to the possible vanishing gradient problem. In fact, if the scoring function can be sufficiently developed in the future, the low-quality spectra can be excluded from the analysis from the very beginning. This advancement will not only improve the data analysis speed, but will also reduce the false positives created by these low-quality spectra.

Chapter 3

Automatic Detection of the Protease Used in Bottom-up Proteomics Experiments

In this chapter, we propose a statistical model to automatically detect the digestive enzyme used in a proteomics experiment by studying the *de novo* sequenced peptides.

3.1 Introduction

All proteomics data analysis software requires the setting of several important parameters for the effective analysis of bottom-up proteomics data. Such parameters include the digestive enzyme (proteases) used in sample preparation, mass error tolerance, and post-translational modifications.

To facilitate automated and streamlined data analysis, it is desirable for the data analysis software to determine these parameters automatically from the MS data, rather than requiring a human user to intervene between the data production and the data analysis. In this work, we propose an algorithm to automatically detect the enzyme used for digestion, by analyzing the distribution of amino acids in peptides *de novo* sequenced with a nonspecific enzyme setting.

3.2 Method

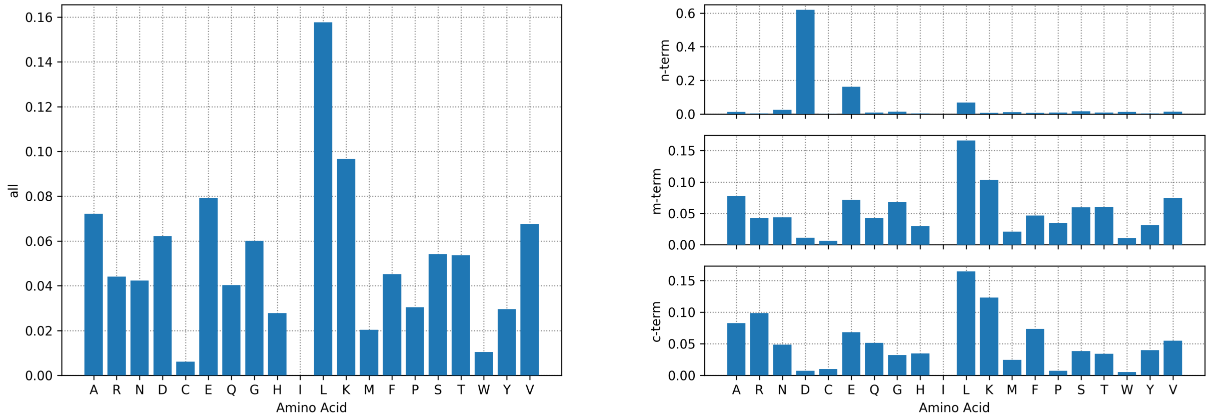
Novor software is used to *de novo* sequence the spectra. Since the enzyme information is unknown at this point, the nonspecific enzyme setting is used. The frequencies of different amino acids at n-term (first amino acid of a peptide), c-term (last amino acid of a peptide), middle (every amino acid between n-term and c-term), and all positions of the *de novo* peptides are examined to detect the used enzyme. For different enzymes, these distributions are very different. Figures 3.1-3.6 illustrate the distribution of amino acids at different positions obtained from training data of different enzymes.

In the training step, for each enzyme, we calculate the above-mentioned expected frequencies per amino acid. Then, for a new set of *de novo* results, the amino acid frequencies at different positions are compared with the ones obtained during training for each known enzyme. The enzyme that maximizes the similarity is selected as the correct enzyme.

3.3 Model

For enzyme e, let $f_n^e\ (\alpha)\,,\ f_m^e\ (\alpha)\,,\ f_c^e\ (\alpha)\,,\ f_t^e\ (\alpha)$ be the expected frequencies of amino acid α at n-term, middle, c-term, and all positions of the *de novo* peptides, respectively. Let $g_n\,(\alpha)\,,\ g_m\,(\alpha)\,,\ g_c\,(\alpha)\,,\ g_t\,(\alpha)$ be the frequencies of a new *de novo* result. Now, for each enzyme e, we define P^e as:

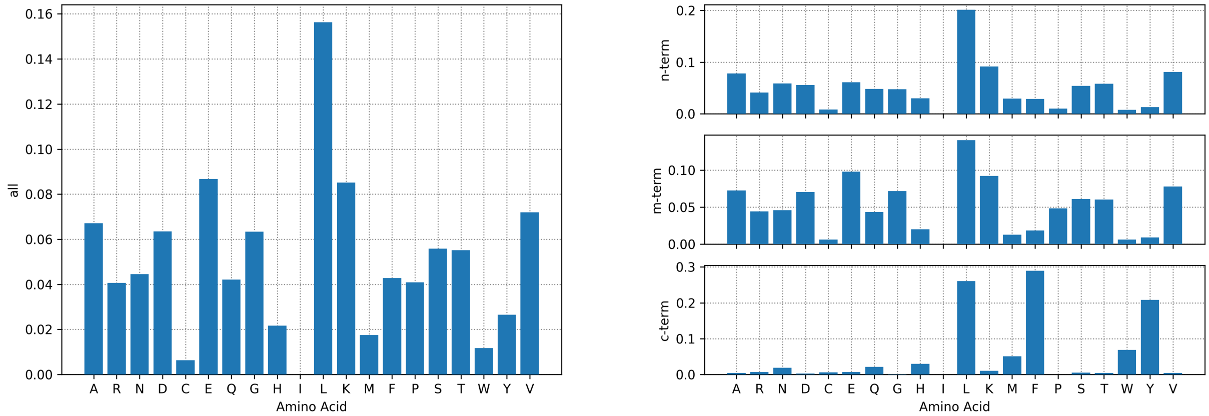
$$P^e = \sum_{i\in\{n,\ m,\ c\}} n_i \cdot \sum_{\alpha\ is\ an\ amino\ acid} g_i\,(\alpha) \cdot \log \frac{f_i^e\,(\alpha)}{f_t^e\,(\alpha)} \tag{3.1}$$



(a) All positions

(b) n-term, middle, and c-term

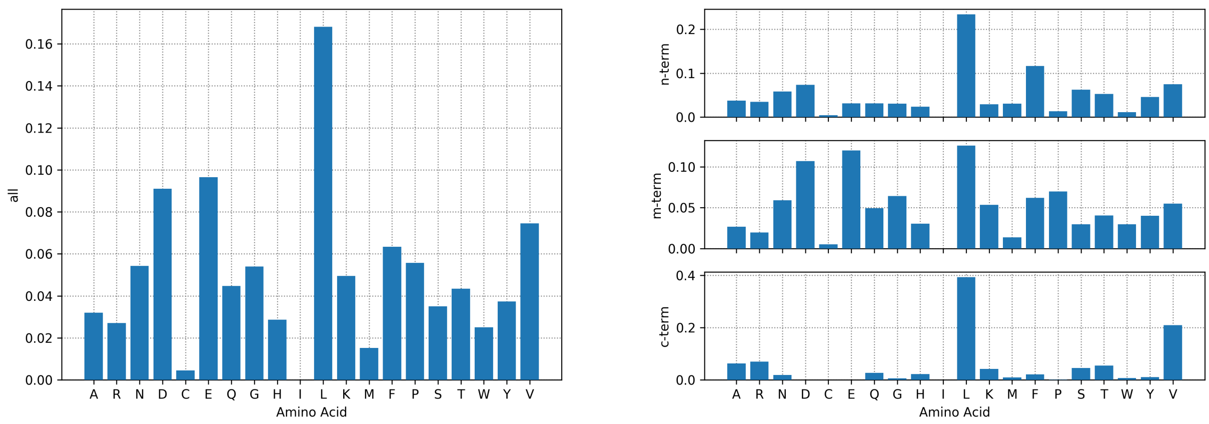
Figure 3.1: AspN enzyme amino acid frequency plots.



(a) All positions

(b) n-term, middle, and c-term

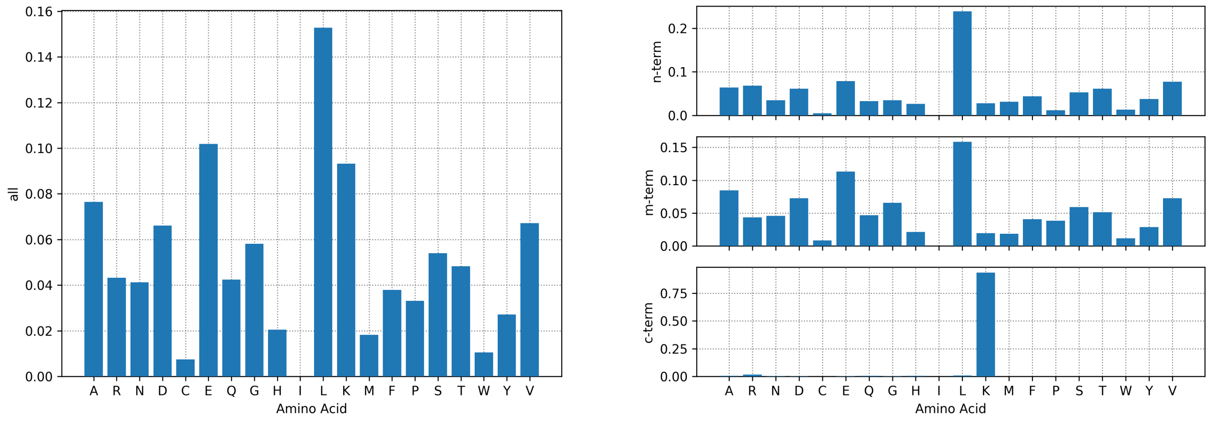
Figure 3.2: Chymotrypsin enzyme amino acid frequency plots.



(a) All positions

(b) n-term, middle, and c-term

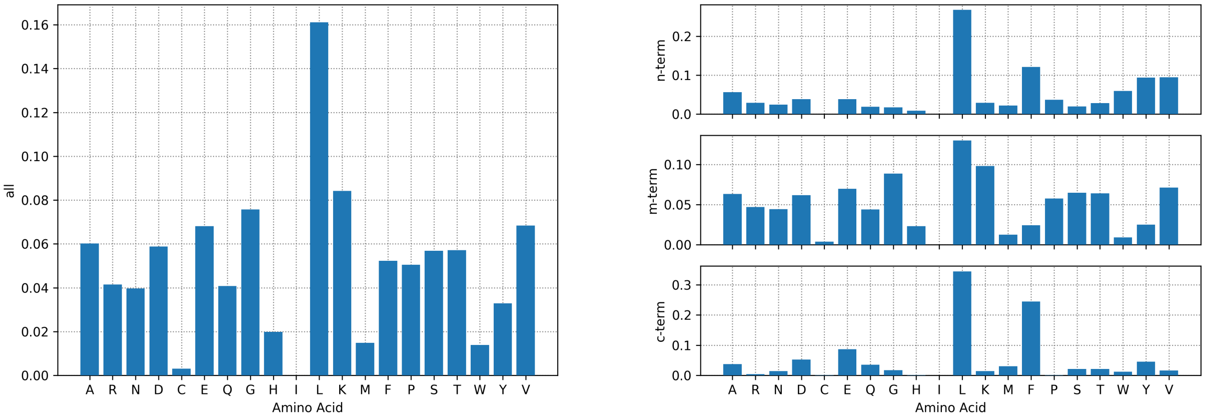
Figure 3.3: Elastase enzyme amino acid frequency plots.



(a) All positions

(b) n-term, middle, and c-term

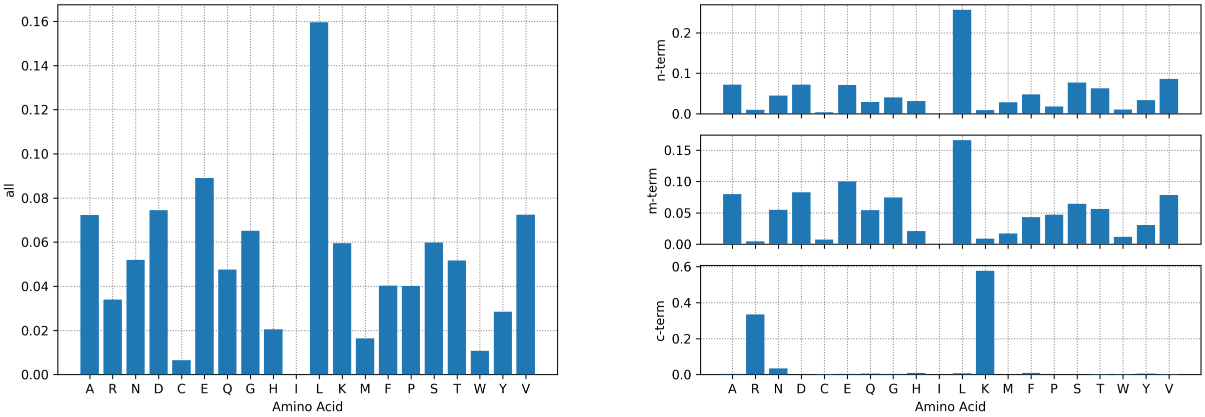
Figure 3.4: LysC enzyme amino acid frequency plots.



(a) All positions

(b) n-term, middle, and c-term

Figure 3.5: Pepsin enzyme amino acid frequency plots.



(a) All positions

(b) n-term, middle, and c-term

Figure 3.6: Trypsin enzyme amino acid frequency plots.

Where n_i is the average number of amino acids in i. P^e is expected to be greater than 0 if the enzyme is correct. Thus, the enzyme that maximizes P^e has the highest probability to be the correct enzyme. One can view $\log \frac{f_i^e(\alpha)}{f_t^e(\alpha)}$ as the score of amino acid α to belong to position $i \in \{n,\ m,\ c\}$ for the enzyme e. The score for an enzyme is simply the weighted sum of the amino acid scores in each position.

3.4 Data

The model only uses high-quality spectra for training and prediction, where a peptide is considered high-quality if it has a *de novo* score above 70. The training dataset includes 19702 *de novo* peptide sequences for 6 different enzymes: AspN (2490), Chymotrypsin (4394), Elastase (2325), LysC (2644), Pepsin (3733), and Trypsin (4116), respectively. The trained model was then tested on 14 additional MS data files.

This model is designed to detect the enzyme from *de novo* sequenced peptides with a nonspecific enzyme. Therefore, it is important that for each enzyme, the pre-trained frequencies used in the prediction model (f^e) have the same characteristics as the input data. It means that the trained distribution of amino acids from each site (n-term, middle, c-term, and total) should come from samples digested with the same enzyme but *de novo* sequenced with Novor's nonspecific enzyme setting.

3.5 Results and Discussion

The method was able to correctly identify the right enzyme for all of the testing datasets. On average, the similarity score between a testing dataset and a model enzyme ranges between -3 and 3, whereas the predicted enzyme with the highest score has a 1.61 margin in comparison to the second highest-scoring enzyme.

test enzyme	AspN	Chymotrypsin	Elastase	LysC	Pepsin	Trypsin	margin
AspN-1	2.372	-1.627	-1.639	-2.424	-1.448	-1.629	3.820
AspN-2	2.012	-1.431	-1.950	-1.258	-1.554	-0.769	2.781
AspN-3	1.886	-1.540	-1.424	-3.190	-1.538	-2.969	3.310
Chymotrypsin-1	-1.581	0.640	-0.904	-4.243	-0.562	-4.539	1.202
Chymotrypsin-2	-1.833	0.537	-1.556	-3.590	-1.617	-2.988	2.093
Chymotrypsin-3	-2.332	1.225	-0.954	-4.639	-0.138	-5.068	1.363
Elastase-1	-0.717	-1.479	-0.248	-2.016	-2.409	-1.045	0.469
Elastase-2	-1.199	-1.859	-0.206	-2.768	-2.083	-3.469	0.993
LysC-1	-1.256	-2.700	-0.507	2.312	-2.501	1.359	0.953
LysC-2	-1.065	-2.059	-0.466	2.965	-2.049	2.948	0.017
LysC-3	-1.507	-2.612	-0.696	1.755	-2.458	1.411	0.344
Pepsin-1	-1.249	-0.473	-2.595	-3.975	0.425	-2.860	0.898
Pepsin-2	-1.024	-0.025	-2.390	-4.293	0.774	-2.616	0.799
Trypsin-1	-0.390	-2.014	0.287	0.933	-2.605	2.949	2.016
Trypsin-2	-0.701	-2.087	0.174	-0.166	-3.921	1.657	1.483
Trypsin-3	-1.012	-2.499	0.459	0.609	-2.943	2.216	1.607

Table 3.1: The score result of each enzyme against different enzyme data sets. Darker green indicates a correct prediction with higher confidence.

Enzymes with more unique digestion rules tend to have a larger score margin, indicating the higher robustness in their detection. These include AspN (score margin = 3.30), Chymotrypsin (1.55), and Trypsin (1.70). In contrast, other enzymes that are either non-specific or similar to another enzyme, such as Elastase (0.73), Pepsin (0.85), and LysC (0.44), had a relatively lower score margin. All scores are available in Table 3.1.

- **test enzyme**: The enzyme used in the test file.
- **AspN** to **Trypsin**: The score of the model for each enzyme.
- **margin**: The difference between the highest and the second-highest scored enzyme.

Since only a subset of several hundred randomly sampled spectra is needed for the analysis, the speed of the enzyme detection analysis is very fast, thanks to the *de novo* sequencing speed of Novor. For each of our testing data, the analysis can be completed in a few seconds on a laptop computer.

While the proposed model was able to identify the correct enzyme for all of the tested data sets (Table 3.1), there are still some cases where the right enzyme is identified with a very small margin. This is one of the main areas to continue and improve this work. One possible solution is to use a second scoring function in such situations. The current model is trained on less than 4500 samples for each enzyme, however, increasing the number of training samples can also improve the accuracy of the model. Another possible way to continue this work is to support more enzymes to be automatically identified, such as ArgC, GluC, LysN, ProteaseAP, and ProteaseK. Furthermore, the top-scored enzyme is always selected as the output of the model, and “Nonspecific” enzyme is not designed in the current model, which can be addressed in future works.

Chapter 4

Conclusion

In this thesis, first we presented SPEQ, a software tool that uses deep learning to predict the quality of an MS/MS spectrum. The prediction accuracy of SPEQ was evaluated by the ROC curves on several different data sets. SPEQ performed better (with higher AUC) than the other tools compared (Figure 2.6a-2.6c). This is still the case when the testing and training data are from independent experiments of two different species (Figure 2.6d).

We have developed the SPEQ tool for spectrum quality assessment based on deep learning, and demonstrated its usefulness. Further improvement of the quality score by the bioinformatics community is needed, and will greatly enhance the usefulness of the quality assessment. The availability of SPEQ may help other proteomics data analyses and support other bioinformatics researchers to further improve the accuracy of spectrum quality assessment. SPEQ is written in Python and the source code is freely available at `https://github.com/sor8sh/SPEQ`. A manuscript of this research has been published in Bioinformatics journal [19].

In the other work presented in this thesis, we proposed an algorithm that uses the distribution of amino acids from *de novo* sequenced peptides to automatically determine the enzyme used in the proteomics experiment. In this algorithm, we used a statistical model to compare the distribution of all amino acids on the n-term, c-term, and the middle of *de novo* peptides sequenced with a nonspecific enzyme setting. Results presented in Table 3.1 support the effectiveness of this algorithm. This work is presented as a poster at the 70th ASMS Conference on Mass Spectrometry and Allied Topics [18].

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, Mar 2003.

[3] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, Mar 2003.

[4] Marshall Bern, David Goldberg, Hayes Mcdonald, and John Yates. Automatic quality assessment of peptide tandem mass spectra. *Bioinformatics (Oxford, England)*, 20 Suppl 1:i49–54, 09 2004.

[5] Alejandro Brenes, Jens Hukelmann, Dalila Bensaddek, and Angus I. Lamond. Multibatch tmt reveals false positives, batch effects and missing values. *Molecular & Cellular Proteomics*, 18(10):1967–1980, 2019.

[6] Roland Bruderer, Oliver M. Bernhardt, Tejas Gandhi, Yue Xuan, Julia Sondermann, Manuela Schmidt, David Gomez-Varela, and Lukas Reiter. Optimization of experimental parameters in data-independent mass spectrometry significantly increases depth and reproducibility of results. *Molecular & cellular proteomics : MCP*, 16(12):2296–2309, Dec 2017.

[7] Jonathan Burbaum and Gabriela M Tobal. Proteomics in drug discovery. *Current Opinion in Chemical Biology*, 6(4):427–433, 2002.

[8] Matthew C. Chambers, Brendan Maclean, Robert Burke, Dario Amodei, Daniel L. Ruderman, Steffen Neumann, Laurent Gatto, Bernd Fischer, Brian Pratt, Jarrett Egertson, Katherine Hoff, Darren Kessner, Natalie Tasman, Nicholas Shulman, Barbara Frewen, Tahmina A. Baker, Mi-Youn Brusniak, Christopher Paulse, David Creasy, Lisa Flashner, Kian Kani, Chris Moulding, Sean L. Seymour, Lydia M. Nuwaysir, Brent Lefebvre, Frank Kuhlmann, Joe Roark, Paape Rainer, Suckau Detlev, Tina Hemenway, Andreas Huhmer, James Langridge, Brian Connolly, Trey Chadick, Krisztina Holly, Josh Eckels, Eric W. Deutsch, Robert L. Moritz, Jonathan E. Katz, David B. Agus, Michael MacCoss, David L. Tabb, and Parag Mallick. A cross-platform toolkit for mass spectrometry and proteomics. *Nature Biotechnology*, 30(10):918–920, Oct 2012.

[9] Igor V. Chernushevich, Alexander V. Loboda, and Bruce A. Thomson. An introduction to quadrupole–time-of-flight mass spectrometry. *Journal of Mass Spectrometry*, 36(8):849–865, 2001.

[10] Jürgen Cox and Matthias Mann. Maxquant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26(12):1367–1372, Dec 2008.

[11] Robertson Craig and Ronald C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 02 2004.

[12] Joshua E Elias and Steven P Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods*, 4(3):207–214, 2007.

[13] Jimmy K Eng, Michael R Hoopmann, Tahmina A Jahan, Jarrett D Egertson, William S Noble, and Michael J MacCoss. A deeper look into comet—implementation and features. *Journal of the American Society for Mass Spectrometry*, 26(11):1865–1874, 2015.

[14] Jimmy K. Eng, Tahmina A. Jahan, and Michael R. Hoopmann. Comet: An open-source ms/ms sequence database search tool. *PROTEOMICS*, 13(1):22–24, 2013.

[15] Kristian Flikka, Lennart Martens, Joël Vandekerckhove, Kris Gevaert, and Ingvar Eidhammer. Improving the reliability and throughput of mass spectrometry-based proteomics by spectrum quality filtering. *Proteomics*, 6:2086–94, 04 2006.

[16] Ari Frank and Pavel Pevzner. Pepnovo: de novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, 77(4):964–973, Feb 2005.

[17] Kris Gevaert and Joël Vandekerckhove. Protein identification methods in proteomics. *ELECTROPHORESIS*, 21(6):1145–1154, 2000.

[18] Soroosh Gholamizoj, Qixin Liu, Noah Reinhardt, Iain Rogers, and Bin Ma. Automatic detection of the protease used in bottom-up proteomics experiments. *Proceedings of the 70th ASMS Conference on Mass Spectrometry and Allied Topics*, June 2022.

[19] Soroosh Gholamizoj and Bin Ma. SPEQ: quality assessment of peptide tandem mass spectra with deep learning. *Bioinformatics*, 38(6):1568–1574, 01 2022.

[20] Anton A. Goloborodko, Lev I. Levitsky, Mark V. Ivanov, and Mikhail V. Gorshkov. Pyteomics—a python framework for exploratory data analysis and rapid software prototyping in proteomics. *Journal of the American Society for Mass Spectrometry*, 24(2):301–304, Feb 2013.

[21] Sam Hanash. Disease proteomics. *Nature*, 422(6928):226–232, Mar 2003.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[23] Patricia Hernandez, Markus Müller, and Ron D. Appel. Automated protein identification by tandem mass spectrometry: Issues and strategies. *Mass Spectrometry Reviews*, 25(2):235–254, 2006.

[24] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, 1998.

[25] Shantanu Jain, Martha White, and Predrag Radivojac. Recovering true classifier performance in positive-unlabeled learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2066–2072. AAAI Press, 2017.

[26] Andrew Keller, Alexey I. Nesvizhskii, Eugene Kolker, and Ruedi Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Analytical Chemistry*, 74(20):5383–5392, Oct 2002.

[27] Yong Kil, Christopher Becker, Wendy Sandoval, David Goldberg, and Marshall Bern. Preview: A program for surveying shotgun proteomics tandem mass spectrometry data. *Analytical Chemistry*, 83(13):5259–5267, 2011.

[28] Sangtae Kim and Pavel A. Pevzner. Ms-gf+ makes progress towards a universal database search tool for proteomics. *Nature communications*, 5:5277–5277, Oct 2014.

[29] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[30] Bin Ma. Novor: Real-time peptide de novo sequencing software. *Journal of The American Society for Mass Spectrometry*, 26(11):1885–1894, Nov 2015.

[31] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. Peaks: powerful software for peptide de novo se-

quencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 17(20):2337–2342, 2003.

[32] Ze-Qiang Ma, Matthew Chambers, Amy Ham, Kristin Cheek, Corbin Whitwell, Hans-Rudolf Aerni, Birgit Schilling, Aaron Miller, Richard Caprioli, and David Tabb. Scanranker: Quality assessment of tandem mass spectra via sequence tagging. *Journal of proteome research*, 10:2896–904, 04 2011.

[33] Brian McDonagh, Giorgos K Sakellariou, Neil T Smith, Philip Brownridge, and Malcolm J Jackson. Differential cysteine labeling and global label-free proteomics reveals an altered metabolic state in skeletal muscle aging. *Journal of proteome research*, 13(11):5008—5021, November 2014.

[34] W. Hayes McDonald and John R. Yates III. Shotgun proteomics and biomarker discovery. *Disease Markers*, 18:99–105, 2002. 2.

[35] Johra Muhammad Moosa, Shenheng Guan, Michael F. Moran, and Bin Ma. Repeat-preserving decoy database for false discovery rate estimation in peptide identification. *Journal of Proteome Research*, 19(3):1029–1036, Mar 2020.

[36] Seungjin Na and Eunok Paek. Quality assessment of tandem mass spectra based on cumulative intensity normalization. *Journal of proteome research*, 5:3241–8, 01 2007.

[37] Alexey I. Nesvizhskii, Franz F. Roos, Jonas Grossmann, Mathijs Vogelzang, James S. Eddes, Wilhelm Gruissem, Sacha Baginsky, and Ruedi Aebersold. Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data: Toward more efficient identification of post-translational modifications, sequence polymorphisms, and novel peptides*. *Molecular & Cellular Proteomics*, 5(4):652–670, 2006.

[38] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[39] Sheng Pan, Ruedi Aebersold, Ru Chen, John Rush, David R. Goodlett, Martin W. McIntosh, Jing Zhang, and Teresa A. Brentnall. Mass spectrometry based targeted protein quantification: Methods and applications. *Journal of Proteome Research*, 8(2):787–797, Feb 2009.

[40] Akhilesh Pandey and Matthias Mann. Proteomics to study genes and genomes. *Nature*, 405(6788):837–846, Jun 2000.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[42] David N. Perkins, Darryl J. C. Pappin, David M. Creasy, and John S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *ELECTROPHORESIS*, 20(18):3551–3567, 1999.

[43] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[44] Dennis W Ruck, Steven K Rogers, and Matthew Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[45] Jussi Salmi, Robert Moulder, Jan-Jonas Filén, Olli S. Nevalainen, Tuula A. Nyman, Riitta Lahesmaa, and Tero Aittokallio. Quality classification of tandem mass spectrometry data. *Bioinformatics*, 22(4):400–406, 12 2005.

[46] Richard Alexander Scheltema, Jan-Peter Hauschild, Oliver Lange, Daniel Hornburg, Eduard Denisov, Eugen Damoc, Andreas Kuehn, Alexander Makarov, and Matthias Mann. The q exactive hf, a benchtop mass spectrometer with a pre-filter, high-performance quadrupole and an ultra-high-field orbitrap analyzer*. *Molecular & Cellular Proteomics*, 13(12):3698–3708, 2014.

[47] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[50] Hanno Steen and Matthias Mann. The abc's (and xyz's) of peptide sequencing. *Nature Reviews Molecular Cell Biology*, 5(9):699–711, Sep 2004.

[51] Stephen Stein. Nist libraries of peptide fragmentation mass spectra, nist standard reference database 1 c, 2008.

[52] Juan A. Vizcaíno, Eric W. Deutsch, Rui Wang, Attila Csordas, Florian Reisinger, Daniel Ríos, José A. Dianes, Zhi Sun, Terry Farrah, Nuno Bandeira, Pierre-Alain Binz, Ioannis Xenarios, Martin Eisenacher, Gerhard Mayer, Laurent Gatto, Alex Campos, Robert J. Chalkley, Hans-Joachim Kraus, Juan Pablo Albar, Salvador Martinez-Bartolomé, Rolf Apweiler, Gilbert S. Omenn, Lennart Martens, Andrew R. Jones, and Henning Hermjakob. Proteomexchange provides globally coordinated proteomics data submission and dissemination. *Nature Biotechnology*, 32(3):223–226, Mar 2014.

[53] Fang-Xiang Wu, Pierre Gagné, Arnaud Droit, and Guy G. Poirier. Quality assessment of peptide tandem mass spectra. *BMC Bioinformatics*, 9(6):S13, May 2008.

[54] Hao Yang, Hao Chi, Wen-Feng Zeng, Wen-Jing Zhou, and Si-Min He. pnovo 3: precise de novo peptide sequencing using a learning-to-rank framework. *Bioinformatics (Oxford, England)*, 35(14):i183–i190, Jul 2019. 31510687[pmid], PMC6612832[pmcid], 5529238[PII].

[55] Jing Zhang, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng Zhang, Gilles A Lajoie, and Bin Ma. Peaks db: de novo sequencing

assisted database search for sensitive and accurate peptide identification. *Molecular & cellular proteomics*, 11(4), 2012.