Ollscoil na Gaillimhe
University of Galway

School of Computer Science
College of Science and Engineering
University of Galway

Machine Learning for *De Novo* Peptide Identification

Kevin McDonnell

Supervised by:

Dr. Enda Howley and Dr. Florence Abram

A thesis submitted for the degree of Doctor of Philosophy

January 2023

Abstract

Proteomics involves the identification and analysis of proteins, therefore providing valuable insight into ecosystem functioning. In this methodology, protein sequences are typically identified using a bottom-up approach whereby short subsequences called peptides are matched to experimental mass spectra using a database search. However, it is reported that on average, 75% of the spectra recovered from experiments remain unidentified. *De novo* peptide identification is an alternative approach to database searching that uses only the spectrum to identify the peptide sequence. This method has undergone significant recent improvements, in part due to the integration of machine learning models into the algorithms.

This thesis explores the strengths and weaknesses of many of the current state-of-the-art *de novo* peptide identification algorithms through an extensive evaluation. As understanding the underlying data is key to this analysis, a comprehensive survey of the characteristics of tandem mass spectra is included alongside the performance of the algorithms. An alternative machine learning architecture is then proposed to address the weaknesses found. The proposed novel CNN-GNN peptide ion encoding module was able to identify more peptide ions than the encoding modules used by state-of-the-art *de novo* peptide identification algorithms in all datasets tested. Finally, the utility of artificial data in the context of *de novo* peptide identification is explored. Artificial spectra were found to be missing critical noise that was present in real data. However, the quantification and introduction of this noise into to artificial spectra increased their similarity to real spectra, significantly improving their potential for use in the training and testing of models. Based on the results of this thesis we recommend specific research avenues for the design and development of the next generation of *de novo* peptide identification algorithms. This thesis not only demonstrates the challenges facing *de novo* peptide identification, but also takes the critical first steps toward overcoming them.

Acknowledgements

Firstly I would like to thank my two supervisors, Dr Enda Howley and Dr Florence Abram. Your advice, support and guidance throughout the last few years has been invaluable. I would also like to thank you for the fun and enthusiasm you brought to the whole process. I hope that this is only the start of our collaboration together.

To those who have passed through Room 307 during my time there, I would like to thank you all. The journey was made much more enjoyable by your friendship and discussion. I was also fortunate to have the support of those in the FEM Lab in Microbiology. The long debates and discussions we had at meetings were always entertaining.

Finally I would like to thank my family. To my parents Geraldine and Michael, I am forever grateful for the love and support you have given me. I could not have not have completed this without you. To Marie and John, thank you for always being there for me and looking out for your little brother.

Contents

List of Figures

List of Tables

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of Galway. This thesis is the result of my own investigations.

Some of the material contained in this thesis has appeared in the following published papers:

1. McDonnell, K., Howley, E., and Abram, F. The impact of noise and missing fragmentation cleavages on *de novo* peptide identification algorithms. Computational and Structural Biotechnology Journal 20 (2022).

2. McDonnell, K., Abram, F., and Howley, E. Application of a novel hybrid CNN-GNN for peptide ion encoding. Journal of Proteome Research (2022).

3. McDonnell, K., Howley, E., and Abram, F. Critical evaluation of the use of artificial data for machine learning based *de novo* peptide identification. Computational and Structural Biotechnology Journal (2023)

Chapter 1

Introduction

1.1 Motivation

Proteomics provides insight into the functional profile of a microbial system through the analysis of proteins [14]. Proteins perform almost all of the functions of a cell from signalling to providing structure [187]. Changes in protein abundance can indicate changes in the environment such as stress to a microbial system [94]. As such, identifying what proteins are being expressed in a cell can help in our understanding of diseases such as cancer [46]. Fundamental to this approach is the quality of the data and the accuracy of the tools used to analyse it [191]. In a proteomics analysis pipeline, proteins are typically digested down into smaller subsequences called peptides before being analysed via tandem mass spectrometry. These smaller sequences have several advantages such as being easier to both fragment and ionise [272]. The spectra which are recovered in this bottom-up approach are then matched to peptides in a protein database [265]. This database contains the possible proteins from organisms that could be present in the sample. The protein sequences in the database are then subjected to *in silico* enzymatic digestion. For each resulting peptide, a list of all possible fragment ions is generated, thereby creating a theoretical spectrum. Observed spectra are labelled with a peptide by finding the closest matching theoretical spectrum and using a scoring function to distinguish the true matches from the false matches [181]. This database search methodology has its limitations however, as on average only 25% of spectra acquire significant peptide matches [95]. This is in part due to the fact that larger protein databases have a greater chance of leading to a random peptide match [150]. Conversely, if a database is too small, the significance estimate can be inaccurate leading to an overestimation of significant matches [176].

De novo peptide identification is a database free alternative whereby algorithms pre-

dict the associated peptide using the spectrum alone [56]. Not being reliant on a database means it has important applications where no reference peptide sequence is available, such as the identification of neoantigens that can be used in immunotherapies [240].

While database searching is still the more popular approach, *de novo* identification has had a lot of recent growth, stimulated in part by improvements in performance due to machine learning [177]. *De novo* peptide identification does not face the same bottlenecks as database searching and so if the current trend continues it could soon become a competitive alternative. It can also be combined with database search methods to improve sensitivity [271].

Machine learning is a process whereby models are designed to learn from data without being explicitly programmed to do so [170]. Prediction tasks generally involve supervised machine learning where the model is trained to predict the desired label/output given an associated set of features. In the context of peptide identification, machine learning models have been used to both predict the spectrum given the peptide sequence [237] as well as identify the peptide sequence given the tandem mass spectrum [241]. These two applications involve the same data but with their features and labels exchanged. Spectrum prediction models have been used to increase the sensitivity of database searches by providing more realistic theoretical spectra with which to compare to the observed spectra [89]. Machine learning models that help to predict the peptide from the spectrum are incorporated all of the current state-of-the-art *de novo* peptide identification algorithms [153, 241, 199]. Novor [153] uses random forests to identify fragmentation sites while DeepNovo [241] and PointNovo [199] employ deep learning to predict the next amino acid in the sequence.

Despite there being different algorithms and approaches for *de novo* peptide identification, there have been very few independent evaluations. Self reported performance is typically limited to a small set of metrics with little exploration on the strengths and limitations of the different algorithms. There is therefore little information available for new research groups to develop novel *de novo* peptide identification algorithms. With machine learning advancing at an exponential rate, the integration of newly developed approaches to *de novo* algorithms will be key to the field's continued improvement. For that to happen, researchers must have an understanding of the algorithms, the data, and how they both interact.

The focus of this thesis is to identify and address some of the challenges associated with *de novo* peptide identification. Specifically, it is to characterise tandem mass spectra in the context of *de novo* peptide identification and understand how we can design better algorithms to perform this task in the future. Understanding the data is key to the design of effective *de novo* peptide identification algorithms and so analysis of real spectra is fundamental to this work. Furthermore, the analysis described in this thesis is

extended to artificial spectra and how they can be used to benefit *de novo* peptide identification. Artificial spectra have previously been used to evaluate *de novo* algorithms but the significance of these results to real data performance has not yet been established. The thesis also includes the first use of graph neural networks (GNNs) in the context of *de novo* peptide identification. A novel architecture, incorporating GNNs, is proposed for the encoding of peptide ions in tandem mass spectra. The performance of this model is then compared to the encoding modules of three state-of-the-art *de novo* peptide identification algorithms. With many diverse and important applications, improvements to *de novo* peptide identification will increase both the utility and adoption of this methodology in future. This thesis provides a foundation for the development of new and improved approaches to *de novo* peptide identification.

1.2 Research Questions

The aim of this thesis is to address the following research questions:

1. What are the main challenges to *de novo* peptide identification?
2. Can we design better encoding modules to address these challenges?
3. Can artificial spectra be leveraged to aid the training and evaluation of *de novo* peptide identification algorithms?

1.3 Hypotheses

Based on the research questions described above I expect to show that:

1. Missing fragmentation cleavages are a major challenge to current *de novo* peptide identification algorithms.
2. Combining the graph-like structure of the data with deep learning through GNNs can provide better peptide ion encoding.
3. Introducing noise to artificial spectra can provide ground truth data for *de novo* peptide identification algorithm evaluation. It could also provide difficult-to-classify examples which are currently rare in the training data.

1.4 Thesis Overview

The next chapter presents a comprehensive summary of the research topics relevant to this research. It provides a background in proteomics and *de novo* peptide identification

as well as machine learning.

Chapter 3 explores the prevalence of noise peaks and missing fragment ion peaks in tandem MS spectra, two of the biggest challenges in peptide identification. An evaluation of current state-of-the-art *de novo* peptide identification algorithms is performed including an analysis of how these challenges affect their performance. Possible solutions to the limitations observed are also proposed.

Chapter 4 presents a novel peptide ion encoding module based on graph neural networks (GNNs). The proposed module is compared to encoding modules employed by other state-of-the-art *de novo* peptide identification algorithms over a range of datasets.

The focus of chapter 5 is to perform a critical evaluation of artificial peptide spectra and their use in the training and evaluation of *de novo* peptide identification algorithms. This includes the quantification of the different types of noise and variability in real spectra and how they could be used to improve the utility of artificial spectra.

Chapter 6 concludes the thesis with a summary of the main contributions as well as a discussion of the limitations of this work and an outline of potential directions for future research.

Chapter 2

Background

2.1 Proteomics

2.1.1 Background

Proteins are large molecules that act like molecular machines and are essential to life on earth [82]. They help carry out numerous functions such catalysing reactions, providing structure to cells and transporting molecules [149]. The information needed to create proteins is coded for by genes in DNA. Following the activation of transcription regulators, sequences of DNA are transcribed into a complementary RNA strand [186]. The nucleic acids that make up the RNA strand are then translated into a protein sequence. Proteins are made up of long chains of building blocks called amino acids. It takes three nucleic acids in a coding region of a gene to encode one amino acid.

There are twenty different types of amino acid, each with unique properties defined by a distinct side chain called an R-group. The amino acids bind together to form long chains. Each protein is defined by the specific order of its constituent amino acids called its primary sequence. The order is defined from the NH_2 (N-terminus) end to the COOH (C-terminus) end. These long chains fold into secondary structures such as helices or sheets (Figure 2.1). The interactions between the amino acids form these secondary bonds creating a unique structure for each unique sequence [11]. The secondary structures further fold up into a protein's tertiary shape. Many proteins can also then interact and bond with one another to give what is known as a quaternary structure. The final 3-dimensional shape of a protein defines its function.

The protein expression of a cell is directly correlated to its environment and the associated stresses [4]. Therefore protein expression profiling can provide a snapshot into ecosystem functioning. Furthermore, in complex environments where many different

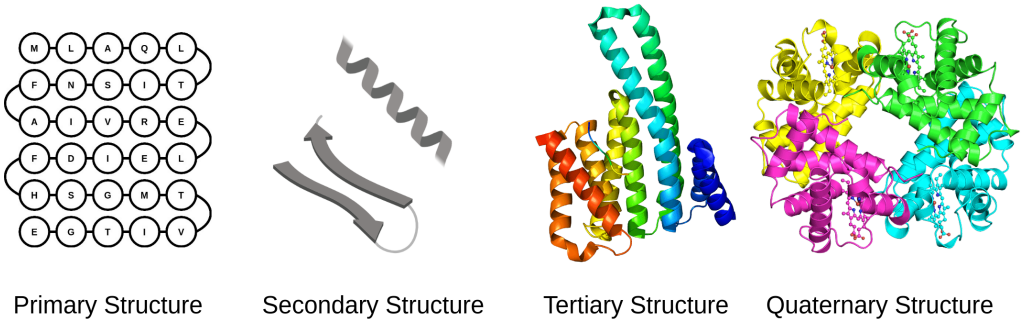

Figure 2.1: The different levels of protein structure.

types of organisms are present protein identification can inform both what functions are being carried out as well as the organism responsible [1]. Just as the genome is the full complement of genetic information of an organism, the proteome is the protein complement expressed by the genome [255]. Similarly, the analysis of the proteome is known as proteomics.

As proteins perform so many molecular functions, understanding and characterising the proteome is a key part of understanding biological systems. Proteomics has therefore applications in a huge variety of areas. For example, the alteration of a system's proteome could be a sign of disease or stress [207]. In the area of healthcare, proteomics can aid in the understanding and diagnosis of many diseases including IBD [62], Alzheimer's disease [244], general morbidity [233] and many heart conditions [147, 156, 6]. It can also be used to identify disease causing bacteria and their antibiotic resistance [29, 217] as well as the interaction between a virus and its host [242, 61]. This kind of research has had huge implications in terms of public health with the recent COVID-19 pandemic caused by the SARS-CoV-2 virus [142].

Proteomics is particularly useful in cancer research where the altered genome of the cancer cell will express a different proteomic profile [137]. Protein biomarkers can be used to help diagnose cancer [210] while cancer cells will respond differently to a drug depending on their profile of expressed proteins [49]. Cancer immunotherapy is a treatment whereby the body's own immune system is used to target the cancer cells [213]. Identification of proteins unique to the cancer cell have been shown to provide a viable target to trigger an immune response [127].

Most microorganism do not live in isolation but as part of complex communities [212]. Meta-omics technologies involve the analysis of these complex communities where the many different types of organisms both compete and cooperate [214]. As proteins are expressed by a cell in response to stimuli and the current needs of the cell, their characterisation allows researchers to see what cells are doing at a particular moment in

time. This is particularly useful in complex communities. While they may perform similar functions, proteins expressed by different organisms vary slightly. Proteomic profiling can therefore show both the constituents of a community alongside the functions each taxon is performing [1]. It may happen that the taxonomic profile remains constant under some stress but the functional profile changes significantly [163].

2.1.2 Methodology

Proteomics can be performed in both a top-down and bottom-up approach. In top down proteomics, proteins are isolated intact before analysis. Then tandem mass spectrometry (MS) is used to fragment the proteins and create a fragmentation spectrum. Proteins can then be identified based on the characteristics of these spectra. However, this strategy is not straightforward due to difficulties in the fractionation, ionization and fragmentation of complete proteins [272].

Bottom-up proteomics has therefore become the more popular approach. It is also called shotgun proteomics, based on its similarities to shotgun sequencing where small DNA fragments are recombined to determine the sequence [91]. In shotgun proteomics and in contrast to the top-down approach, proteins are first digested down into smaller fragments called peptides. The proteins are typically digested using trypsin but other enzymes can also be used [243]. Trypsin cleaves the protein at the C-terminus side of arginine (R) and lysine (K) except when either is followed by a proline (P). This results in a set of peptides where the majority are less than 30 amino acids in length [239]. After digestion, the peptides are fractionated by liquid chromatography (LC). Here the peptides are separated based on their affinity to the mobile phase of the LC column [225]. Similar peptides will therefore pass through at similar rates before entering the next phase. Following the LC column, the now separated peptides enter into a tandem mass spectrometer. This further isolates the peptides by mass before they are fragmented and a unique signature for each acquired [265]. Tandem mass spectrometry will be further discussed in the next section. The spectra produced must then be matched to the originating peptide sequence. This is typically done by comparing the observed spectra to theoretical spectra created from the possible peptides in a protein database.

Shotgun proteomics is the preferred method of analysis as peptides are more soluble and easier to separate than intact proteins [38]. The technique makes it possible to produce large amounts of high quality data but the approach has some computational bottlenecks. The top down approach of proteomics aims to match the observed mass spectra to theoretical spectra created using a protein database. The bottom-up approach introduces another layer of complexity as spectra need to first be mapped to peptides and then to proteins. Two problems that arise from this are the uncertainty when reconnecting

the identified peptides to proteins as well as the amplification of error rates when going from peptide to protein level identification [181]. While the list of peptides is created from a protein database, there is not necessarily a unique mapping between the two [181]. A Peptide can be shared between multiple proteins leading to ambiguity as to which it belongs to. Some peptides are unique to one protein. However, a match with such a peptide does not guarantee the presence of the protein as the algorithms used are far from perfect [118]. Improvements to analysis pipelines are evidently needed, however understanding of the data and how it is created is an essential first step in this process [166].

2.1.3 Mass Spectrometry

Mass spectrometry is now the method of choice for proteomics analysis [181]. New technologies have seen the quality, scale and diversity of the data explode recently. This however has led to challenges, as the analysis pipelines have yet to catch up [253].

Mass spectrometers are a well established technology and have been used in research for many decades [27]. There are now many different types available but they all work on roughly the same principle [277]. A sample is first converted to a gas and ionised (charged) so that it can be accelerated using an electromagnetic field. The acceleration due to the field is inversely proportional to the mass-to-charge ratio (m/z) of the ion. Given the same electromagnetic field heavier ions will accelerate more slowly than lighter ions. Similarly, ions with lower charge will accelerate more slowly than ions with greater charge. These properties are utilised to separate the ions by their m/z value. A detector then measures the different ion abundances and converts them into a spectrum.

The spectrum is a 2-dimensional representation of the observations of the detector with m/z units on the x-axis and intensity on the y-axis. The intensity (height) of the peak indicates the frequency/relative abundance with which that m/z value was detected. No measurement is perfect and so raw MS spectra will not appear as perfect zero-width peaks. Instead, raw spectra must be converted into a list of m/z values corresponding to the centroid of each observed peak. Spectra can also be filtered by eliminating low intensity peaks that are indistinguishable from random noise [10]. The preprocessing method performed can have a large influence on the quality of the results [203]. Following this, mass spectra are then represented by a 2-dimensional list of m/z and intensity pairs.

Tandem mass spectrometry, as the name suggests, uses two mass analysers in series [164]. A sample is ionised and passed into the first mass analyser. Here the ions are separated by m/z so that specific m/z values can be targeted. Using the first mass analyser a specific m/z range is isolated so that only ions within this range remain. This generally targets a singular peak which is then called the parent ion. In the context

of bottom-up proteomics, a peak in the initial mass scan can represent a single peptide due to the combined separation of both the LC column and mass analyser. The isolated peptides then pass to a collision chamber where they are fragmented (see Section 2.1.4). Finally, these fragments pass through a second mass analyser to create a tandem mass spectrum. The peaks now represent the relative abundance of the different fragments of the parent ion. The spectrum also has associated meta data including the m/z and charge of the parent ion, which can also be used by to infer the originating peptide.

2.1.4 Peptide Fragmentation

Peptide fragmentation happens in the collision chamber of a tandem mass spectrometer, between the two mass analysers. It can be done through collision induced dissociation (CID) where the peptides are accelerated into, and collided with, neutral molecules causing them to break into smaller fragments [252]. New technologies provide much greater m/z precision through higher-energy dissociation (HCD) [185]. This method of fragmentation typically results in cleavage at the peptide (CO-NH) bonds in the amino acid chain. As shown in Figure 2.2, these fragments are called b and y ions. Other ions (a,c,x and z) can also occur through the cleavage of different bonds in the chain, but do so in lower frequencies in HCD data. Fragment ions can also lose neutral molecules such as water or ammonia shifting their m/z value by the corresponding mass.

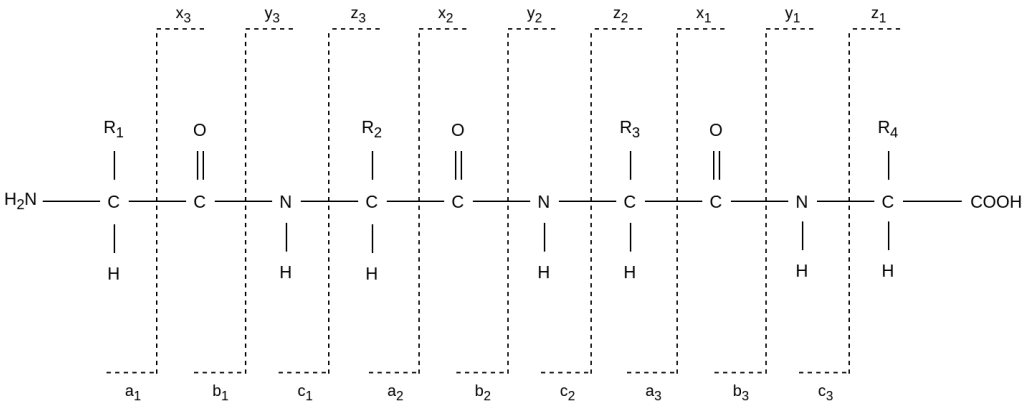

Figure 2.2: Common nomenclature for the possible backbone ions from peptide fragmentation. The chemical structure of a four amino acid peptide is shown. The dotted lines indicate the possible cleavages. N-terminus fragments are listed along the bottom with C-terminus fragments along the top. R_n indicates the side chain of the n^{th} amino acid in the sequence.

While the collision of molecules with the peptide during fragmentation is random, the spectrum that is created is dependant on features of both the data and the experimental

setup. The length of a peptide sequence, its sequence of amino acids, and the position of the bond along the sequence where the cleavage occurs all influence the relative abundance of fragment ions [189, 237]. Fragmentation will occur more often in more energetically favourable scenarios [189]. These will be observed as higher intensity spectrum peaks. Furthermore, the type of mass spectrometer and fragmentation method will also change the fragmentation patterns observed with different mass ranges preferred by different setups [57, 73].

Due to these dependencies, peptides will exhibit partially consistent fragmentation patterns unique to their amino acid sequence and the experimental setup. The spectrum will contain high intensity peaks corresponding to the fragments created from the different cleavages of the peptide (Figure 2.3). These can then be used to deduce the unique sequence of the originating peptide.

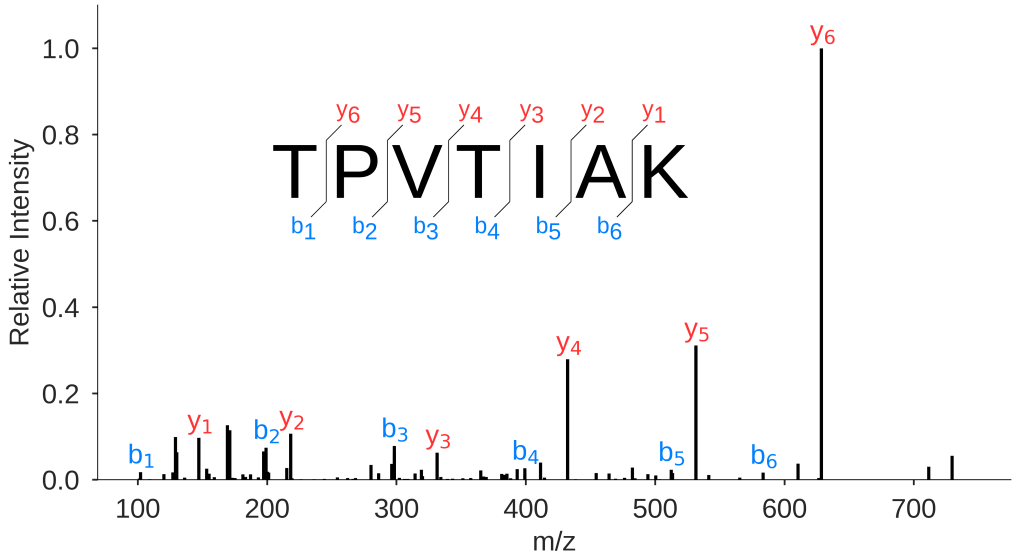

Figure 2.3: Tandem mass spectrum of the TPVTIAK peptide. The peptide and possible cleavages is shown above the spectrum. Matched ions are labelled with the corresponding fragment.

As the mass of each amino acid is known the theoretical m/z values for each fragment can be calculated (Table 2.2). Theoretical spectra can be created from query peptides in a database and compared to the observed spectrum. The peptide can also be reconstructed *de novo*, without the use of a database by identifying the fragment ions through their mass differences alone. However, adding to the complexity of the problem is the possibility of the peptide to undergo post-translational modifications (PTMs). PTMs are alterations to the chemical makeup of a protein [151]. For example, methylation involves

the addition of a methyl group to the side chain of an amino acid. PTMs expand the functional capabilities of proteins beyond the standard amino acids and can be reversible or irreversible. Recent studies have shown that they are involved in the regulation of almost all cellular events [251]. Detection of PTMs from tandem MS is possible but difficult as increasing the number of possible amino acid masses exponentially expands the search space [45].

Amino Acid	Symbol	Mass (Da)	Composition
Glycine	G	57.02146	C_2H_3NO
Alanine	A	71.03711	C_3H_5NO
Serine	S	87.03203	$C_3H_5NO_2$
Proline	P	97.05276	C_5H_7NO
Valine	V	99.06841	C_5H_9NO
Threonine	T	101.04768	$C_4H_7NO_2$
Cysteine	C	103.00919	$C_3H_7NO_2S$
Isoleucine	I	113.08406	$C_6H_{11}NO$
Leucine	L	113.08406	$C_6H_{11}NO$
Asparagine	N	114.04293	$C_4H_6N_2O_2$
Aspartic Acid	D	115.02694	$C_4H_5NO_3$
Glutamine	Q	128.05858	$C_5H_8N_2O_2$
Lysine	K	128.09496	$C_6H_{12}N_2O$
Glutamic Acid	E	129.04259	$C_5H_7NO_3$
Methionine	M	131.04049	C_5H_9NOS
Histidine	H	137.05891	$C_6H_7N_3O$
Phenylalanine	F	147.06841	C_9H_9NO
Arginine	R	156.10111	$C_6H_{12}N_4O$
Tyrosine	Y	163.06333	$C_9H_9NO_2$
Tryptophan	W	186.07931	$C_{11}H_{10}N_2O$

Table 2.1: Amino acid masses and chemical composition.

2.1.5 Database Searching

Database searching is currently the most popular way of identifying peptides in tandem mass spectra. With the exponential increase in the availability of genome sequences, there is now an abundance of protein sequences available [50]. The first step in a database search pipeline is the selection of the appropriate protein sequence database. The database should include all possible proteins that could be found in the sample [180]. For example, for a sample from a pure culture of human cells, one could use the complete human proteome as a database. For unknown samples, one could choose a more general database containing proteins form hundreds or even thousands of different organisms. Databases can be downloaded from online collections such as UniProt which contains millions of proteins and thousands of proteomes, both reviewed and unreviewed [50].

Ion type	**Mass Calculation**
a	$\sum$ AAs - CO + H
b	$\sum$ AAs + H
c	$\sum$ AAs + NH_3 + H
x	$\sum$ AAs + CO + OH
y	$\sum$ AAs + OH
z	$\sum$ AAs - NH_3 +OH
Ion Loss	
H_2O	ion - H_2O
NH_3	ion - NH_3
Internal Fragments	
a	$\sum$ AAs - CO + H
b	$\sum$ AAs + H

Table 2.2: Mass calculation for the different peptide fragment ion types present in tandem mass spectra.

Once a database is defined, the proteins in the database are used to create all possible peptides through an artificial enzymatic digestion. For example, if trypsin was used in the experiment, the proteins would be digested using the known rules outlined earlier. The mass of each of the peptides is also calculated. For a given spectrum, a list of possible peptides is generated from the database by extracting those that match the parent ion mass within a given tolerance [130]. The possible fragment ion peaks for each of the candidate peptides are calculated to create a theoretical spectrum for each peptide. The peaks in each theoretical spectrum are compared to the peaks in the real spectrum with each peptide given a score based on their similarity [69]. Peptides whose theoretical spectra score sufficiently high enough are considered correct matches.

While the technology used in creating the data has seen revolutionary changes in the last 20 years, the methodology and algorithms used have remained largely the same [249]. X!Tandem is one such database search algorithm [52]. It calculates a hyperscore for each peptide-spectrum pair to differentiate correct matches from incorrect ones [72]. For each theoretical peptide, the set of possible peaks $P \in \{0, 1\}$ is predicted. The correlation between the theoretical peaks and the spectrum is calculated by summing the intensities I of the real peaks that match within a given threshold. This is equivalent to a dot product between the two spectra. Finally, this product is multiplied by the factorial of both the number of matched b ions (n_b) and the number of matched y ions (n_y).

$$hyperscore = (n_b! n_y!) \sum_{i=0}^{n} I_i P_i \tag{2.1}$$

A survival function is then defined based on the distribution of the hyperscores of all possible peptides with a particular spectrum [72]. This defines the probability that a spectrum's hyperscore will be higher than a given value by random matching. Using the survival function, the highest scoring peptide for each spectrum is given an expectation value (e-value). The e-value is an indication of how many peptides would have at least that score. Therefore lower e-values indicate that the hyperscore is less likely to have occurred by chance. This can be used to differentiate random matches from true matches.

Databases have seen a large increase with more and more genomes being sequenced. Large databases are a problem for database searches as they increase the probability of a random match [150]. A good scoring function should separate random matches from true matches with correct matches scoring higher. Ideally, a score threshold could then be set above which a match is deemed correct. However, the exact error rate is difficult to determine and so the ideal threshold is not known.

This problem is addressed by estimating the false discovery rate (FDR) by using the scoring function on a decoy database [121]. A decoy database is created by reversing or shuffling the sequences of the actual database being used. This creates a database with the same distribution of peptide lengths and masses but which share few or no peptide sequences [67]. The theoretical spectra from the decoy peptides are then scored against the observed spectra as before. As these decoy peptides are known not to exist, the distribution of scores from the decoy database is assumed to match that of the random false peptide matches. A score threshold is then typically set so that only 1% of peptides above this come from the decoy database giving an estimated FDR of 1%. Large databases increase chance of a random match meaning for the same score threshold there will be more incorrect matches. Therefore to maintain a 1% FDR the minimum threshold must be increased. Correctly matched peptides will then be discarded as they now fall below the accepted threshold. On average, as little as 25% of spectra obtain significant peptide matches [95]. This problem is even more profound for metaproteomics analysis [178]. Metaproteomics involves the functional profiling of mixed communities, where the the databases are much larger due to the increased species diversity.

As the difficulty in search spaces increases exponentially with greater numbers of options, only limited amino acid modifications are considered in database searches. This is despite the fact that PTMs are are ubiquitous in proteins [251]. Increasing the number of amino acids to look for increases both the FDR and runtime of search algorithms [5]. Typically only common modifications such as oxidation (+16 Da) of methionine are considered. The amino acid cysteine is generally modified through carbamidimethylation (+57 Da) during the experimental process and so is only considered in this form [30].

2.1.6 De Novo Peptide Identification

De novo peptide sequencing is the identification of peptides without the use of a database [56]. This strategy relies on the spectrum alone, using the relationship between peaks to deduce the amino acid sequence. Singly charged ions of the same type from neighbouring cleavages will produce peaks separated by the mass of the amino acid between them (Figure 2.2). Ions of different types can be used also (Table 2.2). Moving from peak to peak, the peptide can be built up, one amino acid at a time.

The practice of *de novo* peptide identification started out as the manual labelling of fragment ions. Researchers would assemble the ion series, and therefore the peptide, using a set of learned rules. This process was eventually automated but would still require manual checking due to the inconsistencies of the results particularly at the ends of the peptide [216].

Since then there have been multiple algorithmic approaches to the *de novo* peptide identification problem. Many of these approaches try to model the fragmentation process to aid in their attempt to identify the peptide sequence [7]. If the mapping from peptide to spectrum can be learned, that information can then be used in the reverse mapping using the spectrum to recreate the peptide.

Bartels [18] introduced the idea of modelling the mass spectrum as a graph. In this approach, nodes corresponding to the different ion types are created for each peak. Edges are created between nodes where the mass difference between them is equal to that of an amino acid. As the series of peptide fragment ions are separated by the mass of the constituent amino acids, ions from neighbouring cleavages will be connected in the graph. Provided all cleavages are represented by fragments in the spectrum, the edges of a path through the graph will give a candidate peptide. If peaks are missing they can be represented by a mass-gap. Each node in the graph is given a score based on the fragmentation probability of the associated cleavage [223]. Dynamic programming is then used to find the highest scoring path through the graph. The edges of the highest scoring path will describe the most likely amino acid sequence.

PepNovo updated this methodology with an improved probabilistic scoring function [78]. In this method a probabilistic network is created to model the interaction between amino acids and peaks. The model was trained using database PSMs to learn the weights in the network. Again, a spectrum graph was created, however this time with each vertex scored using the probabilistic network. Finally dynamic programming was used to find the highest scoring antisymmetric path in the graph. The authors require the path to be antisymmetric as it may be possible in the graph to travel partly along the b-ion series before travelling back along the y-ion series created by the same cleavages [57].

NovoHMM is a method of *de novo* peptide sequencing using hidden markov models

[75]. In this method, spectrum peaks are the observable random variables while the originating amino acid sequence is represented by the hidden variables. The hidden markov model consists of a learned set of transition and emission probabilities between the states allowing the it to model the peptide fragmentation process. The best sequence given the observed peaks is then found using the viterbi algorithm [76].

One of the difficulties of the *de novo* sequencing problem is the size of the search space. Unlike database searching, there is no *a priori* knowledge of the sequence and so for a peptide of just length 10, there are 20^{10} possible amino acid combinations to be considered. If internal fragments are included, finding the optimal sequence has been shown to be NP-complete [260].

A common feature of all of the above algorithms is peak scoring using only a small subset of local features such as peaks from neighbouring fragments, followed by a step-by-step approach and dynamic programming. However, peptide fragmentation is a complicated process with the complete amino acid sequence influencing the likelihood of each cleavage [237]. Unfortunately these methods do not therefore effectively model the peptide fragmentation process. The algorithms are forced to employ approximate solutions due to the complexity of the problem [152]. The development of more complicated models is prohibitively computationally expensive. Yet, despite the large size of the search space, *de novo* identification can still include PTMs into the search space [7]. Previous analysis has shown that the integration of PTMs into *de novo* algorithms has a much smaller effect on the running time than that of database methods [81].

The ultimate goal of proteomics is the characterisation of the proteome. However, *de novo* identification of peptides makes it difficult to map the predicted peptides back to proteins. In database methods each matched peptide is mapped to one or several proteins. While this has its own difficulties as outlined earlier, all predicted peptides could possibly exist and are linked to proteins. However, peptides predicted using *de novo* algorithms may not even exist in the protein database. Missing peaks in a spectrum may lead to ambiguity in parts of the predicted peptide, causing a partial match. Instead of looking for direct matches, *de novo* peptides therefore can be matched using the Basic Local Alignment Search Tool (BLAST) [175]. BLAST is a powerful tool that compares the similarity between biological sequences [8]. It can be used to predict the function of a protein by finding other proteins of similar sequence. The tool is unsuitable for short sequences such as peptides and so a more specific alternative, MS BLAST, was developed [219]. Shevchenko *et al.* showed how the tool could be combined with *de novo* peptide identification to characterise the proteomes of organisms without sequenced genomes. Many other tools are also available that perform similar tasks [116, 155]. However, as these methods only use the predicted sequence, information from the spectrum is lost which could be critical to accuracy of the peptide [174]. Furthermore the accuracy of *de*

novo algorithms has also limited the utility of this approach.

Until recently, *de novo* strategies have lagged behind database searching in terms of outright accuracy with the latter consequently vastly more popular. Instead of been used to identify all the peptides in a sample, *de novo* methods have been more commonly used to identify sequence tags used to filter databases or process spectra missed by database searches [231, 230, 220, 101]. Combining *de novo* methods with database searches provides a potential solution to the aforementioned issues with large databases. Most *de novo* algorithms provide a confidence score in each of the amino acids they predict. Short high-scoring sub-sequences can then be used alongside the parent ion mass to filter the database, lowering the chance of a random assignment [181].

De novo peptide identification algorithms are also used when there is no reference database available. This could be because the genome of an organism is not yet sequenced, or the proteins in question have mutations such as the case in certain cancers. Neoantigens are small peptides that can be recognised by the body's immune system. The identification of neoantigens specific to a cancer can therefore be used to develop highly specific immunotherapies [87]. These are types of treatment whereby the immune system is triggered to fight the disease itself. *De novo* peptide identification algorithms offer a viable way of sequencing these peptides [240].

While *de novo* peptide identification has its limitations there is still room for optimism. Machine learning has recently being applied to the area leading to an explosion in algorithm development and accuracy. With continued improvements it may soon become an alternative to the database search approach [177].

2.1.7 Benchmarking Performance

There is no universal benchmark for *de novo* peptide identification algorithm evaluation. Evaluation of models is performed on data from different experimental setups and different research groups with no agreed upon standard [7]. This means the characteristics of the data being used can vary widely. Furthermore, there is also no guarantee that the peptide assignments used in the evaluation are correct. The data is usually obtained from a database search using a specified FDR, however incorrect implementation of the approach can underestimate the error rate by a significant amount [121, 58].

Evaluation is also typically performed by the research group that designed the algorithm. Performance of algorithms can vary widely from dataset to dataset and one cannot assume that the algorithm that performs best on data from one experimental setup will perform best overall [7]. Each algorithm is based on a different learning model which will have its own strengths and weaknesses. While the performance of models is generally reported in papers using amino acid precision and recall as well as peptide accuracy, the

reasons why one algorithm performs better or worse, or the where each model struggles is not explored. Unfortunately, there have been very few independent evaluations of *de novo* peptide identification algorithms to account for this. Muth *et al.* [177] performed one such study where they identified common errors made by state-of-the-art algorithms. The scarcity of independent evaluations means researchers are using tools that lack rigorous analysis. Furthermore, with such limited data on where algorithms go wrong it is difficult for improvements to be made.

Overall, the benefits of proteomics and peptide identification are clear, despite their many limitations outlined earlier. Better benchmarks and standards will help identify and address these limitations, leading to further improvements in the field. With many of the issues relating to data quality having been addressed, computational challenges are now central to the advancement of peptide identification [113]. As is currently the case in many fields of research, machine learning is sure to play a central role in this process.

2.2 Machine learning

2.2.1 Background

The field of machine learning dates back to the mid-twentieth century but has seen an explosion recently with the availability of more powerful hardware and an abundance of data [274]. It now permeates almost all aspects of our lives from healthcare [42], entertainment [23], agriculture [144] and security [259].

Machine learning involves the development of models that can learn patterns from data without explicit direction [169]. Following experience (E) on a given task (P), the model updates its parameters to improve its future performance (P). Data are typically made up of many observations of a particular phenomenon. Each observation will have the same feature types but a unique set of values [274]. Depending on the data used and the requirements of the task, different types of machine learning may be used. In unsupervised learning the model seeks to discern patterns from unlabelled data. An example of unsupervised learning is cluster detection, whereby a model tries to identify observations that are similar to one another in feature space [104].

Supervised learning involves labelled data whereby the model is given a set of features and an output it must predict. The model then tries to replicate the function mapping the features to the desired output. One of the simplest supervised learning models is the k-nearest neighbour model [131]. Prediction of new observations can be done by simply taking a weighted vote of the nearest k labelled observations in feature space. Other models need to iterate over the data multiple times. Linear regression by gradient descent is one such example [84]. First, a random model is created and its performance

measured. The parameters of the model are then updated to reduce its error and the process is repeated. The model continues to iterate over the labelled data until the error stops decreasing.

2.2.2 Data

Data is the cornerstone of machine learning. Much of the success of machine learning in recent years is due to the explosion in quality and scale of available datasets [125]. The quality of any machine learning model produced is reliant on the quality of the data available. Common issues with data include noisy or incorrect labels, missing values, imbalanced classes or bias [97, 40]. Ideally, given enough data a model will generalise over some of these issues. However this is not always possible and steps must then be taken to mitigate them.

Typically in machine learning, datasets are split into three parts; a training set, a validation set and a test set [93]. The training set is used by the model to learn a mapping from input features to output labels. The validation set is not used for training the model but for evaluation. During training, it gives an unbiased indication of the performance of the model. The model can then iterate over the training data until the validation performance stops increasing. The hyperparameters of the model can also be adjusted to maximise the validation performance [48]. These are non-trainable parameters within the model that affect its performance. Only once the training and validation are complete should the model be tested on the test set. This gives an estimate of the real performance of the model as this dataset is completely independent of the training process. It should be noted that the terms test set and validation set can sometimes be interchanged in the literature but the process remains the same.

2.2.3 Bias and Variance

The prediction error of a machine learning model has two main components; bias and variance [136]. The bias can be interpreted as the average error between a model's predictions and actual labels over all possible training sets. It is the error due to the inherent inabilities of the model. The variance is the error caused by differences between models for different training sets. Highly complex models will vary if trained on different subsets of a dataset. The design and choice of a machine learning algorithm often results in a trade-off between bias and variance [65]. Simple models will produce similar results regardless of the training data used (low variance). However, the algorithm may have insufficient degrees of freedom to adequately model the data and so have high bias. This is called underfitting. Alternatively, complex models will have low bias as each model fits the training data very well. They will also have high variance, as the prediction is highly

dependant on the training set and each model may not generalise well. This is called overfitting.

Ideally a model will have low bias and low variance. However, this can be difficult as modifications to a model that decrease one often increase the other [136]. The ideal model is one that balances this trade-off (Figure 2.4). There are steps one can take to identify these issues and to find an adequate model. It is easy to identify overfitting as the performance between the training and validation sets should be similar. If a model performs much better on the training set than the validation set it has overfit and will not be very useful. The model is unable to generalise what it has learned to new data. This can happen when a machine learning model has too many parameters so that it can completely recreate the training data. A model that has underfit can be difficult to recognise as both training and validation error will be similar. However, performance will be poor for both. In such cases, one can try to increase model complexity to see if it leads to increased general performance.

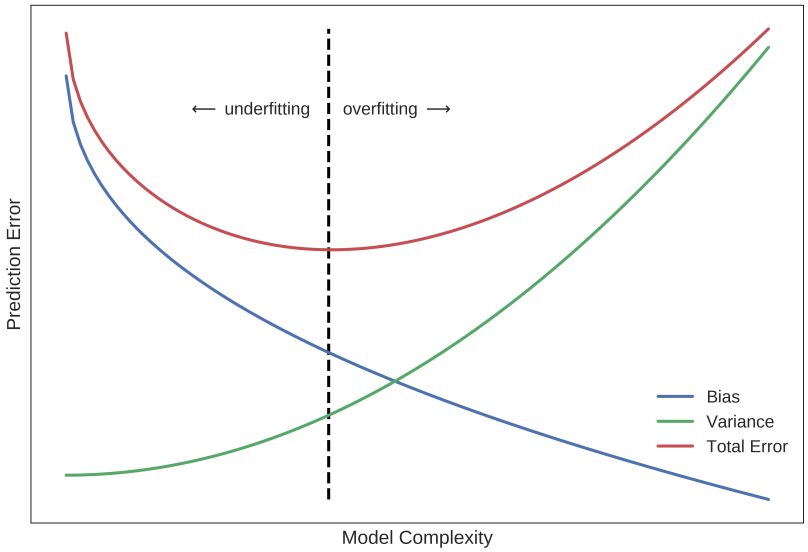

Figure 2.4: Trade-off between bias and variance. Models with low complexity will have high bias while models with high complexity will have high variance. The ideal model will find a balance which minimizes the prediction error.

There can also be bias in the data used to train models. Each dataset is only a small proportion of the total population of possible observations. Misleading relationships between features and labels may appear in the data due to imbalanced sampling, biased labelling or selection bias and therefore not accurately reflect the population [47, 269, 24, 103]. It is important that the training data provides a true representation of the

problem domain as bias in the training data will lead to bias in the model [132]. It is therefore imperative that researchers identify bias in their data and take steps to mitigate it. A simple example is ensuring that the training set is a random sample of the total population. If the data does not contain a complete or accurate representation of the problem domain it is almost impossible for a model to compensate. Understanding the data and its bias is an essential part of creating a machine learning model.

2.2.4 Decision Trees and Random Forests

Decision trees are a type of supervised machine learning algorithm whereby the data is continuously split based on simple rules that the model learns. A metric such as information gain, is used to decide which feature is used to split the data [201]. Using the feature and value that provide the most information gain, the data is divided into two parts. The process of calculation of information gain and feature selection is repeated recursively each time the data is split forming a "tree" of decisions. While this process can be repeated until each dataset contains only one class, this may lead to overfitting. A maximum tree depth can be defined to help ensure the model can generalise to new data. As the rules learned by a decision tree can be easily extracted and visualised the models are highly interpretable and therefore popular for medical applications [245].

This methodology can also be expanded to create an ensemble of decision trees called a random forest (RF) [35]. The first step in this methodology is to create random samples of the data with replacement. This is called bootstrap aggregation (bagging). Also, a subset of the features is randomly sampled for each dataset [110]. A decision tree model is then fit to each data sample. Each of the trees in the collection (forest) give a prediction for a given observation. The class with the majority vote is then assigned to the observation.

Random forests have been applied to a variety of areas from Alzheimer's disease prediction [138] to intruder detection [204]. As they are an ensemble of many decision trees, they are robust to overfitting. While a single tree may have high variance, the collection of trees as a whole will not [234]. They are especially useful when the number of features exceeds the number of observations. As such have been used to help predict survival rates post-cancer using microarray data, where the number of features (genes) may be an order of magnitude larger than the sample size [246].

2.2.5 Artificial Neural Networks

Artificial neural networks (ANNs) have been at the forefront of machine learning's recent success. Like decision trees and random forests, they are a supervised learning algorithm. ANNs are loosely inspired by the neurons of the brain and are made up of layers of nodes with connections that can pass a signal between them [93]. Each node receives multiple

inputs h_i, which are multiplied by respective trained weights w_i (Figure 2.5). These are then summed along with a bias term b before passing through a non-linear activation function f to give the output of that node. This can be then fed into the next layer of the network. Layers of nodes can be stacked so that the output of one provides the input to the next. With inputs $h^{(0)}$ we can define the output of the lth layer as follows:

$$\mathbf{h}^{(l)} = f(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \tag{2.2}$$

Activation functions must be non-linear to allow the model to interpret non-linear relationships in the data. Rectified Linear Units (ReLU) have become the method of choice in many state-of-the-art deep learning models [183]. For values greater than zero that pass through a ReLU function, the value remains unchanged with values below zero set to zero. While also being quicker than many other activation functions, a ReLU's partial linearity helps with vanishing gradients [154]. Vanishing gradients occur in deep networks, where the backpropagated error can decay to zero as the number of layers increases [112]. The final layer of an ANN is called the output layer, and has a single node for each possible class. The final activation function can then be used to output a probability distribution over the possible classes. For binary or multi-label classification the sigmoid function maps each value between zero and one as follows:

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

For multi-class single label problems a general form of the sigmoid function called a softmax is used. The softmax scales a vector such that its sum equals one as follows:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.4}$$

Neural networks are sometimes referred to as deep learning as layers can be stacked on top of one another to increase model complexity and create a deep network. However this means they can be computationally expensive as well as not being easy to interpret. With hundreds or thousands of parameters as well as multiple non-linear functions it is difficult to understand how a model makes a decision. Furthermore, models that are too complex for the problem at hand will tend to overfit.

ANNs typically update their parameters (θ) using gradient descent [209]. The gradient of a function is the partial derivative of the function with respect to all of its parameters. In the case of a neural network, the parameters are the weights of the connections between nodes. The function we want the derivative of is called the cost function $J(\theta)$ and it sums the error over all predictions which are calculated using a loss function $L(\theta)$. The gradient

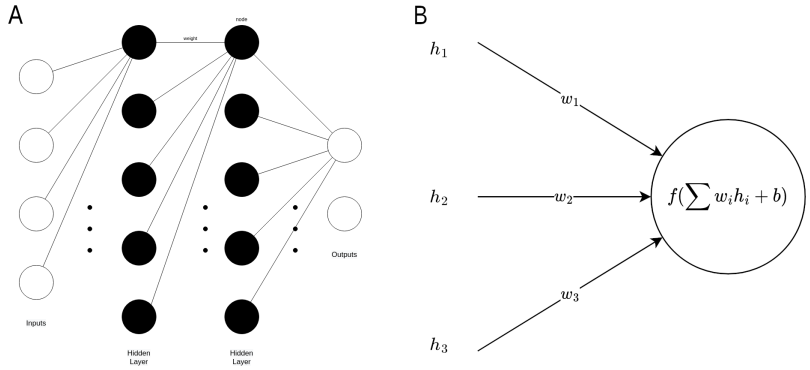

Figure 2.5: Artificial neural network structure. A, A fully connected neural network with two hidden layers (black). Only connections going to the first node in each layer are shown. B, A depiction of a single node in a neural network.

shows the direction in parameter space which will give the greatest increase in the function output value. The parameters in the model are adjusted in the opposite direction to try minimise the cost/error.

Equation 2.5 shows the cross entropy loss for a set of binary labels $\mathbf{y}$ and predictions $\hat{\mathbf{y}}$ [93].

$$J(\theta) = \frac{1}{N}\sum_{i}^{N} L(\theta) = -\sum_{i}^{N} y_i log(\hat{y}_i) + (1 - y_i) log(1 - y_i) \tag{2.5}$$

ANNs the require the gradient of the function $J(\theta)$.

$$\nabla_\theta J(\theta) = \frac{1}{N}\nabla_\theta \sum_{i}^{N} L(\theta) \tag{2.6}$$

Using gradient descent with a given a learning rate η, the parameters are updated using the following [209]:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta) \tag{2.7}$$

As modern datasets can have millions of observations, calculating the gradient can be extremely computationally expensive. A simple alternative is stochastic gradient descent (SGD) [32]. In SGD, the model updates its parameters after calculating the error in a randomly selected observation. This provides an estimate of the true gradient with a much simpler calculation meaning the model can be updated more frequently. This is repeated until the model converges to an optimum. However, as SGD only computes an

approximation of the true gradient there will be a certain amount of error/noise in its gradient prediction [33].

Modern ANNs generally use a compromise between the two approaches by calculating the approximate gradient using a batch of observations. The optimal size of this batch is dependent on the resources available, the architecture of the model, as well as the problem in question and its choice is therefore an important step in algorithm design [16]. Also, the size of the batch and the learning rate are closely linked, as the learning rate defines how much the model parameters change in response to the batch gradient. The interaction and optimisation of these two hyperparameters are the subject of much research [224, 135, 105].

Other hyperparameters include the number of nodes and layers in the network. For more complex ANNs, there are even more model hyperparameters. As many of these change the ANN architecture, they cannot be tuned during training. Traditionally, a grid search is performed, where many models are trained with different combinations of hyperparameters to look for an optimal choice [145]. This can be computationally expensive as many different networks need to be trained.

ANNs have been around for decades with their popularity initially stemming from their ability to easily model many non-linear processes with a simple model [28]. Early applications provided modest results on simple datasets such as zip code recognition [139]. However, increased data availability, increased computational resources as well as the creation of novel architectures have seen the field explode in recent years [222]. Simple fully connected networks are now rarely used but these alternative architectures now dominate the area of machine learning.

2.2.6 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a different type of deep learning architecture, this time inspired by human vision models [86]. Instead of fully connected layers, nodes are only locally connected. Furthermore, weights are shared between nodes, thereby reducing the complexity of the model and increasing its ability to generalise [140]. While this architecture has a lower capacity than a similar ANN, the shared weights mean that the same weight matrix (kernel) effectively passes over the entire input acting like a feature detector. The locally connected nodes mean that this type of network is especially proficient for a euclidean feature space such as images where neighbouring features are related.

CNNs can use multiple kernels that can identify different features. For example, each kernel $\mathbf{w}_k^l$ in the l^{th} layer of the network, is passed over the input moving one pixel at a time (assuming a stride length of 1), performing a convolution on each neighbourhood of

pixels from the previous layer. The value at the position (i, j) in the k^{th} feature map of the l^{th} layer of the network $h^l_{i,j,k}$ can be defined as follows [96]:

$$h^l_{i,j,k} = f(\mathbf{w}^l_k \mathbf{x}^{l-1}_{i,j} + b^l_k) \tag{2.8}$$

where $\mathbf{x}^l_{i,j}$ is the patch centred at location (i, j), b is a bias term and $f()$ is a non-linear function, generally a ReLU.

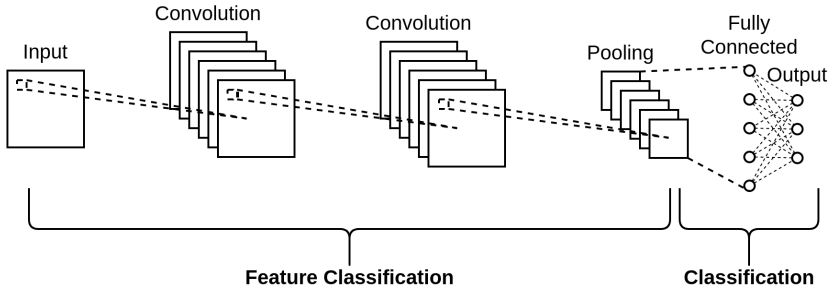

Figure 2.6: Convolutional neural network (CNN) model architecture. The kernels of the CNN act like feature detectors. The fully connected layers interpret these features to make a prediction.

These filters act as feature detectors which is particularly useful for task such as object detection. In this manner, the kernel passes over the entire input producing a feature map (Figure 2.6). This can be interpreted as whether or not that feature was detected at that neighbourhood of pixels on the previous layer. Given the multiple kernels, each "pixel" in output will be represented by a vector with elements defined by the product of each kernel. Initial CNN layers typically learn low level features such as contrasts or lines. When stacked on top of one another, CNNs can combine the information from the lower level features to learn higher level features [270]. Convolutional layers can also be followed by pooling layers that reduce the dimensionality of the output. As in ANNs, many of these layers can be stacked together. Typically the convolutional layers are followed by fully connected layers as above. These transform the CNN layers into the desired output. This characteristic of CNNs have seen them become hugely successful in recent years, particularly in the area of computer vision. A pioneering example of CNNs was LeNet [140] which introduced basic idea of a convolution followed by a non-linear activation and pooling. Since then this basic structure has remained but has increased in depth [109], differentially sized kernels [229], and residual connections [106] among many others.

CNNs are one of the most popular neural network architectures due to their remarkable performance and the abundance of vision tasks they can be applied to. For instance they can be used to help diagnose cardiac arrhythmia [2], cancer [236] and pneomonia [227].

They also have applications in a range of other areas from autonomous driving to fraud detection [85].

2.2.7 Recurrent Neural Networks

Many types of data have a particular sequence to their features such as text or speech data. This proves to be a problem for contemporary neural networks who treat the inputs independently [268]. Furthermore, the desired output of a machine learning model may also be sequential, for example in text translation. In this case it is important that the model knows what it has previously predicted in the sequence as that will influence the next prediction. Recurrent neural networks (RNNS) were developed for this purpose.

The key to RNNs is they have the capability of memory or context given the previous sequence [68]. This is done through the addition of a hidden state vector to the simple ANN (equation (2.2)). Given an input $\mathbf{x}^{(t)}$ for each time t and an initial hidden state h^0 at time $t = 0$, the hidden state is updated as follows:

$$\mathbf{h}^{(t)} = f(\mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}) \tag{2.9}$$

where $\mathbf{W}$ and $\mathbf{U}$ are trainable parameter matrices and $f()$ is a non-linear function. The hidden state encodes the context of the sequence and can be used as the feature vector for another ANN layer to make predictions.

One issue with early RNNs was vanishing gradients [111]. As the size of the sequence gets large the gradients tend to zero making the learning process slow. It also makes it difficult to learn long range dependencies between different tokens in the sequence. More recent RNNs such as long short-term memory networks (LSTMs) use multiple hidden states and more complex update functions to mitigate against the vanishing gradient problem [112]. Due to their capability to encode long range interactions, LSTMs are now ubiquitous in natural language processing and sequential data [268]. They have helped achieve state-of-the-art performance in tasks such as language translation [15], image captioning [262] and playing video games [250].

2.2.8 Graph Neural Networks

Other types of structured data require a different type of network. Graph neural networks (GNNs) are locally connected networks that operate on non-euclidean data that can be structured as graphs [257]. Unlike CNNs, the local connections are defined by connections in the graph instead of proximity in the feature space. GNNs are therefore able to encapsulate and model complex relationships and interdependancies [211]. For a lot of data, relationships between objects is known but cannot be represented easily by an input

vector. The use of GNNs means the connections themselves are part of the model and can then be used to propagate information throughout the graph.

GNNs have been used in a variety of applications with structured data containing complex relationships. They have been shown to be effective in many domains from recommendor systems [172] to the prediction of molecular properties [90]. The complex networks associated with biology are also an area of research where they have a lot of potential. As such, GNNs have being used to predict the interface of proteins [77] and to classify the sub-type of breast cancer using gene expression profiles [206].

A simple GNN can be defined as follows using an update function and an aggregation function [99]. The aggregation function defines how the embeddings of each node's neighbours are combined. This can be as simple as summing the list of vectors or taking their mean to create a new vector. The update function specifies how this new vector is combined with each node's own embedding to create a new embedding. For each iteration/layer of the GNN, the embedding of each node is updated by aggregating its neighbours' embeddings from the previous layer and combining them with its own using the update function. Just like the other neural networks, these layers can be stacked on top of one another. As each aggregation step passes information a distance of one connection, the number of layers in the network can be thought of as the depth of the search space [98]. With each additional layer in the GNN, the embedding of each node incorporates information from a larger and larger neighbourhood.

Giving a formal definition, we define a graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges. With GNNs, each node $u \in \mathcal{V}$, has an input embedding $h_u^{(0)}$. Each node also has a list of neighbours $\mathcal{N}(u)$. The embedding $h_u^{(k)}$ of node u is updated by aggregating its previous embedding $h_u^{(k-1)}$, with the embedding of u's neighbours $\mathcal{N}(u)$, where k is the number of message passing layers i.e. update steps. The simplest GNN takes the sum of the neighbour embeddings for each node, given by the following equation;

$$\mathbf{h}_u^{(k)} = f\left(\mathbf{W}_{self}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{neigh}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right) \tag{2.10}$$

One issue with sum aggregation is its instability and sensitivity to node degree [99]. Alternative aggregation functions to combine the neighbour embeddings can also be used such as mean aggregation or pooling. This will normalise the effects of node degree but thereby may result in a loss of information.

2.2.9 Dynamic Programming

Dynamic programming is a type of optimisation whereby large complex problems can be solved by being broken down into smaller simpler sub-problems. It can be applied to

problems that satisfy the following assumptions [22]:

- Each sub-problem uses the same metric
- The result of one sub-problem is independent of the others
- The result of the complete problem is equal to the sum of the results of the sub-problems

One application of this is in the knapsack problem. For a given capacity and mass-value pairs, the goal is to select the pairs that maximise the value while keeping the total mass below the capacity.

Dynamic programming can also be used to find the longest path in a directed acyclic graph. For a graph with weighted nodes or edges the longest path is equivalent to the highest scoring path. For large graphs it may be computationally impractical or even impossible to compare all possible paths. However, the problem can be broken down into smaller sub-problems. The longest path to a given vertex v can be computed by adding v's score to the highest scoring vertex that points to it. This is repeated for all vertices, starting at nodes with no incoming edges. The highest scoring path is then found by starting at the vertex with the largest path score, and moving to the vertex connected to it with the largest path score.

2.2.10 Metrics

Comparison of machine learning models relies on a choice of metric over which they will be evaluated. Different metrics focus on different types of error. Therefore, the choice of metric should reflect the problem space and the requirements of the user [102].

In a binary classification problem each observation belongs to one of two possible classes which the model can predict. This leaves four possible outcomes. Given two classes called the positive class (P) and the negative class (N), true positives (TP) are observations where the model correctly identifies them as the positive class and true negatives (TN) are when the model correctly classifies observations into the negative class. Conversely, false positives (FP) are where the model incorrectly identifies a negative observation as a positive while false negatives (FN) are when a model misclassifies positive observations as belonging to the negative class.

Different metrics take different combinations of the above values into consideration. This is useful when some of these are deemed more important than others. For example, in a medical setting, a model may accept a false positive over a false negative so that fewer actual cases are missed.

Historically, accuracy was the most popular metric for the evaluation of machine learning models [197]. It is defined as follows:

		True class		
		Positive	Negative	Total
Predicted class	Positive	TP	FP	TP+FP
	Negative	FN	TN	FN+TN
	Total	P	N	

Table 2.3: Confusion matrix for binary classification. The rows represent the classes predicted by the model while the columns represent the actual classes. TP stands for true positive, FP stands for false positive, FN stands for false negative and TN stands for true negative. P represents the total number of observations in the actual positive class while N represents the number in the negative class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

However, this assumes that the cost of misclassification for both classes is equal. Also, it is only defined at a particular threshold. Models generally output a continuous value between 0 and 1 for binary classification. Values greater than a given threshold are assigned to the positive class and lower than the threshold to the negative class. Different models may be preferred for different thresholds.

Using confusion matrix shown in Table 2.3, one can calculate the true positive rate (TPR) and false positive rate (FPR) as follows:

$$TPR = Recall = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{2.12}$$

$$FPR = \frac{FP}{N} = \frac{FP}{TN + FP} \tag{2.13}$$

A way of comparing models over all thresholds is to use a receiver operating characteristic (ROC) curve. It is defined as the plot of a model's FPR vs TPR for all thresholds. The ROC curve shows how the performance of models change as the threshold changes. An ideal classifier will have a convex curve that passes through the point (0,1). Conversely, a model that makes predictions at random will have a curve that is a straight line from (0,0) to (1,1).

The performance of a model with respect to this curve can be captured in a single statistic by taking the area under the ROC curve (AUC/AUROC) [235]. This can be interpreted as the probability that the model will score a randomly selected observation from the positive class higher than a randomly selected observation from the negative class. However, for a dataset with a large proportion of negative observations, a large change in the number of false positives will only produce a small change in the FPR due to the denominator N.

Davis *et al.* proposed using the area under the precision-recall curve (AUPR) instead

for this reason [59]. Recall is equal to TPR defined above, with precision defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.14}$$

The precision-recall (PR) curve also shows the performance over the complete range of thresholds. In PR space an ideal classifier is one that produces a convex curve through the point (1,1). A model that predicts classes at random will have a curve that is a straight line from (0,0) to (1,0). As seen in equations (2.14) and (2.12), AUPR is not directly related to the size of the negative class, N.

The dynamic of the above metrics changes for multiclass classification. Recall becomes the number of correct predictions over the total number of actual observations. Precision is the number of correct predictions over the total number of predictions. In the context of *de novo* peptide prediction, the total number of predictions and observations may not be the same. An amino acid can be considered matched between the real and predicted peptide if mass of the previous amino acids agree within 0.5 Da and the mass of the predicted amino acid agrees with the actual amino acid within 0.1 Da [153, 241, 200, 199]. As shown in Table 2.1 on page 11, there are many amino acids that are quite similar in mass. Amino acid recall is then defined as the number of correctly predicted amino acids over the total number of actual amino acids. Precision is defined as the number of correctly predicted amino acids over the total number of predicted amino acids. Peptide recall/accuracy is then the total number of correct peptides over the total number of actual peptides/spectra. As *de novo* algorithms give a confidence score with each peptide prediction, the number of predicted peptides may differ from the total number of spectra if a score threshold is implemented. Given the definition of amino acid precision and recall AUPR can also be calculated.

2.2.11 Noise and Artificial Data

Noise can be defined as anything that obscures the relationship between features and class labels [108]. This may be due to incomplete or missing features (attribute noise) or erroneous class labels (class noise) [275]. The presence of noise therefore makes supervised learning more difficult as the relationship the model is trying to learn is unclear. However, removal of noise is not necessarily the correct option. If noise is present in the data of interest, removing noise from the training data may lead to worse performance, despite the relationship being clearer [201].

Class noise can pose difficulties when present in training data as it forces the model to try to learn untrue relationships. This can be mitigated by filtering the data to remove false positives thereby reducing the FDR and creating more accurate training data [89].

Generally, attribute noise is not as great a problem as class noise but still severely impacts training [275]. Attribute noise may be caused by measurement error of the feature in question or just random perturbations in the data itself [83].

As noise is naturally present in real data, introducing it artificially can be used to mask the introduction of synthetic artefacts to real data [100]. Addition of noise can also be added to real data to increase performance. In this context it may be used to increase the size of the training set by altering current examples or improve performance through regularisation of the model [39, 258].

Noise also plays an important role in the creation of realistic synthetic data. These are data generated through artificial processes and may not contain the natural noise and variability of real data [190]. Once a relationship between features and classes is learned, artificial data can be generated which inherit the same characteristics [202]. For a given class label, a set of features can be defined which indicate the same relationship as is found in the real data. However, generation of artificial data may be deterministic as it lacks the noise present in real data.

There are multiple ways of generating artificial data. Ideally artificially generated data will share the same statistical properties of the real data [3]. One way of achieving this is to train a model to learn to generate the data from observing the real data [55]. In what can be thought of as the reverse problem of supervised machine learning, models can learn to generate a set of features for a given output. More advanced methods train two networks in opposition, a generator and a discriminator [53]. The generator starts off generating random examples. The discriminator is taught to learn between real and generated examples. However, the error signal from the discriminator is given to the generator so that it can learn to "fool" the discriminator. Given the right set up, the generator learns to create artificial examples indistinguishable from real examples, without ever encountering them.

A major benefit of using artificial data for model testing is that experimental conditions can be controlled allowing for a more systematic evaluation [21]. This can facilitate gaining a better understanding of a model's strength's and weaknesses [31]. Artificially generated data can also be used to train machine learning models [167]. As the training data can be fully controlled, observations can be generated with enough variability to span the complete space of features and labels [238]. The utility of these artificially generated data can be measured by comparing the performance of a model trained on them versus real data. If the performance of the model trained on artificial data is similar to the performance of a model trained on real data it can be assumed the former includes the same patterns and diversity of the latter [221]. Similarly, a artificial data can be evaluated by being used as test data. For a model trained on real data, high test performance on artificial data indicates they are of sufficient precision [221]. While both of these two

properties are highly desirable, they may not always be present together if at all.

2.3 Machine Learning for *De Novo* Peptide Identification

The rise in the capabilities of machine learning in recent years has seen it spread to many areas of science, with proteomics being no exception. The proteomics pipeline has many complex classification problems, alongside large datasets, that make it an ideal candidate for the application of machine learning [129]. Examples include trypsin proteolysis modelling [71] and retention time prediction [89], with the fundamental problem of protein folding recently receiving a lot of attention [126].

Machine learning has also been revolutionary in the task of peptide identification. The Percolator algorithm is a popular machine learning model that helps distinguish correct PSMs from incorrect ones [128]. It is performed as a post-database search step to improve the rate of correct peptide assignment in proteomics pipelines. Percolator uses a support vector machine to differentiate the high-scoring incorrect matches from the high-scoring correct matches.

De novo peptide identification has also greatly benefited from the introduction of machine learning. All state-of-the-art *de novo* peptide identification algorithms now incorporate machine learning models into their pipeline, generally alongside dynamic programming [153, 241, 200, 199]. Spectra labelled with peptides through a database search, typically with a stringent FDR, are used as training data for the algorithms. The machine learning models are then used to identify likely fragmentation sites in the spectrum, or to predict amino acids directly. Three such algorithms will now be described.

2.3.1 Novor

Novor is a *de novo* peptide sequencing algorithm that combines multiple RF models with dynamic programming [153]. Each spectrum is first converted into a mass array. For a given tolerance δ and peptide mass M, the size of the array is defined as $M \times \delta + 1$. Novor uses an RF model to score the probability that a mass in the array defines a fragmentation site. They consider nine possible ion types resulting from each possible fragmentation site. If matched to real peaks, their intensities are used as features for the model. The ions include singly charged b and y ions, their neutral losses of H_2O and NH_3 and their doubly charged version, alongside singly charged a ions. For each ion type the model also includes some expert defined features, namely the number of peaks in the spectrum that have a greater intensity than the current peak (rank), the number of peaks with an intensity greater than half the current intensity (half rank), the number of peaks

in a 50 Da radius that have a greater intensity than the current peak (local rank) and the number of peaks in a 50 Da radius with an intensity greater than half the current intensity (local half rank). Novor uses dynamic programming to identify which collection of fragmentation sites that span the peptide mass return the highest score. These sites are used to indicate an amino acid sequence.

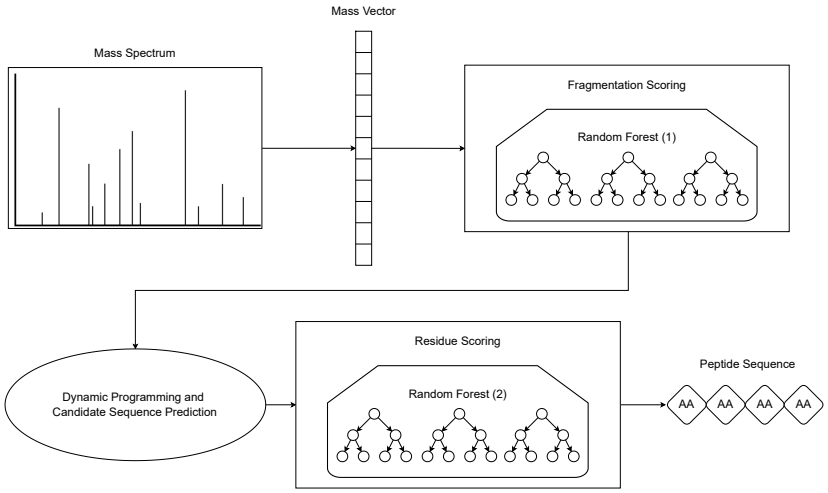

Figure 2.7: Flow diagram of the Novor algorithm.

Novor then uses another RF model to measure the correctness of the amino acid residues in a candidate sequence. This RF used to refine the top scoring sequence predicted using the fragmentation scores. The algorithm greedily selects the top scoring residues based on this second RF and splits the sequence into mass segments surrounding them. For each segment this process is repeated until the remaining mass is small enough that less than 100 possible sequences could account for it. A residue score is calculated for all of these possible sequences with the maximum for each segment kept. The combination of the maximum scoring sub-sequence for each mass gap defines the peptide.

2.3.2 DeepNovo

Another approach was proposed by Tran *et al.* [241]. In their paper their they describe DeepNovo, a deep learning based approach to *de novo* peptide sequencing. They incorporate multiple deep learning modules into their algorithm. Firstly the spectrum is converted into a mass array similar to Novor. Each peak in the spectrum is placed into its encompassing bin.

They then define an ion-CNN which can encode prospective neighbouring ions from the possible next cleavage, given the amino acids already predicted. For a given position

in the spectrum, windows of size 10 of the mass array are extracted for all possible b, y, b-H_2O, y-H_2O, b-NH_3, y-NH_3, b(2+) and y(2+) ions, for all possible amino acids. This results in input tensor of shape 26×8×10. This is then transposed to an 8×10×26 tensor before being transformed by a 1×3×26×32 kernel used by the ion-CNN. A second CNN kernel of shape 1×3×26×32 is then used followed by a ReLU activation function. The resulting tensor is passed through a max-pooling step to take on the shape 8×5×64. Finally a fully connected ANN layer converts this tensor into a 512 vector encoding.

DeepNovo uses another convolutional network called the spectrum-CNN. The spectrum CNN encodes the complete spectrum array. It uses two 1×4 convolutions, both with 4 different filters. These are followed by a ReLU function before the output passes through a max-pooling layer. A fully connected layer then transforms the pooled data into a 512 vector. This vector is used as the initial state of the LSTM, the final deep learning module in the algorithm.

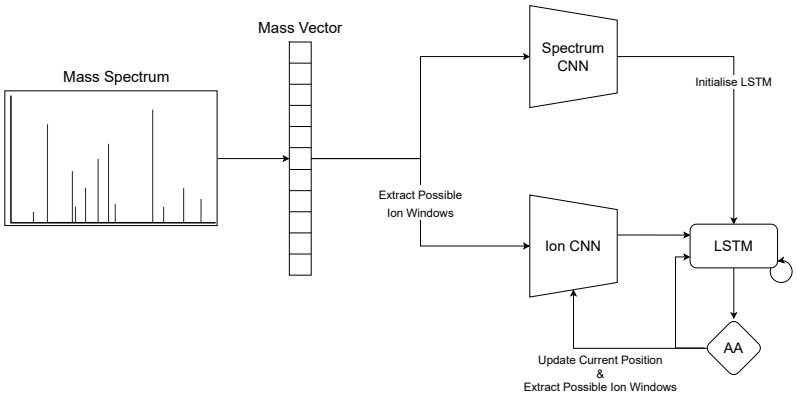

Figure 2.8: Flow diagram of the DeepNovo algorithm.

The LSTM controls the sequence prediction phase for DeepNovo. At each prediction step, the ion-CNN looks for the next possible cleavage site and the output encoding is passed to the LSTM. The LSTM encodes the ion-CNN output along with the previous two amino acid predictions and provides a probability distribution over all possible amino acids with the maximum selected. While LSTMs can encode long sequences, the authors

found this to be the best setup to create a model capable of working for a diverse range of species. However, they also concede that this is an area that requires additional analysis. After each amino acid prediction, the position of the ion-CNN in the spectrum is updated to reflect the predicted sequence and the process is repeated. DeepNovo moves through the spectrum in this step-by-step manner until the complete peptide mass is accounted for. As the difference between the peptide mass and the mass of the predicted gets smaller, there will be a limited combination of amino acids that will equal this value. DeepNovo uses dynamic programming, similar to the knapsack problem, to limit the amino acid predictions to those that will satisfy the mass requirement. It also employs a beam-search which looks at multiple sequences through the spectrum. As the algorithm moves through the spectrum it keeps hold of the top 5 sequences. The probability distribution for each amino acid is calculated for each of the 5 sequences. Of the now 130 (26×5) sequences the 5 highest scoring are retained. Finally, DeepNovo performs the above search beginning at both ends of the spectrum each producing 5 sequences. The amino acid probability distributions of both directions are combined to give 25 sequences. The top scoring sequence of these 25 is then returned as the prediction.

2.3.3 PointNovo

An updated version of DeepNovo has been developed recently, improving upon the performance of its predecessor [200]. It has subsequently been released under the name PointNovo [199]. The new approach is very similar to DeepNovo with the main update coming from the ion-CNN encoding module with some architectural changes. PointNovo draws its inspiration from PointNet, a deep learning architecture for point cloud classification [198]. For PointNet, Qi *et al.* developed a T net module which encodes points from continuous space, negating the need for pixelation. PointNovo uses the T Net module instead of the standard CNN allowing the model to encode exact m/z values instead of the mass bins of the original version. Alongside improved performance, this makes the model readily adaptable to increased precision of future mass spectrometers [199].

Instead of spectrum windows, PointNovo uses the difference between the theoretical ions and the spectrum peaks. Given its current position in the spectrum, PointNovo calculates the location of the possible ions from the next amino acid for all possible amino acids resulting in 26×8 values. This matrix is then subtracted from all the peaks in the spectrum to create a $26\times8\times$N tensor, where N is the maximum number of peaks in the spectrum. The peak intensities are then concatenated to this tensor before being fed to the T Net. The T Net performs a 1-dimensional convolution across this tensor before a fully connected layer transforms the output into a 512 vector. This output encoding is fed to an LSTM for the sequence prediction. Unlike DeepNovo, the initial

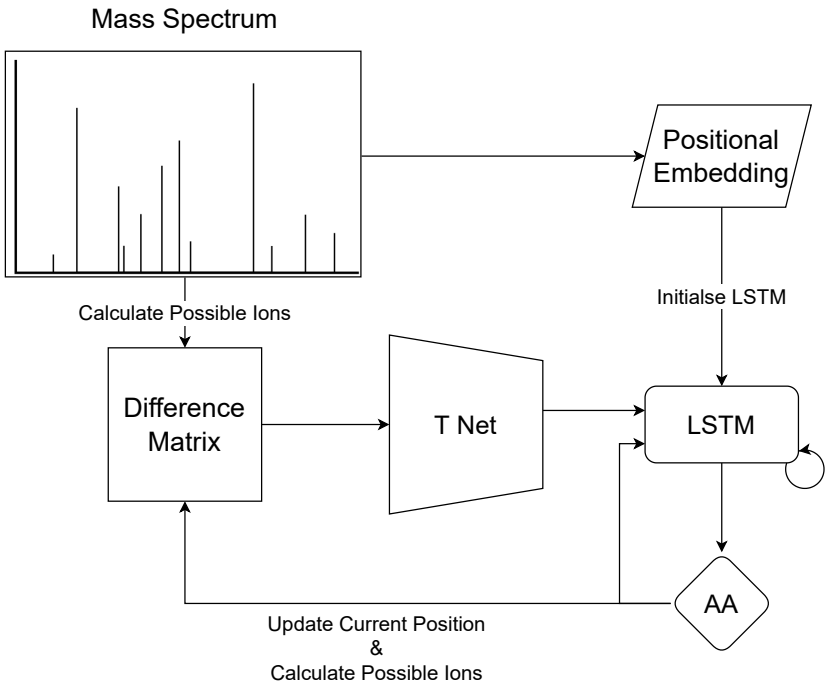

Figure 2.9: Flow diagram of the PointNovo algorithm.

state for PointNovo's LSTM is not provided by a CNN but using positional encodings [247]. This is done by transforming the m/z of each peak in the spectrum into a sinusoidal wave and multiplying each by their intensity before they are all summed. Another fully connected layer transforms this into a 512 vector to provide the initial hidden state. Also unlike DeepNovo, PointNovo's LSTM includes all previous amino acid encodings for each prediction. The LSTM encodes the initial hidden state, the previous predictions and the output of the T Net before producing a probability distribution over all amino acids. After each amino acid prediction the algorithm updates its position and repeats the process. PointNovo also uses the same beam search and dynamic programming as DeepNovo to improve performance.

2.4 Artificial MS/MS Spectra

Although peptides can be fragmented using the collision of particles, the frequency with which peptide bonds break is not random. The abundance of fragment ions is dependent on the size of the fragments, the peptide length, the amino acid sequence, the type of mass spectrometer and the fragmentation method used [189, 237]. Attempts have been

made to model this relationship and therefore predict the relative intensities of fragment ions in tandem MS spectra. Early attempts used strictly mathematical models with very few parameters [273]. Then, advances in database search methods provided sufficient labelled data for machine learning to be applied. Two such machine learning models that have been used to predict peptide fragmentation spectra include bayesian decision trees [66] and neural networks [13]. These models have been very effective when integrated with database search methods. Database search algorithms typically compare the m/z values of theoretical fragment ions to the observed spectra [181]. Correlation between these two is based on the assumption that more intense peaks are better [72]. However, this is not always the case as ions from less energetically favourable cleavages will have a relatively lower intensity [189]. To account for this, the theoretical spectra generally used in a database search can be replaced by artificial spectra with predicted intensities. With accurate prediction of fragmentation patterns, these spectra can by compared by both the m/z and intensity of the peaks, thereby increasing the sensitivity of the database search [66].

The accuracy of spectrum prediction models has increased alongside the advancement of machine learning methods. A recent example, Prosit, reports greater quality fragment ion intensity than experimental spectra [89]. By using the predicted spectra in a database search, the authors were able to increase the peptide identification rate in notoriously difficult metaproteomics data [89]. Prosit consists of two main parts; an encoder and a decoder. The model first encodes the amino acid sequence of the peptide using a bi-directional RNN with gated recurrent memory unit (GRU) cells. This encoding is combined with the output of a dense ANN that encodes the the peptide charge and collision energy. Another bi-directional RNN with GRU cells is used as a decoder. At each step the decoder model uses an attention mechanism to combine the encodings of the peptide sequence so it can focus on particular areas relevant to the particular ion it is predicting. Prosit can predict 6 different ion types (b and y ions from singly to triply charged) for peptides of up to 30 amino acids in length.

The focus of fragment ion intensity prediction so far has been on the improvement of database search accuracy [66, 89, 237]. The additional benefits of being able to match both peak position and height have allowed more peptides to be recalled given the same FDR threshold. However, this research has implications beyond database methods [177].

The field of *de novo* peptide identification is heavily reliant on database peptide identification to create both training and evaluation datasets. However, due to the limitations of this process the data is not 100% accurate. Furthermore, database scoring functions, such as hyperscore, will favour spectra with more fragment ions present [72]. This causes a bias in the training data as high scoring spectra, with more ions matched, are more likely to be present. Data labelled in this way will therefore not be an accurate represen-

tation of the entire population of spectra recovered from the sample. If training data are biased this can be learned by the model creating a biased classifier [132]. However, if the test data are generated in the same way, this may be difficult to detect.

Chapter 3

The Impact of Noise and Missing Fragmentation Cleavages on *De Novo* Peptide Identification Algorithms

The work outlined in this chapter was published in:

McDonnell, K., Howley, E., and Abram, F. The impact of noise and missing fragmentation cleavages on de novo peptide identification algorithms. Computational and Structural Biotechnology Journal 20 (2022), 1402–1412

3.1 Abstract

To enable *de novo* peptide sequencing to realise its full potential, it is critical to explore the mass spectrometry data underpinning peptide identification. In this research we investigate the characteristics of tandem mass spectra using 8 published datasets. We then evaluate two state of the art *de novo* peptide sequencing algorithms, Novor and DeepNovo, with a particular focus on their performance with regard to missing fragmentation cleavage sites and noise. DeepNovo was found to perform better than Novor overall. However, Novor recalled more correct amino acids when 6 or more cleavage sites were missing. Furthermore, less than 11% of each algorithms' correct peptide predictions emanate from data with more than one missing cleavage site, highlighting the issues missing cleavages pose. We further investigate how the algorithms manage to correctly identify peptides with many of these missing fragmentation cleavages. We show how noise negatively impacts the performance of both algorithms, when high intensity peaks are considered. Finally, we provide recommendations regarding further algorithms' improvements and offer potential avenues to overcome current inherent data limitations.

3.2 Introduction and Related Work

Proteomics has become an indispensable tool for biologists in the last few decades with its ability to identify system-wide protein expression. [180]. Its application is wide ranging and encompasses the identification of cancer biomarkers [210] and antigens for immunotherapy [19], as well as mechanisms underlying drought resistance in crops [9] and virulence factors in human pathogens [196, 143]

In proteomics, protein extracts are typically enzymatically digested and analysed using mass spectrometry. The corresponding mass spectra are then matched to peptides, which are short sequences of amino acids. Database search algorithms are commonly used in proteomics and aim to match theoretical peaks predicted from all possible peptides in the relevant protein databases to the peaks in actual spectra. Although database searching is the most popular technique used in protein identification, improved data quality and algorithm design mean *de novo* peptide sequencing is becoming increasingly popular in proteomics [175].

Recent advances in mass spectrometry (MS) have considerably raised the level of data resolution and acquisition in the field of proteomics [253], while the same database search algorithms have dominated the field for the last 20 years [249]. Typically, for shotgun proteomics, following the enzymatic digestion of proteins, the resulting complex peptide mixture is fractionated using liquid chromatography. The corresponding peptide fractions are then analysed using tandem mass spectrometry (MS/MS). Peptides are separated by mass and charge (m/z) in the first mass analyzer. Then, peaks from the resulting spectra are isolated and the associated peptides are passed through a fragmentation chamber to be charged and broken down into smaller pieces (fragment ions). These fragments pass through the second mass analyzer producing fragmentation patterns as the ions are separated. A database search or *de novo* peptide sequencing is then conducted to establish the most likely peptide sequence corresponding to each fragmentation pattern. Two common methods of fragmentation include collision induced dissociation (CID) and higher-energy dissociation (HCD). While similar in methodology, HCD fragmentation provides greater resolution and mass accuracy than CID [185]. Both of these methods fragment peptides by colliding them with gas molecules. This causes the cleavage of the amino acid sequence typically at a peptide (amide) bond resulting in two possible fragments; b and y ions [232]. While b and y ions themselves are the most common, peptide fragments can also suffer neutral losses of ammonia and water molecules producing different peaks with a shifted m/z value. Conventional notation enumerates the b ions according to their fragmentation site from the N-terminus to the C-terminus. Conversely, y ions are numbered from the C-terminus to the N-terminus. Although both ion types are ordered by increasing mass, it means for a peptide of length 20, the b_1 ion is created

from the same cleavage as the y_{19} ion. As the peptide mass is known, the mass of the corresponding y ion can be easily calculated given a b ion and *vice versa*. As these ions contain equivalent information about amino acid composition they can be grouped together. We refer to missing fragmentation cleavages from here on to indicate that neither a b or y ion, or their neutral losses, is present for a given fragmentation/cleavage site along the peptide chain. To refer to our example again, if for a peptide of 20 amino acids, neither the b_1 or y_{19} ions were present, or peaks indicating the loss of ammonia or water from these ions, we would then consider that the first cleavage is missing.

Although popular, database searching is not straightforward due to the irregularity and incompleteness of the peptide fragmentation process which effectively means there is never a perfect match between predicted and actual peaks in the mass spectra. Even with recently developed algorithms and up-to-date, tailored databases, on average, only 25% of spectra are identified leaving the remaining 75% unclassified and thereby discarded [80, 95]. This can be partly attributed to the size of the databases, where a larger number of possible matches increases the false discovery rate [181]. This is particularly problematic for metaproteomics, where databases typically span large species diversity. Peptide identification from mass spectra can also be performed *de novo*, where peptides are identified based on the spectrum alone, thus removing the need for a database. Historically this approach has had a much lower sensitivity than database search methods but recent advances in machine learning and mass spectrometry have seen it become a competitive alternative [177]. Without the use of a database, *de novo* methods are not limited in the same way as matching algorithms are, while also being able to identify post-translational-modifications (PTMs) relatively easily [152]. PTMs expand protein function beyond the standard amino acids by both reversible and irreversible modifications. The importance of PTMs is only starting to be uncovered as evidence suggests they are involved in the regulation of almost all cellular events [251].

When database search algorithms include variable modifications, reflective of PTMs, it exponentially increases their search space as the m/z value of any peak including the modified amino acid will be shifted accordingly. This has the effect of increasing both the FDR and running time of the algorithm [5]. This is not the case for *de novo* peptide sequencing where the number of PTMs being searched may have little or no effect on run time [81].

Although the current state of the art *de novo* algorithms are still not as effective as database searching, the recent availability of big data, and the simultaneous explosion in machine learning means the field is on an upward curve. Two algorithms leading the way are Novor [153] and DeepNovo [241]. They use machine learning and dynamic programming to both learn patterns within the data and simplify the prediction process respectively. How they implement these techniques is quite different however. Novor

models the spectrum as a graph, a traditional approach to *de novo* peptide sequencing [57]. Each node in the graph, which corresponds to a peak in the spectrum, is scored using a random forest model, trained on thousands of other spectra. Edges are created between nodes whose associated masses differ by that of an amino acid. Using dynamic programming, Novor then finds the highest scoring path through the graph, whose edges will classify the amino acids of the peptide. DeepNovo's approach to the problem involves progressing through the spectrum step-by-step using two different deep learning architectures combined. Based on the mass of the predicted sequence so far, a convolutional neural network (CNN) is trained to encode the parts of the spectrum where the next fragment ions might appear. A long short-term memory (LSTM) recurrent neural network uses this encoding, along with all the encodings from the previous predictions, to determine the next amino acid in the sequence. DeepNovo uses dynamic programming to limit the number of possible amino acids it can predict to those that would satisfy the remaining mass of the peptide, given those already predicted.

While *de novo* algorithms continue to improve, their possible uses continue to increase. *De novo* peptide sequencing has been used successfully to both aid and confirm database search results [133, 271, 79]. To aid database methods it can be used to identify amino acid "tags" from a spectrum that can then be used to limit the size of the search space to entries that only include them, thereby decreasing the false discovery rate (FDR). More recently, advanced *de novo* sequencing algorithms like DeepNovo, have been used for neoantigen detection [240]. Antigens are used by the immune system to recognise pathogens and trigger a response [276]. Neoantigens are antigens previously unseen by the immune system, which may be caused by genetic mutations [87]. Identification of these neoantigens is important for the development of cancer immunotherapies as they are not expressed by healthy tissue [193].

If the continual increase in the accuracy of *de novo* sequencing can be sustained, it may also open up the possibility of re-mining available data. The PRIDE Repository [157] contains data from thousands of proteomics experiments and improvements in machine learning and *de novo* peptide sequencing could uncover new insights from previous studies. To enable *de novo* peptide identification to reach its full potential, it is vital to understand the underlying data [166], in order to best design *de novo* algorithms.

Previous studies of *de novo* algorithms have sought to show how these algorithms perform on different datasets while investigating what errors they are making [177, 37]. Here, we investigate the prevalence and effects of missing fragmentation cleavage sites and noise on *de novo* peptide sequencing using real labelled data as well as artificial data. Specifically, we address the following research questions; How prevalent are occurrences of noise and missing fragmentation cleavages in tandem MS data? What are the effects of noise and missing fragmentation cleavages on the performance of *de novo* peptide sequencing

algorithms? How do the current state of the art approaches cope with noise and missing fragmentation cleavages? Finally, based on our findings, we propose approaches that could be implemented in the future to improve *de novo* peptide sequencing algorithms.

3.3 Methods

3.3.1 Data

We analysed data from eight different datasets downloaded from their respective archive on the PRIDE Public Repository [157]. A summary of each is provided in Table 3.1. These include the four used by Muth and Renard (2018) [177]. The eight datasets are made up of six different organisms, distributed between the two fragmentation types, CID and HCD.

To obtain the labelled data required for this research we performed a database search using two popular search algorithms. For each organism, a protein database was downloaded from UniProt (Appendix A Table 1). Just as was done by Muth and Renard (2018), all prokaryotic data were searched against the yeast proteome as well as their own. Accurate FDR estimation requires each spectrum to be compared to multiple peptides [51]. If this condition is not satisfied it can lead to an overestimation of identifications in smaller databases [176]. Therefore the small databases of prokaryotic organisms were augmented to circumvent this issue [177].

MS-GF+ [134] and X!Tandem [52] were used to search the databases through the SearchGUI platform [17]. Carbamidomethylation of cysteine was set as a fixed modification and oxidation of methionine was set as a variable modification. A maximum of two missed tryptic cleavages were allowed. b and y ions were considered with precursor charge bounded between 2 and 4 inclusive. MS-GF+ was set to HCD or CID mode depending on the data being used. Using an FDR of 1%, we extracted the top scoring peptide spectra matches (PSMs) from each dataset. Furthermore, we then selected from these PSMs only those for which MS-GF+ and X!Tandem agreed. The results of these conditions can be found in Table 3.2. The data were then collated into two groups, one for each fragmentation type. This resulted in a split of 25007 HCD spectra and 23821 CID spectra. For the remainder of this research, CID data refers to the four combined CID datasets listed and HCD data refers to the four HCD datasets.

3.3.2 Peptide peak and noise assignment

Using the peptides assigned to the spectra following the database search, each peak was labelled as either a peptide peak or noise. To do this the assigned peptides were artificially

Table 3.1: Overview of the datasets and processing steps used in this study.

Dataset	Pride Archive	Organism	Original Format	Mass Spectrometer	Frag Type	PrecTol	FragTol
MouseCID	PXD000790	*M. musculus*	MGF	LTQ Orbitrap Elite	CID	5 ppm	0.50 Da
YeastCID	PXD002726	*S. cerevisiae*	MGF	LTQ Orbitrap Velos	CID	10 ppm	0.80 Da
EcoliCID	PXD016825	*E. coli*	RAW	LTQ Orbitrap Velos	CID	20 ppm	0.50 Da
StaphAurCID	PXD017932	*S. aureus*	RAW	LTQ Orbitrap Velos	CID	5 ppm	0.60 Da
HeLaHCD	PXD000674	*H. sapiens*	RAW	Q Exactive	HCD	10 ppm	0.02 Da
PyroHCD	PXD001077	*P. furiosus*	RAW	LTQ Orbitrap Velos	HCD	10 ppm	0.06 Da
EcoliHCD	PXD008685	*E. coli*	MGF	Q Exactive	HCD	10 ppm	0.02 Da
StaphAurHCD	PXD023039	*S. aureus*	RAW	Q Exactive	HCD	10 ppm	0.06 Da

Frag Type: Fragmentation Type
PrecTol: Precursor Mass Tolerance
FragTol: Fragment Mass Tolerance

fragmented to create b and y ions along with their neutral losses of ammonia (NH_3) and water (H_2O) using the Pyteomics framework [141]. These are the ion types used by both Novor and DeepNovo. If possible these were matched to peaks in the spectra and labelled as peptide peaks with a tolerance of 0.5 Da for CID data and 0.05 Da for HCD data. Thereby the ions and hence cleavage sites which were not represented in each spectrum were identified and peaks that could not be matched to a fragment ion were classified as noise. For clarity, noise was also considered in its proportion to peptide peaks [177]. When low intensity noise peaks were found not to affect performance, only those above the median of the distribution of noise peaks were included. A median normalised noise intensity value of approximately 7.2e-3 was observed for the CID data and a median of approximately 2.1e-2 for the HCD data. The number of noise peaks above this threshold was recorded for each spectrum. The noise factor was then defined as the number of high intensity noise peaks divided by the number of peptide peaks in each spectrum (#NoisePeaks / #PeptidePeaks).

3.3.3 Algorithms

DeepNovo was downloaded from https://github.com/nh2tran/DeepNovo. Two models were then trained, one for CID data and one for HCD data. These two models used the parameters specified for low resolution and high resolution data in the original paper respectively [241]. The models were also trained using the same data as the original paper found at ftp://massive.ucsd.edu/MSV000081382/. The algorithm was then run through a linux terminal using Python 2.7.17. Novor was operated through the DenovoGUI interface [179] in CID or HCD mode depending on the data. Precursor precision and fragmentation tolerance were kept the same as DeepNovo for a fair comparison. Both algorithms were set to consider carbamidomethylation of cysteine as a fixed modification and oxidation of methionine as a variable modification.

Table 3.2: The number of peptides matched at the 1% FDR level for both X!Tandem and MS-GF+, as well as how many of those were in agreement (Overlap)

Dataset	Overlap	X!Tandem	MS-GF+
MouseCID	12132	15586	13345
YeastCID	534	650	1519
EcoliCID	5716	7210	7752
StaphAurCID	5439	6020	6363
HeLaHCD	4061	4973	4167
PyroHCD	9719	12080	10172
EcoliHCD	5180	5279	5257
StaphAurHCD	6047	8279	6850

3.3.4 Metrics

Amino acid match

For the CID data, two amino acids are considered a match if the prefix mass of the peptide before the prediction is correct to within 0.5 Da and the masses of the amino acids predicted are within 0.1 Da. For HCD data, the tolerance is lowered with an amino acid match requiring the prefix mass of the peptide before the prediction to be correct within 0.05 Da and the masses of the amino acids predicted to be within 0.01 Da.

AA recall

Amino acid recall is defined as the number of amino acids matched divided by the total number of amino acids in the database assigned peptide.

Peptide accuracy

Peptide accuracy corresponds to the number of peptide predictions that correctly match those assigned to the spectra divided by the total number of spectra.

Peak recall

We compare the cumulative masses generated by the amino acids in the PSM's peptide sequence and the predicted peptide sequence which are akin to the position of cleavage sites along the peptide. For CID data a predicted fragmentation cleavage is considered correctly matched if its mass differs by less than 0.5 Da from the corresponding true peptide cleavage. We also compare if the true peptide's cleavage sites are represented with a b or y ion in the spectrum with a tolerance of 0.5 Da. For HCD data the tolerance for both matches is reduced to 0.05 Da.

3.3.5 Confirmatory Analysis

High scoring spectra and artificial spectra were also used in a complementary analysis to confirm the trends observed when evaluating the algorithms with respect to all of the real data.

For high scoring spectra, *de novo* peptides above an acceptable score were extracted for both algorithms separately. High scoring spectra are defined as those with scores above a threshold which gives 90% amino acid recall. This standard was used by Tran *et al.* (2019) when using DeepNovo for antigen identification. Also, similar levels of peptide accuracy or higher amino acid recall were not possible for both algorithms. 90% amino acid recall was achieved in CID data with a score threshold of 0.89 (2740 peptides) and 0.74 (10295 peptides) for Novor and DeepNovo respectively. The thresholds for Novor and DeepNovo in HCD data were 0.67 (13493 peptides) and 0.73 (16898 peptides) respectively.

Artificial data were created to match the distribution of peptides found in the real data. Prosit was downloaded from https://github.com/kusterlab/prosit. A trained HCD Prosit model was then downloaded from https://figshare.com/projects/Prosit/35582. The overlapping HCD peptides matched by both database algorithms were extracted and artificial spectra were created for each using this Prosit model. CID peptides were not considered as there was no available model.

The artificial data were duplicated four more times with each duplicate given a different level of noise. Therefore, for each duplicate each spectra was given additional random noise peaks corresponding to the respective noise factor of that duplicate. Noise factors of 0,4,8,12 and 16 were considered.

3.4 Results

3.4.1 Missing fragmentation cleavage sites are prevalent in mass spectra

It can be difficult to evaluate *de novo* algorithms as there is no such thing as real data that is 100% correctly labelled. Instead we use the results of two database search algorithms that agree at a 1% FDR. We evaluate two state of the art *de novo* algorithms by comparing the database PSMs to their *de novo* predictions. Given the assigned peptide from the database search, we establish which peaks in the spectrum are fragment ions. Those that cannot be attributed to the peptide are classified as noise. We can then quantify what fragmentation cleavage sites are present and how many are missing from the spectrum. Models are also available to create high quality artificial data [89, 237], although they only predict peaks at precise locations directly derived from the peptide sequences. They

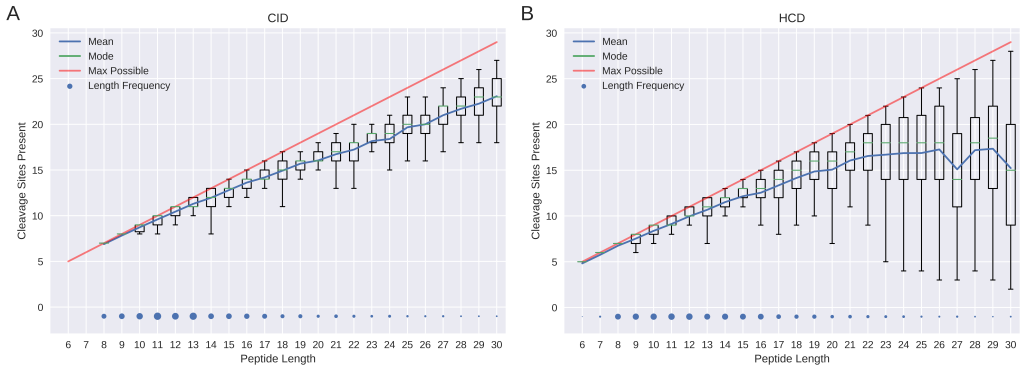

Figure 3.1: Number of cleavage sites present in the spectra. Box plots show the numbers of fragmentation cleavage sites present in the spectra for peptides of length 6 to 30. The combined results of all the CID spectra from this study are shown in A, with the HCD spectra from this study shown in B. The relative numbers of spectra per length are indicated by the blue dots, and the mean number of fragmentation cleavage sites present is shown by the blue line. The mode of each peptide length is highlighted by the green bar and the maximum number that could be present (peptide length - 1) is shown by the red line.

also do not include noise peaks, which affect performance when present in large volumes. We also evaluate the algorithms using these artificial data with additional random noise as a complementary analysis to provide a deeper insight into their performance.

De novo sequencing relies solely on the individual spectrum to identify the peptide that produced it. In contrast to database searching that can match peaks independently, *de novo* algorithms must predict and recreate each cleavage, even if no peaks from it exist in the spectrum. When available, many different fragment ions from one cleavage site serve as stronger evidence for that particular fragment as being correct. When no fragment ions from a cleavage site are present there is no direct evidence for the adjacent amino acids in the spectrum and so these are more difficult to determine.

Figure 3.1 shows the distribution of fragmentation cleavage sites present for all peptide lengths in both CID and HCD data. For both data types, shorter peptides matched by the database search are more likely to have a fragment ion from each cleavage in the spectrum. As the length of the peptide increases the mean number of fragmentation cleavages in the spectra (blue line) deviates from the maximum number possible (red line). The variance, indicated by the box plots, also increases as peptide length increases. This effect is more evident in HCD data. HCD provides higher resolution peaks and the ability to use smaller fragment mass tolerance for the database search. This means random matches are less likely and so fewer matching peaks are needed by the database search algorithms for a significant match.

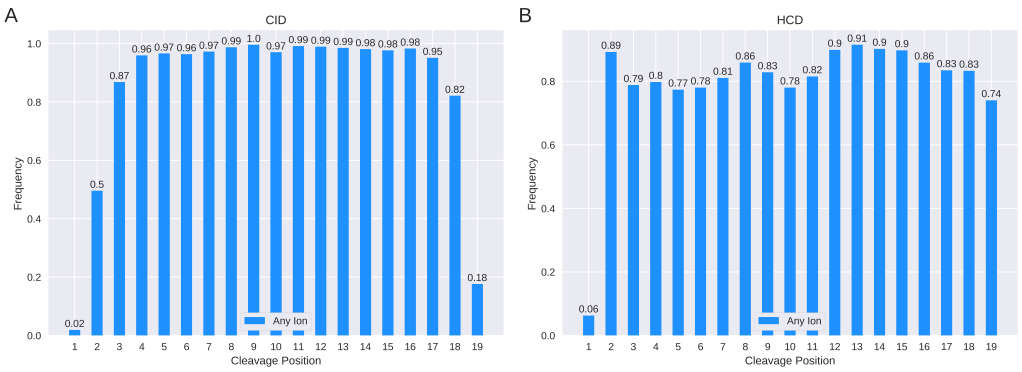

Figure 3.2: Fraction of spectra with one or more ions at each cleavage position. The figure shows the fraction of spectra, for length 20 peptides, that contain one or more ions at each fragmentation cleavage site. A contains all peptides of length 20 from the four CID datasets used in this study with B containing all peptides of length 20 from the four HCD datasets. Numbers on top of the bars indicate their relative frequency.

Both Novor and DeepNovo look for b and y ions, as well as peaks created from their neutral losses of both ammonia and water, to identify peptides. Using chains of fragment ions they can identify amino acids through their mass differences. For both the CID and HCD data, we consider the frequency with which spectra contain any fragment ion from the possible cleavage positions along the peptide backbone. Figure 3.2 shows how likely each cleavage position in a peptide of length 20 identified through database search is to be represented by an ion in the spectra.

Length 20 was chosen as it revealed some interesting patterns with other peptide lengths available in Appendices (Appendix A Figures 1-3). Just 2% of CID spectra and 6% of HCD spectra of peptides of length 20 had an ion from the first fragmentation cleavage site. The first cleavage site also had a below average rate of occurrence in other length peptides (Appendix A Figures 1-3). For peptides of length 14, the median peptide length, fragment ions from the first cleavage appeared in 37% of CID spectra and 33% of HCD spectra (Appendix A Figure 2). While 74% of HCD spectra of length 20 peptides had at least one ion from the last (19^{th}) cleavage site, this number fell to 18% for CID spectra. Fragmentation cleavage sites closer to the centre of the peptides had a much better chance of being represented in the spectra. This trend was shared among all peptide lengths (Appendix A Figures 1-3). For both CID and HCD peptides of length 20, each cleavage site from position 3 to 18 and 2 to 19 respectively, was represented over 74% of the time.

3.4.2 Noise peaks outnumber peptide peaks

Further complicating the identification process is the abundance of peaks in the spectrum which do not belong to the peptide and are classified as noise [173].

The distribution of all peaks in the data is shown in Figure 3.3. Each point represents the mass-to-charge ratio (m/z) and normalised intensity values of a peak in the data. A random selection of 1% of all peaks were used to make the plot readable. The peaks are categorised by those that can be explained by the assigned peptide (peptide peaks) and those that cannot (noise). Both distributions are skewed to the right with very few peaks greater than 1500 m/z. This trend is still observed even when controlling for peptide mass. Noise peaks outnumber those from the peptide approximately 15:1 in the CID data with the ratio being approximately 7:1 in the HCD data. While higher intensity ions are generally seen as more likely to come from a peptide, Figure 3.3 shows how this alone is insufficient evidence. The quantity of noise peaks is equal to or above the quantity of peptide peaks at all intensity levels (Appendix A Figure 4).

Only 6.3% of peaks in the CID data were attributable to the peptide assigned by the database search. Although this number more than doubled to 13% for HCD data as the number of noise peaks reduced, the noise peaks that were present were of a higher average intensity.

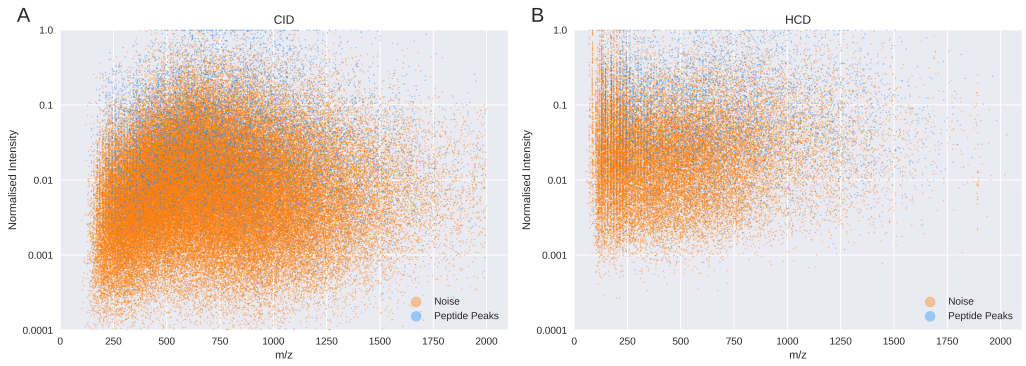

Figure 3.3: Scatter plot of noise and peptide peaks. Scatter plot of the distribution of peak m/z and normalised intensities for both the four CID (A) and four HCD (B) datasets. Peaks attributable to each peptide are shown in blue with noise peaks shown in orange.

3.4.3 De Novo algorithm performance exponentially decreases with increasing peptide length

Figure 3.4A shows the peptide length distribution of the total CID dataset, the number of peptides that had each fragmentation cleavage site represented in the spectrum and the number of peptides that each algorithm predicted correctly. In total, Novor predicted 5768 (24%) of the 23821 peptides correctly while DeepNovo managed 7870 (33%). DeepNovo performed better than Novor for all peptide lengths. Of the 2798 CID peptides of length 11, the most common length, DeepNovo correctly predicted 1243 (44%), while Novor correctly predicted 929 (33%). For length 8 peptides, the shortest peptides in the data, Novor correctly predicted 68% of the peptide sequences correctly while DeepNovo correctly predicted 76%. Novor successfully predicted just 5 peptides of length greater than 20 and none greater than 24. DeepNovo predicted 175 peptides with length greater than 20 and 37 greater than 24.

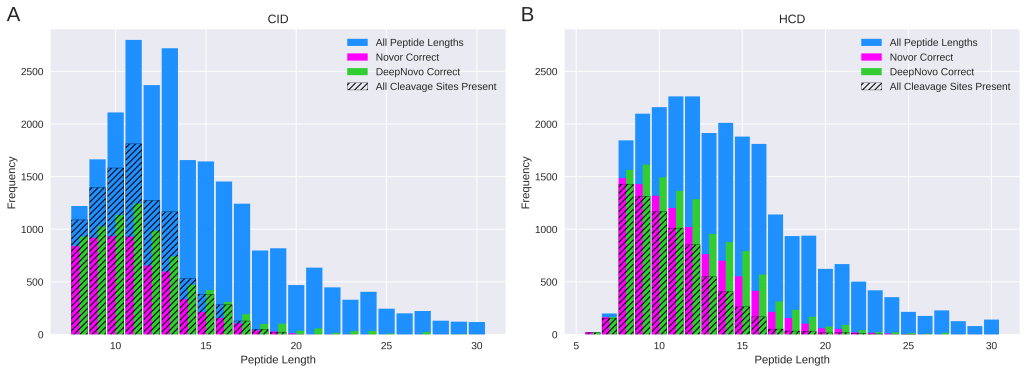

Figure 3.4: Correct peptide prediction distribution. Distribution of the correct peptide predictions of both algorithms for the four CID (A) and four HCD (B) datasets. The total number of peptides in the data of each length is shown in blue, with the number containing a fragment ion from each cleavage site shown by the hatching. Numbers of correct Novor predictions are shown in magenta with correct DeepNovo predictions shown in green

The same distributions are shown for HCD data in Figure 3.4B. DeepNovo predicted more peptides than Novor correctly for almost all peptide lengths. Novor did perform better for lengths 6 and 29, but due to the small sample size at these lengths this cannot be considered as significant. Of the 25007 HCD peptides, DeepNovo predicted 11705 (47%) correctly whereas Novor predicted 9710 (39%) correctly. The accuracy of both algorithms was greater across all peptide lengths compared to the CID data, highlighting how technological advances directly impact on algorithm performance. There were 2262 HCD peptides of length 11, again the most common length, of which DeepNovo correctly

predicted 1305 (60%) and Novor correctly predicted 1202 (53%). DeepNovo and Novor correctly predicted 1564 (85%) and 1485 (81%) of the 1844 length 8 peptides respectively. The relative frequency of correct peptides across the different peptide lengths is shown in Appendix A Figure 5. Here an exponential decrease in peptide accuracy for both fragmentation types is observed as the peptide length increases.

The trends shown in Figure 3.4 are not only the result of the decreased prevalence of fragmentation cleavage sites as peptide length increases. As the number of amino acids in a peptide sets the upper limit on the number of cleavages that can be missing, the two variables are correlated. However, when controlling for the number of missing cleavages, increased peptide length still negatively impacts performance. When the number of fragmentation cleavage sites that are missing is held constant, both algorithms show a linear decrease in peptide accuracy as peptide length increases for both data types (Appendix A Figure 6). For HCD data, Novor correctly predicted 86% of peptides of length 8 when no fragmentation cleavages were missing. It only predicted 36% of peptides of length 16 for the same criterion. DeepNovo's accuracy dropped from 91% to 69% over the same interval when there were no missing fragmentation cleavages.

3.4.4 Increasing number of missing fragmentation cleavage sites exponentially decreases *de novo* peptide algorithm accuracy

Peptide ion peaks may be missing in the MS spectra for a variety of reasons. These include the random nature of the fragmentation collisions, the cut-off of the mass spectrometer or how unfavourable fragmentation at a cleavage site is given the amino acid sequence of the peptide [119, 237].

As shown in Figure 3.5, the majority of the correctly identified peptides had at most one fragmentation cleavage site missing from the spectrum. Fewer than 3.6% of CID peptides correctly identified by Novor and 10% of CID peptides correctly identified by DeepNovo had more than one fragmentation cleavage missing. Novor did not predict any peptide correctly with more than 5 cleavage sites missing. CID spectra with more than one missing fragmentation cleavage account for over 36% of the total number of spectra. For HCD data, spectra with more than one missing fragmentation cleavage accounted for 11% of Novor's correct predictions and 12% of DeepNovo's. HCD spectra with more than one cleavage site missing account for 40% of the total.

To more easily compare the performance of the algorithms we also evaluate them using the relative frequency of the correct peptides. Figure 3.6 shows the peptide accuracy and amino acid recall for the data bins shown in Fig 3.5. For both CID and HCD data, there is an exponential decrease in the peptide accuracy of the algorithms as the number of miss-

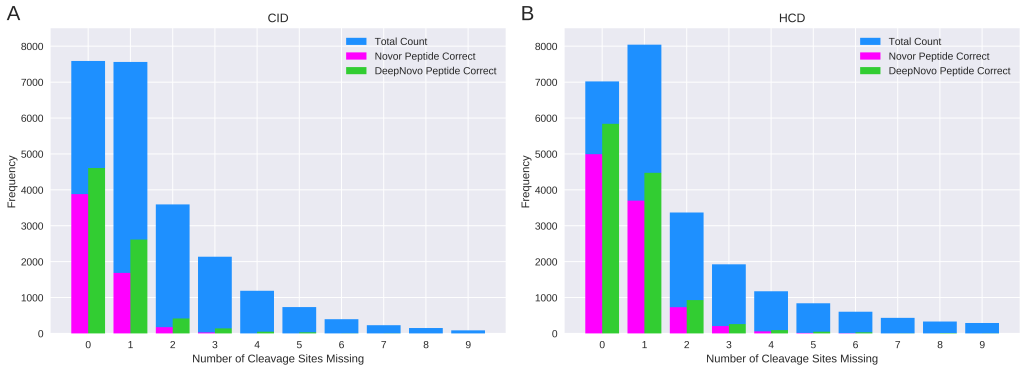

Figure 3.5: Algorithm performance for increasing numbers of missing fragmentation cleavage sites. Bar plot showing the total number of spectra (blue), the total number of peptides correctly predicted by Novor (magenta) and the total number of peptides correctly predicted by DeepNovo (green) for each number of missing fragmentation cleavage sites. The combined CID data are shown in A with the combined HCD data shown in B.

ing fragmentation cleavage sites increases. DeepNovo consistently outperformed Novor in peptide accuracy for both fragmentation types and all numbers of missing cleavage sites. For CID data with 0 missing fragmentation cleavages, DeepNovo predicted 61% of the peptides correctly while Novor only predicted 51% correctly. Neither algorithm predicted any CID peptide with 9 or more missing fragmentation cleavages correctly. The accuracy of both algorithms was higher for HCD data. DeepNovo predicted 83% of peptides correctly while Novor predicted just 71% when no fragmentation cleavages were missing. For 3 missing cleavages the accuracy was 13% and 11% respectively. Once the number of fragmentation cleavage sites that are missing exceeded 3 in CID data, the probability of either algorithm correctly predicting a peptide fell below 4.3% with Novor fairing significantly worse. With HCD data, the peptide accuracy of DeepNovo fell below 7.9% and Novor below 5.0% when more than 3 fragmentation cleavage sites were missing and continued to decrease for greater numbers of missing cleavages.

To further evaluate the performance of the models, we also compare them using amino acid recall. While related to peptide accuracy, amino acid recall gives a finer resolution view of how the algorithms are dealing with missing fragmentation cleavage sites. This is particularly useful for spectra where there are many missing cleavages and peptide accuracy is extremely low.

A clear correlation can be seen in Figure 3.6 between the amino acid recall of both algorithms and the number of fragmentation cleavage sites that are missing. The amino acid recall of both algorithms decreases almost continuously for both fragmentation types as the number of missing cleavages increases.

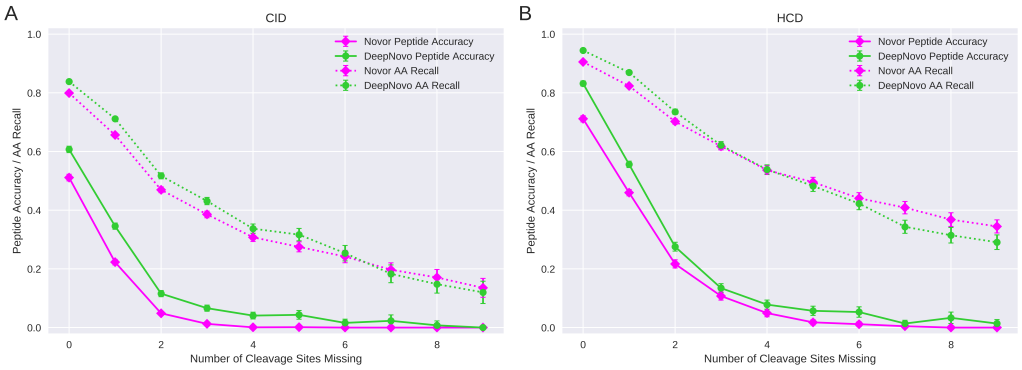

Figure 3.6: Peptide accuracy and amino acid recall. Plots show both algorithms for the different fragmentation types; CID (A) and HCD (B). Peptide accuracy is shown by solid lines with amino acid (AA) recall shown by dotted lines. 95% confidence intervals surround each point with some too small to see.

When no cleavage sites were missing, DeepNovo had an amino acid recall of 84% in CID data and 94% in HCD data. For the same data, Novor had amino acid recalls of 80% and 91% respectively. When there are 4 or fewer missing fragmentation cleavages, DeepNovo outperforms Novor with a greater amino acid recall for both fragmentation types. In contrast, when 5 or more cleavage sites were missing Novor was found to perform best. For spectra with 8 missing fragmentation cleavages, Novor correctly recalled 17% and 37% of the amino acids in CID and HCD data respectively. DeepNovo only recalled 15% and 31% of the amino acids correctly for the same respective data.

When these algorithms are used by researchers, only high-scoring peptides are included in the analysis. Therefore, we also performed a brief analysis using only these high scoring *de novo* peptides. We extracted all peptides above a threshold that gives 90% amino acid recall. The distribution of missing fragmentation cleavage sites in these peptides (Appendix A Figure 7) does not match that of the complete data (Figure 3.5) as both algorithms favour peptides with fewer missing cleavages. Just 1.2% and 11% of Novor's high scoring peptides in CID and HCD data respectively had more than 1 missing fragmentation cleavage site while 9.7% and 18% of DeepNovo's high scoring peptides had more than 1 missing cleavage site for the respective fragmentation types.

To eliminate interactions between features, a further complementary analysis was carried out on artificial HCD data. The distribution of missing fragmentation cleavages in the data is shown in Appendix A Figure 8. Novor correctly predicted 8792 (88%) out of the 9839 artificial peptides with no missing cleavage site. DeepNovo correctly predicted 9342 (95%) of these peptides. Differences in the performance of the algorithms between artificial data and real data may be due to both the more accurate peak placement and

lack of noise in the artificial data. It is difficult to give accurate predictions of peptide accuracy when more than 3 fragmentation cleavages are missing due to the lack of artificial spectra fitting this description.

3.4.5 Impact of noise changes with the number of fragmentation cleavages that are missing

The effect of noise on the accuracy of *de novo* peptide sequencing algorithms is sometimes difficult to elucidate. When viewed alone, the amount of noise in a spectrum did not show a clear negative correlation to performance. This is due to the much stronger influence of the number of missing fragmentation cleavages on algorithm accuracy. Also, much of the noise is at such low intensities that it does not affect the performance of the algorithms. To account for this, in the following analysis we only consider noise above a specific threshold, determined as the median of the noise distribution. We then define the noise factor as the ratio of these high intensity non-peptide noise peaks to peptide peaks. For example, a noise factor of 10 means there are 10 times as many noise peaks as peptide peaks in the spectrum.

Figure 3.7 shows amino acid recall as a function of both the number of fragmentation cleavage sites that are missing and the noise factor for both algorithms and both fragmentation types. Amino acid recall was chosen over peptide accuracy as correct peptides were concentrated to where only zero or one fragmentation cleavage was missing (Figure 3.5). Appendix A Figure 9 shows a similar plot for peptide accuracy. In Figure 3.7 the number of missing cleavage sites increases from top to bottom while the noise factor increases from left to right. As expected, both algorithms perform best when there are very few missing fragmentation cleavages and the noise factor is low.

The distribution of the number of spectra in Figure 3.7 is not uniform. Data points toward the extreme right and bottom of the graph have fewer and fewer spectra in them as these combinations of missing fragmentation cleavages and noise factor are less likely following the database search. White squares are data points where no spectra meet that particular combination. A few outliers near the white squares exhibit unusually high recall, inconsistent with the rest of the graph. These are data points where sample sizes are small and so do not reflect the trends seen in the rest of Figure 3.7.

The relationship between noise and amino acid recall is linked with the absence of fragmentation cleavage sites. As the number of missing cleavage sites increases, fewer and fewer amino acids are correctly recalled from spectra with high noise factors. Performance decreases from where noise peaks and missing fragmentation cleavages are few (top left) to where both noise and missing fragmentation cleavages are more prevalent (bottom right) for each algorithm and fragmentation type. For both fragmentation types, DeepNovo is

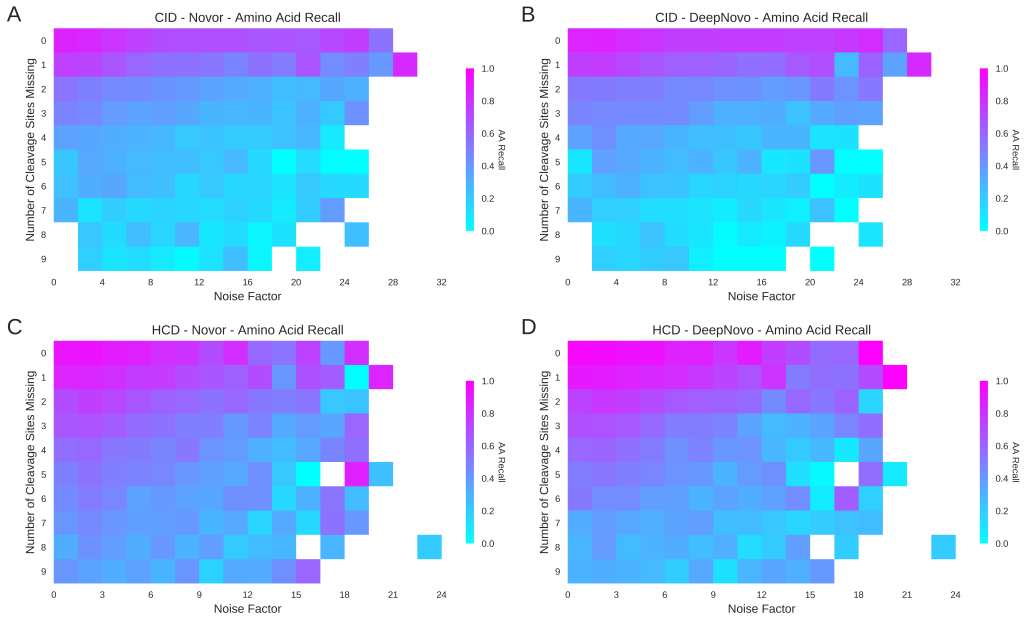

Figure 3.7: Amino Acid recall as a function of the number of missing fragmentation cleavage sites and the Noise Factor. Higher amino acid (AA) recall is shown in pink, with lower recall shown in cyan. Performance of Novor across the two fragmentation types are shown on the left (A and C) with the performance of DeepNovo shown on the right (B and D). CID data are shown on top (A and B) with HCD data shown on the bottom (C and D).

less affected by noise than Novor. Amino acid recall does not fall as sharply as with Novor as the noise factor increases. As seen in Appendix A Figure 9, the peptide accuracy of Novor also decreases rapidly as the noise factor increases for both CID and HCD data. The effect is less acute for DeepNovo but still present. The trend is also much stronger for both algorithms in HCD data where the noise considered is of a higher average intensity and so has a much stronger influence on algorithm prediction.

To isolate the effect of noise from missing fragmentation cleavages we also analysed artificial data with additional noise peaks. To eliminate confounding factors, the artificial data were duplicated and each spectrum in a duplicate was given the same factor of random noise. Appendix A Figure 8 B shows the linear decrease in performance as the noise factor was increased. Again, Deepnovo was less affected than Novor by the increased noise.

3.4.6 *De novo* algorithms can correctly predict amino acids missing from spectra

Earlier analyses showed the ability of both algorithms to correctly predict peptides when fragmentation cleavage sites are missing from the spectra (see Figure 3.6). Although the performance of the algorithms is severely affected as the number of missing fragmentation cleavages increases, the algorithms are still able to make some accurate predictions. When one fragmentation cleavage is missing Novor had a CID peptide accuracy of 22% and a HCD peptide accuracy of 46%, while DeepNovo had a CID peptide accuracy of 35% and HCD peptide accuracy of 56%. To investigate how algorithms deal with missing fragmentation cleavages we compared how often each cleavage site was represented by a fragment ion in the spectra to how often it was correctly identified by the *de novo* algorithms (Figure 3.8).

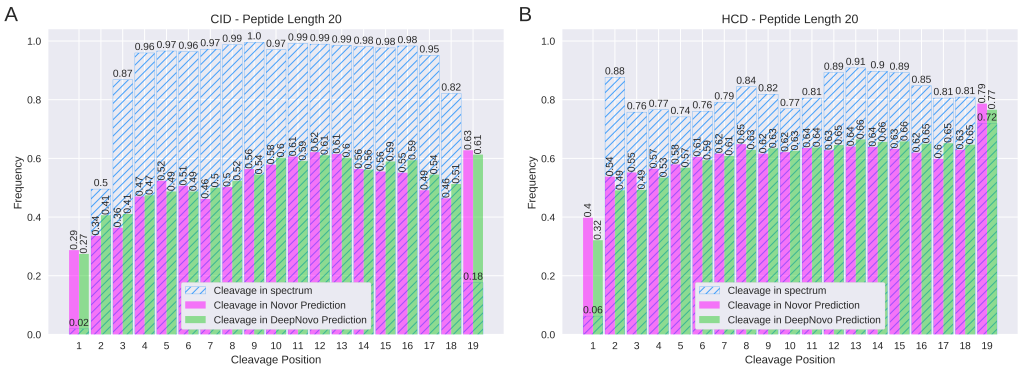

Figure 3.8: Algorithm cleavage site predictions compared to missing cleavage sites. The hatched blue bars represent the fraction of spectra that contain an ion from that cleavage site in the peptide. The magenta (Novor) and green (DeepNovo) bars show the fraction of peptides predicted by each algorithm that contained that same cleavage site. Numbers on top of the bars indicate their value.

As can be seen in Figure 3.8A, CID peptides of length 20 are more likely to be missing a fragmentation cleavage site nearer the end of the peptide. This peptide length was selected as it highlights some interesting characteristics of the two algorithms. Other peptide lengths can be found in Appendices (Appendix A Figures 1-3). As mentioned previously, only 2% of peptides of length 20 in CID data have an ion from the first cleavage position (b_1 or y_{19}) in the spectrum. However, both algorithms account for this fragmentation cleavage with their predicted peptides far more often than it appears in the data. Novor correctly identifies this cleavage position 29% of the time whereas DeepNovo correctly identifies it 27% of the time. Novor correctly predicted the 19^{th} cleavage position in 63% of the peptides while DeepNovo predicted it correctly in 61%. This cleavage site

was represented in only 18% of CID peptides of length 20. Even though both algorithms appear to perform similarly in Figure 3.8A, DeepNovo predicted more than three times as many length 20 CID peptides correctly, when compared to Novor.

The corresponding graph for HCD data is shown in Figure 3.8B. The first cleavage site is only present in 6% of spectra. Yet, Novor accounts for this site in 40% of the data and DeepNovo in 32%. Novor performs better than DeepNovo on the first and last cleavage sites in both HCD and CID data despite DeepNovo performing better overall. DeepNovo's correct predictions are less evenly spread than Novor among all the peptides with a small subset containing most of the recalled amino acids. Other peptide lengths can be found in Appendices for which similar trends were observed (Appendix A Figures 1-3).

3.5 Discussion

De novo peptide sequencing is a growing field with machine learning fuelling its development. Historically, effective design of *de novo* algorithms was difficult with previous methods relying on human expert knowledge. Including this knowledge in the design of machine learning algorithms is not straightforward as it is difficult to capture and may significantly increase the complexity of the corresponding algorithms [153]. In fact, most of the fragmentation rules identified by researchers are not included in proteomics identification tools [166]. Machine learning may allow algorithms to learn these features automatically as they uncover patterns in the data. However, the design of algorithm architectures that would facilitate this learning is non-trivial and requires a deep understanding of the data and fragmentation process.

As shown in our analysis and others [177], the performance of modern algorithms on artificial data far exceeds that of real data. Not only does this mean that analysis of algorithms on artificial data is not directly applicable to real data but it also highlights how the current bottlenecks lie with features of the data and the algorithms' inability to cope with them. To elucidate some of these data features and show how they might be addressed, we evaluated two state of the art *de novo* sequencing algorithms on both real and artificial MS/MS data. We determined both the prevalence and effects of missing fragmentation cleavages and noise on *de novo* sequencing algorithms. We also investigated how the state of the art algorithms overcome these features.

We firstly analysed the performance of DeepNovo and Novor with respect to peptide length to ensure it did not confound later observations. Like in previous studies [37, 177], an increase in peptide length was found to negatively affect performance. Furthermore, we demonstrate the peptide accuracy exponentially decreases in response to an increase in length. This is likely due to the fact that *de novo* algorithms must predict each amino

acid, meaning the likelihood of at least one incorrect prediction increases with the number of amino acids.

Missing fragmentation cleavages were found to be the main problem with the data which *de novo* sequencing algorithms must overcome. The vast majority of peptides correctly identified come from spectra with zero or one missing fragmentation cleavages (Figure 3.5). As the number of missing cleavage sites increases, identification becomes more difficult with correct peptides from both algorithms becoming non-existent. Consequently, almost all of the peptides scored highly by the *de novo* algorithms have zero or one missing fragmentation cleavage sites (Appendix A Figure 7). Fragmentation cleavages were found to be more likely to be missing for longer peptides with larger mass values (Appendix A Figure 3) and toward the ends of the peptide.

Similar to other studies [37], we found amino acid recall tended to be better toward the middle of the peptide. Figure 3.8 shows that the reduced cleavage prediction accuracy is a result of the reduced prevalence of fragment ions from those cleavages. This in turn leads to reduced amino acid recall. A clear relationship can be seen between the presence of a fragmentation cleavage in the spectra and the presence of that cleavage in the predicted peptide. These missing cleavages explain the equal mass multi-amino acid substitutions observed by these studies [37, 177]. A missing cleavage leaves a mass-gap in the chain of peptide fragments which can be filled by a number of equal mass amino acid sequences.

Noise has historically been seen as a major problem in *de novo* peptide sequencing [171]. For graph based methods in particular it also increases the complexity of the *de novo* sequencing problem exponentially by increasing the number of nodes and edges [44]. These additional peaks only become a major problem when present in large quantities and if of high intensity. We found that DeepNovo is better able to deal with high intensity noise compared to Novor in both real and artificial data.

Novor and DeepNovo employ a range of techniques including machine learning and dynamic programming as they attempt to overcome these challenges.

Both algorithms step through the spectrum one amino acid at a time. Using machine learning they try to learn what function of the features at that particular point distinguish peptide peaks from non-peptide peaks. The success of this approach is seen in their effectiveness against noise. Also, DeepNovo only considers nearby peaks when making each amino acid prediction, meaning many of the noise peaks are inconsequential. Similarly, Novor scores each peak independently thus limiting the effect of noise. Unlike DeepNovo however, Novor creates a graph of all peaks. As with all graph based *de novo* algorithms, this increases the complexity of the solution space. The difference in approaches is highlighted by DeepNovo's greater performance with respect to noise.

A major strength of Novor's graph based approach is its amino acid recall when many fragmentation cleavages are missing, where it outperforms DeepNovo. When many

cleavage sites are missing, it is almost guaranteed that the highest scoring path in the graph does not reflect the correct peptide. This is shown by Novor's low peptide accuracy in this range. However, the algorithm still manages to incorporate short subsequences of fragment ions that are present into the highest scoring path. While the complete path may not be present, these short subsequences will still be scored highly by the algorithm and so are likely to appear in the highest scoring path giving rise to a partially correct peptide. DeepNovo, on the other hand, has no means of rejoining the correct path once an incorrect step is made and so has more complete matches but fewer partial matches than Novor for this type of data. This is the reason DeepNovo maintains a higher peptide accuracy than Novor while having a lower amino acid recall at greater numbers of missing fragmentation cleavages.

The independent scoring of graph nodes means Novor cannot encapsulate the long range relationships of peptide fragmentation. Tiwary *et al.* (2019) showed that the entire peptide composition will have an impact on the peak intensity of each fragment ion [237]. Therefore accurate amino acid prediction will require the consideration of fragment ions from the entire peptide. DeepNovo uses an LSTM to keep track of fragment ions already encountered. It can then take advantage of their encodings for aiding the prediction of amino acids further along the peptide. This is particularly useful when DeepNovo is presented with a mass-gap caused by missing fragmentation cleavages. It can leverage the information encoded from the spectrum it has already encountered to replace the absence of peaks in its current position and make accurate predictions. The largest mass-gap correctly traversed by DeepNovo in this research spanned seven cleavage sites compared to a maximum of three for Novor. DeepNovo also uses dynamic programming, similar to the knapsack problem, to make up for the fact it can only see as much as one amino acid ahead at a time. As the true mass of the correct peptide is known, DeepNovo limits the number of amino acids to consider at each step by only allowing those that are possible given the remaining mass of the peptide. This is particularly useful at the end of the peptide where the number of options will be significantly reduced.

While algorithms are improving, our analysis has uncovered some limitations in their approaches. Unlike their database counterparts, the performance of the algorithms is not independent of the peptide length as both algorithms build the peptides up from individual components. Step-by-step predictions and independent peak scoring simply do not encapsulate all the necessary information from the fragment process. The whole of the peptide, and hence the whole of the spectrum is needed for exact amino acid prediction [237]. DeepNovo does incorporate some long range interactions, but only for the final amino acids predicted and only if those already predicted are correct. Graphs are the most suitable way to capture all the complex interactions but Novor's machine learning is not applied over the graph but only on the peak scores. Thus, complete spectrum encoding

would encapsulate the complex nature of peptide fragmentation leading to more accurate predictions. The combination of the strengths of these two aforementioned models can be harnessed using graph neural networks (GNNs) [211]. In GNNs, features of each node, such as its m/z value and intensity, can be encoded with a neural network and passed along the edges of the graph to other nodes. Through this mechanism, peaks from one end of the graph could influence the prediction of amino acids at the other. Similar applications have been shown such as the Graph2Seq model [263]. This model was shown to be extremely effective in tasks such as path finding, where an optimal sequence is predicted from a complex graph. This is similar to *de novo* sequencing where the sequence would be the peptide. Graph2Seq uses a GNN to encode the graph before an attention based LSTM is used to predict each element in a sequence. While each node will share information with its neighbours, the use of attention means the model can focus on multiple relevant parts of the graph at one time. In that way, a *de novo* peptide model could learn to focus on those peaks shown to be related to the sequence [237].

De novo algorithms may also benefit from a pre-processing step that removes noise peaks. Previous noise removal algorithms focus on a peak's intensity and its rank among the other peaks [173, 64]. As shown in Figure 3.3, intensity alone is an insufficient discriminator and peak interactions must be considered. Denoising spectra needs the same long range interactions, amino acid predictions does. Both tasks are essentially trying to find a function that distinguishes between peptide peaks and non-peptide peaks. Machine learning can learn such functions as shown by the performance of both algorithms. However, the noise peaks causing the problems for these algorithms are still scored highly and their removal requires more intelligent systems. The incorporation of long range interactions into a noise removal model would provide increased resolution, which in turn should improve the *de novo* algorithms' performance in their current state.

3.6 Conclusion

The availability of large datasets and the addition of machine learning has led to notable advances in *de novo* peptide sequencing algorithms. Real data analyses revealed that noise peaks are far more abundant than peptide peaks while most peptides have missing fragmentation cleavages. DeepNovo was found to perform best overall with Novor surpassing it only for amino acid recall when many cleavages were missing. Missing fragmentation cleavages were found to be the biggest obstacle for both algorithms with both peptide length and noise also affecting performance. DeepNovo's recurrent neural network helped counteract the effect of missing fragmentation cleavages. Future *de novo* algorithms may benefit from a complete spectrum encoding that encapsulates the long range dependencies of peptide fragmentation. While the quality of data is increasing,

improvements in *de novo* peptide identification algorithms could allow new insights from past research. Future *de novo* algorithms will also benefit from the advances in the field of machine learning. Recently developed machine learning algorithms, such as graph neural networks, may help better capture the intricate relationships of peptide fragmentation thereby advancing performance in this space.

Chapter 4

Application of a Novel Hybrid CNN-GNN for Peptide Ion Encoding

The work outlined in this chapter was published in:

McDonnell, K., Abram, F., and Howley, E. Application of a novel hybrid CNN-GNN for peptide ion encoding. Journal of Proteome Research (2022).

4.1 Abstract

Almost all state-of-the-art *de novo* peptide sequencing algorithms now use machine learning models to encode fragment peaks and hence identify amino acids in MS spectra. Previous work has highlighted how the inherent MS challenges of noise and missing peptide peaks detrimentally affect the performance of these models. In the present research we extracted and evaluated the encoding modules from 3 state-of-the-art *de novo* peptide sequencing algorithms. We also propose a CNN-GNN machine learning model for encoding peptide ions in tandem MS spectra. We compared the proposed encoding module to those used in the state-of-the-art *de novo* peptide sequencing algorithms by assessing their ability to identify b ions and y ions in MS spectra. This included a comprehensive evaluation in both real and artificial data across various levels of noise and missing peptide peaks. The proposed model performed best across all datasets using two different metrics (AUC and average precision). The work also highlighted the effect of including additional features such as intensity rank in these encoding modules as well as issues with using the AUC as a metric. This work is of significance to those designing future *de novo* peptide identification algorithms as it is the first step towards a new approach.

4.2 Introduction

Proteins are large macromolecules which perform essential functions for all life on earth [82]. They are composed of long chains of amino acids which define their structure and consequently their function. As they are fundamental to all living organisms, the accurate identification of these proteins has wide ranging significance from the detection of cancer [114] to the optimisation of resource recovery from food waste products [184]. When profiling protein expression (proteomics), the identification process typically involves the enzymatic digestion of the proteins down into smaller sequences of amino acids called peptides. These peptides are then characterised using liquid chromatography tandem mass spectrometry (LC-MS/MS). In this process, peptides of a particular sequence are separated and isolated using liquid chromatography and a mass analyzer. They are then fragmented using collision based methods such as high energy collision dissociation (HCD). For each fragmented peptide sequence, the resulting fragments pass through a second mass analyzer, producing spectra with a unique fragmentation pattern for that sequence. The originating peptides can then be identified from the spectra using a database search.

During the fragmentation process the peptides are generally split between amino acids at the peptide (amide) bonds [232]. Cleavage at a peptide bond results in b ions and y ions. Other ions, such as a ions, appear when cleavage occurs at other bonds along the amino acid chain. Fragment ions can then suffer neutral losses of both ammonia and water thereby shifting the m/z of their peaks. Ions can also be doubly charged resulting in m/z values approximately half that of their singly charged counterparts. As they are made up of subsequences of amino acids, two singly charged peaks of the same ion type from neighbouring peptide bonds will be separated in the spectrum at a distance equal to the mass of the amino acid between them. Database search methods work by creating the theoretical peaks for each peptide in the database given the possible fragmentation sites. They then compare these to the peaks in the spectra. A peptide is assigned to a spectrum if its set of theoretical peaks significantly matches the observed array of peaks in that spectrum.

Although widely used, database search methods may only utilise a small fraction of the MS spectra recovered during an experiment [80, 95]. This is partly due to the large protein databases needed to cover the complete set of proteins being investigated [181]. Larger databases increase the probability of a false positive peptide match therefore increasing the false discovery rate (FDR). To account for this, search algorithms must adopt more stringent criteria for a positive match, thereby excluding many correct but lower scoring matches. For metaproteomics experiments, where there are a large number of possible organisms and therefore even larger databases, the problem is even worse[107].

De novo peptide sequencing is becoming a competitive alternative to these database search methods [177]. In this strategy, peptides are identified using the spectra alone. This alleviates the need for a database and its associated challenges. An important use of *de novo* peptide sequencing is the identification of neoantigens for cancer immunotherapy [70], where peptides specific to a tumour may not be available in a database. The field has benefited tremendously from advancements in machine learning in recent years, with machine learning models now incorporated into almost all state-of-the-art *de novo* identification algorithms due to their unrivalled pattern recognition capabilities [153, 241, 199]. In this context, machine learning models encode meaningful parts of the spectrum which may help infer the amino acid sequence. The models can learn to differentiate between peptide ions and noise peaks which could be up to 28 times as prevalent [161]. From this, the algorithms infer the fragmentation sites and thereby the amino acid sequence. However, previous analysis has shown how difficult this inference is due to the aforementioned levels of noise and the even greater challenge of missing ion peaks [161]. McDonnell *et al.* found that increasing numbers of fragmentation sites without any representative ion caused an exponential decrease in the accuracy of *de novo* algorithms.

Three such state-of-the-art *de novo* peptide sequencing algorithms that use machine learning are Novor, DeepNovo and PointNovo [153, 241, 199]. Novor uses a random forest (RF) model to score likely fragmentation sites using ions from neighbouring cleavages. DeepNovo and PointNovo use convolutional neural networks (CNNs) to encode these neighbouring ions to infer the next amino acid in the sequence. However, as shown in our previous work, these algorithms do not fully encapsulate the fragmentation process [161]. This is in part due to limitations of their encoding modules. New approaches are needed that can account for this shortcoming.

With many types of machine learning models available to encode peptide ions and each part of a large and complex algorithm, it can be difficult to select the appropriate one when designing *de novo* peptide sequencing algorithms. Therefore we extract the encoding models from DeepNovo, PointNovo and Novor and perform a comprehensive evaluation on their ability to identify peptide ions. Also, we propose a novel encoding module, a hybrid CNN-GNN and compare it to the modules used in these state-of-the-art algorithms. This is the first step towards a new *de novo* sequencing approach. The impact of neighbour independent features on these models is also investigated. Finally we identify issues with the common evaluation metric Area Under the receiver operating characteristic Curve (AUC).

4.3 Background and Related Work

Peptides are made up of building blocks called amino acids and when they are fragmented using HCD, they generally cleave between these amino acids at the peptide bonds [218]. Fragmentation at a peptide bond results in b ions and y ions, depending on which side of the cleavage the fragment is from. The amino acid sequence of proteins, and hence peptides, are by convention always ordered from the N-terminus to the C-terminus with b ions relating to the N-terminus fragment and y ions relating to the C-terminus fragment. Fragmentation sites are also ordered from the N-terminus to the C-terminus. Fragmentation along the peptide chain means that the peaks of fragment ions of the same type appear at intervals from one another, equal to the mass of the constituent amino acids. Through this relationship, the sequence of amino acids can be identified by looking for a sequence of spectrum peaks separated by amino acid masses. Identification of these fragment ions is therefore essential to the *de novo* prediction of peptides.

As they cannot rely on a database to know which spectrum peaks correspond to fragment ions, *de novo* algorithms look at features of each peak as well as their relationship to other peaks to distinguish likely candidates. The amino acid sequence is then built up by moving from one peptide peak (fragment ion) to the next. The step size between peaks indicates the amino acid in the sequence.

Before fragmentation the ionised peptides are separated by their mass in the first mass analyzer. The ions that are selected for fragmentation are called the "parent ions" as they are broken down into fragment ions. While the peptide sequence is not known, its mass can be inferred from the parent ion. The mass of complementary b ions and y ions that came from the same fragment site can then be identified using the following formulae:

$$y = (M + H)^{1+} - b + H \tag{4.1}$$

$$b = (M + H)^{1+} - y + H \tag{4.2}$$

where y is the mass of the y ion, b is the mass of the b ion, M is the mass of the peptide and H is the mass of a hydrogen atom.

These can be generalised to the formula:

$$m(ion_i) = M - m(comp\_ion_i) + 2H \tag{4.3}$$

where $m(ion_i)$ is the mass of an ion from the i^{th} fragmentation site and $m(comp\_ion_i)$ is the mass of the complementary ion from the i^{th} fragmentation site.

Ions of the same type from neighbouring fragmentation sites can be identified using

the following formula:

$$m(ion_{i+1}) = m(ion_i) + m(AA) \tag{4.4}$$

where $m(AA)$ is the mass of an amino acid.

Combining equations (4.3) and (4.4) to look for complementary ions from a neighbouring fragmentation site we get the following:

$$m(ion_{i+1}) = M - m(comp\_ion_i) + 2H + m(AA) \tag{4.5}$$

Some ions lose a neutral molecule of H_2O or NH_3. While the charge is maintained the resulting m/z is shifted by the corresponding mass of the lost molecule. This can then be incorporated into the above equations by subtracting this mass from the ion in question. The following shows the case for the loss of H_2O:

$$m(ion_{i+1} - H_2O) = m(ion_i) + m(AA) - H_2O \tag{4.6}$$

Ions can also be doubly charged (ion^{2+}). To convert a doubly charged peak to its singly charged form one only needs to double its m/z value and subtract the mass of the extra hydrogen nucleus (proton):

$$m(ion_i) = 2 \times (m(ion_i^{2+})) - H \tag{4.7}$$

Ions of other types can occur if fragmentation occurs at bonds other than the amide bond. While these are less likely to occur under HCD conditions, they can easily be incorporated into the search space. N-terminus ions can be calculated with respect to the b ion by taking into account the relevant atomic differences. For instance a ions can be identified by subtracting the mass of CO (28 Da) from the corresponding b ion. In a similar way, the other C-terminus ions can be identified with respect to the corresponding y ion.

The *de novo* peptide identification problem can be solved by creating a graph with every peak as a node [18]. The above equations are then used to create connections between nodes from potential neighbouring fragmentation sites. Passing through the graph from zero to the mass of the parent ion, a peptide sequence will emerge from the amino acid connections used to create the path.

De novo peptide sequencing is not without its difficulties however. Peaks not attributable to the peptide (noise) account for the vast majority of peaks in tandem MS spectra [161]. Furthermore, peaks corresponding to all possible fragment ions from a peptide may not appear in the spectrum. This is particularly problematic when no ion from a fragmentation site is present leading to ambiguity in the order or identification of

the amino acids. The absence of fragment ions can be partly attributed to the fact that the cleavage of some peptide bonds may be less energy favourable than others [123].

The machine learning modules of *de novo* peptide sequencing algorithms create encodings which are used to indicate the likelihood of a fragmentation site or particular amino acid at that position. The modules encode the intensity of the peaks and those near them defined by equations (4.4) and (4.5) as well as other features in the spectrum.

Tiwary *et al.* [237] showed that there is a significant relationship between the ion intensity and the complete amino acid sequence, not just the neighbouring cleavages. However, in many *de novo* algorithms, only peaks from possible neighbouring peptide bond cleavages are considered as the number of possible locations grows exponentially for cleavages farther away. These long-range interactions should be considered in future *de novo* peptide identification algorithms. Graph neural networks (GNNs) are a great way to encapsulate the long chain-like structure of the peptide without an exponential increase in complexity, but so far no algorithm has utilised them in the context of *de novo* peptide identification [161].

Novor is a *de novo* peptide sequencing algorithm that uses an RF machine learning model [153]. The RF model encodes related peaks to score the likelihood of a fragmentation site. It also uses other features of peaks such as their intensity rank to influence its decision. Using the above equations, the mass difference between fragmentation sites can be used to identify amino acids. Novor then uses dynamic programming to find the highest scoring combination of fragmentation sites that fulfil the peptide mass.

DeepNovo is a more recent *de novo* algorithm using both CNNs and a long short-term memory network (LSTM) to encode peptide ion peaks as it steps through the spectrum [241]. Starting at one end of the peptide, the algorithm looks to identify each amino acid in the sequence one-by-one. Given its current position, the CNN encodes the sections of the spectra where the next possible peptide peaks may occur (equations (4.4) and (4.5)). The output of the CNN is then passed to an output layer or LSTM to identify next most likely amino acid. Dynamic programming is also used to limit the number of possible amino acids that can be predicted. Following this, the current position is updated to either that of the b_n ion or y_n ion given the n amino acids already predicted and the direction of prediction. While DeepNovo's LSTM can use information from previously traversed peaks, its encoding module does not have any way of looking more than 1 amino acid ahead. Also, as stated in the original paper, the LSTM only uses the previous two amino acids to influence its decision. This is done by resetting the LSTM using the output of a second CNN which transforms the spectrum into a vector encoding. At each prediction step, this vector is then used to initialise the LSTM before the previous two amino acid predictions are encoded. This was found by the authors to reduce overfitting [241]. Although DeepNovo does use a beam search to explore a greater number of possible

sequences, each is generated using this limited information.

An updated version of DeepNovo has been released called PointNovo [199]. The methodology of this approach is very similar to its predecessor. However, the CNN used to encode the spectrum windows is replaced by a T Net which uses absolute peak differences and not spectrum sections. The T Net is essentially a 1-dimensional CNN with a kernel size of 1. The T Net encodes both the difference between each peak and the theoretical position of neighbouring fragment ions as well as the intensity of each peak. Again, this encoding can be either passed to an LSTM or an output layer to predict the next amino acid in the sequence. Unlike its predecessor, the LSTM used by PointNovo encodes all previous amino acid predictions. Also, as it uses the difference values and not discretised windows, PointNovo is more robust to different resolution mass spectrometers [199].

Our previous work showed limitations in the approaches of modern algorithms [161]. The peptide accuracy of the models was found to decrease exponentially with increasing numbers of missing fragmentation cleavages. Also, DeepNovo showed a much steeper decline in amino acid recall than Novor as the number of missing fragmentation cleavages increased. This led to Novor performing better for spectra with more than 4 missing cleavages. This previous work highlighted the need to explore different methodologies to address the problems of *de novo* peptide sequencing while also showing a potential avenue of exploration.

The work described here constitutes the first step in the exploration of such new methodologies with the use of graph neural networks (GNNs) for peptide ion encoding. GNNs can capture the graph like structure of the peptide fragmentation process as well as having the pattern recognition capabilities of neural networks. Therefore their architecture can encode more spectrum information than CNNs alone. However the following evaluation does not involve the integration of GNNs into the aforementioned algorithms. The GNN module proposed updates all nodes simultaneously. This is in contrast to the step-like architecture of DeepNovo and PointNovo and so it does not easily fit into their algorithm. Furthermore, the code of Novor is not open source and so we cannot integrate our model into their architecture either. Nevertheless we wish to benchmark this approach against other encoding modules used in this space. As such we compare a novel CNN-GNN approach with the encoding modules of the three state-of-the-art *de novo* peptide sequencing algorithms on their ability to identify peptide ions.

This means that the encoding modules employed by DeepNovo and PointNovo will not be doing exactly what they were designed to do. While it may seem likely that peptide ion identification and amino acid identification are related, this has not been explicitly shown in this research. Nonetheless, *de novo* encoding modules should be designed to learn features of spectra that link observed fragmentation patterns to the corresponding

peptides. As outlined earlier, identification of the chain of backbone ions can elucidate the peptide. Therefore, effective encoding modules should be able to identify ions from this chain. While a step-by-step approach is currently the state of the art, perhaps a more complete spectrum encoding is required. This research proposes integrating the long-range relationships between backbone ion peaks into the encoding process through the use of GNNs. The aim is to highlight the potential of GNNs in the context of peptide ion identification and hence *de novo* peptide identification.

4.4 Methods

4.4.1 Benchmark Datasets

Real HCD tandem mass spectra, collated by Tran *et al.* [241], were used in this evaluation. HCD data provides greater resolution and mass accuracy than other fragmentation methods [185]. The data are available to download at ftp://massive.ucsd.edu/MSV000081382/. The data are made up of tandem mass spectra from 9 different organisms from 9 different research groups [188, 182, 41, 205, 194, 159, 215, 115, 54]. The spectra were labelled by Tran *et al.* with peptides using a database search with a 1% FDR threshold against the UniProt database [12]. These peptides are assumed to be correct, with their fragment ions serving as ground truth labels in this research (see next section). Table 4.1 shows a summary of the datasets. More details including the precursor and fragment tolerances used are available in the original paper [241].

Data partitioning into training, validation and test sets was also carried out Tran *et al.*. This was done by having separate partitions for each organism. The training and validation data for each organism type were made up of spectra from the other 8 datasets. Then, testing was then done on spectra from the organism itself, essentially performing a leave-one-out cross-validation. Each test set was made up of approximately 10000 spectra from a single organism. 9 models were trained for each encoder type tested in this evaluation, one for each organism type.

4.4.2 Peak Classification

Theoretical peaks were created for each peptide using the Pyteomics module [141]. These were then compared to the peaks in the corresponding spectra. Peaks were labelled as peptide peaks if they matched the theoretical peaks within a tolerance of 0.05 Da.

The ions considered were b ions, y ions, b-H_2O ions, y-H_2O ions, b-NH_3 ions, y-NH_3 ions, b(2+) ions and y(2+) ions. Peaks that could not be assigned to one of these ion types were labelled as noise. If multiple peaks fell within the tolerance only the peak with

the smallest error was assigned an ion. The number of possible/theoretical peaks, given these ion types, is known for each peptide. Each spectrum was then classified by the Fraction of theoretical peptide Peaks Present (FPP) by matching them to the observed spectrum peaks. The number of peaks that remained unmatched (noise) were compared to the number of identified peaks. The relative proportion of these for each individual spectrum we define as the Noise Ratio (NR).

Dataset	Organism	Mean FPP	Mean NR
Yeast	*Saccharomyces cerevisiae*	0.37	6.3
Human	*Homo sapiens*	0.24	5.0
Mouse	*Mus musculus*	0.21	4.1
Bacillus	*Bacillus subtilis*	0.35	6.4
ClamBacteria	*Candidatus Thiodiazotropha endoloripes*	0.22	3.7
Honeybee	*Apis mellifera*	0.35	7.4
Ricebean	*Vigna mungo*	0.28	6.0
Tomato	*Solanum lycopersicum*	0.30	4.4
M. mazei	*Methanosarcina mazei*	0.31	6.5
All Data		0.29	5.6

Table 4.1: Summary of real datasets used. FPP is the Fraction of peptide Peaks Present in the spectra. NR is the ratio of noise peaks to peptide peaks.

4.4.3 Artificial Datasets

Additionally, models were also evaluated on artificial data created using the Prosit pipeline [89]. Prosit creates artificial spectra with accurate representations of both the mass and intensities of peptide peaks. As it is a well studied model organism, *Saccharomyces cerevisiae* was selected as the basis for the artificial data. The yeast proteome (UP000002311) was downloaded from Uniprot on 02/07/2021 [12]. Protein sequences were artificially digested using the Pyteomics parser [141]. Artificial spectra were created for each unique peptide (188694) using the Prosit pipeline. To create a manageable dataset size and match the real data test set sizes, a random sample of 10000 spectra were selected as the artificial dataset.

The artificial dataset has a mean fraction of peaks present (FPP) equal to 0.36. The FPP of the artificial dataset is not 1.0 as Prosit does not predict peaks that would be very unlikely to appear in real spectra. However, while the value closely matches that of the real Yeast dataset (0.37), the distribution of ion types and numbers matched are quite different (data not shown). The models were then evaluated on the artificial data with adjusted levels of noise. The Prosit dataset was duplicated 5 times. As it was duplicated the peaks present were the same for each dataset with only the level of noise changing. Then to span the range of values in real data, additional noise was added at ratios to the

peptide peaks of 0, 1, 5, 10 and 15.

The m/z values of the artificial noise peaks were randomly sampled from a uniform distribution between zero and the mass of the peptide attributed to the spectrum. The intensity values of the artificial noise peaks were randomly sampled from a distribution approximating the noise intensity values in the real yeast dataset. This approximation is a log-normal distribution. The mean and standard deviation of the natural log of this distribution is -4.4 and 1.5 respectively.

4.4.4 Ion Identification

Machine learning modules in Novor, DeepNovo and PointNovo all encode fragment ions in reference to the positions of possible b ions and y ions. These encodings then inform either the likelihood that the current peak is from a fragmentation site or the prediction of the next amino acid in the sequence. In the case of DeepNovo and PointNovo, their algorithms move to the position of the next b ion and y ion in the spectrum depending on the latest amino acid predicted and the direction of prediction. In this research, models are evaluated on their ability to identify these b ions and y ions using the features listed in the next section. Each module was used to encode the relevant spectrum sections to vector of length 512, the same size used as DeepNovo. This was then followed by a single output node to give a score to each peak that it is one of the two ions. Models that could encode more information should learn to better distinguish these ions from the other peaks.

4.4.5 Model Features

Features used by the state-of-the-art models were extracted from the spectra as follows. Assuming each peak corresponds to a b ion or y ion, sections of the spectrum surrounding the following locations were identified; the location of all possible peptide ion peaks from cleavages in front of the current ion (equations (4.4) and (4.5)), the location of all possible peptide ion peaks from cleavages behind the current ion (equations (4.4) and (4.5) with changed signs), and the location of the possible complementary ion peak in the spectrum (equation (4.3)). Neutral loses of H_2O and NH_3 as well as double protonation were considered for each ion type. To align with the precision of the algorithms, 0.1 Da windows surrounding the exact positions (0.05 Da each side) were extracted and peaks were placed into the relevant bins, each of size 0.01 Da. The forward and backward ion features have the shape (#AA, #ion types, window size) where #AA is the number of possible amino acids and their modifications (26), #ion types is the number of ion types (8; b ions, y ions, b-H_2O ions, y-H_2O ions, b-NH_3 ions, y-NH_3 ions, b(2+) ions and y(2+) ions) and window size is 10 (0.1/0.01). The features used by the T Net are slightly

different in that instead of discretised windows the absolute difference is used. This is to correspond with the features and module used in the original paper [199].

Alongside neighbouring peaks, Novor also uses expert selected features to enhance model accuracy [153]. These include the peak intensity rank, peak intensity half rank, local intensity rank and local intensity half rank. The addition of these features was included in the models denoted by "+F". For more information on these features see the original paper [153].

4.4.6 Random Forest Model

An RF model was created to emulate the scoring function of the Novor algorithm [153]. The RF model is used by Novor to measure the probability each peak is from a real fragmentation site. The RF evaluated in this research is an approximation of the model used by Novor as the source code is not released. A further description of the Novor algorithm can be found in the original paper [153]. The RF model was trained with the above features (see previous section) as input and a single variable output. It was created using the scikit-learn [192] library. The number of trees in the model was set to 1000. There was no maximum depth for the trees. As Novor uses the additional features in its algorithm, the RF model was only evaluated with these features included (RF+F).

4.4.7 CNN Model

Similarly a CNN model was created to emulate the encoding module of DeepNovo's algorithm [241]. The original code can be found at `https://github.com/nh2tran/DeepNovo`. A complete description of the DeepNovo algorithm can be found in the original paper [241]. Table 4.2 shows a summary of the CNN module. Just like DeepNovo, the first convolution layer in the module has a 1×3 filter and a stride of 1 in both directions. The second layer has a 1×2 filter again with a stride of 1 in both directions. the Max_pool layer has a filter size of 1×3 and strides of (1,2). Finally the layers are flattened and passed to two consecutive dense layers of size 512. It should be noted that in addition to the amino acids, DeepNovo encodes Start, End and Pad tokens. These are set to zero and ignored.

Three of these CNN modules were used in the the complete model, one for each spectrum window (forward, backward and complementary). The output of the three CNNs are concatenated and passed to a 512 dense layer before a single node output layer. The model was created with (CNN+F) and without (CNN) the additional Novor features. The CNN models were trained using focal loss[146] as recommended by Tran *et al.*[240], and the Adam optimisation algorithm.

Layer	Output Shape
Input Window	(None,26,8,10)
Transpose	(None,8,10,26)
Conv2D_1	(None,8,10,64)
Conv2D_2	(None,8,10,64)
Max_pool	(None,8,5,64)
Dense	(None,512)
Dense	(None,512)

Table 4.2: Structure of each CNN module as used by DeepNovo

4.4.8 T Net Model

A T Net module was created to emulate the encoding module of PointNovo [199]. Code for the original implementation can be found at `https://github.com/volpato30/DeepNovoV2` with further information in the original paper [199]. While PointNovo essentially uses the same features as DeepNovo it formats them differently. Instead of spectrum windows the T Net model uses the differences from the theoretical values of each possible amino acid. The T net encodes the differences and intensities of all the possible amino acids with each peak in the spectrum. A sequence of three one-dimensional CNNs converts the input into a 64 dimensional vector followed by two consecutive dense layers of size 512.

Three T Net modules were combined with one each for forward, backward and complementary ions. These were then combined and condensed using a further 512 dense layer as with the CNN above.

PointNovo uses 12 ions which includes a ions, a-H_2O ions, a-NH_3 ions and a(2+) ions in addition to the 8 ions listed above. Therefore the T Net was included in both 8 (Tnet8+F) and 12 (Tnet12+F) ion versions. Both versions include the additional features denoted by "+F". All T Net models were trained using focal loss and the Adam optimisation algorithm.

4.4.9 CNN-GNN Hybrid Model

The proposed encoding module uses a graph neural network (GNN) to capture the long-range interactions of tandem peptide spectrum graphs (Figure 4.1). In a GNN with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the embedding for each node $u \in \mathcal{V}$, is defined as $h_u^{(k)}$, where k is the number of message passing layers i.e. update steps. The embedding $h_u^{(k)}$ is updated by aggregating the embedding of u's neighbours $\mathcal{N}(u)$. The proposed GNN uses a mean aggregation of the neighbour embeddings for each node, given by the following equation;

$$h_u^{(k)} = ReLU\left(W_{self}^{(k)} h_u^{(k-1)} + W_{neigh}^{(k)} \frac{\sum_{v \in \mathcal{N}(u)} h_v^{(k-1)}}{|\mathcal{N}(u)|} + b^{(k)}\right) \tag{4.8}$$

Where W is a weight matrix and b is bias vector. The architecture was inspired by the encoder module of the Graph2Seq model [263].

The CNNs described above are used to create the node embeddings. As before, at each peak, spectrum windows which encompass the possible neighbouring amino acids are passed to the three CNNs. Again, the output of the three CNNs is concatenated before passing through a 512 dense layer. This provides the initial node embeddings $h_u^{(0)}$. A graph is then created with a node for each peak and edges between nodes where equations (4.4) and (4.5) are satisfied. An aggregate path length of 4 is used as a compromise between complexity and performance. At each node the embeddings of its neighbour and itself are combined and then updated using the above formula (equation (4.8)), specifying the new embeddings for each node. The process is repeated 4 times resulting in a 512 vector encoding for each node. These GNN encodings are then passed to a single node output layer to provide the peak score. The CNN node embedding and aggregation steps of the GNN are all trained together. For simplicity, the CNN-GNN hybrid will be referred to as just GNN for the rest of the paper. The model was created both withRealDataAUPR (GNN+F) and without (GNN) the additional Novor features. Like the other neural network models, all GNN models were trained using focal loss and the Adam optimisation algorithm.

4.4.10 Model Evaluation

Both the area under the precision-recall curves (AUPR) and the area under the receiver operating characteristic curve (AUC) were considered as metrics in this research. AUPR summarises the precision-recall curve into a single number. Both recall (equation (4.9)) and precision (equation (4.11)) are independent of the number of true negatives (TN). This makes AUPR a more informative metric when the negative class (noise) vastly outnumber the positive class (peptide ions) [59]. AUPR is difficult to calculate however, as a linear interpolation between points in PR space leads to an overestimation of performance [59]. Therefore we use an approximation of the AUPR, namely the average precision over all score thresholds [266]. Average precision was calculated using the the metrics module for Scikit-learn [192].

AUC was also used in the evaluation. The receiver operating characteristic (ROC) is the plot of the true positive rate (TPR; equation (4.9)) against the false positive rate (FPR; equation (4.10)). AUC, the area under this curve, is a popular metric for binary classification tasks such as those described in this research as it also captures the perfor-

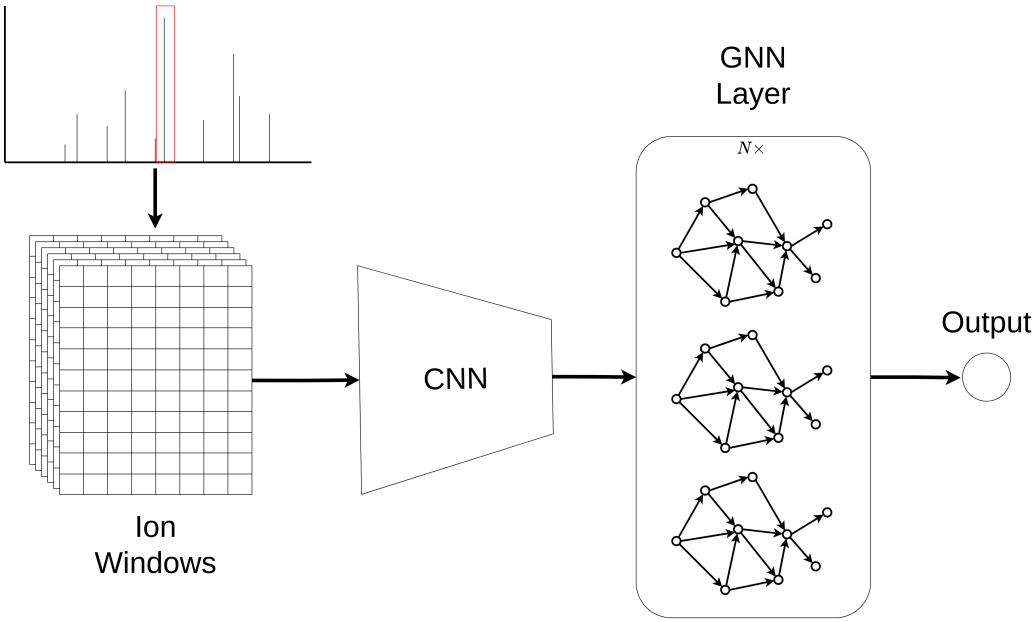

Figure 4.1: Diagram of the CNN-GNN Hybrid Model

mance of a model in a single statistic [158]. AUC can be interpreted as the probability that a model will score a randomly selected positive example higher than a randomly selected negative example.

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{4.9}$$

$$FPR = \frac{FP}{N} = \frac{FP}{TN + FP} \tag{4.10}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.11}$$

4.4.11 Hardware Specifications

All models were trained on a 16 GB linux machine with an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz and Nvidia GeForce GTX 1650 4 GB GPU. Random forest models were created using the Scikit-learn 0.24.1 module while deep learning models were created using Tensorflow 1.13.1.

4.4.12 Code Availability

The artificial data and deep learning models used in this research are available at `https://github.com/KevinMcDonnell6/MSencoding`.

4.5 Results and Discussion

4.5.1 Performance on Benchmark Datasets

All models were evaluated on 9 real tandem MS datasets from 9 different organisms. Table 4.3 shows the average precision of the models on all 9 datasets. The GNNs with and without the additional features were the top two performing models in all datasets. The GNN+F was found to perform best in 8 of the 9 datasets with the standard GNN performing slightly better on the Tomato data. The RF+F was the worst performing model in all but one of the 9 datasets with the CNN marginally worse on the Human dataset.

The results demonstrate the strength of the graph approach in assisting peptide peak identification. The graph can encapsulate more information as long range interactions are passed through and encoded by the model. The GNN maintained a significantly higher average precision than the other models despite the variation between the datasets of organism type, FPP and NR (Table 4.1). This indicates a robustness in the GNN architecture regardless of the data characteristics it is faced with. However, as of yet they are unused in the context of *de novo* peptide identification. The advantage of GNNs is that they can capture the inherent graph-like nature of peptide fragmentation patterns. Consequently, these results suggest their utility in *de novo* peptide sequencing should be explored further.

The results also show the utility of Novor's additional features. These expert features devised by Novor include intensity rank, intensity half rank, local intensity rank and intensity half rank. The CNN+F performed substantially better than the standard CNN for all datasets. The superiority of models that utilise these features highlight that there is still a place for expert knowledge in the field of *de novo* peptide sequencing. Machine learning is often seen as being able to provide a solution to all problems. However, care must be taken when designing architectures and/or features that best fit the problem at hand. In this context area experts can still play an integral part in machine learning algorithm design. Also, while these features are useful in distinguishing peptide peaks from noise, it is unclear whether or not they can assist in amino acid prediction as well. The features were designed and utilised by Novor to identify fragmentation sites. Future algorithms similar to DeepNovo may benefit from expert features designed for their step-

Dataset	RF+F	CNN	CNN+F	Tnet8+F	Tnet12+F	GNN	GNN+F
Yeast	0.7170	0.7368	0.7873	0.8110	0.8092	0.8612	**0.8853**
Human	0.7235	0.7234	0.7926	0.8048	0.8081	0.8472	**0.8679**
Mouse	0.6848	0.7287	0.7648	0.7942	0.8041	0.8466	**0.8641**
Bacillus	0.6572	0.6878	0.7558	0.7619	0.7759	0.8333	**0.8567**
ClamBacteria	0.7049	0.7146	0.7864	0.7872	0.8070	0.8290	**0.8548**
Honeybee	0.6375	0.6563	0.7248	0.7418	0.7660	0.8004	**0.8299**
Ricebean	0.6636	0.6754	0.7435	0.7408	0.7534	0.8249	**0.8516**
Tomato	0.7403	0.7666	0.8246	0.8365	0.8428	**0.9017**	0.9002
M. mazei	0.6613	0.6901	0.7529	0.7693	0.7813	0.8490	**0.8586**

Table 4.3: Average precision values for each model on all 9 real datasets

by-step amino acid prediction approach.

4.5.2 The Effect of Missing Peaks

To further investigate the above results, the performance of the models was compared for spectra with varying amounts of noise and peptide peaks present. In our previous work these were found to be the main challenges when identifying peptides *de novo* [161].

To do so, all 9 real datasets were merged and each spectrum was assigned a Noise Ratio (NR) and Fraction of theoretical Peaks Present (FPP) based on the amount of ion peaks identified (see section Peak Classification). The FPP values spanned the range from 0 to 0.8 with spectra grouped into bins encompassing each 0.1 span. The NR values ranged from 0 to 30 with spectra grouped into the designated bin for each 1.0 increase. The final bin encompasses the NR range from 14 to 30 as very few spectra matched this range.

Figure 4.2A shows how the performance of the models changes with respect to the fraction of possible peptide peaks present. Spectra were grouped into bins corresponding to fraction of peaks present. Average precision was then calculated for each bin as shown. When many peptide peaks are missing, the performance of the models is worst. As the fraction of peptide peaks present increases, so does the average precision.

Consistent with the overall results from the 9 datasets, the GNNs were the best performing models across all of the data. Even when many peaks were missing (FPP$<$0.1) the graph approach performed best. While the complete chain of ions may not be available, partial chains help inform ion identification. This aligns with our previous findings where despite many fragmentation sites not being represented in the data, Novor was able to make use of the partial sequences of sites that were [161]. *De novo* algorithms should be designed so that they can utilise partial sequences and not rely on complete ion chains to be present as these data account for only a fraction of the total.

The advantage of the additional expert features is again evident with those models utilising them performing better than their counterparts for almost all of the data. The

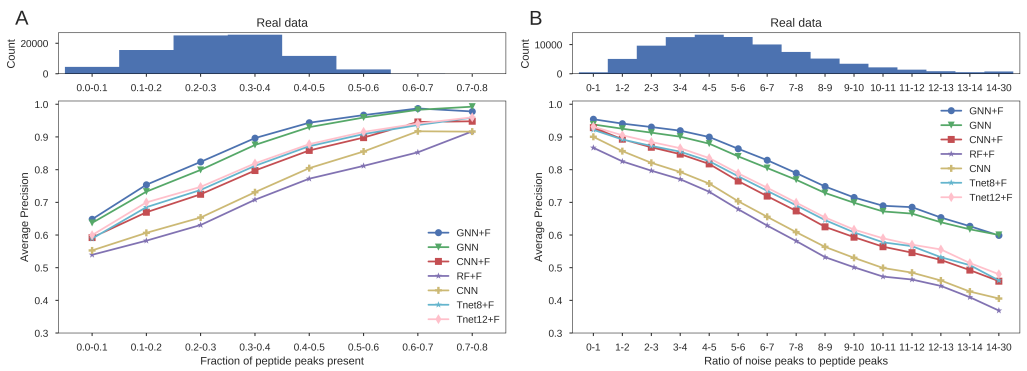

Figure 4.2: Performance of models with respect to the fraction of peptide peaks present and noise ratio. Average precision is shown for spectra matching the different grading of both features.

CNN+F performs better than the CNN for all data across the range of peptide peak prevalence. The GNN without these features surpasses the GNN+F only when very few peaks are missing. While there is very little data in this range the performance of these models converges as the fraction of peaks present increases. The advantage of the features becomes minimal when the graph is almost complete and information can pass between peaks instead of relying on the independent features. Conversely, the CNN+F maintains its advantage over the CNN in the same range. These models are not able to encode the long range interactions and hence the additional features have a greater impact.

4.5.3 The Effect of Noise

The models were also evaluated with respect to the ratio of noise to peptide peaks in the spectra (Figure 4.2B). Noise is defined as any peak that could not be attributed to a b ion or y ion in the database assigned peptide, either singly or doubly charged, or singly charged with a neutral loss of water or ammonia. Spectra were again binned, this time corresponding to their noise ratio. Average precision was calculated for each bin as shown.

The GNN+F was found to be the best performing model for almost all noise ratios with the GNN without additional features the next best. The increase in precision of the GNNs over the other models was greatest when noise ratios were high.

When many noise peaks are present, some of them may appear by chance at mass values equal to an amino acid away from a peptide peak. It is very difficult for the CNN, T Net and RF to distinguish these from actual peptide peaks. The GNNs have the advantage of being able to encapsulate long-range relationships between peaks in the

spectrum and so are better able to distinguish the real peptide peaks from noise. Real peptide peaks are likely to appear as part of a chain of peaks, which is extremely rare for noise.

Dataset	RF+F	CNN	CNN+F	Tnet8+F	Tnet12+F	GNN	GNN+F
FPP0.36 NR0	0.9865	0.9972	0.9972	0.9948	0.9974	**0.9984**	0.9981
FPP0.36 NR1	0.9805	0.9943	0.9947	0.9964	0.9966	0.9964	**0.9977**
FPP0.36 NR5	0.9341	0.9759	0.9727	0.9875	0.9791	0.9866	**0.9915**
FPP0.36 NR10	0.8749	0.9403	0.9268	0.9648	0.8091	0.9702	**0.9791**
FPP0.36 NR15	0.8191	0.8937	0.8774	0.9223	0.6163	0.9491	**0.9611**

Table 4.4: Average precision values for all artificial datasets. FPP stands for Fraction of peptide Peaks Present and NR stands for Noise Ratio

The effect of noise was also investigated using artificial data (Table 4.4). Artificial data provide a way to evaluate the models on the same data while changing only the number of non-peptide peaks. The yeast proteome downloaded from Uniprot [12] was used to create a list of peptides. A dataset of the corresponding spectra was created using the Prosit pipeline [89]. This dataset was duplicated and then the noise ratio was artificially set to the specified levels for each duplicate. The noise ratio of each spectra was artificially assigned thereby controlling for the correlations between variables observed in real data (Appendix B Figure 1). The models prepared for the real yeast data, which were therefore not trained on any yeast spectra or peptides, were used in the evaluation.

Table 4.4 shows the average precision values for the yeast models on the 5 artificial datasets. Like the real data, the GNNs were the top two performing models for each dataset. The GNN+F was the best model for all datasets except when there was no additional noise and the GNN performed marginally better.

4.5.4 CNN-GNN Hyperparameter Comparison

The GNN+F model was also trained and tested with different hyperparameters, such as the number of message passing layers, the direction of neighbouring nodes to aggregate and the aggregation function.

Table 4.5 shows how average precision increases with increasing numbers of message passing layers but with decreasing magnitude. The GNN model is designed such that setting the model to have 0 message passing layers is equivalent to the CNN model. As shown in Table 4.5 increasing the the number of layers from 0 to 2 gives an initially large increase in average precision of 11%. Further increases from 2 to 4 and 4 to 6 give more modest improvements of 1.2% and 0.88% respectively.

Table 4.5 also shows how mean aggregation was found to give better average precision than sum aggregation when there are 4 message passing layers. Mean aggregation is more stable to differing node degrees as it normalises the inputs maintaining the same scaling.

Models also tended to converge to their optimum quicker using mean aggregation (data not shown).

Models with 4 layers were also also trained using only the forward or backward connections in the graph. In this context, forward connections refer to nodes from the apparent succeeding cleavage (equations (4.4) and (4.5)), with backward connections from the preceding cleavage. The results show that both 4 layer unidirectional models exhibited similar average precision to each other but lower than the 4-layer model using both directions. The average precision of the 4-layer single direction models is very similar to that of the model utilising both directions over only 2 layers. For any given peak, each of these 3 models are using neighbour encodings that span a distance of 4 hops. Although each model would ultimately be using different neighbours the results remain consistent.

#Layers	Aggregation Fn	Direction	Average Precision
0	Mean	Fw & Bw	0.7873
2	Mean	Fw & Bw	0.8748
4	Mean	Fw & Bw	0.8853
6	Mean	Fw & Bw	0.8931
4	Sum	Fw & Bw	0.8759
4	Mean	Fw	0.8766
4	Mean	Bw	0.8740

Table 4.5: Average precision values for different GNN+F models on the yeast dataset. The number of aggregation layers is denoted by #Layers, the aggregation function is specified under Aggregation Fn and the directions information could flow is highlighted under Direction.

4.5.5 Problems with AUC

During the evaluation of the models, AUC was also used as a metric to compare their performance across the different datasets. However, this analysis resulted in some unusual findings.

Figure 4.3 shows how the AUC changes for each model on the real data when binned by the fraction of peptide peaks present and the noise ratio. There is an initial increase in AUC as the noise ratio increases from 0 to 5 for all models (Figure 4.3B). This is in contrast to previous findings which showed increasing noise levels resulted in worse performance [161, 177].

Upon further investigation it was found that this portion of the data had a much lower than average fraction of peptide peaks present, which may contribute to the lower than expected performance overall (Appendix B Figure 1A). As shown in the Figure 4.2A, lower levels of peptide peaks present correlate to lower overall performance. However, if this was the case it did not affect the average precision (Figure 4.2B).

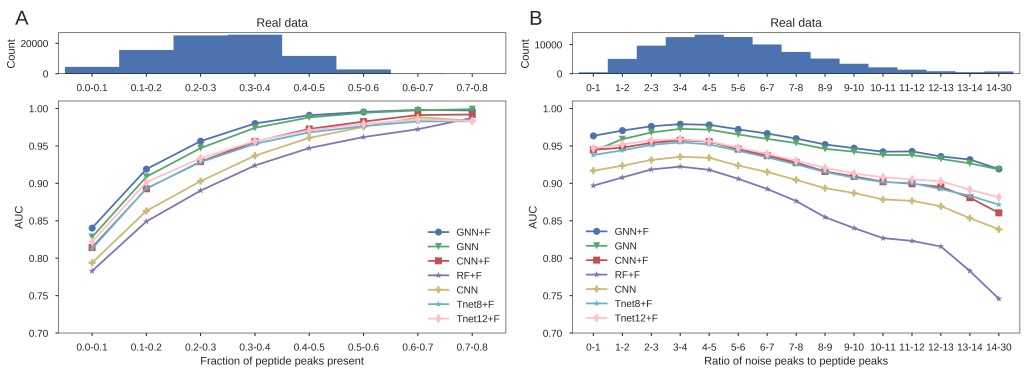

Figure 4.3: Performance of models with respect to the fraction of peptide peaks present and noise ratio. AUC is shown for spectra matching the different grading of both features.

To account for these correlations the AUC was calculated for each model on the artificial data where only the noise level was changed (Table 4.6). Again an initial increase in the AUC was observed for each model as noise was added to the data.

Dataset	RF+F	CNN	CNN+F	Tnet8+F	Tnet12+F	GNN	GNN+F
FPP0.36 NR0	0.9609	0.9918	0.9916	0.9841	0.9920	**0.9954**	0.9946
FPP0.36 NR1	0.9861	0.9960	0.9961	0.9973	0.9974	0.9969	**0.9984**
FPP0.36 NR5	0.9813	0.9941	0.9924	0.9964	0.9951	0.9948	**0.9976**
FPP0.36 NR10	0.9736	0.9896	0.9839	0.9917	0.9863	0.9918	**0.9957**
FPP0.36 NR15	0.9670	0.9836	0.9758	0.9875	0.9762	0.9883	**0.9932**

Table 4.6: AUC values for all artificial datasets. FPP stands for Fraction of peptide Peaks Present and NR stands for Noise Ratio

Investigation into the definition of AUC provided some insights as to why this is the case. The additional noise introduces many easy to classify negative examples. This increases the size of the negative class (N) while having little effect on the number of false positives (FP) (see equation (4.10)). This lowers the false positive rate thereby inflating the AUC. Conversely, neither the precision or recall are proportional to the size of the negative class (equations (4.9) and (4.11)). This is why average precision does not show similar trends. Further discussion can be found in Appendix B.

4.5.6 Time Evaluation

Increased performance may come at a cost as models become more complex. The training times of all the models were compared with respect to the training time per spectra. This did not include time taken to process the data.

Figure 4.4 shows the results of the time evaluation. RF was the most efficient algorithm taking 14 μs per spectrum. Both CNNs showed similar results to each other. The

addition of the added features increased training time by 1 μs per spectrum from 38 μs per spectrum to 39 μs per spectrum. Both took over twice as long as the RF. The GNNs took longer to train both due to their added complexity and the difficulty in parallelising their computation. Again the addition of the extra features resulted in a 1 μs per spectrum increase in training time. Both GNNs took over 3 times longer per spectrum to train than the RF. The T Net models took substantially longer to train than any of the other models with Tnet8+F and Tnet12+F taking 236 μs and 354 μs per spectrum respectively. This is due to the different way the models interpret the spectra. The other deep learning models take in candidate ion windows which are processed outside the model. Instead the T Net models use the difference from the expected ion values with some of the processing taking place inside the model itself. While this means the T Net models are slower, it shortens their data processing time making the combined time comparable to the CNNs and GNNs. The T Net data processing took approximately 2.8 μs per spectrum whereas the CNN and GNN data processing took approximately 97 μs per spectrum.

The added features caused marginal increases in training times. However, as shown in the performance evaluations earlier, they can cause large increases in prediction accuracy (Table 4.3). This was particularly evident for the CNN which showed a substantial improvement with the addition of the extra features. Due to their increased complexity, the GNN and GNN+F took longer to train. However, there was a substantial difference in performance (Table 4.3). For most accurate identification, the GNN is shown to be best suited. Nonetheless, for real-time peptide identification, the RF may be preferable due to its increased speed. When choosing the appropriate encoding module for their *de novo* algorithm, researchers must decide on the trade-off between accuracy and training time.

4.6 Conclusion

We propose a new CNN-GNN hybrid module for peptide ion encoding. Our model was found to be more effective at identifying peptide peaks in MS spectra than the encoding modules used by the state-of-the-art algorithms, Novor, DeepNovo and PointNovo. The CNN-GNN was better able to distinguish peptide peaks than the other modules over all levels of noise and peptide peaks present in the data. The ability of our GNN based model to incorporate long range ion relationships yielded significantly increased performance over the other models in all datasets.

Our results suggest that there is potential for exploring the use of GNNs in *de novo* peptide sequencing algorithms. However, it is still unknown if this will improve peptide identification rates of current, state-of-the-art algorithms. To test this the encoding module described here would need to be integrated within an architecture capable of sequence

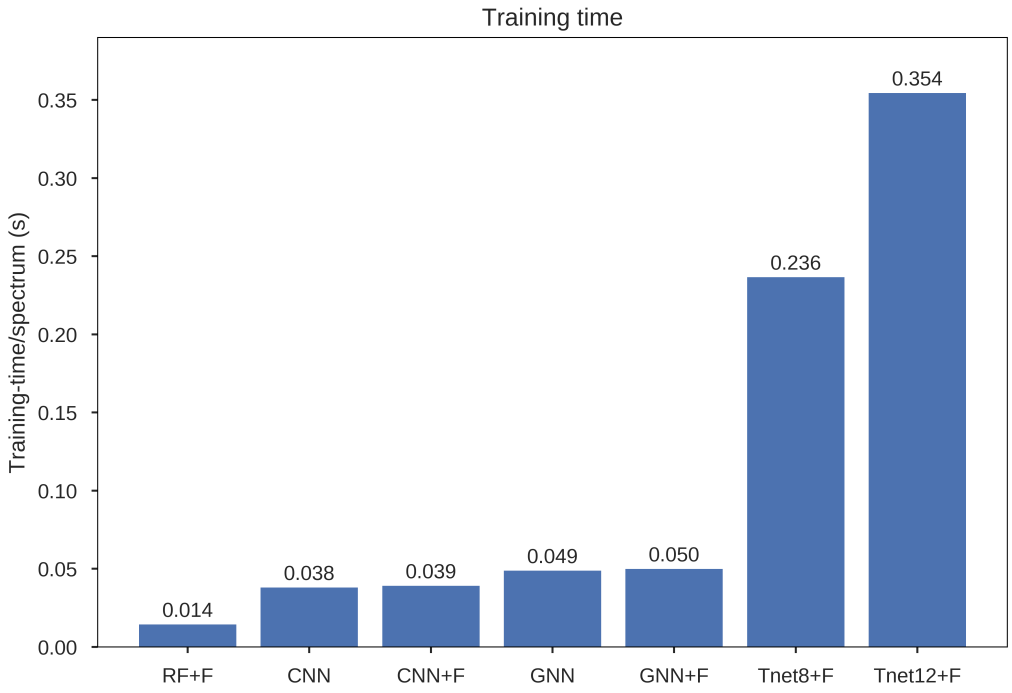

Figure 4.4: Comparison of the training times of the seven models.

prediction. One option would be combining it with an LSTM, although LSTMs require sequential vector inputs while GNNs produce an encoding for each node which do not have a natural order. Therefore, another module, such as an attention mechanism, would be required to allow the model to condense the collection of node encodings into the required sequence. Again, there are multiple ways in which this could be done, and their investigation requires further research.

The GNN module proposed in this work shows a considerable improvement in average precision (>20%) over the RF used by Novor for all datasets. These results would suggest that Novor's methodology may become more competitive if the random forest scoring module was replaced with a more effective machine learning algorithm such as a CNN-GNN hybrid.

Finally, this research also showed the utility of Novor's additional expert features for peptide ion identification when peaks are isolated and there are few if any connections between them. Hence our work highlights the importance of expert domain knowledge in the design of *de novo* sequencing models. Overall this research uncovers limitations in peptide ion encoding from state-of-the-art *de novo* algorithms and the presented CNN-GNN hybrid model offers a promising alternative by embedding spectral features more comprehensively.

Chapter 5

Critical Evaluation of the Use of Artificial Data for Machine Learning Based *De Novo* Peptide Identification

The work outlined in this chapter was published in:

McDonnell, K., Howley, E., and Abram, F. Critical evaluation of the use of artificial data for machine learning based *de novo* peptide identification. Computational and Structural Biotechnology Journal (2023)

5.1 Abstract

Peptide identification in proteomics typically relies on matching high resolution tandem mass spectra to a protein database but can also be performed *de novo*. While artificial spectra have been successfully incorporated into database search pipelines to increase peptide identification rates, little work has been done to investigate the utility of artificial spectra in the context of *de novo* peptide identification. Here, we perform a critical analysis of the use of artificial data for the training and evaluation of *de novo* peptide identification algorithms. First, we classify the different fragment ion types present in real spectra and then estimate the number of spurious matches using random peptides. We then categorise the different types of noise present in real spectra. Finally, we transfer this knowledge to artificial data and test the performance of a state-of-the-art *de novo* peptide identification algorithm trained using artificial spectra with and without relevant noise addition. Noise supplementation increased artificial training data performance from 30% to 77% of real training data peptide recall. While real data performance was not fully replicated, this work provides the first steps towards an artificial spectrum framework

for the training and evaluation of *de novo* peptide identification algorithms. Further enhanced artificial spectra may allow for more in depth analysis of *de novo* algorithms as well as alleviating the reliance on database searches for training data.

5.2 Introduction and Related Work

Proteomics can provide valuable insight into the functional profile of a biological system through the identification of the proteins present at the time of sampling [26]. This process is typically performed using a bottom-up strategy, whereby proteins are first digested down in smaller sub-sequences of amino acids called peptides [272]. Peptides are then detected using tandem mass spectrometry (MS) before being mapped back to the corresponding protein. Peptide identification using tandem MS can be accomplished through two main algorithmic approaches; database searching or *de novo* identification [181]. Database searching has been the dominant method for the last few decades, with a higher peptide identification rates than its alternative. In this approach, theoretical spectra are first created for each peptide in the protein database. Each peptide is then given a score based on the similarity between its theoretical spectrum and the observed spectrum. Finally, significant peptide spectrum matches (PSMs) are used to infer proteins identification. Database searching is not without its limitations however. In this approach only 25% of spectra receive significant matches [95]. This is partly because larger database sizes lead to an increased probability of random matches [150]. To limit the number of these false positives, the score threshold for an acceptable PSM must be increased, meaning many correct matches are lost. In this context, *de novo* peptide identification offers a promising database free alternative.

De novo peptide identification relies on the spectrum alone to determine the originating peptide sequence [152]. Models are designed to recognise the patterns associated with peptide fragmentation, such as mass differences between ions, to identify amino acids. With an abundance of data now available on repositories such as PRIDE [124], training complex machine learning models for *de novo* peptide identification has become the norm. Indeed, machine learning models are now an integral part of all current state-of-the-art *de novo* peptide identification algorithms [241, 199, 153]. The aim of these models is to learn the fragmentation patterns from the observed spectra that are indicative of the labelled peptide. They are generally trained and tested on real tandem MS spectra, labelled using a database search. However, database searches can be prone to errors meaning the quality of the training data may be suboptimal. Even though target-decoy methods are used to estimate the false discovery rate of database searches, this strategy can still underestimate the number of incorrect matches [121]. This means that *de novo* methods lack ground truth data with which they can be evaluated.

Peptide identification is not straightforward due to the complexity of the peptide fragmentation process. Different fragmentation patterns are observed depending on a multitude of factors such as the amino acid composition of the peptide, the peptide length, the peptide charge and the method of excitation [189]. Between each two amino acids along the peptide chain, there are three different bonds where cleavage can occur. Cleavages at these bonds vary in frequency depending on how energetically favourable they are [189]. For common collision based fragmentation methods such as higher energy collisional dissociation (HCD), cleavage at the amide bond is most common, resulting in b and y ions [63]. Depending on their charge state, fragment ions will be observed at different mass-to-charge ratios (m/z). They can also lose neutral molecules of ammonia or water causing a further shift in their observed m/z. The b ions are conventionally numbered from N-terminus to C-terminus, with y ions numbered from C-terminus to N-terminus.

The score a PSM receives through a database search is dependent on the number of fragment ions matched [72]. The PSMs that receive the highest scores and are used will therefore be biased toward those with greater numbers of matched ions. This means the training data for *de novo* models, which are labelled using a database search, will tend to have fewer fragmentation cleavages missing. In our previous research missing fragmentation cleavages were found to pose a significant challenge to *de novo* algorithms [161]. While spectra with many missing cleavages will inevitably be more difficult to characterise, a lack of these spectra in the training set will exacerbate the problem.

Artificial data may provide a solution to this issue. In many areas of machine learning artificial data are used to train models where data are scarce, low in diversity or biased [226, 43, 261]. This allows for full control over the creation of the data, and therefore over what a model learns from. Artificial data are created such that they match the statistical properties of real data and in doing so, capture the patterns present in the real data [3]. Shmelkov *et al.* developed a method to evaluate the utility of artificially generated data [221]. They trained a classification model on synthetic data created using a generative adversarial network (GAN) and compared the performance to a model trained on real data. Artificial training data that could best replicate the performance of the real data and could therefore replicate the inherent relationships were deemed to be the most useful. Similar methods have been employed by other groups to measure artificial data quality [261, 264, 34].

Artificial data can also be used for evaluation purposes [254, 20]. For example, artificial test data can be designed to contain scenarios which are unlikely to appear in a real dataset. This is extremely useful to test a system against important but rare events [117]. Furthermore, the flexibility of artificial data means the performance of models can be tested across a wide range of scenarios.

In the context of peptide identification, there are many models available to create artificial peptide spectra [66, 13, 60]. Prosit is a state-of-the-art, open-source, spectrum prediction model [89]. It uses machine learning to capture the relationship between fragment ion intensity and peptide sequence, producing high quality artificial spectra. The model is capable of predicting the b and y ion fragment peaks of peptides up to length 30. While the theoretical spectra traditionally used in a database search only contain the m/z location and an arbitrary intensity, Prosit's spectra are highly correlated to the observed fragmentation pattern. The authors successfully used this additional information to increase the number of identified peptides by up to 35% compared to the m/z alone [89].

To advance the field of *de novo* peptide identification, a greater understanding of both the strengths and limitations of current algorithms is required. Artificial test data would facilitate this by providing spectra with specific characteristics allowing researchers to understand how their algorithms perform for varying levels of complexity and noise. Artificial data have previously been used to evaluate peptide identification models but only to a limited degree and are generally used as a secondary analysis with unvalidated additions of noise [25, 177, 161]. While artificial spectra have proven useful for database identification, their relevance to *de novo* identification has never been addressed. Furthermore, using artificial spectra as training data for *de novo* models may help circumvent the current bias associated with database labelling. To address the knowledge gaps, we evaluate the utility of artificial data in the context of *de novo* peptide identification. We first analyse real data and categorise the different forms of noise which can be present. We also estimate the rate of spurious ion matches in the mass spectra using random non-matching amino acid sequences. Then, through the addition of noise, we modify artificial spectra to increase its similarity to real spectra. Finally, we assess the utility of the modified artificial spectra by using them to train the state-of-the-art *de novo* peptide sequencing model PointNovo [199], and compare the difference in performance to the model trained on real spectra.

5.3 Methods

5.3.1 Real Spectra

The real spectra used in this research come from 9 different organisms and 9 different research groups (Table 5.1) [188, 182, 41, 205, 194, 159, 215, 115, 54]. All experiments were conducted with a Thermo Scientific Q-Exactive mass spectrometer by the respective research groups. The raw data were combined and processed by Tran *et al.* [241] using the PEAKS DB software [271] with their respective proteome database. The data were

then filtered using a 1% false discovery rate threshold. More information on the data processing and experiments can be found in the original papers [188, 182, 41, 205, 194, 159, 215, 115, 54, 241]. As processing of the MS/MS spectra was carried out prior to this research, the effect of this on the subsequent analysis is not considered.

The spectra were also partitioned into different datasets by Tran *et al.* [241] to create 9 separate training, validation and testing sets, one for each organism. Each test set consists of spectra from the respective organism while the training and validation sets for each organism are composed of spectra from the other 8. The resulting MGF files were downloaded from `ftp://massive.ucsd.edu/MSV000081382/`.

In this research, the characteristics of real data were derived from a subsample of 50,000 spectra from the nine organisms, while model training was conducted on the entire yeast partition. These choices were made to limit computational resources while still addressing our research aims. The yeast partition was selected as *Saccharomyces cerevisiae* is a model organism and well characterised.

To allow for a meaningful comparison, the real data were then filtered to exclude spectra which could not be replicated by Prosit. The Prosit pipeline considers carbamidomethylation of cysteine as a fixed modification and oxidation of methionine as the only variable modification. It also cannot predict spectra for peptides with more than 30 amino acids. Hence, PSMs in the datasets longer than this or containing other modifications were removed prior to analysis.

5.3.2 Artificial Spectra

Artificial spectra in this experiment were created using the Prosit pipeline [89]. The source code was downloaded from https://github.com/kusterlab/prosit. A pretrained model was downloaded from https://figshare.com/projects/Prosit/35582. The precursor charges and peptide sequences of the spectra were extracted from each dataset. These were then used by Prosit to create an artificial copy of the real datasets used in this study (Table 5.1).

5.3.3 Peak Matching

Theoretical fragment ions were created for each of the database assigned peptides using the Pyteomics Python module [141]. These were then compared to the observed peaks in the spectra collated and processed by Tran *et al.* [241]. Observed peaks that fell within 0.05 Da of a theoretical peak were considered matched (maximum from Table 5.1). If multiple observed peaks satisfied this condition, the peak with the smallest mass difference from the theoretical ion was considered to be the correct match.

Dataset	Organism	Accession	FragTol (Da)	Reference
Yeast	*Saccharomyces cerevisiae*	PXD003868	0.05	Seidel *et al.* [215]
Human	*Homo sapiens*	PXD004424	0.02	Cypryk *et al.* [54]
Mouse	*Mus musculus*	PXD004948	0.05	Nevo *et al.* [182]
Bacillus	*Bacillus subtilis*	PXD004565	0.05	Reuß *et al.* [205]
ClamBacteria	*Candidatus Thiodiazotropha endoloripes*	PXD004536	0.05	Peterson *et al.* [194]
Honeybee	*Apis mellifera*	PXD004467	0.05	Hu *et al.* [115]
Ricebean	*Vigna mungo*	PXD005025	0.05	Paiva *et al.* [188]
Tomato	*Solanum lycopersicum*	PXD004947	0.05	Mata *et al.* [159]
M. mazei	*Methanosarcina mazei*	PXD004325	0.05	Cassidy *et al.* [41]

Table 5.1: Details of nine real datasets used. Accession indicates the PRIDE accession number. FragTol indicates the error tolerance for fragment ions used by Tran *et al.* in the database search [241].

Fragment ions can occur from single bond cleavages along the backbone of the peptide. In this research, three such backbone ion types were considered (a, b and y) and are referred to as backbone ions for the remainder of this manuscript. These ions can also lose neutral molecules of NH_3 and H_2O which we refer to as neutral losses. Therefore we consider twelve ion types in total, namely a ions, b ions, y ions, a-H_2O ions, b-H_2O ions, y-H_2O ions, a-NH_3 ions, b-NH_3 ions, y-NH_3 ions, a(2+) ions, b(2+) ions and y(2+) ions. Loss of H_2O was only considered for C-terminus ions as well as others containing aspartic acid, glutamic acid, serine or threonine [189, 228]. Loss of NH_3 was only considered for fragments containing the amino acids arginine, lysine, glutamine or asparagine [189, 228].

Internal fragments were also created for each peptide. These are defined as the amino acid sequence arising from two backbone cleavages and can occur in two possible types; a and b [120]. To this end, all k-mers from the second amino acid to the second last were identified and the sum of their masses calculated. The mass of a hydrogen atom was added to give the mass of b-type internal fragments [166]. These masses were then duplicated with the combined mass of a carbon and oxygen removed to create the set of a-type internal fragments.

5.3.4 Random Peptides

The number of peaks matched in the spectra that may have occurred by chance was estimated by creating a random peptide for each spectrum. Peptides of the same length were generated by randomly sampling amino acids with probabilities proportional to their prevalence in the set of assigned peptides used in this study (data collated by Tran *et al.* [241]). To generate tryptic peptides the final amino acid in each sequence was set to arginine or lysine, alternative to the last amino acid of the database assigned peptide. This method was used to ensure as little overlap in fragment ion masses as possible while generating random tryptic peptides with the same distribution of amino acids as those observed in real data. Alternative methods explored can be found in Appendix C with results in Appendix C Table 1. Theoretical backbone ions and internal fragments were then created for each random peptide, as was done with the real peptides (see section Peak Matching). These were then matched to the spectra with a tolerance of 0.05 Da.

5.3.5 Data Modification

During this research both real and artificial spectra were modified. Four different spectrum features that can contain noise or variability in their observation were identified; m/z, intensity, presence/absence of fragment ions and unknown peaks. The effect of the addition and removal of each of these noise types was analysed.

In real data, removal of m/z jitter was performed by resetting each peak to its expected value. To do this the m/z value for each theoretical ion of the assigned peptide was calculated using the Pyteomics package [141]. The m/z of the closest matched peak was then assigned the theoretical m/z value (see section Peak Matching).

The m/z jitter was reintroduced using two different methods, each approximating the real noise distribution. Method one consisted of a mixture distribution of two normal distributions, both with means of 0 and standard deviations of 1e-2 and 1e-3 respectively in a 1:1 ratio. Method two consisted of a mixture distribution of a Laplace distribution with a mean of 0 and a scale parameter of 2.5e-3 as well as a uniform distribution between -0.05 and 0.05 with a 12:1 ratio. Jitter in the m/z values was introduced to peaks by taking random samples from the respective distributions and adding them to the expected theoretical m/z.

When real intensity was modified it was replaced by the Prosit predicted value. However, Prosit did not predict an intensity for all peaks matched in the real data. If a fragment ion was considered unlikely enough Prosit would not create a corresponding peak. If this occurred and the peak was matched in the real spectrum the intensity was left unchanged.

With peaks ordered by intensity, we found there was a linear relationship between

the number of peaks to be removed from the Prosit spectra to match the number of missing fragmentation cleavages in real spectra and the peptide length. Using a linear regression model we could approximate the mean number to remove by length using the formula $n = max(l - 5, 0)$, where n is the number of peaks removed and l is the length of the peptide. Peaks were removed with the lowest intensity first as these were the least abundant ions and therefore most likely to be missing.

Unknown peaks were introduced using three different methods. These methods introduced peaks as singly charged ions as these account for most of the observed peaks. The first method involved the creation of random combinations of amino acids and calculating the sum of their masses. These were introduced as singly charged peaks with their m/z value equal to their mass. The second method involved the creation of internal fragments from the peptide assigned to the spectrum. For each peptide, all possible internal fragments of type a and b were created (see section Peak Matching). This created a population of m/z values equal to their masses as these were also only considered as singly charged peaks. The third method introduced peaks as a combination of the above methods. For each method, the number of peaks introduced to each spectrum was set equal to the number observed in the equivalent real spectrum for a fair comparison. Unknown peaks were then introduced by randomly sampling from the respective set of created peaks. In the case of the combined method, the number of internal fragments was defined to match their observed occurrence while random combinations of amino acids made up the remaining amount. Intensity values for artificial non-peptide peaks were sampled from a log-normal distribution estimated from unknown peaks in real data in our previous research [160]. The natural log of this distribution has a mean of -4.4 and a standard deviation of 1.5.

5.3.6 PointNovo

PointNovo is the current state-of-the-art in *de novo* peptide identification [199]. It was used to evaluate the utility of modified and unmodified spectra for training *de novo* models. PointNovo is the updated version of DeepNovo [241] and was previously released as DeepNovoV2 [200]. The source code was downloaded from https://github.com/volpato30/DeepNovoV2. Models were trained on the Yeast partition of the labelled data collected by Tran *et al.*. In this partition the test data come from *Saccharomyces cerevisiae* while the training and validation data come from the other 8 organisms. The models were trained with validation testing occurring every 300 steps as per the original code and the parameters were saved for the lowest validation loss.

5.3.7 Metrics

The metrics used to evaluate the trained models are those used by PointNovo [199]. Firstly, the predicted amino acid and the actual amino are required to have a mass difference of less than 0.1 Da. If this condition is met and the difference between the combined mass of the previously predicted amino acids and the combined mass of the previous actual amino acids is less than 0.5 Da, then an amino acid is considered matched. Amino acid precision is then defined as the total number of matched amino acids over the total number predicted. Amino acid recall is the total number of matched amino acids over the total number of actual amino acids. Similarly, peptide recall is the total number of correct peptides over the total number of spectra.

5.4 Results

A comparison between the performance of a model trained using real and artificial data can be used as an indicator of the quality of the latter [221]. Table 5.2 shows the performance of PointNovo given three different training and test set combinations of real and artificial spectra. The artificial data are duplicates of the real data, generated using Prosit [89]. The performance of the model when trained and tested on real spectra is the baseline for this research. It provides a reference with which to compare to the model performance when trained using artificial spectra. If artificial training data can reproduce the test performance of real training data it can be considered an adequate replacement [264]. It should be noted that the performance reported here using real training data differs slightly from that reported in the original paper [199] as the real data used in this experiment has been filtered to match the capabilities of Prosit (see Section Real Spectra)

Train Data	Test Data	AA Recall	AA Precision	Peptide Recall
Real Spectra	Real Spectra	0.7160	0.7158	0.4971
Prosit Spectra	Real Spectra	0.3764	0.3743	0.1487
Real Spectra	Prosit Spectra	0.9277	0.9286	0.7238

Table 5.2: Performance of PointNovo [199] on real and artificial spectra. The real spectra are from the yeast partition dataset collated by Tran *et al.* [241]. The artificial spectra are from a duplicate dataset created using Prosit [89]. Test data are composed of *Saccharomyces cerevisiae* spectra with training data made up of spectra from 8 other organisms. AA stands for amino acid.

When PointNovo was trained on artificial spectra created using Prosit, its test performance on real data dropped dramatically (Table 5.2). Peptide recall fell by 70% with amino acid recall and precision falling by 47% and 48% respectively (Table 5.2). In contrast, a model trained on real data and tested on artificial data appears to perform much

better than the baseline of real test data. For the artificial test set created using Prosit, peptide recall was 46% greater than the real data test performance with both amino acid recall and precision both increased by 30% (Table 5.2). These results indicate that current artificial spectra models are not an adequate representation of tandem mass spectra for *de novo* evaluation. While they may provide accurate predictions of fragment ions, they lack the noise and random variation associated with real spectra. Artificial spectra provide a much simpler representation for the model to learn which is highlighted by the reduced performance when used for training, and the increased performance when used for testing. We therefore examined the distinctive characteristics of real spectra to appropriately modify artificial spectra.

5.4.1 Classification of Peaks

Due to the complexity of tandem mass spectra resulting from peptide fragmentation, many algorithms and models only consider backbone ions attributable to the peptide. Likewise, artificial spectra prediction algorithms, such as Prosit, only train their models to predict b and y fragment ions [89]. While these ions are important as they can reveal the peptide, they make up only a fraction of the total number of peaks [161]. Little work has been done to classify the other peaks in the spectra in the context of *de novo* peptide identification, despite their overwhelming majority. Here we aim to classify as many peaks as possible using the data collated by Tran *et al.* [241].

Not all possible peptide fragments will appear in the matched spectra. This is because the creation of some fragments will be more energetically favourable than others [189]. Table 5.3 shows the numbers of fragment ions matched in a sample of 50,000 spectra. This sample size was used to limit computational resources. The number of possible ions were calculated for each ion type using the peptides assigned during the database search conducted by Tran *et al.* [241].

As with any matching task there is a probability that some of the matches will occur by chance through the random alignment of a non-fragment peak with the position of an expected fragment ion peak. To estimate the occurrence of this phenomenon, a random peptide was created for each spectrum and all matching fragment ions identified. The most abundant ions were b and y backbone ions as expected (Table 5.3) [63]. They accounted for approximately 3% and 6% of the total peaks in the spectra respectively. Both of these ion types also matched the largest fraction of their possible peaks. 42% of all possible b ions were matched as well as 78% of all possible y ions (Table 5.3). Of the b and y ion matches observed, we estimate the fraction of random matches to be 11% and 10% respectively. The other backbone ion type analysed, a ions, were matched in much smaller numbers accounting for 1.8% of all peaks as they only matched 24% of those

Ion Type	#Possible	#Matched	Fraction matched	#Random	$\frac{\#\text{Random}}{\#\text{Matched}}$
Backbone	1934880	929072	48%	117596	13%
a	644960	154436	24%	35959	23%
b	644960	272490	42%	29728	11%
y	644960	502146	78%	51909	10%
Charge 2+	1934880	160515	8%	78457	49%
a(2+)	644960	43336	7%	19616	45%
b(2+)	644960	50799	8%	16811	33%
y(2+)	644960	66380	10%	21224	32%
Ion Loss	1842317	497932	27%	57651	12%
a-H20	143726	26641	19%	7138	27%
b-H20	143726	51607	36%	5050	10%
y-H20	644960	216682	34%	43334	20%
a-NH3	303529	39709	13%	8372	21%
b-NH3	303529	65583	22%	6093	9%
y-NH3	302847	97710	32%	8471	9%
Int. Frag.	7755678	1661562	21%	934571	56%
b	3877839	1031737	27%	474481	46%
a	3877839	629825	16%	460090	73%

Table 5.3: The number of matched peaks of different ion types in a sample of 50,000 HCD PSMs with a matching tolerance of 0.05 Da. The data are from 9 different organisms and research groups, collated by Tran *et al.* [241]. Columns indicate the number of possible ions of each type from the assigned peptides (#Possible), the number of these possible ions that were matched in the spectra (#Matched), the fraction of the possible ions that were matched (Fraction Matched), the number of ions from random peptides that were matched (#Random), and the ratio of the number of ions matched from the random peptides to the number of ions matched from the assigned peptides (#Random/#Matched).

possible (Table 5.3). The estimate for the fraction of matches which occurred randomly was also higher for a ions at 23%.

Fragment ions attributable to the database assigned peptide make up only a fraction of the peaks in MS/MS spectra (Figure 5.1). Backbone ions (a,b,y), both singly and doubly charged, were found to account for approximately 12% of all the peaks in the spectra. Of this 12%, 18% were estimated to be random (2% of the total). The fraction of peaks accounted for by backbone ions is almost half of what was estimated in previous work for HCD spectra from an LTQ Orbitrap Velos mass spectrometer [168]. Similarly, fewer matches were also observed for neutral loss ions in our experiment. Backbone ions with a neutral loss of a water or ammonia molecule accounted for just 5% of the total (Figure 5.1).

A substantial proportion of the peaks in the spectra could be attributable to internal fragments (18%, Figure 5.1). This is in part due to the large number of possibilities for this ion type; for a peptide of length 10 with 10 unique amino acids, there are 28 possible

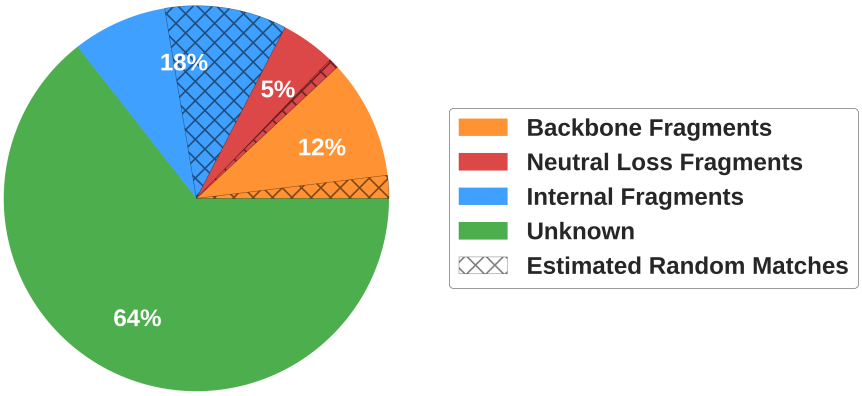

Figure 5.1: Fraction of peaks accounted for a sample of 50,000 HCD spectra. Percentages indicate the fraction of the total number of peaks each segment represents. Hatching indicates the proportion of each ion type estimated to have been matched by chance. The data are from 9 different organisms and research groups, collated by Tran *et al.* [241].

unique internal fragments of each ion type. This is compared to 9 possible backbone fragments of each ion type for the same peptide length. Also, the number of possible internal fragments grows exponentially for longer peptides. Furthermore, we consider two ion types (a and b) for each internal fragment which will double the number of possible peaks [166, 165]. Table 5.3 shows that 21% of the possible internal fragments were matched to a peak in the spectra. However, many of the internal fragments from the random peptides were also matched in the spectra. By comparing the number of internal fragments matched for both the assigned and random peptides, 49% of the actual internal fragment matches were estimated to have occurred by chance (Figure 5.1).

5.4.2 Distribution of m/z Error

The observed m/z values of peptide fragment ions may differ slightly from their theoretical values. These errors may be random or systematic [36]. Systematic errors are caused by biased measurements that result in repeatedly observed errors. To quantify the measurement error of the matched peaks, the mass difference between each matched ion and its expected value was recorded. A mass tolerance of 0.05 Da was used, so each peak matched fell within this error range [161]. Figure 5.2 shows the distribution of the difference in mass between the observed and theoretical values for singly charged, b and y ions. It also shows the error distribution for the peaks matched to the randomly

generated peptides. In general the real peptide jitter is centered around zero indicating random measurement error.

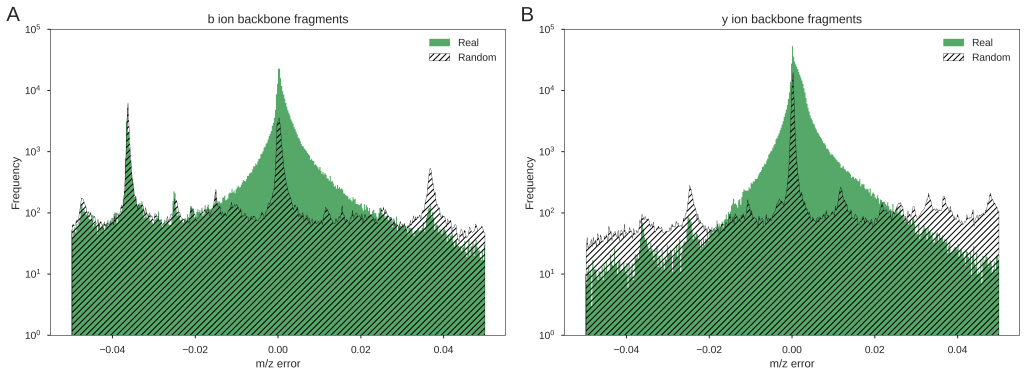

Figure 5.2: Distribution of error in matched peak m/z for singly charged b and y ions from a sample of 50,000 HCD spectra. The data are from 9 different organisms and research groups, collated by Tran *et al.* [241]. A shows the error distribution of matched b ions. B shows the error distribution of matched y ions. Error for ions from the real peptides are shown in green, with errors from the random peptides in black hatching.

There is a considerable difference between the error distributions of matched peaks from real and random peptides (Figure 5.2). It should be noted figure uses a log scale on the y axis. The b ions from real peptides had a mean squared error over 4 times lower than the random peptides ($1.6e^{-4}$ vs $7.1e^{-4}$). The y ions from the real peptides had a mean squared error of 3.4^{-5} with random peptides over 9 times larger at 3.2^{-4}. Notably, the random distribution matches the real distribution at the tails, especially for the b ions. This suggests that many of the real matches have occurred by chance. A similar method to estimate the number of spurious matches has been used by Goloborodko *et al.* [92]. Without using random peptides, they assumed the spurious matches formed a uniform distribution over the entire window below which the frequency never fell. Our random peptide matches show a similar distribution with a constant rate across the whole window, validating their approach.

The random distribution of y ions in Figure 5.2 B shows a very large spike at 0.00 m/z error. This is partially attributable to the prevalence of y_1 ions of arginine and lysine present in most spectra (Appendix C Table 2). The random distribution in Figure 5.2 B also has higher tails compared to the distribution of real y ions. However, we were unable to account for this difference.

For both plots in Figure 5.2, multiple significant secondary peaks in the error distribution can be observed along the x-axis, in addition to the peak at zero. These are particularly pronounced in Figure 5.2A showing the b ions. The two largest secondary

peaks in the distribution occur around 0.036 Da each side of the origin. This is approximately the mass difference between lysine (K) and glutamine (Q) which may explain this phenomenon. While they share most of their constituent atoms, lysine has an additional CH_4 while glutamine has an additional oxygen. Actual spectrum peaks and the possible fragment peaks that share a similar difference in chemical composition and consequently mass, will therefore produce the observed secondary peak in the distribution.

5.4.3 Abundance of Different Ion Types

Missing peaks, and hence missing fragmentation cleavages were found to be the greatest challenge *de novo* peptide identification algorithms must overcome [161]. Missing peaks occur when the abundance of a fragment ion, which is represented by peak intensity, is below the detection limit. The abundance of a fragment ion is dependent on how energetically favourable the corresponding cleavage is, which in turn depends on the cleavage position, peptide sequence and method of fragmentation [189, 237].

Figure 5.3 shows a comparison of the distribution of the presence and absence of fragment ions from 12 different ion types in both real and artificial spectra with assigned peptides of length 10 (median length of the data used).

On average, the artificial data were found to have more fragment ions present than the real data for the ion types that are predicted by Prosit [89]; b and y ions, both singly and doubly charged. In particular, b ions are much more prevalent in the artificial spectra. For y ions, low ion numbers are consistently more prevalent in artificial spectra while some higher ion numbers are more prevalent in real spectra (Figure 5.3). Both of these differences contribute to the observation that artificial spectra have fewer missing cleavages than real spectra [161], as at least one fragment ion is present for most cleavage sites. The reason for the increased number of fragments in artificial spectra may be partly due to the fact that artificial data do not contain measurement error or background noise meaning low intensity fragment ions are not lost. Hence, each ion predicted by Prosit will be present in the artificial spectra.

The overall trends for the b and y ion series share some similarities between the real and artificial data (Figure 5.3). For both data types in the b ion series, the b_1 ion has a low frequency before a large increase to the frequency of the b_2 ion. The frequency of the b ions then decreases for increasing ion numbers. Also, both data types show a generally decreasing frequency in the y ion series with the y_1 ion as the most frequently observed and the y_9 ion as the least frequently observed.

The difference in the number of missing peaks between artificial and real data was found to be related to the length of the peptide. A greater deviation in ion distributions was observed between real and artificial spectra for longer peptides (Appendix C Figures

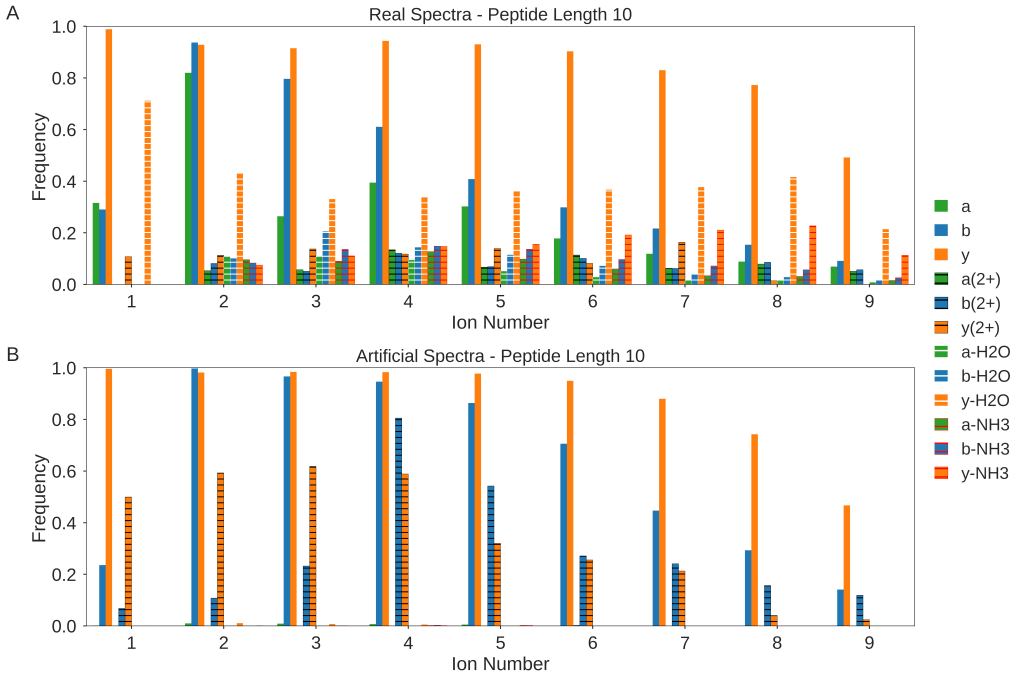

Figure 5.3: Comparison of the distribution of 12 different ion types in real versus artificial spectra for length 10 peptides in a sample of 50,000 HCD spectra. Frequency denotes the fraction of spectra where each ion was present. The real data (A) are from 9 different organisms and research groups, collated by Tran *et al.* [241]. The artificial spectra (B) are from a duplicate dataset created using Prosit [89]. Ions of the same type share the same base colour with different colour hatching indicating different charge states or neutral losses.

1-3).

5.4.4 Differences in Peak Intensity

Prosit produces extremely accurate fragment ion intensity predictions with a reported median spectral angle of 0.92 [89]. It uses a deep learning model composed of multiple recurrent neural network layers and fully connected layers to encode the peptide sequence and make the prediction. However, the Prosit model is deterministic and so for a particular amino acid sequence, it will produce the exact same spectrum every time. This is not the case in real spectra where variability is typically observed between spectra of the same peptide [248].

We compared the intensity predictions of Prosit to those observed in the real spectra for peptides of length 10 (Appendix C Figure 4) by determining the distribution of differ-

ences in intensity between Prosit and the real spectra. Only b and y ions both singly and doubly charged were used as these are the only frequently observed ion types predicted by Prosit. All intensities in real data were normalised to the maximum intensity of the fragment ions and not the spectrum, to provide a fair comparison with Prosit. However, in real data, a fragment ion may be the most intense peak in approximately half of the spectra [161]. The median difference between the artificial and real intensity values was less than 0.05 for all ion types from length 10 peptides, indicating a very low prediction error. Similar trends were observed for other length peptides (Appendix C Figures 5-7). This evaluation confirms the high accuracy reported by Prosit in the original manuscript [89].

5.4.5 Quantifying Internal Fragments

Internal fragments are caused by the cleavage at two or more backbone bonds in a peptide [189]. This results in fragments whose amino acids are not a sequence beginning at one end of the peptide, but instead are an internal sequence. Internal fragments have not yet been utilised in *de novo* peptide identification algorithms as their inclusion was found to make algorithms prohibitively complex [260]. This is despite their prevalence in tandem mass spectra as shown in Figure 5.1. Here we analyse which internal fragments are observed in real spectra and which may have been matched by chance.

Figure 5.4A shows the frequency of occurrence of different length b-type internal fragments in real spectra, and the estimated frequency of randomly matched internal fragments using random peptides. Internal fragments of length two had the greatest frequency of all internal fragment lengths for both the actual and random peptides. The fraction of possible unique length two internal fragments matched by the actual peptides (60%) was also greater than any other internal fragment length for all peptide lengths (Figure 5.4B).

Notably, the fraction of b-type internal fragments of length one that were matched for the assigned peptides (21%) was almost exactly the same as the fraction of length one fragments that were matched for the random peptides (21%). This trend was also observed for a-type internal fragments (Appendix C Figure 8).

5.4.6 Identification of Unknown Peaks

Most of the peaks in tandem MS spectra come from ions of unknown origin (Figure 5.1). These peaks are generally referred to as noise which can be chemical or electrical in nature [122]. The molecular structure of these ions can be used to investigate their origin. As molecules are made of atoms, most of their mass comes from protons and neutrons, together known as nucleons. The dalton (Da) is defined as 1/12th the mass of a carbon-12

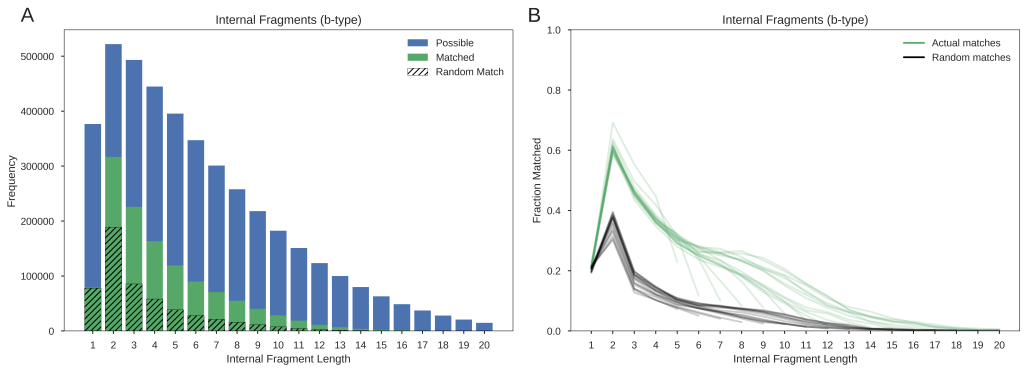

Figure 5.4: The number of b-type internal fragments matched by length in a sample of 50,000 HCD spectra. The data are from 9 different organisms and research groups, collated by Tran *et al.* [241]. A shows the counts of possible unique internal fragment masses (blue), matched internal masses (green), matched random internal masses (black hatch). B shows the fraction of the total number of possible internal fragments matched by the actual peptides (green) and the random peptides (black). Each individual line represents a different peptide length.

atom, the average mass of one of its nucleons. Therefore, if chemical noise is present, it should appear at roughly integer multiples of the dalton while electrical noise will not. Very few peaks were found to fit the criteria for electrical noise and instead almost all appeared to cluster at approximately integer multiples as expected for chemical noise (Appendix C Figure 9A). However, the clusters drifted off the integer units for larger m/z values (Appendix C Figure 9B).

To investigate the nature of this phenomenon in tandem mass spectra from shotgun proteomics we looked at the distribution of m/z values when plotted against the m/z modulo 1 (Figure 5.5). The modulo operation shows the remainder of the division after the modulus (in this case 1) has been divided in evenly.

The distribution of m/z versus m/z modulo 1 shows clear patterns for both the matched and unmatched peaks (Figure 5.5). There are two clear streaks in Figure 5.5A that wrap around from the top of the plot to the bottom. The wrapping is caused by the modulo operator. If the mass of these ions were integer multiples of the dalton, there would be a horizontal line across the plot. However, the streaks appear at a slope of approximately 1.0005. While this is only a slight deviation, it is consistent meaning larger ions appear significantly far from the integer values. This value agrees with previous reports of the average distance between peaks in tandem mass spectra [88, 74].

The binding energy of atoms can differ depending on the make-up of their nucleus. This means their masses can deviate from these integer values of the dalton. The dalton is normalised to the binding energy of carbon. With different binding energies for differ-

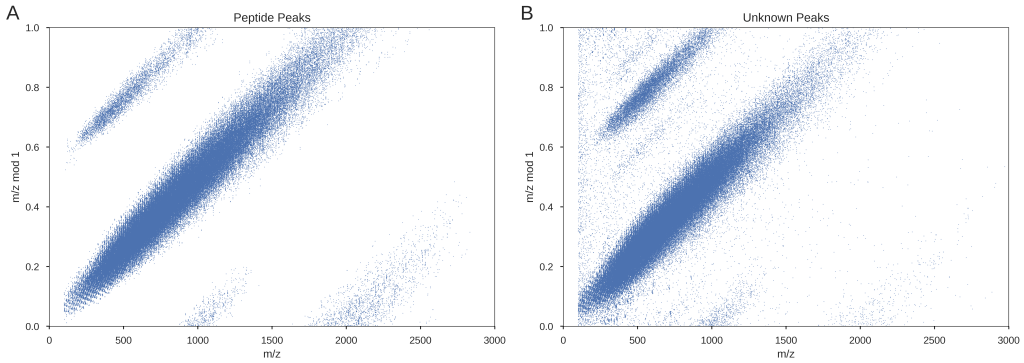

Figure 5.5: Distribution of m/z values vs m/z modulo 1 for peptide fragment peaks and unknown peaks in a sample of 50,000 HCD spectra. The data are from 9 different organisms and research groups, collated by Tran *et al.* [241]. A shows the distribution of the m/z values from peaks attributable to the database assigned peptide. B shows the distribution of the m/z from all other peaks.

ent atoms, almost all areas of the plot could be reached using different combinations of atomic masses. However, most of these atoms are extremely rare so instead we checked if these unknown peaks come from atoms common in biological molecules; i.e. molecules composed mostly of hydrogen, carbon, nitrogen, oxygen and sulphur. Appendix C Figure 10 is an m/z vs m/z modulo 1 plot showing random masses using equal ratios of these atoms, peptide ratios of these atoms, and carbohydrate ratios of these atoms using the formula $C_n(H_2O)_x$ [195]. The relative ratios of hydrogen, carbon, nitrogen, oxygen and sulphur in peptides were calculated using the Pyteomics module [141]. The streak shown in the real data matches the distribution of the peptide ratio (Appendix C Figure 10). This would suggest that the noise observed in the real data is composed of peptide-like fragments. Calculating the average mass per nucleon given the ratio of these elements observed in peptides equates to 1.0005, equal to the slope we calculated earlier.

The parallel streaks observed in these distributions are caused by different charges (Appendix C Figure 11). As only two charge types are considered for the peptide ions in Figure 5.5A, only two streaks are found. Conversely, there are more streaks in Figure 5.5B showing the presence of (3+) ions. Figure 5.5B has also a background level of noise which does not fit into any of the streaks which may be electrical in nature.

Finally we check to see if the observed streaks could also be obtained by contaminant metabolites. The human metabolome was downloaded from `https://hmdb.ca/downloads` [256]. Appendix C Figure 12 shows the ions of these metabolites for different charges. While there is an overlap between the metabolites and our observed peaks, the pattern is clearly different. While peptides have a limited set of possible masses corresponding

to combinations of the 20 amino acids, metabolites have much greater variety leading to the wider observed streaks. Again, this suggests that the unknown peaks are of peptide origin, albeit not the assigned peptide.

5.4.7 Evaluation of Modified Artificial Training Data

We then aimed to improve the similarity between Prosit generated spectra and real spectra through the addition of artificial noise. First real training data were modified to observe the effect of each noise type on PointNovo performance [199]. Table 5.4 shows the performance of the model when the different types of noise and variability, not present in artificial data, are removed from the real training data.

Noise Type Removed	AA Recall	AA Precision	Peptide Recall
m/z Jitter	0.6503	0.6497	0.4274
Intensity Variation	0.7150	0.7149	0.4916
Missing Peaks	0.6578	0.6547	0.4436
Non-Backbone Peaks	0.4957	0.4872	0.2452

Table 5.4: Performance of PointNovo [199] when trained using modified real spectra. The training data had noise removed from the four different spectrum attributes separately. The data are from the yeast partition dataset, collated by Tran *et al.* [241]. Test data are composed of *Saccharomyces cerevisiae* spectra with training data made up of spectra from 8 other organisms.

The removal of variation associated with the intensity of a peptide peak had the least effect with peptide recall decreasing by just 1% (Table 5.4). For this analysis, intensity values for the matched fragment ions were replaced with the Prosit equivalent. The small reduction indicates the high quality of the predicted intensity, agreeing with our earlier analysis (Appendix C Figures 4-7). As the reduction in performance was so small we did not try to modify the intensity further.

Artificial spectra generated by Prosit only contain backbone ions [89]. The removal of non-backbone peaks from real spectra, leaving only these backbone ions, resulted in a decrease in peptide recall of 51% (Table 5.4). These peaks account for over two thirds of the total number of peaks in tandem MS spectra (Figure 5.1). Without them, the model only needs to learn a mapping from the backbone peaks to the peptide, making the task much simpler. The model then overfits the data leading to reduced test performance. This supports the results observed for artificial training data (Table 5.2).

To test if the pattern for non-backbone peaks was unique to the assigned peptide, the non-backbone peaks were then shuffled between spectra of similar parent mass. The spectra were grouped into 100 bins of approximately 25 Da each. Non-backbone peaks were then swapped between spectra in the same bin. Although there are many duplicate

spectra in the dataset, this resulted in non-backbone peaks being reassigned to spectra of the same peptide in <1% of cases. Training PointNovo on these data resulted in a reduction in peptide recall of 7% when testing on unmodified real data (Appendix C Table 3). While this reduction in performance is much less than the reduction caused by removing all non-backbone peaks from the training data, it does suggest that some relationship exists between the assigned peptide and the non-backbone peaks.

Three different models of artificial non-backbone peaks were tested. These models were used to reintroduce the peaks back into the real data after they were removed. The performance of PointNovo was then compared using training data with non-backbone peaks removed versus training data with non-backbone peaks reintroduced artificially. Firstly, the non-backbone peaks were modelled as ions resulting from random peptide fragments since non-backbone peaks of unknown origin were found to be made up of amino acids (Figure 5.5B). To create a non-backbone peak, a random number and selection of amino acids was sampled, with their combined mass defining the m/z value. The peak intensity was sampled from the distribution reported in our previous work [160]. The artificial addition of these peaks to the training data increased the peptide recall by 43% on real test data compared to training data with these peaks removed (Appendix C Table 3). Non-backbone peaks were also modelled solely as internal fragments as these also account for a large proportion of the non-backbone ions in tandem mass spectra (Figure 5.1). Initially, all internal fragments were created for the matched peptide. Then a random sample of these were used to define the m/z values of the new peaks and the intensity values were sampled using the previously described distribution. Addition of these peaks to the training data with non-backbone peaks removed increased the peptide recall by 49% (Appendix C Table 3). Finally, non-backbone ions were introduced as a mixture of both random amino acids and internal fragments. Internal fragments were again sampled randomly but only enough to match their observed frequency. The remaining peaks were then added as random selections of amino acid sequences as before. Addition of non-backbone peaks using this combined method increased peptide recall by 58% compared to the training data with none present (Appendix C Table 3).

The presence or absence of a peak is related to its intensity as lower intensity ions are by definition less likely to appear in spectra. While the intensities of the peaks were found to be accurate, many peaks predicted by Prosit did not appear in real spectra. The addition to the training data of the peaks predicted by Prosit but absent from the real data caused an 11% reduction in peptide recall when tested on unmodified real data (Table 5.4).

The removal of m/z jitter associated with peptide peaks from the training data caused a 14% reduction in peptide recall (Table 5.4). As shown in Figure 5.2, peaks do not appear exactly at the expected value. To remove the associated jitter, matched peaks were set

to the expected m/z value. The m/z jitter was then reintroduced using two different distributions. The first is a mixture distribution of two normal distributions centered at zero with means of 1e-2 and 1e-3 in a 1:1 ratio. The reintroduction of this artificial jitter to the real training data resulted in a 6% increase in peptide recall (Appendix C Table 3). The jitter was also reintroduced using a mixture distribution of a Laplace distribution with a scale parameter of 2.5e-3 and a uniform distribution between -0.05 and 0.05 with a 12:1 ratio. The introduction of this jitter resulted in a 5% increase in peptide recall (Appendix C Table 3).

Using the understanding gained from modifying the real data by removing and replacing the different types of noise, the artificial spectra were then modified to improve their utility. The performance of PointNovo was assessed using real test data and modified artificial spectra as training data (Figure 5.6).

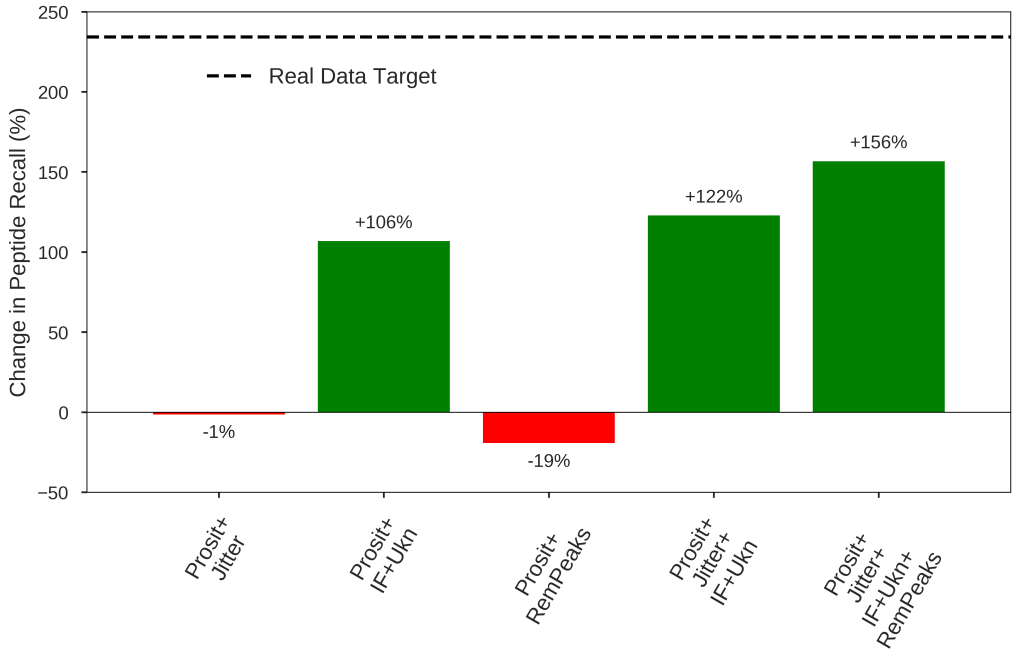

Figure 5.6: Change in performance of PointNovo [199] when trained on artificial spectra and tested on real spectra. The labels on the x-axis indicate the additions to the Prosit [89] generated training data. The real test spectra are from the yeast partition dataset, collated by Tran *et al.* [241]. Jitter signifies addition of m/z noise. IF indicates the addition of internal fragment noise peaks. Ukn indicates the addition of random peptide fragment noise peaks. RemPeaks indicates the removal of some of the lowest intensity peaks. The dashed line shows the performance of PointNovo trained on real spectra.

Modifying the artificial training data with each of the different types of artificial noise

improved the peptide recall by 156% compared to the unchanged artificial training data (Figure 5.6). To put this in reference to real data, the peptide recall using artificial training data increased from 30% of the real training data value to 77% by adding these changes. Notably, not all of the introductions of variability improved performance when introduced alone. Both the introduction of jitter and the removal of peaks reduced performance by 1% and 19% respectively when they were the only changes to the artificial data. However, combined with the addition of internal fragments and unknown peaks they make significant improvements (Figure 5.6).

Furthermore, it should be noted that PointNovo takes 12 ion types into account; a, b, y, a-H_2O, b-H_2O, y-H_2O, a-NH_3, b-NH_3, y-NH_3, a(2+), b(2+) and y(2+) ions. Only 4 of these overlap with the ion types predicted by Prosit; b, y, b(2+) and y(2+) ions. The peptide recall reached using artificial spectra in Figure 5.6 is over 3/4 of the peptide recall reached when using real spectra, despite only using 1/3 of the fragment types. Further extension of the ion types predicted by Prosit would likely yield an increase in performance.

5.5 Discussion

Artificial data have been used extensively in machine learning to both train and test different models. In the context of *de novo* peptide identification it has only been used in the evaluation of models [25, 177, 161]. However, how applicable this approach is to real data performance has never been fully analysed.

In this research, PointNovo was first trained and tested on different combinations of unmodified real and artificial data. Testing on artificial data compared to real data testing reported greatly inflated performance. The artificial spectra had more peaks corresponding to fragment ions as well as no noise when compared to real spectra. This meant that the artificial spectra were much easier to classify than real spectra. Conversely, when training on artificial data the model began to overfit as the data had a much lower level of complexity. Testing this model on real data showed markedly decreased performance as the model was unable to deal with the noise of real spectra. It is evident from these tests that artificial spectra do not currently replicate the true complexity of real spectra.

To quantify these differences, the peaks attributable to the database assigned peptide in real tandem MS spectra were first classified. Previous studies have also aimed to categorise these peaks [168, 218]. However, the work presented here describes different data from a different mass spectrometer. Furthermore, random peptides were used to estimate the number of spuriously matched peaks and include quantification of the different forms of noise associated with real spectra. The real data were also compared to artificial

spectra created using the Prosit pipeline.

By matching the peaks to fragment ions attributable to the assigned peptides from a database search, 36% of peaks in the spectra were accounted for (Figure 5.1). However, 13% of peaks in the spectra were also matched to fragment ions attributable to a random peptide (Figure 5.1). Random peptides were used in this research to provide an estimate of the number of spurious fragment ion matches. Taking this into consideration, 77% of the peaks in the spectra could not be attributed to the database assigned peptides.

The creation of random peptides also allowed us to estimate the number of spurious matches for each ion type. This should guide informed selection of ion types that should be included in future *de novo* models. If an ion does not appear very often and is likely to be matched by chance, it may not be very useful for peptide prediction. While an ion type that appears more often than chance can still provide additional information toward the peptide prediction process, it must be balanced, however, with a trade-off in model complexity. Indeed, the inclusion of each additional ion type increases the model complexity, making training and prediction slower, as well as requiring more computational resources.

Internal fragments are not currently utilised in *de novo* peptide identification algorithms. However, 60% of all possible internal fragments of length 2 were matched in the spectra. While further research is required to assess how this might be realised, the inclusion of internal fragments could be explored in future *de novo* algorithms. Previous work highlighted the need for complete spectrum encoding to counteract missing fragmentation cleavages [161], a strategy that would inherently include internal fragments into the prediction process. A recent approach to *de novo* peptide identification using transformers, Casanovo, encodes every peak in the spectrum thereby also including internal fragments [267]. With this additional information Casanovo reported a mean improvement of 1.3% over PointNovo in peptide precision [267].

The creation of representative artificial data necessitates the inclusion of all peaks present in the spectra, not only backbone ions. Liu *et al.* created a model, PredFull, that attempted to completely recreate tandem MS spectra [148] using a bin size of 0.1 Da, which is large in light of the precision of modern mass spectrometers. The Q-exactive used to create the data for this research has a maximum precision of $<$1 ppm [208]. Training PointNovo using PredFull spectra and testing on real spectra resulted in a peptide recall of 0.20. This was better than Prosit alone (0.15) but not better than the peptide recall achieved in this work through our modifications (0.38).

As artificial data are synthetic, they lack much of the noise associated with real data. In this research, four types of noise that only appear in real data were categorised; m/z jitter, missing peaks, intensity and unknown peaks. Despite being deterministic, Prosit was found to predict the intensity with a mean error of less than 0.05. Using this de-

terministic artificial intensity in the training data was found to have very little effect on model performance (1% reduction). Therefore the artificial intensity was not further modified. As there is such a large diversity of peptides in the training data, variability in intensity between spectra of the same peptide is of minimal importance.

Using a novel plot of m/z versus m/z modulo 1, this work was able to provide evidence as to the origin of the unknown peaks in MS/MS spectra. The mean m/z to nucleon ratio of the unknown peaks was found to be indicative of molecules made of amino acids. This suggests that the unknown peaks in tandem MS spectra are due to peptide contaminants or coeluting peptides. This information was then used to create a model to recreate these peaks in artificial spectra.

Finally, the models of three different types of noise were combined with the Prosit predicted spectra. This increased the peptide recall of a model trained on artificial spectra from 30% to 77% of the peptide recall of a model trained on real spectra. While substantial improvements are still required to realise the full potential of artificial data in the context of *de novo* peptide identification, this work has made significant progress in this area.

Continued improvements would transform artificial spectra into a tool for the systematic and comprehensive analysis of *de novo* algorithms. Artificial data can facilitate evaluation at adjustable levels of data complexity such as increased noise. Observing how the performance of a model is impacted by changes in specific data characteristics would provide valuable insights into its strengths and weaknesses. Currently, models must be tested on real spectra where the effects of different data characteristics are difficult to separate [161]. Furthermore, supplementing the training data with these modified spectra may also help reduce the bias associated with database labelled spectra. While artificial spectra may not completely replace real training data, they can provide diverse or rare examples to the model to help improve performance. Difficult to classify spectra will by definition appear less often in the training data obtained using a database search. This will in turn make it difficult for *de novo* models to learn to classify such spectra. Also, as *de novo* performance is always benchmarked using this subset of high-scoring spectra, the extent of this issue is not easy to quantify. The artificial generation of difficult-to-classify spectra, those with many missing fragmentation cleavages [161], may offer both a way to identify this bias as well as alleviate it through supplementation of the training data. A problem to note here is that models used to generate artificial spectra also currently rely on database labelled spectra as training data. These models will therefore likely have their own bias. Further research is needed to see if the addition of noise will diversify these artificially generated spectra enough to mitigate against their own inherent bias when used as training or test data for *de novo* peptide identification models.

5.6 Conclusion

This work provides a critical analysis of artificial data in the context of *de novo* peptide identification algorithms. It presents a comprehensive survey of the different peptide fragment ions matched in tandem MS spectra and provides evidence for the origins of unmatched peaks. While this research shows the current limitations of using artificial data to train or evaluate *de novo* peptide identification algorithms, it also highlights its future potential. The inclusion of additional noise was shown to significantly increase the utility of artificial spectra for model training. High quality artificial spectra could help alleviate the reliance of current algorithms on a database search for generating training data. Furthermore, such artificial spectra could allow for the quantification of the effects of specific data characteristics on *de novo* peptide identification. A greater understanding of the challenges facing *de novo* algorithms is necessary to he design of more robust future models. This work represents the first step toward such a new approach to *de novo* peptide identification model training and evaluation.

Chapter 6

Conclusion

De novo peptide identification has many applications in proteomics and has recently experienced substantial growth. The adoption of machine learning has led to a reinvigoration of the field, with several new algorithms and approaches being published. The field however, is lacking critical independent analyses of these many approaches.

The aim of this thesis was to address the following research questions:

1. What are the main challenges to *de novo* peptide identification?
2. Can we design better encoding modules to address these challenges?
3. Can artificial spectra be leveraged to aid the training and evaluation of *de novo* peptide identification algorithms?

From the research described in this thesis we can answer these questions with the following responses:

1. Through the analysis of different data characteristics and their effects on performance, missing fragmentation cleavages were found to be the greatest challenge facing current state-of-the-art *de novo* algorithms.
2. A novel CNN-GNN encoding module proposed in this thesis was evaluated on its ability to identify peptide ions. It was shown to perform better than the encoding modules used in the current state-of-the-art *de novo* algorithms for both increased noise and missing cleavages.
3. Artificial spectra were determined to be missing the noise and variability of real spectra that would make them useful for *de novo* peptide identification. However, augmentation of the artificial spectra through addition of this noise improved their utility significantly.

6.1 Summary of Contributions

6.1.1 Main Challenges to *De Novo* Peptide Identification

This thesis presented a comprehensive evaluation of two state-of-the-art *de novo* peptide identification algorithms (Chapter 3). This included an exploration and description of the characteristics of database assigned PSMs which are used for the training and evaluation of *de novo* algorithms. The data were found to be heavily biased toward spectra with very few missing fragmentation cleavages. The study also showed missing fragmentation cleavages to be the feature of the data that *de novo* algorithms found most challenging (RQ1). This was in part due to the algorithms' step-by-step approach. The number of noise peaks observed in the spectra dwarfed those attributable to the assigned peptide. While this noise also negatively affected performance, the effect was difficult to quantify due to the dominating effect of the missing cleavages. Prior to this research there were no independent evaluations that looked specifically at the characteristics of real data and how they affect *de novo* peptide identification performance. This thesis provides an instructive insight into the challenges and limitations currently facing *de novo* algorithms. The understanding gained from this work of the data and algorithms used in *de novo* peptide identification may help researchers develop new approaches in the future. It may also provide insight into the limitations of current tools to those researchers looking to use them to analyse their data. Furthermore, the findings of this research serve as the foundation for the other contributions described in this thesis.

6.1.2 CNN-GNN Peptide Ion Encoding

Different machine learning encoding modules were explored in this thesis with the aim of addressing the challenges to *de novo* peptide identification. GNNs were proposed as a model as they are capable of encoding long range relationships such as those present in MS/MS spectra. This includes the relationship between fragment ion abundance and the complete peptide sequence. This research represents the first step toward GNN peptide identification by demonstrating the ability of this architecture for peptide ion encoding (Chapter 4). A CNN-GNN hybrid model was shown to outperform encoding modules used by all state-of-the-art *de novo* algorithms at peptide ion identification. The proposed model performed best over the complete range of missing cleavages and noise present in the data (RQ2). Before this, the utility of GNNs in the context of *de novo* peptide identification had not yet been shown. While more work is needed to incorporate the encoding module into a complete peptide prediction model, the potential of GNNs is now clear. Furthermore, despite the increased performance of the GNN, including expert

selected features as an input resulted in greater average precision. In fact, the inclusion of these expert features increased the performance of all models, showing the importance of expert knowledge in *de novo* peptide identification algorithm design. Even when using complex machine learning models there is still a need to understand both the data and the problem domain when designing *de novo* peptide identification algorithms.

6.1.3 Utility of Artificial Spectra in *De Novo* Peptide Identification

Previous to the research described in this thesis (Chapter 5), there did not exist an extensive comparison of real and artificial spectra in the context of *de novo* peptide identification. Models that produce artificial spectra provide the means of generating high quality test data that can be used to evaluate models. However, how reflective such analysis is of real data performance has not previously been shown. This research identified key differences between real and artificial spectra which influence the performance of *de novo* peptide identification models (RQ3). Underpinning this analysis was a survey, not only of the fragment ions present in each, but also the noise present in the real spectra and absent from the artificial spectra. Four types of noise were identified and classified for four respective features of the spectra; peak m/z, peak intensity, missing fragment ions and unassigned peaks. A novel plot was introduced to identify the source of unassigned peaks in the spectra, which showed that the vast majority are of peptide origin. A random peptide model was also used to estimate the number of spurious peptide ion matches in the data. This highlighted which ion types were most likely to be matched at random and so were less effective in the identification of peptides. Compared to real spectra, evaluation of models on artificial spectra showed inflated performance. Training of models on artificial spectra before testing on real spectra resulted in much lower performance, further highlighting their differences. This research demonstrated how unaugmented artificial spectra are not representative enough of real spectra for use in *de novo* peptide identification. However, by introducing models of the four noise types into artificial spectra, the performance of a *de novo* peptide identification model trained on these artificial data and tested on real data was increased by 142%. This was despite only using stochastic models of noise and a limited number of ion types. While the work showed that artificial spectra in their current form cannot be used to make inferences about real data performance, it also demonstrated how they could be improved through the introduction of noise.

6.2 Impact

This thesis as a whole provides a foundation in both the underlying data as well as the key algorithms for *de novo* peptide identification. The contributions of the thesis outline some of the current limitations and challenges facing the field as well as how they might be addressed. The work also provides a fundamental understanding of *de novo* peptide identification and so could be used by research groups looking to break into the field. The current state-of-the-art *de novo* algorithms are all based on approaches from the same research groups that have been incrementally improved over time. Opening the field to new research groups increases the likelihood of new innovation and development in algorithm design.

This thesis shows how step-by-step approaches struggle with missing fragmentation cleavages. Groups developing new approaches should look toward full-spectrum encoding as recommended in Chapter 3. A new GNN based encoding module is then proposed in Chapter 4 that can deal with these missing cleavages more easily than other methods. With all source code available, research groups could build their new *de novo* algorithm around this model thereby capitalising on its superior peptide ion encoding ability. Finally, the current limitations of artificial data in the context of *de novo* peptide identification and how its utility could be improved are highlighted in Chapter 5. The work provides a warning to research groups looking to use presently available artificial spectra as a means of testing their models. However, it also provides a framework whereby artificial spectra can be augmented to improve their likeness to real spectra and thus their utility.

Clearly, innovation is required for the development of the next generation of *de novo* peptide identification algorithms as shown by the limitations of current approaches exposed in this work. However, this thesis also provides insight into the possible solutions to these limitations as well as models that do not suffer from the same constraints. With many new areas of research exposed and the first steps taken toward a new methodology, this work provides a foundation for other research groups to develop new approaches that will help *de novo* peptide identification realise its full potential.

6.3 Limitations

6.3.1 Computational Cost of Full Spectrum Encoding

Through this research it is recommended that future *de novo* algorithms should consider complete spectrum encoding. This is in contrast to algorithms such as DeepNovo that only focus on small sections of the spectrum at one time [241]. However, such an architecture

would come at a much greater computational cost. The current step-by-step approaches are successful as they simplify the peptide prediction problem to amino acid prediction, one at a time. While we have shown this to be limiting in terms of accuracy, it makes the algorithms very efficient. Learning a mapping from the complete spectrum to the complete peptide is much more challenging and so it may require more complex models, more training time and more computational resources. However, this assumes the use of similar machine learning models to those currently employed. If more innovative models could be devised that better encode the spectrum than are currently available, perhaps the increase in model complexity could be mitigated.

6.3.2 Peptide Ion Encoding

The CNN-GNN model proposed in this research was only evaluated on its ability to distinguish fragment ions from noise. The proposed model is incompatible with the other step-by-step *de novo* algorithms considered as there is no natural order for the nodes in the graph. Hence, it was not incorporated into any of these algorithms for testing. While the problem of ion identification is related to peptide prediction, the utility of the proposed model on the latter remains untested. It is therefore an open problem to incorporate this module into a *de novo* peptide prediction model. One possible approach would be to use an attention mechanism combined with an LSTM. Again, as they do not have a natural ordering, the output of a GNN does not easily fit with an LSTM for sequence encoding or prediction. However, an attention mechanism allows the LSTM to focus on relevant nodes at each prediction step, thus removing the need for an order. A CNN-GNN encoder coupled to an LSTM with attention decoder is an architecture for *de novo* peptide identification that could be explored.

6.3.3 Artificial Data

The results of this research indicate the potential to improve the performance of models trained on artificial spectra. However, as the data augmentation presented was unable to completely replicate the observed noise, it is still unclear if training with augmented artificial spectra can compete with the performance of models trained on real spectra. Instead of completely replacing real spectra datasets, artificial spectra might only be useful to supplement them. In this context they could provide training examples of spectra with characteristics that are scarce following a database search. Furthermore, the artificial spectra were only evaluated when training PointNovo [199]. Current step-by-step *de novo* algorithms, such as PointNovo, rely on relatively simple models that may see a benefit from the inclusion of artificial spectra in their training dataset. As algorithms evolve and become more complex it may be increasingly difficult to create spectra with

sufficient fidelity to encapsulate the patterns which these new models will be capable of learning.

6.3.4 Random Peptide Model

The random peptide model used to estimate the number of spurious matches will only provide information at the dataset level. As only a single random peptide is used per spectrum it does not provide spectrum specific information. Creating multiple random peptides per spectrum could help provide a peak specific matching probability but this would require a lot of computational resources and is beyond the scope of this research.

6.3.5 Noise Models

The noise models used in this research to augment the artificial spectra were stochastic in nature. This means that the added noise was randomly sampled so that the overall distributions of both the real and artificial data features were similar. The additional noise was therefore independent of the spectra. While this improved the utility of the artificial spectra significantly, this approach is incapable of capturing more complex relationships if they are present. Only the internal fragments were spectrum specific as they were created using the assigned peptide. A more accurate model of the noise in peptide spectra would require a learned model that could predict the spectrum entirely and not just the fragment ions. PredFull [148] is such an algorithm that predicts the spectrum in its entirety. However, as shown in Chapter 5, this model has a low resolution which limits its utility in this context. As PredFull uses a discretised spectrum, increasing the resolution would make the model prohibitively complex. For more precise ion prediction, a model would need to be capable of making these predictions without a limit on their number. This would indicate the need for sequential models, such as LSTMs or transformers, that have this capability to be incorporated into the prediction stage of the algorithm.

6.4 Future Work

6.4.1 Complete Spectrum Encoding and Graph Neural Networks

Since the publication of the work presented in Chapter 3, a model has been developed using a complete spectrum encoding for *de novo* peptide prediction as was recommended. Casanovo is a transformer based encoder-decoder model for *de novo* peptide prediction [267]. The model outperformed both DeepNovo [241] and Pointnovo [199] in peptide precision, by 8% and 1% respectively. While the margin is small, Casanova achieved this without the use of dynamic programming. In their approach, peaks are encoded using a

positional embedding before they are inputted into the transformer model. This converts the peaks into a series of sinusoidal waves. The long range interactions and possible links between peaks are therefore lost when using this approach. In contrast, GNNs can capture these specific interactions, providing a simpler problem for the model to learn. While our work on GNN encodings shows their potential in this area, more work is needed to incorporate them into a complete peptide prediction algorithm.

6.4.2 Database Peptide Scoring

The structure of the GNN encoding model lends itself for use as a database scoring model. By trying to identify fragment ions every peak is given a score. These could be then used to distinguish between true and false PSMs. During this research preliminary tests were performed where the sum of the scores of the matched peaks for each peptide was used as a PSM score (data not shown). The number of peptides recovered using this score with a 1% FDR threshold was then calculated. While the results were indeed better than summing the intensities alone, and similar to current simpler scoring models such as hyperscore, this methodology was not successful in retrieving more significant peptide matches than the popular database search algorithm X!Tandem [52]. However, these tests were performed on limited data and a more rigorous evaluation would be required to explore this in greater detail. For example, different ways of combining the scores could be explored. Furthermore, our model was optimised to distinguish between peptide and non-peptide peaks and not to distinguish between correct and incorrect peptides. A more specific training pipeline, focused on peptide spectrum matching, may help improve the model's ability at this task.

6.4.3 Artificial Data

A model capable of creating realistic artificial spectra could provide training examples with many missing cleavages, the type of spectra *de novo* algorithms struggle with the most [162]. This in turn may help improve the accuracy of these algorithms for such difficult cases. It would also provide a means of generating quality PSMs for model evaluation. However, despite the improvements presented in this thesis, artificial spectra are still not representative enough for use in the evaluation or training of *de novo* peptide identification models. If models trained on artificial data cannot replicate the performance of real data then they do not capture the same complex patterns. Therefore, it cannot be assumed that trends observed when evaluating models on artificial data will be relevant to real data performance. The work in improving artificial spectra utility could be built upon in the future by designing a model that can learn to completely replicate the tandem MS spectra, including the noise. The design of such a model could allow the adjustment

of the levels of the different noise types and provide a framework where models could be fairly evaluated under all possible feature combinations. Such data may also alleviate the dependence of *de novo* models on training data obtained through a database search. The feature distribution of current training data is limited by the characteristics of the spectra matched by the database search. As shown in this research, this is heavily skewed towards spectra with few missing fragmentation cleavages.

6.5 Final Remarks

De novo peptide identification has seen significant recent improvements as it is benefiting from increasing data quality and the incredible capacity of current machine learning models to perform pattern recognition. However, care must be taken when implementing such models as the requirements of the task and the characteristics of the data must be fully understood. Future *de novo* peptide identification algorithms need to be designed with a clear understanding of the data and how best it can be modelled. This research provides a foundation for understanding the challenges *de novo* peptide identification currently faces. Furthermore, the work presented in this thesis on encoding modules and artificial data provide the first step towards new approaches to this problem. Overall, this thesis illuminates the path toward the development of a new generation of *de novo* peptide identification models that will increase their potential for use in proteomics research.

Chapter 7

Appendices

A Supplementary Information (Ch. 3)

Dataset	Organism	Database Size (Protein Number)	Date Downloaded	Yeast Control
MouseCID	Mus musculus	17082	01/07/20	No
YeastCID	Saccharomyces cerevisiae	6049	01/07/20	No
EcoliCID	Escherichia coli	4438	28/02/21	Yes
StaphAurCID	Staphylococcus aureus	2607	12/07/21	Yes
HeLaHCD	Homo sapiens	20286	28/06/20	No
PyroHCD	Pyrococcus furiosus	4020	01/07/20	Yes
EcoliHCD	Escherichia coli	4438	28/02/21	Yes
StaphAurHCD	Staphylococcus aureus	2607	12/07/21	Yes

Table A.1: Database details. Breakdown of the protein databases downloaded from Uniprot used in this research.

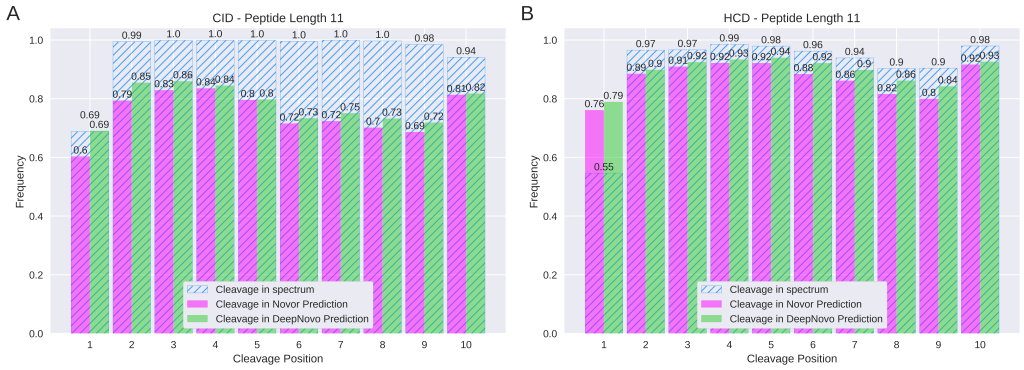

Figure A.1: Algorithms' cleavage predictions for length 11 peptides compared to cleavages in spectra. 11 was found to be the most common peptide length. The hatched blue bars represent the fraction of spectra that contain an ion from that cleavage site in the peptide. The magenta (Novor) and green (DeepNovo) bars show the fraction of peptides predicted by each algorithm that contained that same cleavage site. Numbers on top of the bars indicate their value.

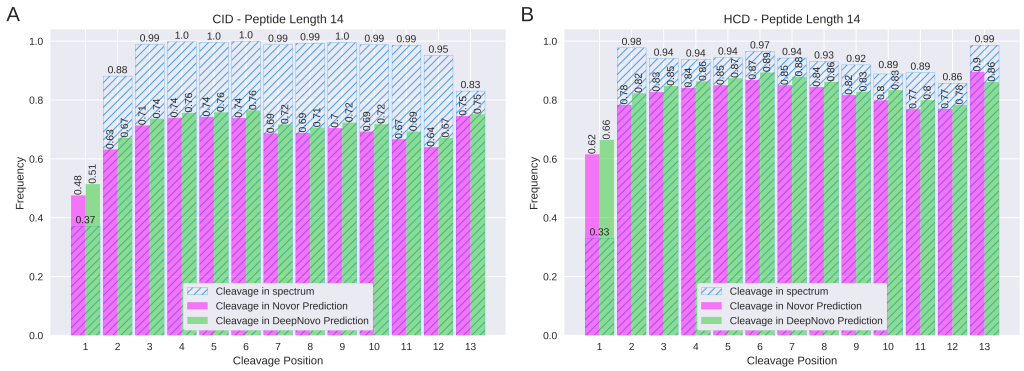

Figure A.2: Algorithms' cleavage predictions for length 14 peptides compared to cleavages in spectra. 14 was found to be the median peptide length. The hatched blue bars represent the fraction of spectra that contain an ion from that cleavage site in the peptide. The magenta (Novor) and green (DeepNovo) bars show the fraction of peptides predicted by each algorithm that contained that same cleavage site. Numbers on top of the bars indicate their value.

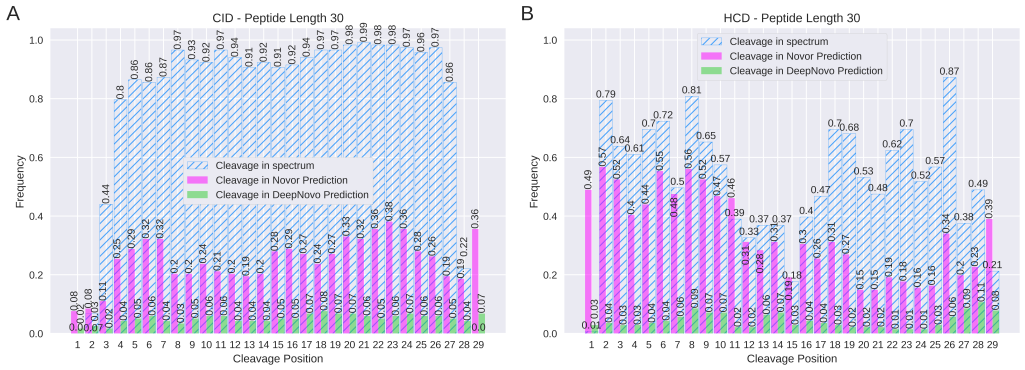

Figure A.3: Algorithms' cleavage predictions for length 30 peptides compared to cleavages in spectra. The hatched blue bars represent the fraction of spectra that contain an ion from that cleavage site in the peptide. The magenta (Novor) and green (DeepNovo) bars show the fraction of peptides predicted by each algorithm that contained that same cleavage site. Numbers on top of the bars indicate their value.

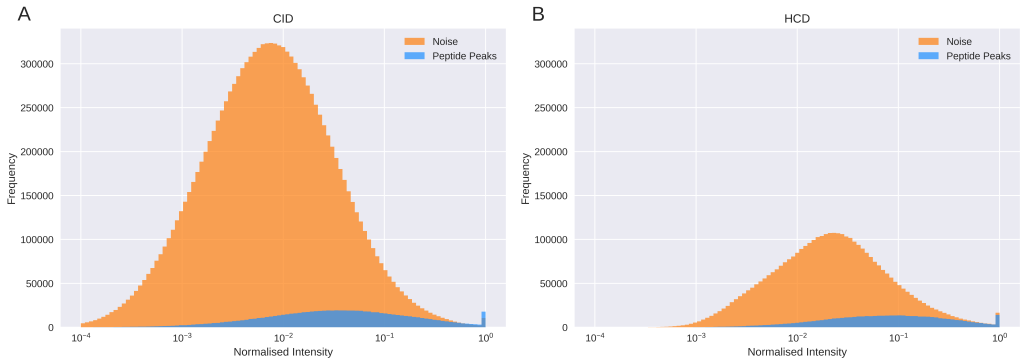

Figure A.4: Intensity distributions spectra peaks. Distributions of the normalised intensities of both noise and peptide peaks for CID (A) and HCD (B) data.

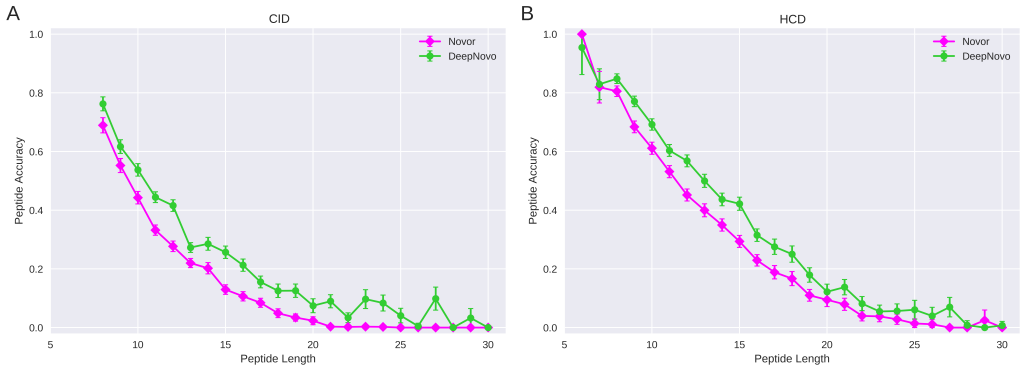

Figure A.5: Peptide accuracy of the algorithms vs peptide length. Peptide accuracy of Novor and DeepNovo for all peptide lengths. A shows peptide accuracy in CID data while B shows peptide accuracy in HCD data. 95% confidence intervals surround each point.

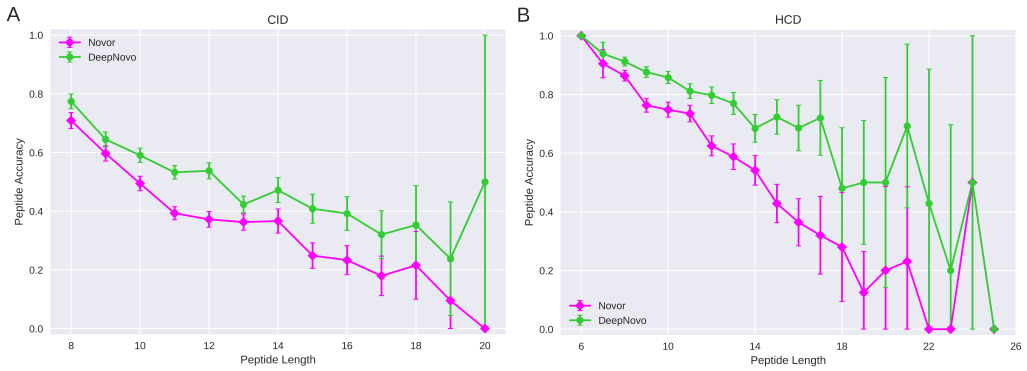

Figure A.6: Peptide accuracy of the algorithms vs peptide length when no cleavages are missing. Peptide accuracy of Novor and DeepNovo for all peptide lengths and when each cleavage in the peptide has at least one ion in the spectrum. A shows peptide accuracy in CID data while B shows peptide accuracy in HCD data. 95% confidence intervals surround each point.

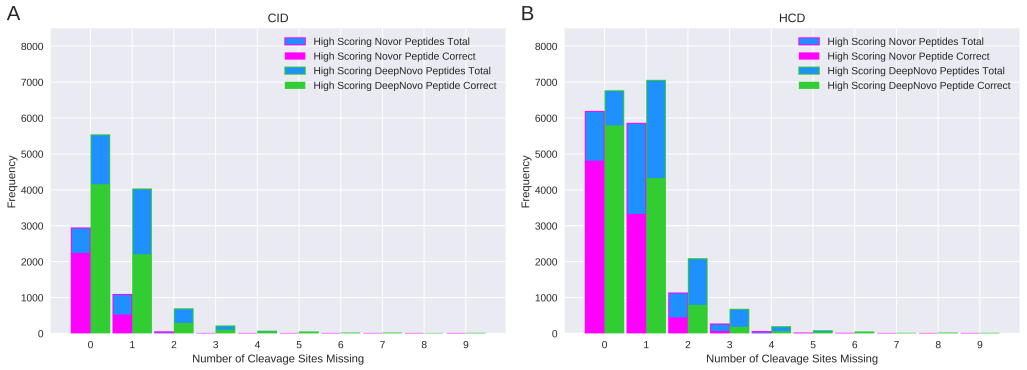

Figure A.7: Algorithm performance for increasing numbers of missing cleavages in high scoring peptides. Bar plot showing the number of correctly predicted high-scoring peptides by Novor (magenta) and DeepNovo (green) as well as the total number of high-scoring peptides returned by each algorithm (blue with surrounding colour) for each number of missing cleavage sites. High-scoring CID peptides are shown in A with high-scoring HCD peptides shown in B.

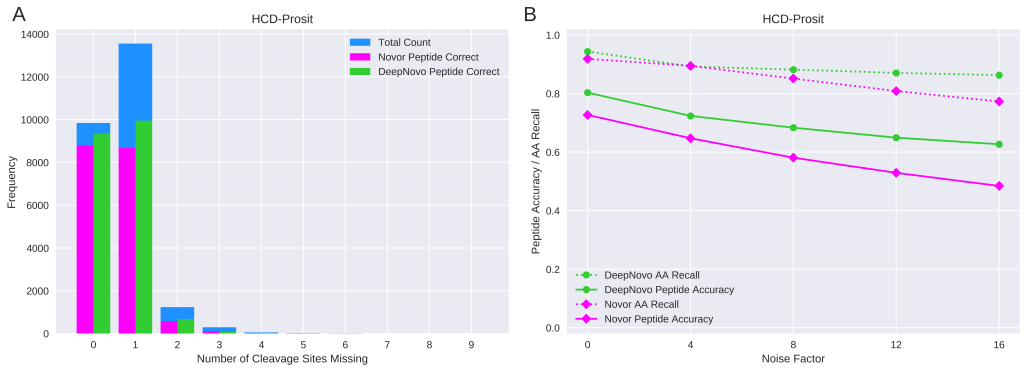

Figure A.8: Algorithm performance on artificial HCD data. Bar plot of algorithm performance with respect to missing fragmentation cleavages in artificial data is shown in A. The plot shows the total number of spectra (blue), the total number correctly identified by Novor (magenta) and the total number correctly identified by DeepNovo (green) for each number of missing cleavages. The performance of the algorithms with respect to increasing levels of random noise in artificial data is shown in B. Solid lines indicate peptide accuracy while dashed lines show amino acid (AA) recall.

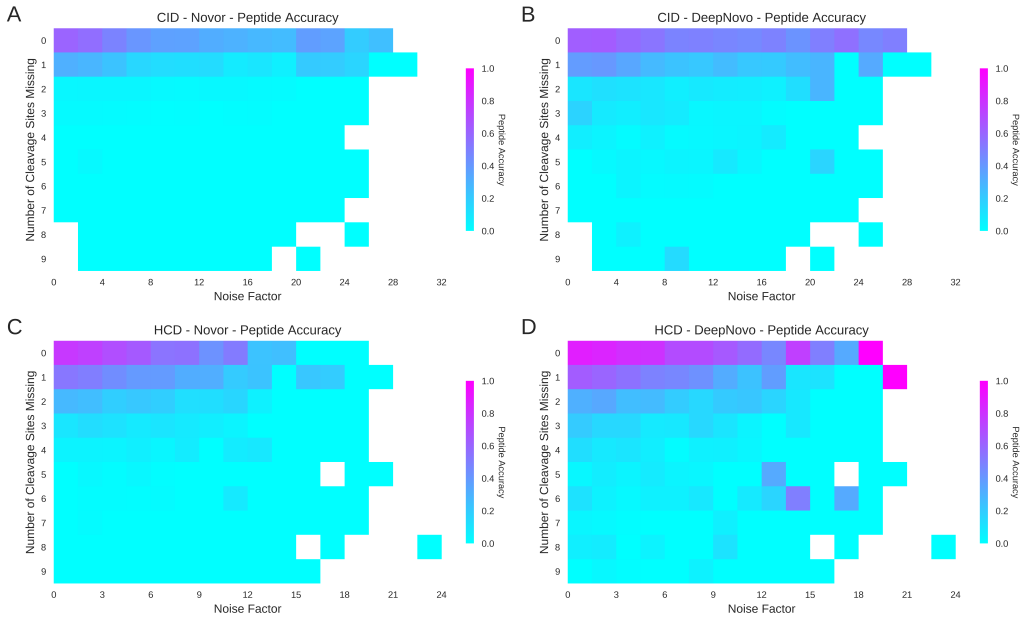

Figure A.9: Peptide accuracy as a function of the number of missing cleavages and the Noise Factor. Higher peptide accuracy is shown in pink, with lower accuracy shown in cyan. Performance of Novor across the two fragmentation types are shown on the left (A and C) with the performance of DeepNovo shown on the right (B and D). CID data are shown on top (A and B) with HCD data shown on the bottom (C and D).

B Supplementary Information (Ch. 4)

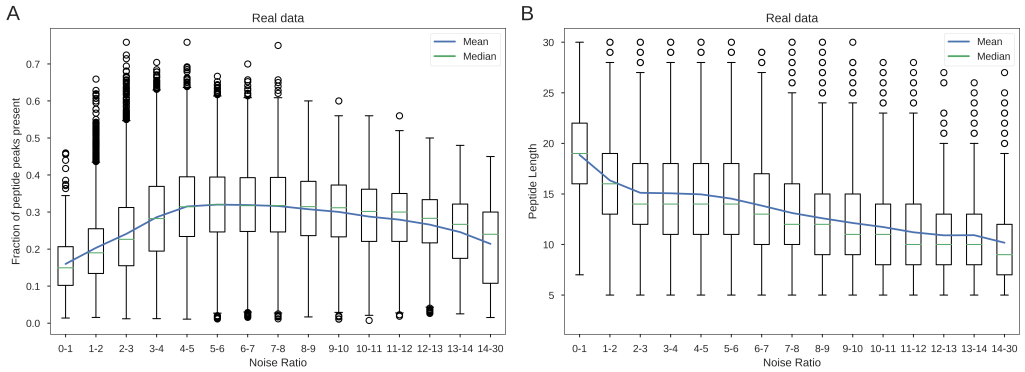

Figure B.1: Correlation of features in real tandem MS data. The correlation between the fraction of peaks present and the noise ratio in the real data used in this study is shown in A. The correlation between the length of the peptide and the noise ratio in the spectra for the same data is shown in B. Box plots indicate the distribution of spectra while the blue line indicates the mean and the green lines indicate the modes.

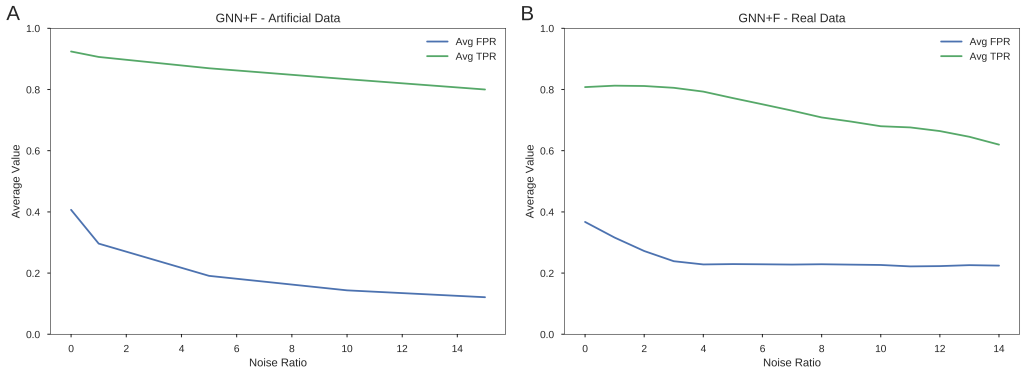

Figure B.2: Impact of noise on the TPR and FPR of the GNN+F in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

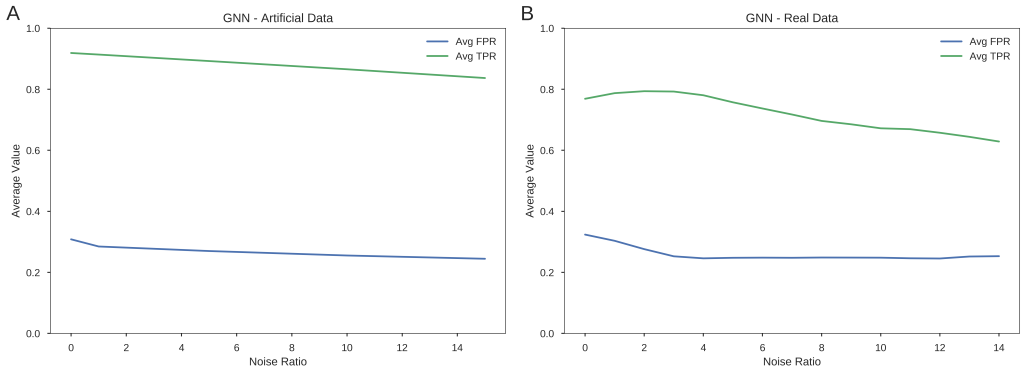

Figure B.3: Impact of noise on the TPR and FPR of the GNN in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

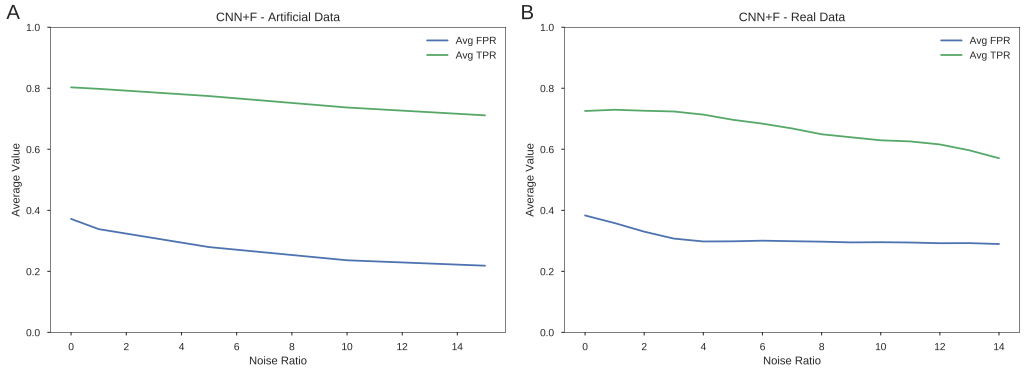

Figure B.4: Impact of noise on the TPR and FPR of the CNN+F in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

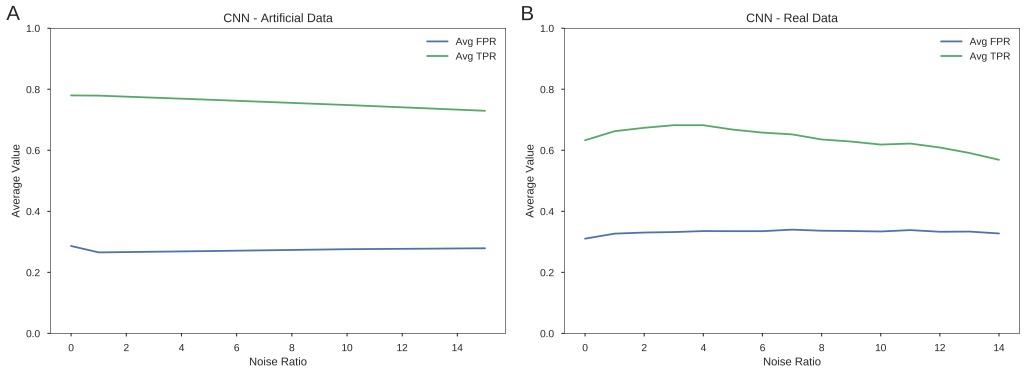

Figure B.5: Impact of noise on the TPR and FPR of the CNN in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

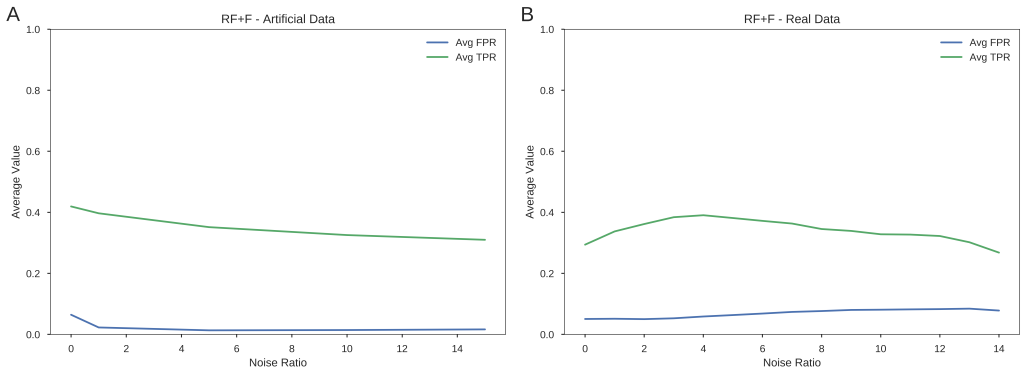

Figure B.6: Impact of noise on the TPR and FPR of the RF+F in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

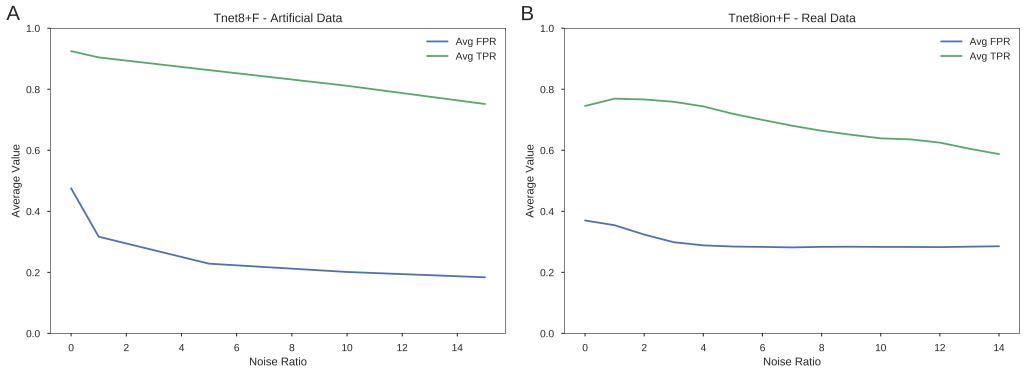

Figure B.7: Impact of noise on the TPR and FPR of the Tnet8+F in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

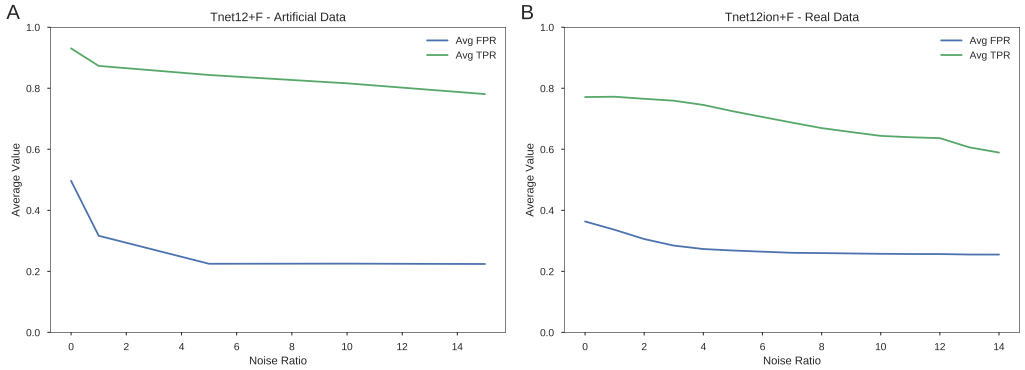

Figure B.8: Impact of noise on the TPR and FPR of the Tnet12+F in artificial and real data. The average TPR is shown in green and the average FPR is shown in blue.

Dataset	RF+F	CNN	CNN+F	Tnet8+F	Tnet12+F	GNN	GNN+F
Yeast	0.9231	0.9354	0.9525	0.9527	0.9557	0.9702	**0.9776**
Human	0.9127	0.9193	0.9454	0.9449	0.9467	0.9585	**0.9655**
Mouse	0.8638	0.8977	0.9168	0.9221	0.9277	0.9452	**0.9511**
Bacillus	0.9078	0.9258	0.9466	0.9407	0.9454	0.9660	**0.9725**
Clam Bacteria	0.8969	0.8996	0.9337	0.9299	0.9373	0.9453	**0.9568**
Honeybee	0.8958	0.9125	0.9393	0.9329	0.9396	0.9552	**0.9654**
Ricebean	0.9092	0.9205	0.9419	0.9361	0.9431	0.9646	**0.9719**
Tomato	0.9036	0.9267	0.9464	0.9484	0.9512	0.9726	**0.9729**
M. Mazei	0.9006	0.9230	0.9436	0.9415	0.9466	0.9673	**0.9713**

Table B.1: AUC values for each model on all 9 real datasets

B.1 Further Discussion on AUC

Appendix B Figure 2A shows the average FPR and TPR for the GNN+F as noise was increased in the artificial data. It shows a sharp decrease in the FPR when noise is initially increased. The additional noise gives "easy" to classify examples to the models, thereby increasing AUC. Further additional noise does not keep having the same magnitude of an effect on the FPR and it levels off. The TPR drops consistently as the noise ratio increases. Additional noise makes the prediction of the positive class more difficult leading to this decrease. The decreasing TPR dominates the trends in AUC as the FPR levels off leading to the observed decrease in AUC for ratios of additional noise greater than 1. Corresponding trends were found for the other algorithms (Appendix B Figures 3-8). A somewhat similar pattern was observed when investigating real data (Appendix B Figure 2B). Decreases in the FPR are present at low noise followed by a levelling off just like in the artificial data for almost every model (Appendix B Figures 3-8). Unlike the artificial data however, an increase in the TPR is observed at low noise ratios. This could be due to the aforementioned difference in the fraction of peptide peaks present for these data (Appendix B Figure 1A). These low noise data are also correlated with increased peptide length which the models find more difficult to successfully classify (Appendix B Figure 1B). For noise ratios above 5 when the fraction of peaks present stops increasing, TPR decreases just as it did in the artificial data for increasing noise. While these AUC results may not have been expected, the rank order of the models remained fairly consistent with that of average precision (Appendix B Table 1). This suggests AUC did a reasonable job at sorting the models based on competence. However, the analysis shows how care should be taken when comparing AUC across datasets, particularly if the class distribution is different between datasets.

C Supplementary Information (Ch. 5)

C.1 Estimating Random Matches

Alongside the method used to estimate the number of random matches in the main manuscript, we also investigated several others. The first of these involved the generation of a random peptide from amino acids not present in the database assigned peptide but of the same length (R_NoShare). This generates highly unlikely, non-tryptic peptides which give a lower bound estimate of how often fragment ions are assigned by chance. This is shown by the relatively lower values observed in Supplementary Table C.1.

The second method was the scrambling of database assigned peptide but maintaining the last amino acid (R_Scramble). This is a method used in the generation of decoy databases and maintains the same amino acid composition as the original set of peptides. With the same amino acids used, this may lead to an overestimation of the randomly matched internal fragments. Indeed this method matched the largest number of internal fragments of those used (Supplementary Table C.1). With many ions shared with the original peptide, this likely gives an upper bound to the number of randomly matched ions.

Finally, we also randomly shuffled the spectra while maintaining the original peptides to estimate the spurious matches (R_Spectrum). This method has the advantage of searching for the same theoretical fragment ions as the original search. Database assigned peptides were compared to the spectra of peptides of similar mass ($\Delta m < 25$Da) so that the fragment ions spanned the same m/z range as the observed ions.

Randomly generated tryptic peptides had a larger relative amount of y ion matches than non-tryptic peptides, even though the last amino acid in the sequences were different to those of the assigned peptides. This is partly explained by the frequency with which both arginine and lysine y_1 ions are matched in the spectra (Supplementary Table C.2). Despite the assigned peptides only ending with in R or K (excluding rare exceptions), y_1 ions for both amino acids were present in almost all spectra. Similar to Figure 5 in the main manuscript, this would indicate that more peptides are present in the spectra than those detected by the database search.

Ion Type	#R_NoShare	#R_NoShare / #Matched	#R_Scramble	#R_Scramble / #Matched	#R_Spectrum	#R_Spectrum / #Matched
Backbone	45373	5%	169940	18%	128288	14%
a	20765	13%	47786	31%	39176	25%
b	13146	5%	45971	17%	32020	12%
y	11462	2%	76183	15%	57092	11%
Charge 2+	24405	15%	103154	64%	84001	52%
a(2+)	9379	22%	22438	52%	20312	47%
b(2+)	7444	15%	18763	37%	17468	34%
y(2+)	7582	11%	24524	37%	22536	34%
Ion Loss	27124	5%	65725	13%	60316	12%
a-H20	1047	4%	8782	33%	6876	26%
b-H20	573	1%	7348	14%	4837	9%
y-H20	12580	6%	56065	26%	45880	21%
a-NH3	5436	14%	10324	26%	9647	24%
b-NH3	3465	5%	7694	12%	6839	10%
y-NH3	4023	4%	12941	13%	9923	10%
Int Frags	626938	38%	1075432	65%	957554	58%
b	274051	27%	562521	55%	485171	47%
a	352887	56%	512911	81%	472383	75%

Table C.1: Estimates of the number of randomly matched peaks of different ion types in a sample of 50,000 HCD PSMs with a matching tolerance of 0.05 Da. The data are from 9 different organisms and research groups, collated by Tran *et al.*. Columns indicate the number of ions from each method that were matched (#R_Type), and the ratio of the number of ions matched from the random peptides to the number of ions matched from the assigned peptides (#R_Type/#Matched). R_NoShare: Random sample of amino acids not present in assigned peptide, R_Scramble: Assigned peptides are scrambled while keeping the same last amino acid, R_Spectrum: Assigned peptides are compared to randomly selected spectra.

	K	R
Last AA in Assigned Peptide	28948	19557
y_1 Ion Matched in Spectrum	47957	45928

Table C.2: The number of arginine and lysine y_1 fragments matched in a sample of 50,000 HCD PSMs with a matching tolerance of 0.05 Da. The data are from 9 different organisms and research groups, collated by Tran *et al.*.

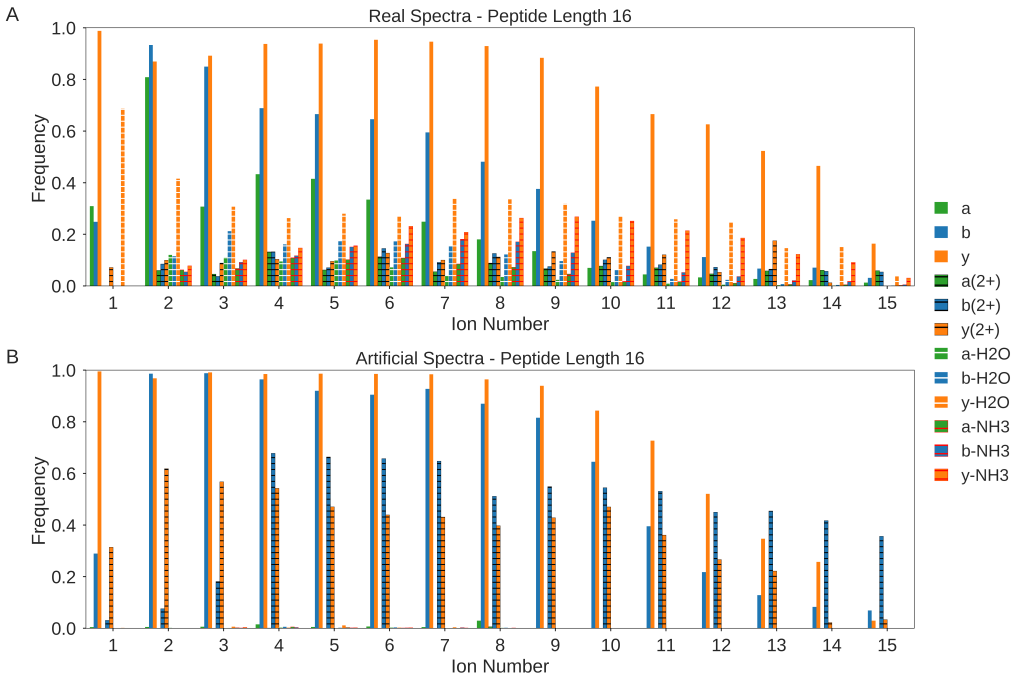

Figure C.1: Distribution of the presence of 12 different ion types in real and artificial spectra for length 16 peptides. Ions of the same type share the same base colour with different colour hatching indicating different charge states or neutral losses.

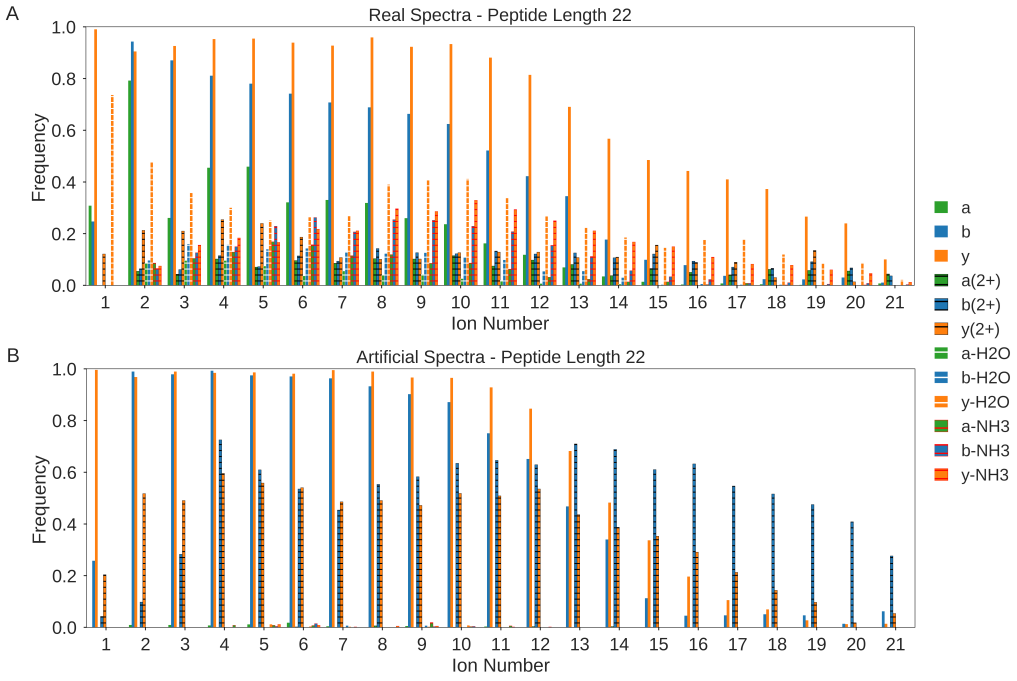

Figure C.2: Distribution of the presence of 12 different ion types in real and artificial spectra for length 22 peptides. Ions of the same type share the same base colour with different colour hatching indicating different charge states or neutral losses.

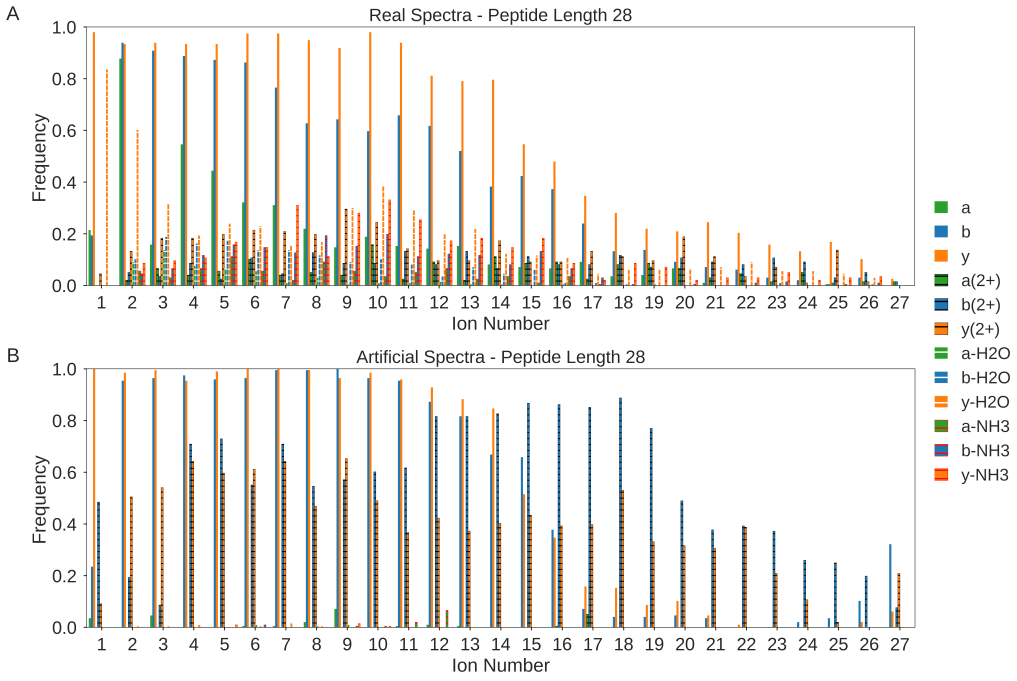

Figure C.3: Distribution of the presence of 12 different ion types in real and artificial spectra for length 28 peptides. Ions of the same type share the same base colour with different colour hatching indicating different charge states or neutral losses.

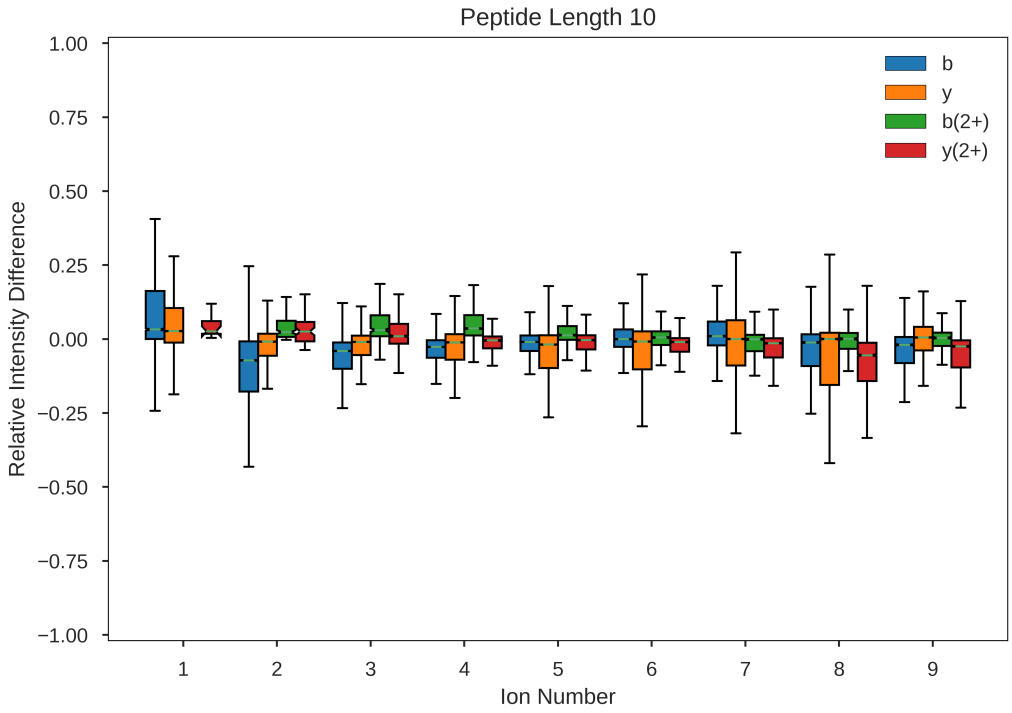

Figure C.4: Distribution of the difference in relative intensity predicted by Prosit and the observed value for length 10 peptides. All real intensities are normalised to the maximum fragment ion intensity matched.

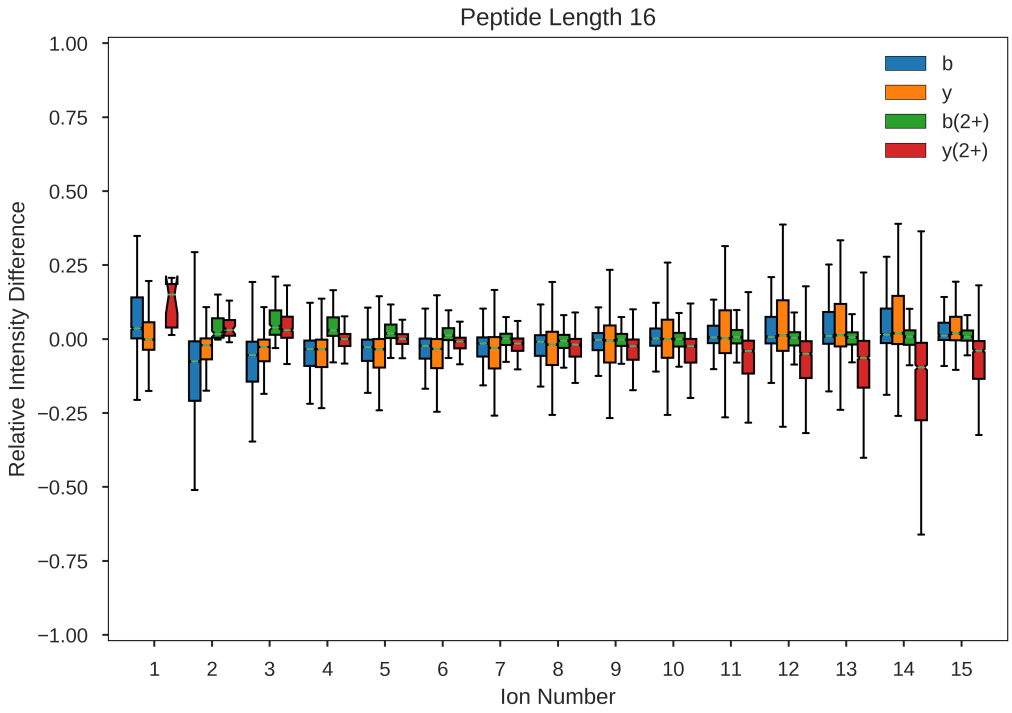

Figure C.5: Distribution of the difference in relative intensity predicted by Prosit and the observed value for length 16 peptides. All real intensities are normalised to the maximum fragment ion intensity matched.

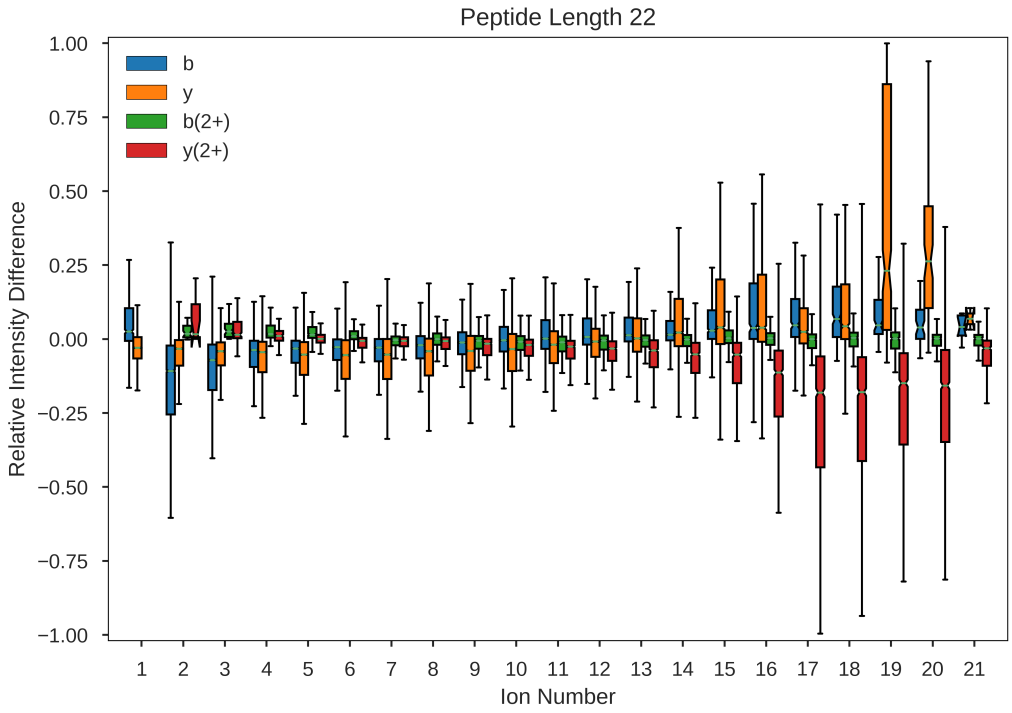

Figure C.6: Distribution of the difference in relative intensity predicted by Prosit and the observed value for length 22 peptides. All real intensities are normalised to the maximum fragment ion intensity matched.

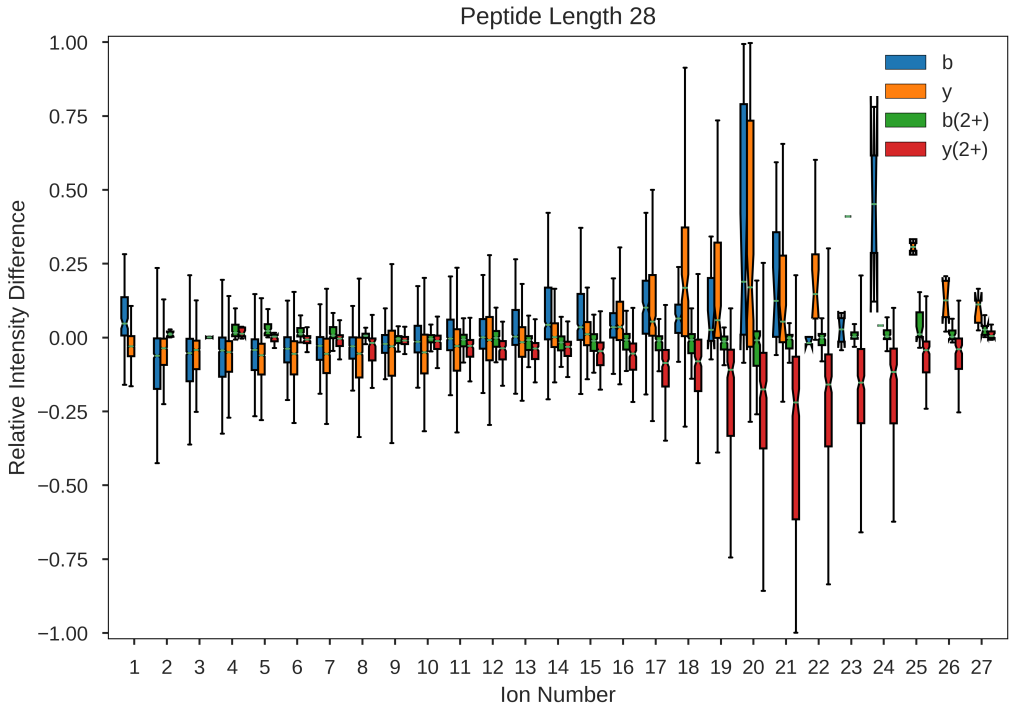

Figure C.7: Distribution of the difference in relative intensity predicted by Prosit and the observed value for length 28 peptides. All real intensities are normalised to the maximum fragment ion intensity matched.

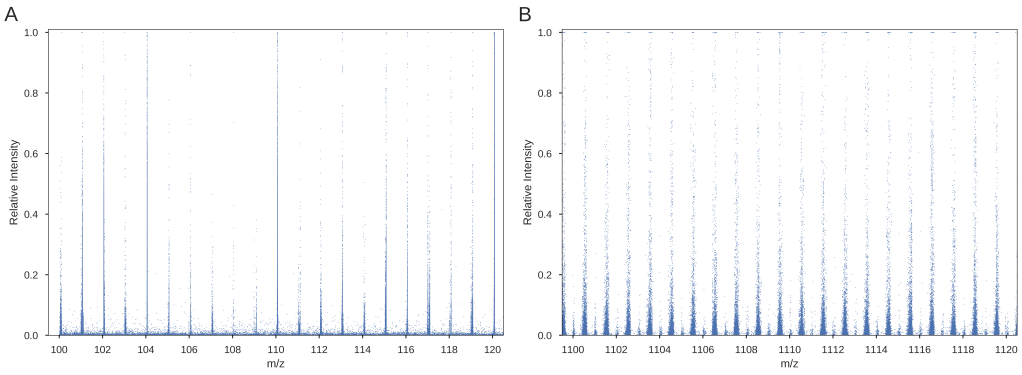

Figure C.8: Distribution of m/z values vs relative intensity values for peaks in a sample of 50,000 spectra. A shows peaks with m/z values between 100 and 120. B shows peaks with m/z values between 1100 and 1120.

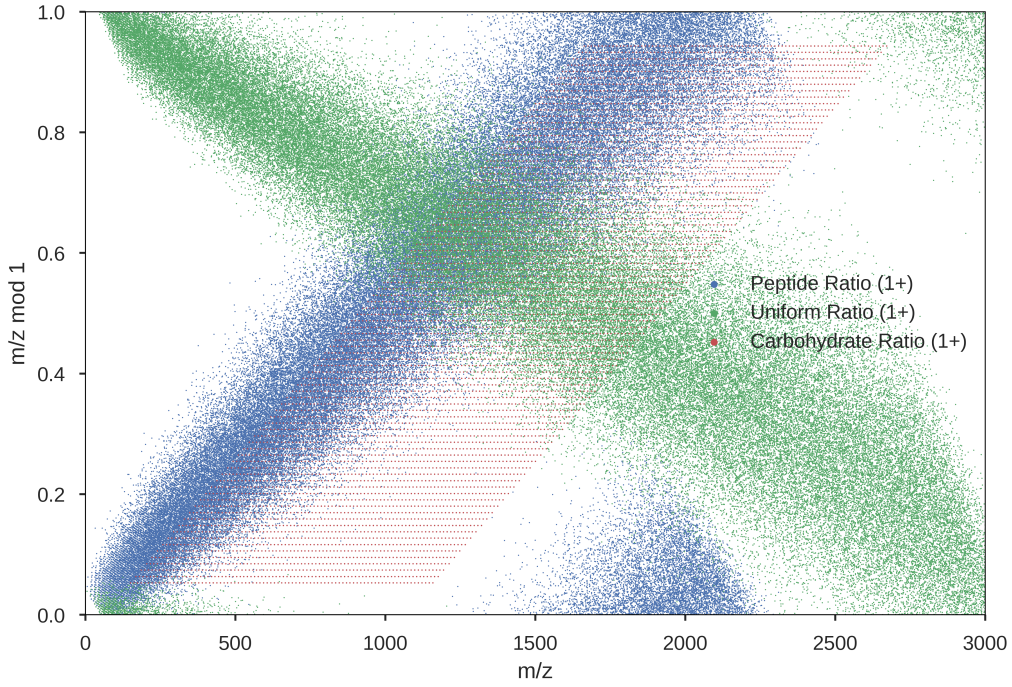

Figure C.9: Distribution of m/z values vs m/z modulo 1 for molecules with different ratios of hydrogen, carbon, nitrogen, oxygen and sulphur.

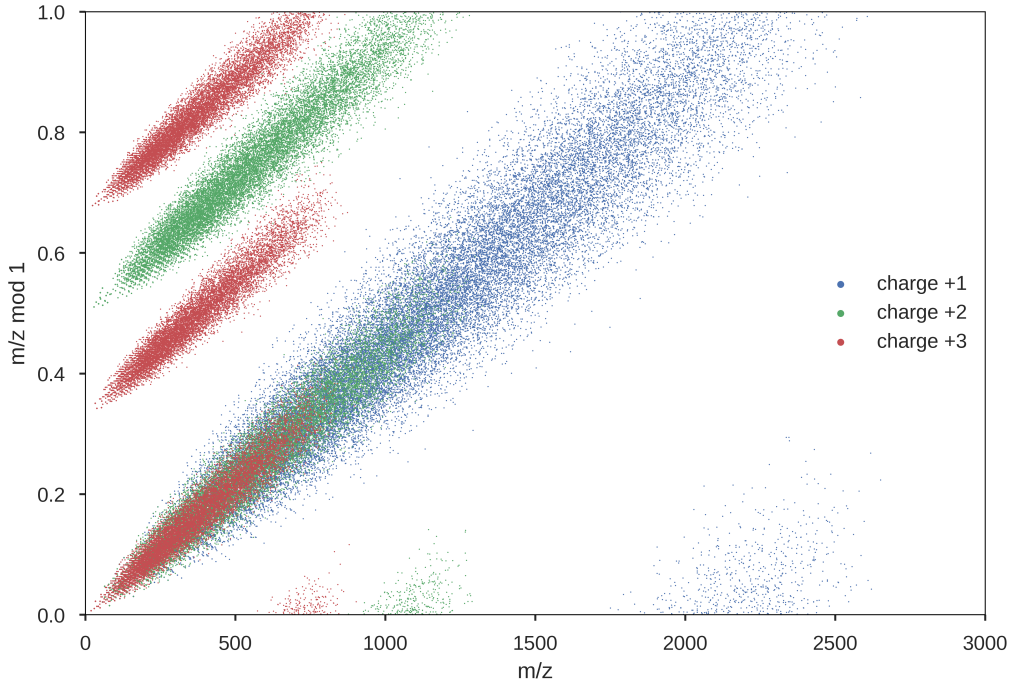

Figure C.10: Distribution of m/z values vs m/z modulo 1 for random peptide fragment peaks of different charges.

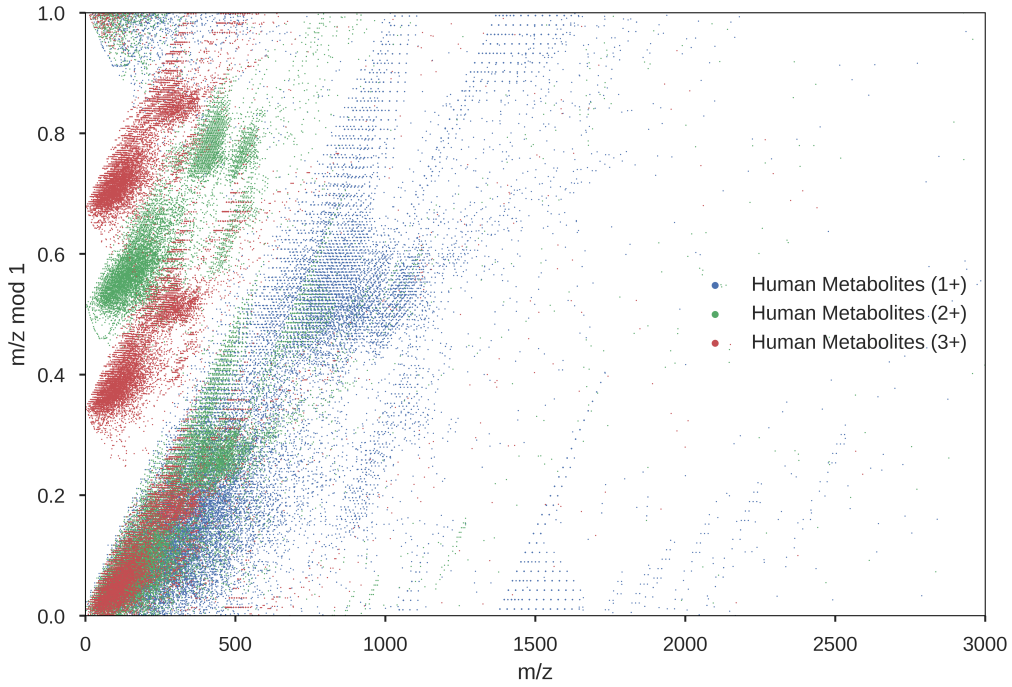

Figure C.11: Distribution of m/z values vs m/z modulo 1 for human metabolites of different charges.

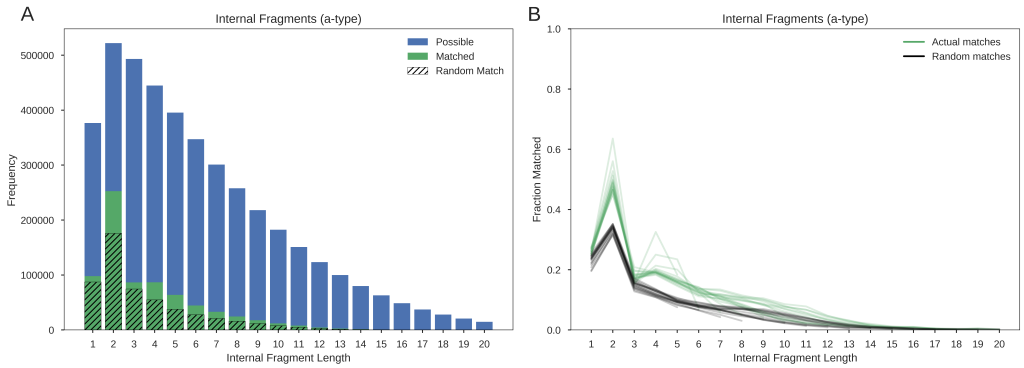

Figure C.12: The number of a-type internal fragments matched by length. A shows the counts of possible unique internal fragment masses (blue), matched internal masses (green), matched random internal masses (black hatch). B shows the fraction of the total number of possible internal fragments matched by the actual peptides (green) and the random peptides (black). Each individual line represents the different peptide lengths.

Bibliography

[1] Abram, F. Systems-based approaches to unravel multi-species microbial community functioning. *Computational and structural biotechnology journal 13* (2015), 24–32.

[2] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., and San Tan, R. A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine 89* (2017), 389–396.

[3] Achuthan, S., Chatterjee, R., Kotnala, S., Mohanty, A., Bhattacharya, S., Salgia, R., and Kulkarni, P. Leveraging deep learning algorithms for synthetic data generation to design and analyze biological networks. *Journal of Biosciences 47*, 3 (2022), 1–11.

[4] Adams, C., and Rinne, R. Strees protein formation: Gene expression and environmental interaction with evolutionary significance. *International Review of Cytology 79* (1982), 305–315.

[5] Ahrné, E., Müller, M., and Lisacek, F. Unrestricted identification of modified proteins using ms/ms. *Proteomics 10*, 4 (2010), 671–686.

[6] Al-Saleh, A., Alazzoni, A., Al Shalash, S., Ye, C., Mbuagbaw, L., Thabane, L., and Jolly, S. S. Performance of the high-sensitivity troponin assay in diagnosing acute myocardial infarction: systematic review and meta-analysis. *Canadian Medical Association Open Access Journal 2*, 3 (2014), E199–E207.

[7] Allmer, J. Algorithms for the de novo sequencing of peptides from tandem mass spectra. *Expert review of proteomics 8*, 5 (2011), 645–657.

[8] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology 215*, 3 (1990), 403–410.

[9] Alvarez, S., Roy Choudhury, S., and Pandey, S. Comparative quantitative proteomics analysis of the aba response of roots of drought-sensitive and drought-tolerant wheat varieties identifies proteomic signatures of drought adaptability. *Journal of proteome research 13*, 3 (2014), 1688–1701.

[10] Andreev, V. P., Rejtar, T., Chen, H.-S., Moskovets, E. V., Ivanov, A. R., and Karger, B. L. A universal denoising and peak picking algorithm for lc-ms based on matched filtration in the chromatographic time domain. *Analytical chemistry 75*, 22 (2003), 6314–6326.

[11] Anfinsen, C. B. Principles that govern the folding of protein chains. *Science 181*, 4096 (1973), 223–230.

[12] Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research 32*, suppl_1 (2004), D115–D119.

[13] Arnold, R. J., Jayasankar, N., Aggarwal, D., Tang, H., and Radivojac, P. A machine learning approach to predicting peptide fragmentation spectra. In *Biocomputing 2006*. World Scientific, 2006, pp. 219–230.

[14] Aslam, B., Basit, M., Nisar, M. A., Khurshid, M., and Rasool, M. H. Proteomics: technologies and their applications. *Journal of chromatographic science 55*, 2 (2017), 182–196.

[15] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[16] Balles, L., Romero, J., and Hennig, P. Coupling adaptive batch sizes with learning rates. *arXiv preprint arXiv:1612.05086* (2016).

[17] Barsnes, H., and Vaudel, M. Searchgui: a highly adaptable common interface for proteomics search and de novo engines. *Journal of proteome research 17*, 7 (2018), 2552–2555.

[18] Bartels, C. Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical & environmental mass spectrometry 19*, 6 (1990), 363–368.

[19] Bassani-Sternberg, M., Bräunlein, E., Klar, R., Engleitner, T., Sinitcyn, P., Audehm, S., Straub, M., Weber, J., Slotta-Huspenina, J., Specht, K., et al. Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry. *Nature communications 7*, 1 (2016), 1–16.

[20] Behjati, R., Arisholm, E., Bedregal, M., and Tan, C. Synthetic test data generation using recurrent neural networks: a position paper. In *2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)* (2019), IEEE, pp. 22–27.

[21] Belanche, L. A., and González, F. F. Review and evaluation of feature selection algorithms in synthetic problems. *arXiv preprint arXiv:1101.2320* (2011).

[22] Bellman, R. E., and Dreyfus, S. E. *Applied dynamic programming*, vol. 2050. Princeton university press, 2015.

[23] Bennett, J., Lanning, S., et al. The netflix prize. In *Proceedings of KDD cup and workshop* (2007), vol. 2007, New York, p. 35.

[24] Berk, R. A. An introduction to sample selection bias in sociological data. *American sociological review* (1983), 386–398.

[25] Bern, M., and Goldberg, D. De novo analysis of peptide tandem mass spectra by spectral graph partitioning. *Journal of Computational Biology 13*, 2 (2006), 364–378.

[26] Bessarabova, M., Ishkin, A., JeBailey, L., Nikolskaya, T., and Nikolsky, Y. Knowledge-based analysis of proteomics data. *BMC bioinformatics 13*, 16 (2012), 1–19.

[27] Biemann, K. Mass spectrometry. *Annual review of biochemistry 32*, 1 (1963), 755–780.

[28] Bishop, C. M. Neural networks and their applications. *Review of scientific instruments 65*, 6 (1994), 1803–1832.

[29] Biswas, S., and Rolain, J.-M. Use of maldi-tof mass spectrometry for identification of bacteria that are difficult to culture. *Journal of microbiological methods 92*, 1 (2013), 14–24.

[30] Boja, E. S., and Fales, H. M. Overalkylation of a protein digest with iodoacetamide. *Analytical chemistry 73*, 15 (2001), 3576–3582.

[31] Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowledge and information systems 34* (2013), 483–519.

[32] Bottou, L. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[33] Bottou, L., and Bousquet, O. The tradeoffs of large scale learning. *Advances in neural information processing systems 20* (2007).

[34] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863* (2018).

[35] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[36] Brenton, A. G., and Godfrey, A. R. Accurate mass measurement: terminology and treatment of data. *Journal of the American Society for Mass Spectrometry 21*, 11 (2010), 1821–1835.

[37] Bringans, S., Kendrick, T. S., Lui, J., and Lipscombe, R. A comparative study of the accuracy of several de novo sequencing software packages for datasets derived by matrix-assisted laser desorption/ionisation and electrospray. *Rapid Communications in Mass Spectrometry: An International Journal Devoted to the Rapid Dissemination of Up-to-the-Minute Research in Mass Spectrometry 22*, 21 (2008), 3450–3454.

[38] Brown, K. A., Melby, J. A., Roberts, D. S., and Ge, Y. Top-down proteomics: challenges, innovations, and applications in basic and clinical research. *Expert review of proteomics 17*, 10 (2020), 719–733.

[39] Brown, W. M., Gedeon, T. D., and Groves, D. I. Use of noise to augment training data: a neural network method of mineral–potential mapping in regions of limited known deposit examples. *Natural Resources Research 12* (2003), 141–152.

[40] Caiafa, C. F., Sun, Z., Tanaka, T., Marti-Puig, P., and Solé-Casals, J. Machine learning methods with noisy, incomplete or small datasets, 2021.

[41] Cassidy, L., Prasse, D., Linke, D., Schmitz, R. A., and Tholey, A. Combination of bottom-up 2d-lc-ms and semi-top-down gelfree-lc-ms enhances coverage of proteome and low molecular weight short open reading frame encoded peptides of the archaeon methanosarcina mazei. *Journal of proteome research 15*, 10 (2016), 3773–3783.

[42] Chen, P.-H. C., Liu, Y., and Peng, L. How to develop machine learning models for healthcare. *Nature materials 18*, 5 (2019), 410–414.

[43] Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F., and Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering 5*, 6 (2021), 493–497.

[44] Chen, T., Kao, M.-Y., Tepel, M., Rush, J., and Church, G. M. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology 8*, 3 (2001), 325–337.

[45] Chen, Y., Chen, W., Cobb, M. H., and Zhao, Y. Ptmap—a sequence alignment software for unrestricted, accurate, and full-spectrum identification of post-translational modification sites. *Proceedings of the National Academy of Sciences 106*, 3 (2009), 761–766.

[46] Cho, W. Contribution of oncoproteomics to cancer biomarker discovery. *Molecular cancer 6*, 1 (2007), 1–13.

[47] Chouldechova, A., and Roth, A. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).

[48] Claesen, M., and De Moor, B. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127* (2015).

[49] Cohen, A. A., Geva-Zatorsky, N., Eden, E., Frenkel-Morgenstern, M., Issaeva, I., Sigal, A., Milo, R., Cohen-Saidon, C., Liron, Y., Kam, Z., et al. Dynamic proteomics of individual cancer cells in response to a drug. *science 322*, 5907 (2008), 1511–1516.

[50] Consortium, U. Uniprot: a hub for protein information. *Nucleic acids research 43*, D1 (2015), D204–D212.

[51] Cottrell, J. S. Protein identification using ms/ms data. *Journal of proteomics 74*, 10 (2011), 1842–1851.

[52] Craig, R., and Beavis, R. C. Tandem: matching proteins with tandem mass spectra. *Bioinformatics 20*, 9 (2004), 1466–1467.

[53] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. Generative adversarial networks: An overview. *IEEE signal processing magazine 35*, 1 (2018), 53–65.

[54] Cypryk, W., Lorey, M., Puustinen, A., Nyman, T. A., and Matikainen, S. Proteomic and bioinformatic characterization of extracellular vesicles released from human macrophages upon influenza a virus infection. *Journal of Proteome Research 16*, 1 (2017), 217–227.

[55] Dahmen, J., and Cook, D. Synsys: A synthetic data generation system for healthcare applications. *Sensors 19*, 5 (2019), 1181.

[56] Dančík, V., Addona, T. A., Clauser, K. R., Vath, J. E., and Pevzner, P. A. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology 6*, 3-4 (1999), 327–342.

[57] Dančík, V., Addona, T. A., Clauser, K. R., Vath, J. E., and Pevzner, P. A. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology 6*, 3-4 (1999), 327–342.

[58] Danilova, Y., Voronkova, A., Sulimov, P., and Kertész-Farkas, A. Bias in false discovery rate estimation in mass-spectrometry-based peptide identification. *Journal of proteome research 18*, 5 (2019), 2354–2358.

[59] Davis, J., and Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 233–240.

[60] Degroeve, S., and Martens, L. Ms2pip: a tool for ms/ms peak intensity prediction. *Bioinformatics 29*, 24 (2013), 3199–3203.

[61] Di Carli, M., Benvenuto, E., and Donini, M. Recent insights into plant–virus interactions through proteomic analysis. *Journal of proteome research 11*, 10 (2012), 4765–4780.

[62] Di Narzo, A. F., Telesco, S. E., Brodmerkel, C., Argmann, C., Peters, L. A., Li, K., Kidd, B., Dudley, J., Cho, J., Schadt, E. E., et al. High-throughput characterization of blood serum proteomics of ibd patients with respect to aging and genetic factors. *PLoS genetics 13*, 1 (2017), e1006565.

[63] Diedrich, J. K., Pinto, A. F., and Yates III, J. R. Energy dependence of hcd on peptide fragmentation: stepped collisional energy finds the sweet spot. *Journal of the American Society for Mass Spectrometry 24*, 11 (2013), 1690–1699.

[64] Ding, J., Shi, J., Poirier, G. G., and Wu, F.-X. A novel approach to denoising ion trap tandem mass spectra. *Proteome Science 7*, 1 (2009), 1–10.

[65] Domingos, P. A unified bias-variance decomposition. In *Proceedings of 17th international conference on machine learning* (2000), Morgan Kaufmann Stanford, pp. 231–238.

[66] Elias, J. E., Gibbons, F. D., King, O. D., Roth, F. P., and Gygi, S. P. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature biotechnology 22*, 2 (2004), 214–219.

[67] 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), 207–214.

[68] Elman, J. L. Finding structure in time. *Cognitive science 14*, 2 (1990), 179–211.

[69] Eng, J. K., McCormack, A. L., and Yates, 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*, 11 (1994), 976–989.

[70] Erhard, F., Dölken, L., Schilling, B., and Schlosser, A. Identification of the cryptic hla-i immunopeptidome. *Cancer immunology research 8*, 8 (2020), 1018–1026.

[71] Fannes, T., Vandermarliere, E., Schietgat, L., Degroeve, S., Martens, L., and Ramon, J. Predicting tryptic cleavage from proteomics data using decision tree ensembles. *Journal of proteome research 12*, 5 (2013), 2253–2259.

[72] Fenyö, D., and Beavis, R. C. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical chemistry 75*, 4 (2003), 768–774.

[73] Finehout, E. J., and Lee, K. H. An introduction to mass spectrometry applications in biological research. *Biochemistry and molecular biology Education 32*, 2 (2004), 93–100.

[74] Fischer, B., Grossmann, J., Roth, V., Gruissem, W., Baginsky, S., and Buhmann, J. M. Semi-supervised lc/ms alignment for differential proteomics. *Bioinformatics 22*, 14 (2006), e132–e140.

[75] Fischer, B., Roth, V., Roos, F., Grossmann, J., Baginsky, S., Widmayer, P., Gruissem, W., and Buhmann, J. M. Novohmm: a hidden markov model for de novo peptide sequencing. *Analytical chemistry 77*, 22 (2005), 7265–7273.

[76] Forney, G. D. The viterbi algorithm. *Proceedings of the IEEE 61*, 3 (1973), 268–278.

[77] Fout, A., Byrd, J., Shariat, B., and Ben-Hur, A. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems 30* (2017).

[78] Frank, A., and Pevzner, P. Pepnovo: de novo peptide sequencing via probabilistic network modeling. *Analytical chemistry 77*, 4 (2005), 964–973.

[79] Frank, A., Tanner, S., Bafna, V., and Pevzner, P. Peptide sequence tags for fast database search in mass-spectrometry. *Journal of proteome research 4*, 4 (2005), 1287–1295.

[80] Frank, A. M., Monroe, M. E., Shah, A. R., Carver, J. J., Bandeira, N., Moore, R. J., Anderson, G. A., Smith, R. D., and Pevzner, P. A. Spectral archives: extending spectral libraries to analyze both identified and unidentified spectra. *Nature methods 8*, 7 (2011), 587–591.

[81] Frank, A. M., Savitski, M. M., Nielsen, M. L., Zubarev, R. A., and Pevzner, P. A. De novo peptide sequencing and identification with precision mass spectrometry. *Journal of proteome research 6*, 1 (2007), 114–123.

[82] Frauenfelder, H., and McMahon, B. Dynamics and function of proteins: the search for general concepts. *Proceedings of the National Academy of Sciences 95*, 9 (1998), 4795–4797.

[83] Freeder, S., Lenz, G. S., and Turney, S. The importance of knowing "what goes with what": Reinterpreting the evidence on policy attitude stability. *The Journal of Politics 81*, 1 (2019), 274–290.

[84] Friedman, J., and Popescu, B. E. Gradient directed regularization for linear regression and classification. Tech. rep., Citeseer, 2003.

[85] Fu, K., Cheng, D., Tu, Y., and Zhang, L. Credit card fraud detection using convolutional neural networks. In *International conference on neural information processing* (2016), Springer, pp. 483–490.

[86] Fukushima, K., and Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[87] Garcia-Garijo, A., Fajardo, C. A., and Gros, A. Determinants for neoantigen identification. *Frontiers in immunology 10* (2019), 1392.

[88] Gay, S., Binz, P.-A., Hochstrasser, D. F., and Appel, R. D. Modeling peptide mass fingerprinting data using the atomic composition of peptides. *ELECTROPHORESIS: An International Journal 20*, 18 (1999), 3527–3534.

[89] Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods 16*, 6 (2019), 509–518.

[90] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning* (2017), PMLR, pp. 1263–1272.

[91] Gilmore, J. M., and Washburn, M. P. Advances in shotgun proteomics and the analysis of membrane proteomes. *Journal of proteomics 73*, 11 (2010), 2078–2091.

[92] Goloborodko, A. A., Gorshkov, M. V., Good, D. M., and Zubarev, R. A. Sequence scrambling in shotgun proteomics is negligible. *Journal of the American Society for Mass Spectrometry 22*, 7 (2011), 1121–1124.

[93] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[94] Gottesman, S. Trouble is coming: Signaling pathways that regulate general stress responses in bacteria. *Journal of Biological Chemistry 294*, 31 (2019), 11685–11700.

[95] Griss, J., Perez-Riverol, Y., Lewis, S., Tabb, D. L., Dianes, J. A., Del-Toro, N., Rurik, M., Walzer, M., Kohlbacher, O., Hermjakob, H., et al. Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets. *Nature methods 13*, 8 (2016), 651–656.

[96] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. Recent advances in convolutional neural networks. *Pattern recognition 77* (2018), 354–377.

[97] Gupta, S., and Gupta, A. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science 161* (2019), 466–474.

[98] Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems 30* (2017).

[99] Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 14*, 3 (2020), 1–159.

[100] Han, X., Hu, Y., Foschini, L., Chinitz, L., Jankelson, L., and Ranganath, R. Deep learning models for electrocardiograms are susceptible to adversarial attack. *Nature medicine 26*, 3 (2020), 360–363.

[101] Han, Y., Ma, B., and Zhang, K. Spider: software for protein identification from sequence tags with de novo sequencing error. In *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.* (2004), IEEE, pp. 206–215.

[102] Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., and Asadi, H. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology 212*, 1 (2019), 38–43.

[103] Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems 29* (2016).

[104] Hartigan, J. A., and Hartigan, J. *Clustering algorithms*, vol. 209. Wiley New York, 1975.

[105] He, F., Liu, T., and Tao, D. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. *Advances in Neural Information Processing Systems 32* (2019).

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

[107] Heyer, R., Schallert, K., Zoun, R., Becher, B., Saake, G., and Benndorf, D. Challenges and perspectives of metaproteomic data analysis. *Journal of biotechnology 261* (2017), 24–36.

[108] Hickey, R. J. Noise modelling and evaluating learning from examples. *Artificial Intelligence 82*, 1-2 (1996), 157–179.

[109] Hinton, G. E., Krizhevsky, A., and Sutskever, I. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25*, 1106-1114 (2012), 1.

[110] Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (1995), vol. 1, IEEE, pp. 278–282.

[111] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6*, 02 (1998), 107–116.

[112] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[113] Hoopmann, M. R., and Moritz, R. L. Current algorithmic solutions for peptide-based proteomics data generation and identification. *Current opinion in biotechnology 24*, 1 (2013), 31–38.

[114] Hristova, V. A., and Chan, D. W. Cancer biomarker discovery and translation: proteomics and beyond. *Expert review of proteomics 16*, 2 (2019), 93–103.

[115] Hu, H., Bienefeld, K., Wegener, J., Zautke, F., Hao, Y., Feng, M., Han, B., Fang, Y., Wubie, A. J., and Li, J. Proteome analysis of the hemolymph, mushroom body, and antenna provides novel insight into honeybee resistance against varroa infestation. *Journal of proteome research 15*, 8 (2016), 2841–2854.

[116] Huang, L., Jacob, R. J., Pegg, S. C.-H., Baldwin, M. A., Wang, C. C., Burlingame, A. L., and Babbitt, P. C. Functional assignment of the 20 s proteasome from trypanosoma brucei using mass spectrometry and new bioinformatics approaches. *Journal of Biological Chemistry 276*, 30 (2001), 28327–28339.

[117] Huang, S., and Ramanan, D. Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 2243–2252.

[118] Huang, T., Wang, J., Yu, W., and He, Z. Protein inference: a review. *Briefings in bioinformatics 13*, 5 (2012), 586–614.

[119] Huang, Y., Triscari, J. M., Tseng, G. C., Pasa-Tolic, L., Lipton, M. S., Smith, R. D., and Wysocki, V. H. Statistical characterization of the charge state and residue dependence of low-energy cid peptide dissociation patterns. *Analytical chemistry 77*, 18 (2005), 5800–5813.

[120] Hunt, D. F., Yates 3rd, J., Shabanowitz, J., Winston, S., and Hauer, C. R. Protein sequencing by tandem mass spectrometry. *Proceedings of the National Academy of Sciences 83*, 17 (1986), 6233–6237.

[121] Jeong, K., Kim, S., and Bandeira, N. False discovery rates in spectral identification. *BMC bioinformatics 13*, 16 (2012), 1–15.

[122] Johnson, J. V., and Yost, R. A. Tandem mass spectrometry for trace analysis. *Analytical Chemistry 57*, 7 (1985), 758A–768A.

[123] Johnson, R. S., Martin, S. A., and Biemann, K. Collision-induced fragmentation of (m+ h)+ ions of peptides. side chain specific sequence ions. *International Journal of Mass Spectrometry and Ion Processes 86* (1988), 137–154.

[124] Jones, P., Côté, R. G., Martens, L., Quinn, A. F., Taylor, C. F., Derache, W., Hermjakob, H., and Apweiler, R. Pride: a public repository of

protein and peptide identifications for the proteomics community. *Nucleic acids research 34*, suppl_1 (2006), D659–D663.

[125] Jordan, M. I., and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science 349*, 6245 (2015), 255–260.

[126] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature 596*, 7873 (2021), 583–589.

[127] Kalaora, S., Barnea, E., Merhavi-Shoham, E., Qutob, N., Teer, J. K., Shimony, N., Schachter, J., Rosenberg, S. A., Besser, M. J., Admon, A., et al. Use of hla peptidomics and whole exome sequencing to identify human immunogenic neo-antigens. *Oncotarget 7*, 5 (2016), 5110.

[128] Käll, L., Canterbury, J. D., Weston, J., Noble, W. S., and MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature methods 4*, 11 (2007), 923–925.

[129] Kelchtermans, P., Bittremieux, W., De Grave, K., Degroeve, S., Ramon, J., Laukens, K., Valkenborg, D., Barsnes, H., and Martens, L. Machine learning applications in proteomics research: how the past can boost the future. *Proteomics 14*, 4-5 (2014), 353–366.

[130] Keller, A., Nesvizhskii, A. I., Kolker, E., and Aebersold, R. Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Analytical chemistry 74*, 20 (2002), 5383–5392.

[131] Keller, J. M., Gray, M. R., and Givens, J. A. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics SMC-15*, 4 (1985), 580–585.

[132] Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 9012–9020.

[133] Kim, S., Gupta, N., Bandeira, N., and Pevzner, P. A. Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. *Molecular & Cellular Proteomics 8*, 1 (2009), 53–69.

[134] Kim, S., and Pevzner, P. A. Ms-gf+ makes progress towards a universal database search tool for proteomics. *Nature communications 5*, 1 (2014), 1–10.

[135] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[136] Kohavi, R., Wolpert, D. H., et al. Bias plus variance decomposition for zero-one loss functions. In *ICML* (1996), vol. 96, pp. 275–83.

[137] Lawrence, M. S., Stojanov, P., Mermel, C. H., Robinson, J. T., Garraway, L. A., Golub, T. R., Meyerson, M., Gabriel, S. B., Lander, E. S., and Getz, G. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature 505*, 7484 (2014), 495–501.

[138] Lebedev, A., Westman, E., Van Westen, G., Kramberger, M., Lundervold, A., Aarsland, D., Soininen, H., Kłoszewska, I., Mecocci, P., Tsolaki, M., et al. Random forest ensembles for detection and prediction of alzheimer's disease with a good between-cohort robustness. *NeuroImage: Clinical 6* (2014), 115–125.

[139] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation 1*, 4 (1989), 541–551.

[140] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[141] Levitsky, L. I., Klein, J. A., Ivanov, M. V., and Gorshkov, M. V. Pyteomics 4.0: five years of development of a python proteomics framework. *Journal of proteome research 18*, 2 (2018), 709–714.

[142] Li, G., He, X., Zhang, L., Ran, Q., Wang, J., Xiong, A., Wu, D., Chen, F., Sun, J., and Chang, C. Assessing ace2 expression patterns in lung tissues in the pathogenesis of covid-19. *Journal of autoimmunity 112* (2020), 102463.

[143] Li, J., Guo, M., Tian, X., Liu, C., Wang, X., Yang, X., Wu, P., Xiao, Z., Qu, Y., Yin, Y., et al. Virus-host interactome and proteomic survey of pmbcs from covid-19 patients reveal potential virulence factors influencing sars-cov-2 pathogenesis. *BioRxiv* (2020).

[144] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. Machine learning in agriculture: A review. *Sensors 18*, 8 (2018), 2674.

[145] Liashchynskyi, P., and Liashchynskyi, P. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059* (2019).

[146] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2980–2988.

[147] Lindsey, M. L., Mayr, M., Gomes, A. V., Delles, C., Arrell, D. K., Murphy, A. M., Lange, R. A., Costello, C. E., Jin, Y.-F., Laskowitz, D. T., et al. Transformative impact of proteomics on cardiovascular health and disease: a scientific statement from the american heart association. *Circulation 132*, 9 (2015), 852–872.

[148] Liu, K., Li, S., Wang, L., Ye, Y., and Tang, H. Full-spectrum prediction of peptides tandem mass spectra using deep neural network. *Analytical chemistry 92*, 6 (2020), 4275–4283.

[149] Lodish, H. F., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D., and Darnell, J. *Molecular cell biology*, vol. 4. WH Freeman and company New York, 2006.

[150] López-Ferrer, D., Martínez-Bartolomé, S., Villar, M., Campillos, M., Martín-Maroto, F., and Vázquez, J. Statistical model for large-scale peptide identification in databases from tandem mass spectra using sequest. *Analytical Chemistry 76*, 23 (2004), 6853–6860.

[151] Lothrop, A. P., Torres, M. P., and Fuchs, S. M. Deciphering post-translational modification codes. *FEBS letters 587*, 8 (2013), 1247–1257.

[152] Lu, B., and Chen, T. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *Drug Discovery Today: BioSilico 2*, 2 (2004), 85–90.

[153] Ma, B. Novor: real-time peptide de novo sequencing software. *Journal of the American Society for Mass Spectrometry 26*, 11 (2015), 1885–1894.

[154] Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (2013), vol. 30, Atlanta, Georgia, USA, p. 3.

[155] Mackey, A. J., Haystead, T. A., and Pearson, W. R. Getting more from less: algorithms for rapid protein identification with multiple short peptide sequences. *Molecular & Cellular Proteomics 1*, 2 (2002), 139–147.

[156] Marshall, K. D., Edwards, M. A., Krenz, M., Davis, J. W., and Baines, C. P. Proteomic mapping of proteins released during necrosis and apoptosis from

cultured neonatal cardiac myocytes. *American Journal of Physiology-Cell Physiology 306*, 7 (2014), C639–C647.

[157] Martens, L., Hermjakob, H., Jones, P., Adamski, M., Taylor, C., States, D., Gevaert, K., Vandekerckhove, J., and Apweiler, R. Pride: the proteomics identifications database. *Proteomics 5*, 13 (2005), 3537–3545.

[158] Mason, S. J., and Graham, N. E. Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 128*, 584 (2002), 2145–2166.

[159] Mata, C. I., Fabre, B., Hertog, M. L., Parsons, H. T., Deery, M. J., Lilley, K. S., and Nicolai, B. M. In-depth characterization of the tomato fruit pericarp proteome. *Proteomics 17*, 1-2 (2017), 1600406.

[160] McDonnell, K., Abram, F., and Howley, E. Application of a novel hybrid cnn-gnn for peptide ion encoding. *Journal of Proteome Research* (2022).

[161] McDonnell, K., Howley, E., and Abram, F. The impact of noise and missing fragmentation cleavages on de novo peptide identification algorithms. *Computational and Structural Biotechnology Journal 20* (2022), 1402–1412.

[162] McDonnell, K., Howley, E., and Abram, F. Critical evaluation of the use of artificial data for machine learning based de novo peptide identification. *Computational and Structural Biotechnology Journal* (2023).

[163] McDonnell, K., Waters, N., Howley, E., and Abram, F. Chordomics: a visualization tool for linking function to phylogeny in microbiomes. *Bioinformatics 36*, 4 (2020), 1309–1310.

[164] McLafferty, F. W. Tandem mass spectrometry. *Science 214*, 4518 (1981), 280–287.

[165] Medzihradszky, K. F. Peptide sequence analysis. *Methods in enzymology 402* (2005), 209–244.

[166] Medzihradszky, K. F., and Chalkley, R. J. Lessons in de novo peptide sequencing by tandem mass spectrometry. *Mass spectrometry reviews 34*, 1 (2015), 43–63.

[167] Melville, P., and Mooney, R. J. Creating diversity in ensembles using artificial data. *Information Fusion 6*, 1 (2005), 99–111.

[168] Michalski, A., Neuhauser, N., Cox, J., and Mann, M. A systematic investigation into the nature of tryptic hcd spectra. *Journal of proteome research 11*, 11 (2012), 5479–5491.

[169] Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., and Waibel, A. Machine learning. *Annual review of computer science 4*, 1 (1990), 417–433.

[170] Mitchell, T. M., and Mitchell, T. M. *Machine learning*, vol. 1. McGraw-hill New York, 1997.

[171] Mo, L., Dutta, D., Wan, Y., and Chen, T. Msnovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry. *Analytical chemistry 79*, 13 (2007), 4870–4878.

[172] Monti, F., Bronstein, M., and Bresson, X. Geometric matrix completion with recurrent multi-graph neural networks. *Advances in neural information processing systems 30* (2017).

[173] Mujezinovic, N., Schneider, G., Wildpaner, M., Mechtler, K., and Eisenhaber, F. Reducing the haystack to find the needle: improved protein identification after fast elimination of non-interpretable peptide ms/ms spectra and noise reduction. *BMC genomics 11*, 1 (2010), 1–8.

[174] Muth, T., Benndorf, D., Reichl, U., Rapp, E., and Martens, L. Searching for a needle in a stack of needles: challenges in metaproteomics data analysis. *Molecular BioSystems 9*, 4 (2013), 578–585.

[175] Muth, T., Hartkopf, F., Vaudel, M., and Renard, B. Y. A potential golden age to come—current tools, recent use cases, and future avenues for de novo sequencing in proteomics. *Proteomics 18*, 18 (2018), 1700150.

[176] Muth, T., Kolmeder, C. A., Salojärvi, J., Keskitalo, S., Varjosalo, M., Verdam, F. J., Rensen, S. S., Reichl, U., de Vos, W. M., Rapp, E., et al. Navigating through metaproteomics data: a logbook of database searching. *Proteomics 15*, 20 (2015), 3439–3453.

[177] Muth, T., and Renard, B. Y. Evaluating de novo sequencing in proteomics: already an accurate alternative to database-driven peptide identification? *Briefings in bioinformatics 19*, 5 (2018), 954–970.

[178] Muth, T., Renard, B. Y., and Martens, L. Metaproteomic data analysis at a glance: advances in computational microbial community proteomics. *Expert review of proteomics 13*, 8 (2016), 757–769.

[179] Muth, T., Weilnbock, L., Rapp, E., Huber, C. G., Martens, L., Vaudel, M., and Barsnes, H. Denovogui: an open source graphical user interface for de novo sequencing of tandem mass spectra. *Journal of proteome research 13*, 2 (2014), 1143–1146.

[180] Nesvizhskii, A. I. Protein identification by tandem mass spectrometry and sequence database searching. *Mass Spectrometry Data Analysis in Proteomics* (2007), 87–119.

[181] 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), 2092–2123.

[182] Nevo, N., Thomas, L., Chhuon, C., Andrzejewska, Z., Lipecka, J., Guillonneau, F., Bailleux, A., Edelman, A., Antignac, C., and Guerrera, I. C. Impact of cystinosin glycosylation on protein stability by differential dynamic stable isotope labeling by amino acids in cell culture (silac). *Molecular & Cellular Proteomics 16*, 3 (2017), 457–468.

[183] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378* (2018).

[184] Nzeteu, C., Joyce, A., Thorn, C., McDonnell, K., Shirran, S., O'Flaherty, V., and Abram, F. Resource recovery from the anaerobic digestion of food waste is underpinned by cross-kingdom microbial activities. *Bioresource Technology Reports 16* (2021), 100847.

[185] Olsen, J. V., Macek, B., Lange, O., Makarov, A., Horning, S., and Mann, M. Higher-energy c-trap dissociation for peptide modification analysis. *Nature methods 4*, 9 (2007), 709–712.

[186] Orphanides, G., and Reinberg, D. A unified theory of gene expression. *Cell 108*, 4 (2002), 439–451.

[187] O'Connor, C. M., Adams, J. U., and Fairman, J. Essentials of cell biology. *Cambridge, MA: NPG Education 1* (2010), 54.

[188] Paiva, A. L., Oliveira, J. T., de Souza, G. A., and Vasconcelos, I. M. Label-free proteomic reveals that cowpea severe mosaic virus transiently suppresses the host leaf protein accumulation during the compatible interaction with cowpea (vigna unguiculata [l.] walp.). *Journal of Proteome Research 15*, 12 (2016), 4208–4220.

[189] Paizs, B., and Suhai, S. Fragmentation pathways of protonated peptides. *Mass spectrometry reviews 24*, 4 (2005), 508–548.

[190] Patki, N., Wedge, R., and Veeramachaneni, K. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2016), IEEE, pp. 399–410.

[191] Patterson, S. D. Data analysis—the achilles heel of proteomics. *Nature biotechnology 21*, 3 (2003), 221–222.

[192] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research 12* (2011), 2825–2830.

[193] Peng, M., Mo, Y., Wang, Y., Wu, P., Zhang, Y., Xiong, F., Guo, C., Wu, X., Li, Y., Li, X., et al. Neoantigen vaccine: an emerging tumor immunotherapy. *Molecular cancer 18*, 1 (2019), 1–14.

[194] Petersen, J. M., Kemper, A., Gruber-Vodicka, H., Cardini, U., Van Der Geest, M., Kleiner, M., Bulgheresi, S., Mussmann, M., Herbold, C., Seah, B. K., et al. Chemosynthetic symbionts of marine invertebrate animals are capable of nitrogen fixation. *Nature microbiology 2*, 1 (2016), 1–11.

[195] Pigman, W. *The Carbohydrates: Chemistry and Biochemistry Physiology*. Elsevier, 2012.

[196] Pocsfalvi, G., Cacace, G., Cuccurullo, M., Serluca, G., Sorrentino, A., Schlosser, G., Blaiotta, G., and Malorni, A. Proteomic analysis of exoproteins expressed by enterotoxigenic staphylococcus aureus strains. *Proteomics 8*, 12 (2008), 2462–2476.

[197] Provost, F. J., Fawcett, T., Kohavi, R., et al. The case against accuracy estimation for comparing induction algorithms. In *ICML* (1998), vol. 98, pp. 445–453.

[198] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660.

[199] 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*, 5 (2021), 420–425.

[200] Qiao, R., Tran, N. H., Xin, L., Shan, B., Li, M., and Ghodsi, A. Deepnovov2: Better de novo peptide sequencing with deep learning. *arXiv preprint arXiv:1904.08514* (2019).

[201] Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

[202] Raghunathan, T. E. Synthetic data. *Annual review of statistics and its application 8* (2021), 129–140.

[203] Renard, B. Y., Kirchner, M., Monigatti, F., Ivanov, A. R., Rappsilber, J., Winter, D., Steen, J. A., Hamprecht, F. A., and Steen, H. When less can yield more–computational preprocessing of ms/ms spectra for peptide identification. *Proteomics 9*, 21 (2009), 4978–4984.

[204] Resende, P. A. A., and Drummond, A. C. A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR) 51*, 3 (2018), 1–36.

[205] Reuss, D. R., Altenbuchner, J., Mäder, U., Rath, H., Ischebeck, T., Sappa, P. K., Thürmer, A., Guérin, C., Nicolas, P., Steil, L., et al. Large-scale reduction of the bacillus subtilis genome: consequences for the transcriptional network, resource allocation, and metabolism. *Genome research 27*, 2 (2017), 289–299.

[206] Rhee, S., Seo, S., and Kim, S. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. *arXiv preprint arXiv:1711.05859* (2017).

[207] Rossing, K., Mischak, H., Dakna, M., Zürbig, P., Novak, J., Julian, B. A., Good, D. M., Coon, J. J., Tarnow, L., Rossing, P., et al. Urinary proteomics in diabetes and ckd. *Journal of the American Society of Nephrology 19*, 7 (2008), 1283–1290.

[208] Roy-Lachapelle, A., Solliec, M., Sinotte, M., Deblois, C., and Sauvé, S. High resolution/accurate mass (hrms) detection of anatoxin-a in lake water using ldtd–apci coupled to a q-exactive mass spectrometer. *Talanta 132* (2015), 836–844.

[209] Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[210] Sallam, R. M. Proteomics in cancer biomarkers discovery: challenges and applications. *Disease markers 2015* (2015).

[211] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks 20*, 1 (2008), 61–80.

[212] Schloss, P. D., and Handelsman, J. Metagenomics for studying unculturable microorganisms: cutting the gordian knot. *Genome biology 6*, 8 (2005), 1–4.

[213] Schuster, M., Nechansky, A., and Kircheis, R. Cancer immunotherapy. *Biotechnology Journal: Healthcare Nutrition Technology 1*, 2 (2006), 138–147.

[214] Segata, N., Boernigen, D., Tickle, T. L., Morgan, X. C., Garrett, W. S., and Huttenhower, C. Computational meta'omics for microbial community studies. *Molecular systems biology 9*, 1 (2013), 666.

[215] Seidel, G., Meierhofer, D., Sen, N. E., Guenther, A., Krobitsch, S., and Auburger, G. Quantitative global proteomics of yeast pbp1 deletion mutants and their stress responses identifies glucose metabolism, mitochondrial, and stress granule changes. *Journal of proteome research 16*, 2 (2017), 504–515.

[216] Seidler, J., Zinn, N., Boehm, M. E., and Lehmann, W. D. De novo sequencing of peptides by ms/ms. *Proteomics 10*, 4 (2010), 634–649.

[217] Seng, P., Rolain, J.-M., Fournier, P. E., La Scola, B., Drancourt, M., and Raoult, D. Maldi-tof-mass spectrometry applications in clinical microbiology. *Future microbiology 5*, 11 (2010), 1733–1754.

[218] Shao, C., Zhang, Y., and Sun, W. Statistical characterization of hcd fragmentation patterns of tryptic peptides on an ltq orbitrap velos mass spectrometer. *Journal of proteomics 109* (2014), 26–37.

[219] Shevchenko, A., Sunyaev, S., Loboda, A., Shevchenko, A., Bork, P., Ens, W., and Standing, K. G. Charting the proteomes of organisms with unsequenced genomes by maldi-quadrupole time-of-flight mass spectrometry and blast homology searching. *Analytical chemistry 73*, 9 (2001), 1917–1926.

[220] Shilov, I. V., Seymour, S. L., Patel, A. A., Loboda, A., Tang, W. H., Keating, S. P., Hunter, C. L., Nuwaysir, L. M., and Schaeffer, D. A. The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Molecular & Cellular Proteomics 6*, 9 (2007), 1638–1655.

[221] Shmelkov, K., Schmid, C., and Alahari, K. How good is my gan? In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 213–229.

[222] Shrestha, A., and Mahmood, A. Review of deep learning algorithms and architectures. *IEEE access 7* (2019), 53040–53065.

[223] Siegel, M. M., and Bauman, N. An efficient algorithm for sequencing peptides using fast atom bombardment mass spectral data. *Biomedical & environmental mass spectrometry 15*, 6 (1988), 333–343.

[224] Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489* (2017).

[225] Snyder, L. R., Kirkland, J. J., and Dolan, J. W. *Introduction to modern liquid chromatography*. John Wiley & Sons, 2011.

[226] Solyman, A., Zhenyu, W., Qian, T., Elhag, A. A. M., Toseef, M., and Aleibeid, Z. Synthetic data with neural machine translation for automatic correction in arabic grammar. *Egyptian Informatics Journal 22*, 3 (2021), 303–315.

[227] Stephen, O., Sain, M., Maduh, U. J., and Jeong, D.-U. An efficient deep learning approach to pneumonia classification in healthcare. *Journal of healthcare engineering 2019* (2019).

[228] Sun, S., Yu, C., Qiao, Y., Lin, Y., Dong, G., Liu, C., Zhang, J., Zhang, Z., Cai, J., Zhang, H., et al. Deriving the probabilities of water loss and ammonia loss for amino acids from tandem mass spectra. *Journal of proteome research 7*, 01 (2008), 202–208.

[229] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.

[230] Tabb, D. L., Ma, Z.-Q., Martin, D. B., Ham, A.-J. L., and Chambers, M. C. Directag: accurate sequence tags from peptide ms/ms through statistical scoring. *Journal of proteome research 7*, 9 (2008), 3838–3846.

[231] Tabb, D. L., Saraf, A., and Yates, J. R. Gutentag: high-throughput sequence tagging via an empirically derived fragmentation model. *Analytical chemistry 75*, 23 (2003), 6415–6421.

[232] Tabb, D. L., Smith, L. L., Breci, L. A., Wysocki, V. H., Lin, D., and Yates, J. R. Statistical characterization of ion trap tandem mass spectra from doubly charged tryptic peptides. *Analytical chemistry 75*, 5 (2003), 1155–1163.

[233] Tanaka, T., Basisty, N., Fantoni, G., Candia, J., Moore, A. Z., Biancotto, A., Schilling, B., Bandinelli, S., and Ferrucci, L. Plasma proteomic biomarker signature of age predicts health and life span. *Elife 9* (2020), e61073.

[234] Tang, C., Garreau, D., and von Luxburg, U. When do random forests fail? *Advances in neural information processing systems 31* (2018).

[235] Tharwat, A. Classification assessment methods. *Applied Computing and Informatics* (2020).

[236] Ting, F. F., Tan, Y. J., and Sim, K. S. Convolutional neural network improvement for breast cancer classification. *Expert Systems with Applications 120* (2019), 103–115.

[237] 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), 519–525.

[238] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (2017), IEEE, pp. 23–30.

[239] Tran, B. Q., Hernandez, C., Waridel, P., Potts, A., Barblan, J., Lisacek, F., and Quadroni, M. Addressing trypsin bias in large scale (phospho) proteome analysis by size exclusion chromatography and secondary digestion of large post-trypsin peptides. *Journal of proteome research 10*, 2 (2011), 800–811.

[240] 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*, 1 (2019), 63–66.

[241] 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 114*, 31 (2017), 8247–8252.

[242] Trauger, S. A., Junker, T., and Siuzdak, G. Investigating viral proteins and intact viruses with mass spectrometry. *Modern Mass Spectrometry* (2003), 265–282.

[243] Tsiatsiani, L., and Heck, A. J. Proteomics beyond trypsin. *The FEBS journal 282*, 14 (2015), 2612–2626.

[244] Tsuji, T., Shiozaki, A., Kohno, R., Yoshizato, K., and Shimohama, S. Proteomic profiling and neurodegeneration in alzheimer's disease. *Neurochemical research 27*, 10 (2002), 1245–1253.

[245] Valdes, G., Luna, J. M., Eaton, E., Simone, C. B., Ungar, L. H., and Solberg, T. D. Mediboost: a patient stratification tool for interpretable decision making in the era of precision medicine. *Scientific reports 6*, 1 (2016), 1–8.

[246] Van Wieringen, W. N., Kun, D., Hampel, R., and Boulesteix, A.-L. Survival prediction using gene expression data: a review and comparison. *Computational statistics & data analysis 53*, 5 (2009), 1590–1603.

[247] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[248] Venable, J. D., and Yates, J. R. Impact of ion trap tandem mass spectra variability on the identification of peptides. *Analytical chemistry 76*, 10 (2004), 2928–2937.

[249] Verheggen, K., Ræder, H., Berven, F. S., Martens, L., Barsnes, H., and Vaudel, M. Anatomy and evolution of database search engines—a central component of mass spectrometry based proteomic workflows. *Mass spectrometry reviews 39*, 3 (2020), 292–306.

[250] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature 575*, 7782 (2019), 350–354.

[251] Wang, Y.-C., Peterson, S. E., and Loring, J. F. Protein post-translational modifications and regulation of pluripotency in human stem cells. *Cell research 24*, 2 (2014), 143–160.

[252] Wells, J. M., and McLuckey, S. A. Collision-induced dissociation (cid) of peptides and proteins. *Methods in enzymology 402* (2005), 148–185.

[253] White, F. M. The potential cost of high-throughput proteomics. *Science signaling 4*, 160 (2011), pe8–pe8.

[254] Whiting, M. A., Haack, J., and Varley, C. Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software. In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2008), pp. 1–9.

[255] Wilkins, M. R., Sanchez, J.-C., Gooley, A. A., Appel, R. D., Humphery-Smith, I., Hochstrasser, D. F., and Williams, K. L. Progress with proteome projects: why all proteins expressed by a genome should be identified and how to do it. *Biotechnology and genetic engineering reviews 13*, 1 (1996), 19–50.

[256] Wishart, D. S., Guo, A., Oler, E., Wang, F., Anjum, A., Peters, H., Dizon, R., Sayeeda, Z., Tian, S., Lee, B. L., et al. Hmdb 5.0: the human metabolome database for 2022. *Nucleic Acids Research 50*, D1 (2022), D622–D631.

[257] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems 32*, 1 (2020), 4–24.

[258] Xie, Z., Wang, S. I., Li, J., Lévy, D., Nie, A., Jurafsky, D., and Ng, A. Y. Data noising as smoothing in neural network language models. *arXiv preprint arXiv:1703.02573* (2017).

[259] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., and Wang, C. Machine learning and deep learning methods for cybersecurity. *Ieee access 6* (2018), 35365–35381.

[260] Xu, C., and Ma, B. Complexity and scoring function of ms/ms peptide de novo sequencing. In *Computational Systems Bioinformatics* (2006), World Scientific, pp. 361–369.

[261] Xu, D., Yuan, S., Zhang, L., and Wu, X. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 570–575.

[262] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (2015), PMLR, pp. 2048–2057.

[263] Xu, K., Wu, L., Wang, Z., Feng, Y., Witbrock, M., and Sheinin, V. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823* (2018).

[264] Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A., and Bennett, K. P. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing 416* (2020), 244–255.

[265] Yates III, J. R. Mass spectrometry and the age of the proteome. *Journal of Mass Spectrometry 33*, 1 (1998), 1–19.

[266] Yilmaz, E., and Aslam, J. A. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (2006), pp. 102–111.

[267] Yilmaz, M., Fondrie, W., Bittremieux, W., Oh, S., and Noble, W. S. De novo mass spectrometry peptide sequencing with a transformer model. In *International Conference on Machine Learning* (2022), PMLR, pp. 25514–25522.

[268] Yu, Y., Si, X., Hu, C., and Zhang, J. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation 31*, 7 (2019), 1235–1270.

[269] Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning* (2004), p. 114.

[270] Zeiler, M. D., and Fergus, R. Visualizing and understanding convolutional networks (2013). *arXiv preprint arXiv:1311.2901* (2013).

[271] Zhang, J., Xin, L., Shan, B., Chen, W., Xie, M., Yuen, D., Zhang, W., Zhang, Z., Lajoie, G. A., and Ma, B. Peaks db: de novo sequencing assisted database search for sensitive and accurate peptide identification. *Molecular & cellular proteomics 11*, 4 (2012).

[272] Zhang, Y., Fonslow, B. R., Shan, B., Baek, M.-C., and Yates III, J. R. Protein analysis by shotgun/bottom-up proteomics. *Chemical reviews 113*, 4 (2013), 2343–2394.

[273] Zhang, Z. Prediction of low-energy collision-induced dissociation spectra of peptides. *Analytical chemistry 76*, 14 (2004), 3908–3922.

[274] Zhou, Z.-H. *Machine learning*. Springer Nature, 2021.

[275] Zhu, X., and Wu, X. Class noise vs. attribute noise: A quantitative study. *The Artificial Intelligence Review 22*, 3 (2004), 177.

[276] Zinkernagel, R. M., and Hengartner, H. Regulation of the immune response by antigen. *Science 293*, 5528 (2001), 251–253.

[277] Zubarev, R. A., and Makarov, A. Orbitrap mass spectrometry, 2013.