Western Graduate&PostdoctoralStudies

Western University

Scholarship@Western

Electronic Thesis and Dissertation Repository

7-7-2023 10:30 AM

DpNovo: A DEEP LEARNING MODEL COMBINED WITH DYNAMIC PROGRAMMING FOR DE NOVO PEPTIDE SEQUENCING

Yizhou Li,

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science
© Yizhou Li 2023

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Recommended Citation

Li, Yizhou, "DpNovo: A DEEP LEARNING MODEL COMBINED WITH DYNAMIC PROGRAMMING FOR DE NOVO PEPTIDE SEQUENCING" (2023). *Electronic Thesis and Dissertation Repository*. 9373.
https://ir.lib.uwo.ca/etd/9373

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

Abstract

De novo peptide sequencing is an efficient approach to identifying peptides from tandem mass spectrometry (MS/MS). Compared with the database search methods, *De novo* peptide sequencing is particularly effective in identifying novo peptide sequences. This thesis presents a new *De novo* sequencing model comprising two deep learning models and a dynamic programming algorithm, namely DpNovo. The deep learning model learns features of a spectrum and gives scores to each peak, and the dynamic programming is capable of determining the optimal amino acid sequence path with the highest accumulated score. Finally, the predicted sequence with mass values representing uncertain mass intervals will be provided. DpNovo is capable of reconstructing charge-two peaks in their charge-one positions, which significantly improves the accuracy of predicting high-charged spectra. Besides, the dynamic programming algorithm ensures that accurate predictions can be made even in cases where some signal peaks are missing. In terms of performance, DpNovo has been tested on both the NIST and ProteomeXchange databases. The deep learning model has demonstrated an excellent ability to identify both signal and noise peaks. Additionally, the accuracy of peptide sequence prediction obtained through the dynamic programming algorithm is comparable to those of other proposed *De novo* sequencing models.

Keywords: *De novo*, machine learning, dynamic programming

Summary for Lay Audience

Tandem mass spectrometry (MS/MS) is an important tool for identifying peptides. The peptide identification approaches can analyze the tandem mass spectrometry of fragments to infer the original peptide sequence. There are mainly two methods for peptide identification: database search and *De novo* sequencing. Database search uses a reference database to match experimental data with theoretical spectra, while *De novo* sequencing does not rely on a database and can identify novo amino acid sequences. Recently, machine learning technology is utilized in *De novo* sequencing, and many machine learning-based approaches have been proposed. In this thesis, we proposed a new model called DpNovo, which combines deep learning and dynamic programming. The deep learning model assigns scores to each peak and the dynamic programming algorithm is capable of finding the optimal amino acid sequence with the highest accumulated score. DpNovo is capable of reconstructing charge-two peaks in their charge-one positions, which greatly enhances the accuracy of predicting high-charged spectra. Additionally, the dynamic programming algorithm employed by DpNovo ensures that it can make precise predictions even in situations where some signal peaks may be absent. The training dataset is acquired from the NIST database, and testing was conducted on the NIST and the ProteomeXchange database. The deep learning model has displayed proficiency in detecting both signal and noise peaks. Moreover, the precision of peptide sequence prediction is equivalent to that of other proposed *De novo* sequencing models.

Acknowlegements

First, I appreciate the help from Professor Kaizhong Zhang, whose generous support has proven invaluable in helping me navigate and overcome various challenges that have arisen throughout my academic journey. Due to the limited opportunities to interact with professors during my undergraduate studies, I gained little exposure to the academic community. However, during my graduate studies, Professor Zhang willingly invested his time in exchanging ideas with me. It was through his mentorship that I was able to learn a great deal from him.

Second, I am also deeply grateful for my girlfriend and friends who have provided me with care and companionship while I am away from home in a foreign land.

Although my graduate studies only spanned two years, I gained a wealth of knowledge and experienced tremendous personal growth during this time. I sincerely hope that I can carry this growth with me into my future endeavours. Finally, I extend my gratitude to the University of Western Ontario.

Contents

List of Figures

List of Tables

List of Supplementary

Chapter 1

Introduction

1.1 Tandem Mass Spectrometry

Mass spectrometry is a powerful technique for the analysis of unknown compounds. In a mass spectrometer, molecules are ionized into positively charged molecular ions. These ions exhibit different masses, leading to varying times required to reach the detector. The key value in this process is the mass-to-charge (m/z) ratio, where m denotes the mass of a specific ion and z represents the absolute value of the number of electrons charged on that ion. In addition to the m/z ratio, mass spectrometry provides information about the abundance or intensity of ions for each m/z. Analyzing the output of a mass spectrometer enables the comprehensive examination of an ion's structure and composition. However, single-stage mass spectrometry has limitations when it comes to analyzing high-mass molecules or complex structures, due to issues such as overlapping spectra or ionization efficiency. To address these challenges, tandem mass spectrometry was developed.

Analyzing tandem mass spectrometry (MS/MS) is one of the most effective methods for protein identification and plays a critical role in proteomics research. This technique has a range of applications, including the identification of unknown compounds, protein identification and characterization, biomolecule quantification, and the analysis of post-translational modifications [4]. In tandem mass spectrometry, two mass analyzers are typically used. The first mass analyzer is used to select the ion that needs to be analyzed, and the second mass analyzer is used to further analyze the selected ions. The process begins with the proteolytic cleavage of proteins into peptides. Subsequently, a peptide ion is selected by the first mass analyzer, termed the precursor ion. This precursor ion is then introduced into a collision cell filled with a neutral, inert gas such as argon or helium. Inside the collision cell, ions undergo collision-induced dissociation (CID), leading to the fragmentation of the peptide into a set of smaller fragment ions. There are many other activation processes [5], apart from CID, including Post-Source Decay (PSD) [6], Surface-Induced Dissociation (SID), Electron-Capture Dissociation (ECD) [7], Infrared Multiphoton Dissociation (IRMPD), Blackbody Infrared Radiative Dissociation (BIRD) [8], and Higher-energy Collisional Dissociation (HCD) [9]. These fragment ions are then separated by their mass using a second mass analyzer. The resulting pattern of fragment ions, often referred to as a ”ladder,” can be employed to infer the original peptide sequence, as the mass difference between two related fragment ions corresponds to the molecular weight of an amino acid residue [10].

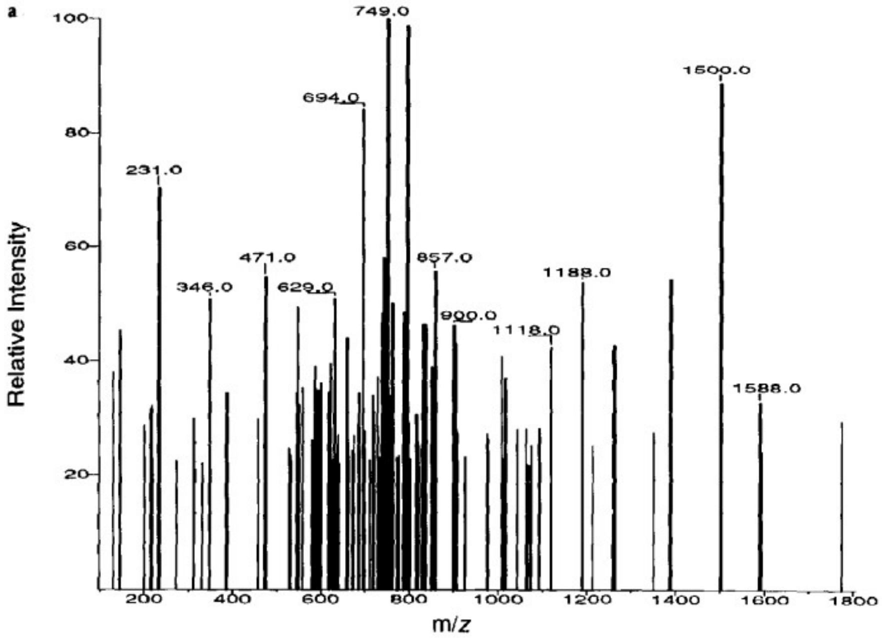


Figure 1.1: Example of tandem mass spectrum for peptide ‘DLRSWTAADTAAQLSQ’. Source: [1]

The precision of the spectrum varies depending on the activation process utilized. Higher-energy Collisional Dissociation (HCD) has contributed significantly to the improved accuracy of peptide sequencing due to its enhancement of mass spectral precision [11]. Higher-energy Collisional dissociation is a fragmentation technique employed in mass spectrometry, involving the collision of high-energy ions with the analyte molecule, causing it to dissociate into smaller fragments. To generate high-energy ions in HCD, the precursor ion is typically accelerated and collided with a gas such as helium or nitrogen. This collision causes the precursor ion to fragment into smaller ions, which can then be analyzed by the mass spectrometer to determine their mass. One of the advantages of HCD is that it can provide high-precision analysis, and there are several reasons: 1. HCD generates multiple ion fragments: The high-energy collisions in HCD cause molecules to produce numerous ion fragments, including *b* and *y*-ions. 2. HCD generates high-resolution data [12]. Overall, HCD is a powerful analytical tool that offers high-precision analysis for complex molecules, making it an essential technique in modern mass spectrometry [13].

1.2 Peptide Identification

Tandem mass spectrometry-based peptide identification can be classified into two methods: (1) database search, and (2) *De Novo* sequencing. As the name implies, database search relies on a reference database for peptide identification, while *De Novo* sequencing is capable of identifying novel or previously unobserved proteins or peptides that are not present in the database.

Over the years, several database searching methods have been proposed, such as SEQUEST [1], Mascot [14], pFind [15], X! Tandem [16], PEAKS DB [17], etc. While the details of the search strategies are different, the main idea of these approaches is to compare the experimental spectra with the theoretical spectra reconstructed from a database. Database search can be efficient when the target peptides are present in the database. However, if the target peptide sequence is absent from the database, a database search might not be feasible. Moreover, as the database expands, the time required for searching can become increasingly extensive.

We will use the SEQUEST algorithm as an example to introduce the database searching method. First, some preprocessing of the spectrum is implemented to enhance the precision. The mass values of fragmented ions are converted into rounded nominal values. Furthermore, only the top 200 ions with the highest abundance are kept to eliminate the noise. The other details of this step will not be presented here. Secondly, the mass of the candidate peptide sequences will be compared to the mass of the target ion, and those candidate peptide sequences whose mass falls within specified mass tolerances will be retained for further comparison. Usually, the mass tolerance will be set at ± 0.05% or $1u$. Thirdly, a scoring method is designed to give scores to those candidates that are in the mass tolerance. This scoring function will give scores based on the abundance, continuity, and presence of certain ions. Then the top 500 candidates are selected to do some further comparisons. Fourthly, the theoretical spectra of these 500 candidates are reconstructed based on some criteria, and figure 1.2 is an example of a reconstructed spectrum. The final step is comparing the theoretical spectra with the experimental spectra, a cross-correlation method is used to achieve the comparison. Then the final prediction will be made according to the result of the cross-correlation scoring method.

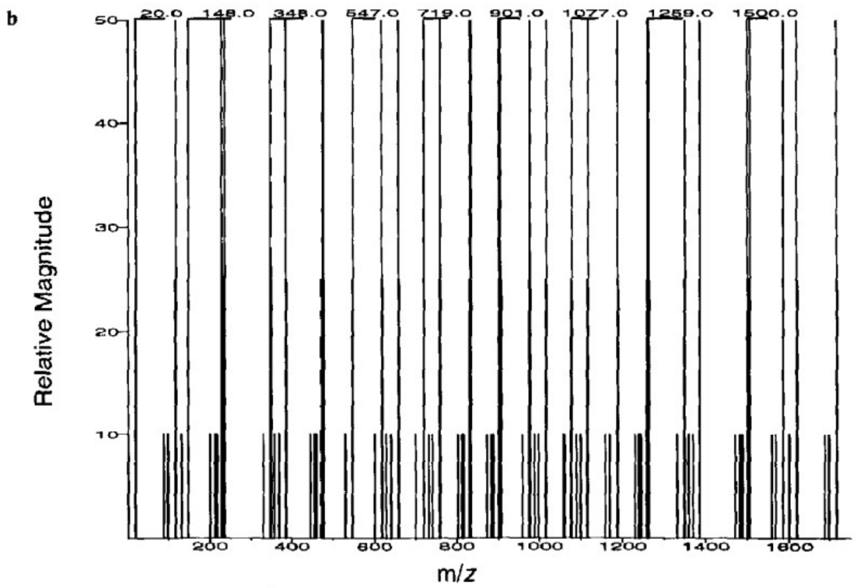


Figure 1.2: The example of a reconstructed spectrum for the amino acid sequence ‘DLRSWTAADTAAQISQ’. Source: [1]

De Novo peptide sequencing is a method that can sequence peptides without the aid of a database and is highly effective in identifying novo amino acid sequences. Besides, the result of the *De Novo* sequencing can be used to validate the result from the database search. The main idea of *De Novo* sequencing is calculating the mass difference between two fragmented ions, which may be equal to the mass of certain amino acids. Several methods have been presented for *De Novo* sequencing, including Pepnovo [18], Peaks [19], pNovo [20], UniNovo [21], NovoHMM [22], and Msnovo [23]. These methods use different approaches to achieve peptide sequencing, including graph-based approaches, hidden Markov models, probabilistic networks, statistical models, and support vector machines. *De Novo* sequencing can be used in many situations since there is no restriction on the database. In recent years, deep learning technology has been integrated into *De Novo* sequencing, for example, DeepNovo [12]. DeepNovo utilizes convolutional neural networks (CNN) to learn the spectrum features and long short-term memory (LSTM) networks to learn the sequence features and predict the next amino acid based on the previous mass and the feature of the next position. *De Novo* sequencing considerably improves the flexibility of spectrum identification. However, there are still some limitations to *De Novo* sequencing. Firstly, certain algorithms necessitate substantial computational resources. Secondly, *De Novo* sequencing has the potential to generate false positives, resulting in the prediction of incorrect or implausible amino acid sequences. Furthermore, *De Novo* sequencing can only predict the amino acid sequence without providing any information about the corresponding protein.

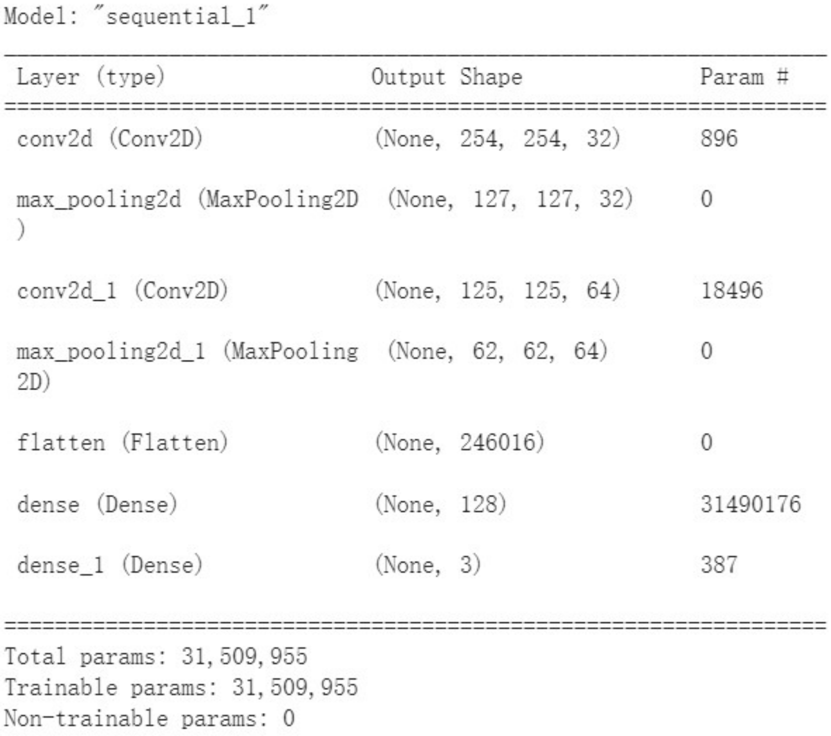
The quality of the experimental spectra is critical for accurate peptide identification. The spectral resolution, signal-to-noise ratio, and the presence of missing or noisy peaks can all impact the precision of the prediction, whether using *De Novo* sequencing or a database search approach. In peptide identification, the activation process used can significantly influence the quality of the spectra obtained. Techniques like HCD, which provide high-resolution data with minimal noise, are highly desirable for achieving reliable and precise identification of peptides.

1.3 Deep Learning

With the help of big data, deep learning technologies have developed rapidly in recent years. Deep learning can create a unified representation of data by automatically learning features from multiple heterogeneous sources and mapping them to a commonly hidden space [24]. There are various deep learning models, for example, autoencoder, restricted Boltzmann machine, convolutional neural networks, recurrent neural networks, deep neural networks, and generative adversarial networks. The earliest implementation of deep learning is image processing. In 2014, Dong et al. [25] proposed a deep convolutional neural network for image classification. And in the 2016 ImageNet Competition, CNNH [26] reached an accuracy of 97%. Deep learning models can be applied to various domains, such as image processing, speech recognition, and natural language processing. With the exponential growth of available data, the application of deep learning techniques is expected to become increasingly widespread across diverse industries and domains.

1.3.1 Convolutional Neural Network

The convolutional neural network is one type of artificial neural network and is extremely efficient in the pattern recognition area within images. The basic idea in image classification is using a convolutional neural network (CNN) to extract the relevant features behind the image, and fully connected layers are used following the convolutional neural network to do the classification of these features. Traditional deep neural networks may struggle to effectively identify images due to their high computational complexity, rendering them impractical for real-world applications. However, the use of CNNs, which employ specialized convolutional kernels to learn and extract features from images, has significantly reduced computational complexity and greatly improved performance in image recognition tasks. Additionally, customized convolutional kernels can extract local features from images based on varying requirements. A basic CNN comprises three distinct layers: convolutional layers, pooling layers, and fully-connected layers. Figure 1.3 provides an example of a basic convolutional network. In the following section, we will review the fundamentals of a basic CNN.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 max_pooling2d (MaxPooling2D  (None, 127, 127, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 125, 125, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 64)       0
 2D)

 flatten (Flatten)           (None, 246016)            0

 dense (Dense)               (None, 128)               31490176

 dense_1 (Dense)             (None, 3)                 387

=================================================================
Total params: 31,509,955
Trainable params: 31,509,955
Non-trainable params: 0
```

Figure 1.3: The structure of a basic convolutional network. This is a sequential neural network architecture consisting of four layers: two convolutional layers (conv2d) and two max pooling layers (max_pooling2d). The first convolutional layer has 32 filters and a kernel size of 3x3, while the second convolutional layer has 64 filters and a kernel size of 3x3. The max-pooling layers are used to downsample the feature maps. The output of the final pooling layer is flattened and passed through two dense layers (Dense), with the first dense layer having 128 neurons and the second dense layer having 3 neurons, which represents the number of classes in the output.

Convolutional layers are a crucial component of convolutional neural networks (CNNs), which are widely used for image-processing tasks such as object recognition and segmentation. The learnable kernel, a small matrix with varying sizes and weights, is the key element of the convolutional layer that extracts features from the input image. The kernel moves from the top-left corner of the image, scanning the entire image while multiplying each element of the kernel with the corresponding pixel value in the image and summing the results. The kernel then moves one stride to the right, and the process repeats, extracting features. The outcome is a new image that represents the features extracted by the kernel. In CNNs, the input typically consists of pixel values from a grayscale or RGB image. If the image is an RGB image, a 3-channel convolutional kernel is used to extract features from each color channel. Multiple convolutional kernels can be combined to extract a broader range of features. In addition to the learnable kernel that extracts features, several other layers and functions in CNNs work together to enhance the quality of the results.

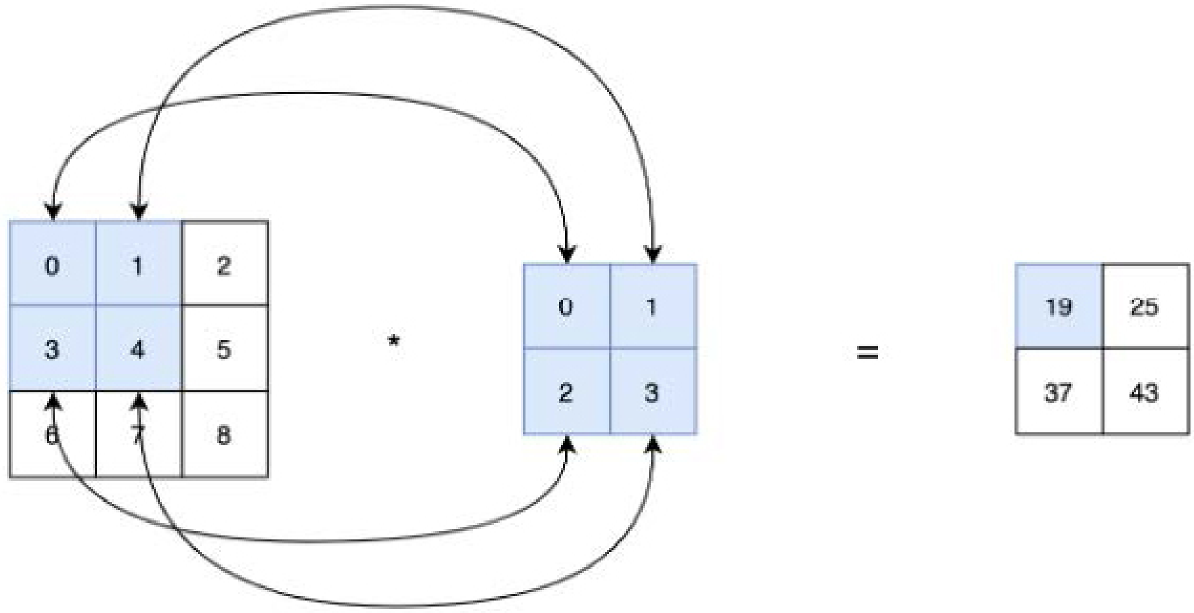


Figure 1.4: An example of convolutional operation. The size of the convolutional kernel is (2,2).

The pooling layers are aiming to decrease the size of the feature maps while preserving the salient features in the data. Pooling is often performed after convolutional layers in CNN, and this process is beneficial to decrease the computational complexity of the number of parameters in the training process and helps to reduce the over-fitting situations. There are many ways to do pooling, for example, max-pooling or overlapping pooling. Max-pooling is the most commonly used pooling method in CNNs [27]. It divides the input feature map into non-overlapping rectangular regions and takes the maximum value from each region, reducing the spatial size of the feature map while retaining the most important features. Overlapping pooling allows for overlap between the rectangular regions, which can preserve more spatial information, but can increase the number of parameters in the network [28].

Additionally, activation functions are frequently employed in convolutional layers. Several activation functions exist, with ReLU (Rectified Linear Unit) being the most commonly used. Activation functions are applied element-wise to the output of each neuron in the convolutional layer, introducing non-linearity in the network and enabling it to learn complex patterns in the data [29].

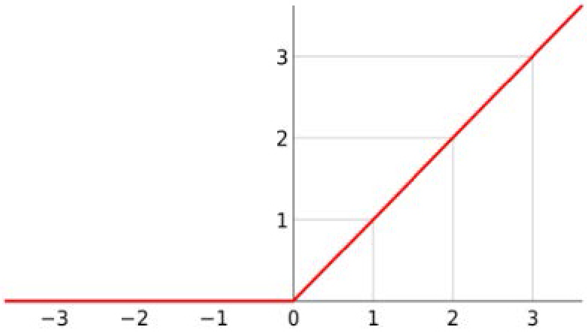


Figure 1.5: The plot of the ReLU function.

Zero-padding is a method used to maintain the spatial dimensions of the input image. This technique involves adding additional rows and columns of zeros at the borders of the input image prior to applying the convolutional filter. The main objective of zero-padding in convolutional neural networks is to maintain the spatial size of the input image and ensure that the resulting feature map has the same dimensions as the input [30]. Without zero-padding, the convolutional kernel would only be able to process a portion of the input image, resulting in a smaller output feature map. By adding zeros around the image, zero-padding enables the kernel to apply to all areas of the input, producing an output with the same spatial dimensions as the original image.

Moreover, batch normalization is a technique used in deep neural networks to improve their performance. Traditionally, neural networks normalize their inputs using the mean and standard deviation of the entire training set. However, this can cause problems during training, as the distribution of the input to each layer can shift significantly with each mini-batch. Batch normalization addresses this problem by normalizing the inputs to each layer for each mini-batch, rather than the entire training set [31].

A flatten layer is commonly employed prior to the fully connected layer. The primary function of the flatten layer is to reshape the output from a convolutional layer, which is generally a 3D tensor representing feature maps, into a 1D vector suitable for input to a classifier or other layers requiring 1D input.

Fully-connected layers, also known as dense layers, are typically positioned at the end of a neural network architecture. These layers accept feature maps generated by preceding convolutional layers as inputs and produce outputs corresponding to the number of classes or labels in a classification task. Each neuron in a fully-connected layer is connected to all neurons in the adjacent layers, with each connection possessing an associated weight and bias. Usually, multiple fully-connected layers are connected together in a deep neural network to enhance the network's representational power [32]. This enables the network to learn increasingly complex features and relationships among features, resulting in improved performance and accuracy for classification tasks.

Within the context of classification tasks, fully-connected layers play an essential role in integrating the features extracted from inputs and making predictions based on the learned representations [33]. The final layer of such a network typically employs a *softmax* activation function, which transforms the outputs from the previous layer into a probability distribution. Each probability value signifies the likelihood that the input belongs to a specific class. The model then selects the class with the highest probability as its final prediction.

In summary, while the specific architecture of a CNN may vary depending on the task and data, the fundamental concept of using convolutional layers to extract features and then using those features for classification remains the same.

1.3.2 Network Training

The primary objective of neural network training is to optimize the trainable parameters within the network. Through the learning process, these parameters can be adjusted to improve performance. During the training process, each iteration yields a result, which is then compared to the ground truth labels using a loss function. There are many choices of loss functions, which depend on the specific requirements and context of the task at hand. For example, Mean Squared Error (MSE): Commonly used for regression problems, MSE measures the average squared difference between the predicted and actual values [34]. Cross-Entropy Loss: Widely used in classification tasks, cross-entropy loss measures the similarity between the predicted probability distribution and the true distribution of class labels. This comparison generates a loss value, which carries with it a gradient [35]. Utilizing the backpropagation algorithm, the direction of this gradient can be computed, and the parameters can be adjusted accordingly [36]. Through multiple iterations of training, the loss function is optimized to converge towards a local minimum or global minimum, resulting in a model that is well-trained and performs optimally on the training data.

Various techniques are employed during the training process to improve model performance, such as early stopping, which can help prevent overfitting [37]; Dropout: a regularization technique that randomly deactivates a fraction of neurons during training, reducing the model's reliance on individual neurons and promoting generalization [38]; Data Augmentation, which involves creating new training examples by applying various transformations to existing data, such as rotation, scaling, or flipping, to increase the diversity of the training dataset and reduce overfitting [39]; and Learning Rate Scheduling, a technique that adjusts the learning rate during training, often starting with a larger value and decreasing it over time to allow the model to converge more efficiently [40]. These techniques, when combined appropriately, can help improve model generalization and prevent overfitting, leading to better overall performance.

With the advancement of machine learning libraries in Python, there is no longer a need to manually compute the loss and gradients. By specifying the desired loss function and training parameters such as epochs, batch size, and so on, these libraries can automatically perform training and adjust the model's parameters. Moreover, with the aid of GPUs, convolution operations can be executed and accelerated, significantly increasing the speed of computation.

1.4 Research in This Thesis

In this thesis, we develop a method called DpNovo, which combines deep learning and dynamic programming to achieve *De Novo* sequencing. DpNovo represents each spectrum as a histogram and the deep learning model is used to assign scores to each peak. The deep learning model comprises two convolutional neural network models: the first model makes an initial prediction, and the output from the first model substitutes the values in the mass map. The second CNN model extracts features from the mass map and makes predictions, then assigns scores to each peak. Subsequently, a dynamic programming algorithm is employed to accumulate scores along different paths, and backtracking is executed from the end position to obtain the path with the highest accumulated score. DpNovo performs well even though there are various missing peaks or when the charge of the spectrum exceeds two.

The training data source is the NIST (National Institute of Standards and Technology) [41] database. The NIST Mass Spectrometry Data Center, which operates under the Biomolecular Measurement Division (BMD), specializes in creating and evaluating mass spectral libraries while also offering associated software tools. The dataset used here is NIST *H.sapiens* Orbitrap HCD, human_hcd_tryp_best spectra. We selected some spectra at random to serve as the training set. And there are two testing data sources: the NIST database and the ProteomeXchange database [42]. The ProteomeXchange database is a public data repository that allows researchers to globally share and access mass spectrometry-based proteomics data. Due to the large size of the original dataset, some subsets of spectra were randomly selected to serve as the testing sets. The detail of the ways of selecting the sets and the size of the sets will be explained in Section 3.1.

Regarding the performance of the deep learning model, the accuracies for identifying signal peaks and noise peaks for the testing set in NIST *H.sapiens* Orbitrap_HCD data were 93.58% and 89.84%, respectively.

Regarding the performance of the whole *De Novo* sequencing algorithm, we use Amino Acid Precision, Amino Acid Recall and Peptide Recall to do the evaluation. The detail of these three evaluation criteria will be explained in Section 3.2. In the NIST dataset, the test had been done on three species, *M.musculus*, *H.sapiens* and *C.griseus*. The amino acid recall and amino acid precision reached 81.1% and 81.3% respectively, and the peptide recall reached 49.2% for *H.sapiens*. The result on other species for example *C.griseus* reached 88.5% for amino acid recall, which means that this model has a strong generalization ability on different species. In the ProteomeXchange database, the amino acid recall and precision reached 59.2% and 59.5% respectively, and the peptide recall reached 16.7% for *H.sapiens* Q-Exactive_HCD data. The recall and precision results for amino acid identification are comparable to those of other *De Novo* sequencing models, such as PEAKS and DeepNovo. However, the peptide recall is relatively low due to various reasons, and the reasons will be explained in the following chapter.

Lastly, when generating the final prediction, undetermined mass intervals caused by missing peaks are replaced with the corresponding mass values, and all possible amino acid sequences that satisfy the mass intervals are provided.

Chapter 2

Methods

2.1 Tandem Mass Spectrum

2.1.1 Peptide Fragmentation and Mass Relationships

In the process of tandem mass spectrometry, proteins are first broken down into numerous peptides by proteases, with trypsin being the most commonly used protease. Trypsin typically cleaves peptide bonds on the carboxyl side of lysine (*K*) or arginine (*R*) amino acid residues, breaking down proteins into smaller peptide fragments [43]. Consequently, the *C*-terminal of the precursor ion selected by the first mass spectrometer ends with either lysine or arginine if the protease is trypsin. As mentioned earlier, the precursor ion generates several fragmented ions under specific conditions, such as CID. A fragmented ion primarily consists of three complementary types: (a-ion, *x*-ion), (*b*-ion, *y*-ion), and (*c*-ion, *z*-ion), determined by the fragmentation position [44].

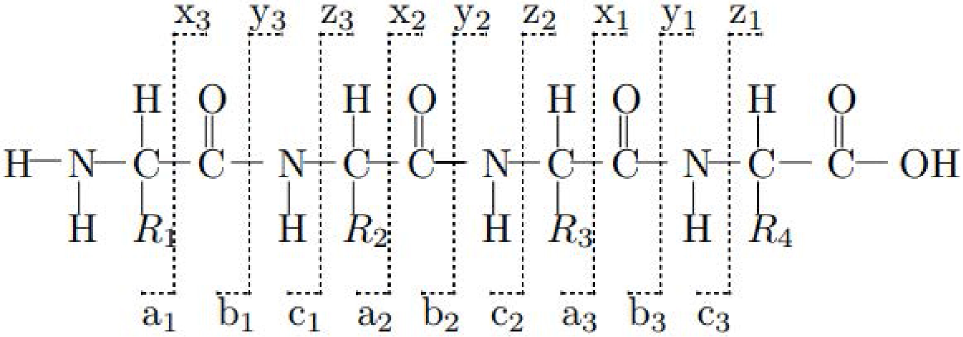


Figure 2.1: The example of different positions of fragmentation. Source: [2]

Electron transfer dissociation (ETD) and collision-induced dissociation (CID) are the two most prevalent methods for peptide fragmentation. Under CID conditions, b and y-ions can be generated, while ETD produces c and z-ions. A *b*-ion (or b-series ion) is a fragment ion formed by the cleavage of a peptide bond N-terminal to the ionized amino acid residue, whereas the y-ion is formed C-terminal to the ionized amino acid residue. High energy collisional dissociation (HCD) is a CID technique associated with Thermo Scientific Orbitrap instruments.

In HCD conditions, there are primarily three types of ions produced: a-ions, b-ions, and y-ions. Besides, various types of additional ions, such as immonium ions, internal fragment ions, neutral losses, gains, and isotopic ions, are commonly observed in addition to the target ions. These ions can be generated due to a variety of factors, including ionization reactions, chemical reactions within the mass spectrometer, the molecular structure of the sample, and experimental conditions such as ion source temperature and ionization energy. For example, immonium ions may be formed due to the presence of specific functional groups in the sample, while internal fragment ions can be produced by fragmentation reactions.

The mass relationship between *y*-ion, *b*-ion, residue mass, parent mass, and charge state are important in *De Novo* sequencing and need to be clarified. In Figure 2.2, it can be observed that the b-ion obtains an additional H^+, and y-ion obtains an additional H_2O and H^+.

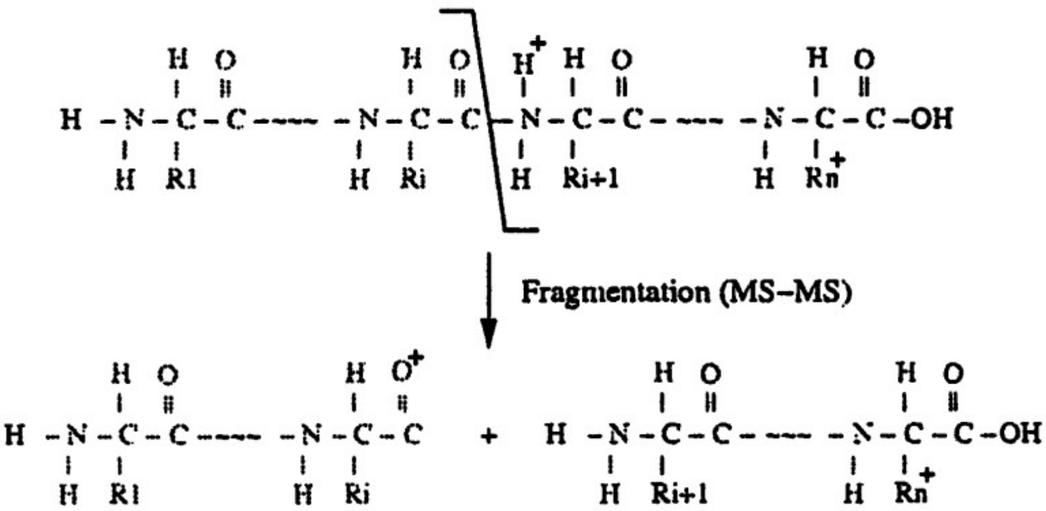


Figure 2.2: A doubly charged peptide molecule is fragmented into a b-ion and a y-ion. Source: [3]

We use α to denote an amino acid and $|\ \alpha\ |$ to denote the residue mass of α. The monoisotopic mass and average mass can be seen in Supplementary 1. In this thesis, we use the monoisotopic mass to represent the mass of each amino acid. Given an amino acid sequence $S = \alpha_1\alpha_2\alpha_3 \ldots \alpha_k$, define $|\ S\ | = |\ \alpha_1\ | + |\ \alpha_2\ | + ... + |\ \alpha_k\ |$. The actual mass of the peptide S is $18 + |\ S\ |$, which is because of an additional H_2O. b_i denotes the b-ion of P with i amino acids, the mass of $b_i = |\ \alpha_1\alpha_2...\alpha_i\ |_b$ can be computed with $|\ b_i\ | = 1 + \sum_{1\leq j\leq i} |\ a_j\ |$. y_i denotes the y-ion of P with i amino acids, the mass of $y_i = |\ \alpha_{k-i+1}...\alpha_{k-1}\alpha_k\ |_y$ can be computed with $|\ y_i\ | = 19 + \sum_{k-i+1\leq j\leq k} |\ a_j\ |$ [3]. Then the following equation can be computed:

$$|\ b_i\ | + |\ y_{k-i}\ | = 20 + |\ S\ |, \quad 1 \leq i \leq k \tag{2.1}$$

And we can call b_i and y_{k-i} a pair of complementary ions. If we consider the charge state, the relationship between parent mass, charge state, and residue mass can be described mathematically as followed:

$$\textit{Parent mass} = (\ 18 + |\ P\ | + \textit{charge state}\ )\ /\ \textit{charge state} \tag{2.2}$$

This relationship is useful when we calculate the dynamic programming matrix, as it enables the determination of both the start and end positions of a path. Furthermore, when extracting features, the complementary position can be computed.

2.1.2 Signal Peak and Noise Peak

To provide a clear illustration, the peaks in the spectrum corresponding to the *a*-ion, *b*-ion, and *y*-ion are referred to as the ‘signal peak’, while the ‘noise peak’ represents the peaks of immonium ions, internal fragment ions, neutral losses, gains, and isotopic ions from the precursor and product ions, as well as the fragment that cannot be identified by the spectrometry. Fragments that cannot be identified are primarily attributed to the presence of mixed peptides during the precursor ion selection process, where the noise likely originates from other peptides. The experiments described in this section were conducted using a subset of the NIST H. sapiens Orbitrap_HCD spectra. The subset consisted of 1000 randomly selected spectra, totalling 98,359 peaks.

It can be observed in Figure 2.3 that the majority of peaks of a spectrum are noise peaks, accounting for 86.5%. As a result, only a very small fraction of the peaks (signal peaks) is beneficial for the final prediction.

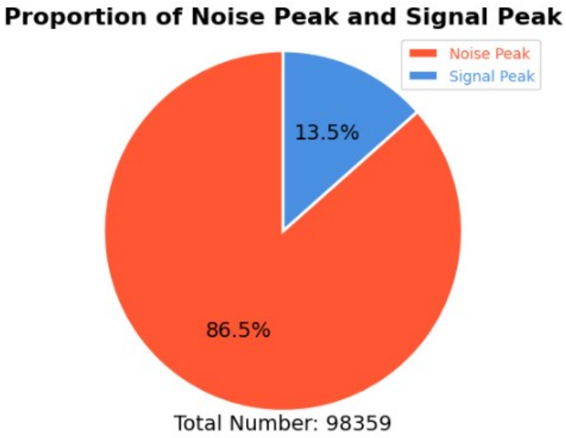


Figure 2.3: The proportion of signal peak and noise peak in a total of 98,359 peaks.

Figure 2.4 shows the relative proportions of y-ions, *b*-ions, and a-ions in the same dataset, and there are a total of 13,278 signal peaks. It can be observed that y-ion is the most predominant ion, accounting for over half of the ions, reaching 53.8%, while *b*-ion accounts for 33.0%, and a-ion only accounts for 13.2%.

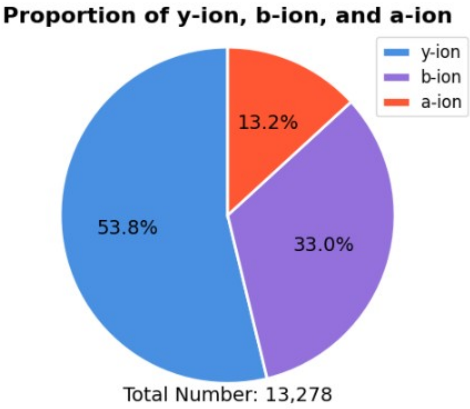
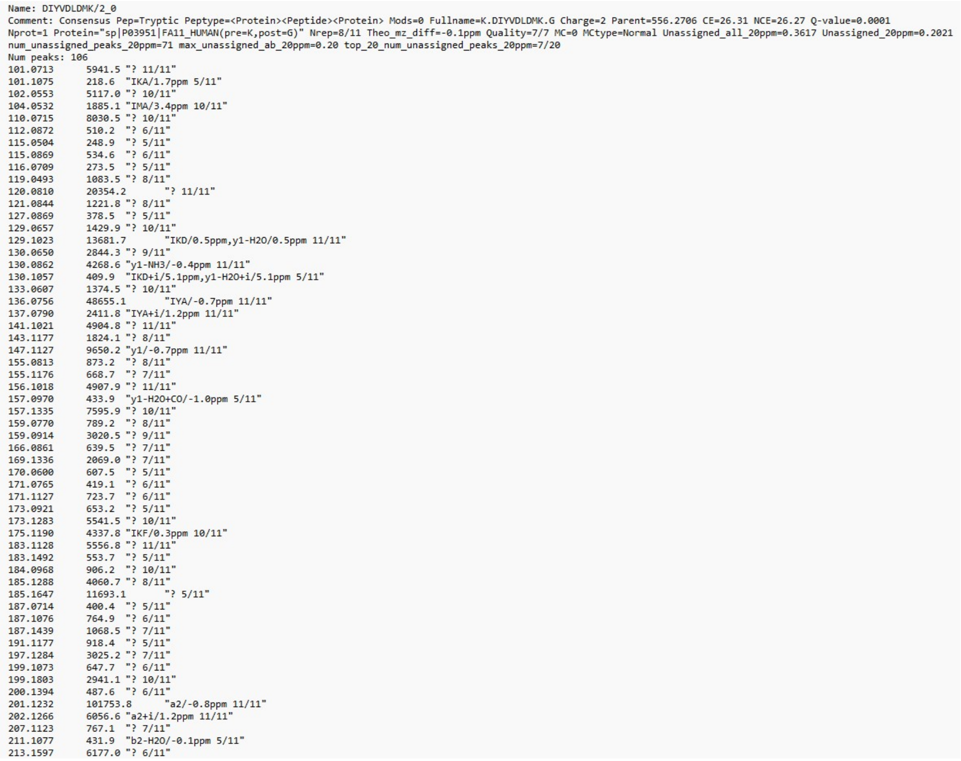


Figure 2.4: The proportion of *a*-ion, *b*-ion, and *y*-ion in a total of 13,278 signal peaks.

2.1.3 Spectrum and Mass Map

In this study, the NIST H. sapiens Orbitrap_HCD spectra are selected as the training data set. The spectrum is in ASCII text (MSP format), and an example of a peptide sequence "DIYVDLDMK" is shown in Figure 2.5, and the rest part of the spectrum can be seen in Supplementary 2. An annotated spectrum contains the name of the precursor ion, the modification, charge state, parent ion mass, and the abundance of each peak, as well as their mass-to-charge value.



```
Name: DIYVDLDMK/2_0
Comment: Consensus Pep=Tryptic Peptype=<Protein><Peptide><Protein> Mods=0 Fullname=K.DIYVDLDMK.G Charge=2 Parent=556.2706 CE=26.31 NCE=26.27 Q-value=0.0001
Nprot=1 Protein="sp|P03951|FA11_HUMAN(pre=K,post=G)" Nrep=8/11 Theo_mz_diff=-0.1ppm Quality=7/7 MC=0 MCtype=Normal Unassigned_all_20ppm=0.3617 Unassigned_20ppm=0.2021
num_unassigned_peaks_20ppm=71 max_unassigned_ab_20ppm=0.20 top_20_num_unassigned_peaks_20ppm=7/20
Num peaks: 106
101.0713	5941.5 "? 11/11"
101.1075	218.6  "IKA/1.7ppm 5/11"
102.0553	5117.0 "? 10/11"
104.0532	1885.1 "IMA/3.4ppm 10/11"
110.0715	8030.5 "? 10/11"
112.0872	510.2  "? 6/11"
115.0504	248.9  "? 5/11"
115.0869	534.6  "? 6/11"
116.0709	273.5  "? 5/11"
119.0493	1083.5 "? 8/11"
120.0810	20354.2	      "? 11/11"
121.0844	1221.8 "? 8/11"
127.0869	378.5  "? 5/11"
129.0657	1429.9 "? 10/11"
129.1023	13681.7	      "IKD/0.5ppm,y1-H2O/0.5ppm 11/11"
130.0650	2844.3 "? 9/11"
130.0862	4268.6 "y1-NH3/-0.4ppm 11/11"
130.1057	409.9  "IKD+i/5.1ppm,y1-H2O+i/5.1ppm 5/11"
133.0607	1374.5 "? 10/11"
136.0756	48655.1	      "IYA/-0.7ppm 11/11"
137.0790	2411.8 "IYA+i/1.2ppm 11/11"
141.1021	4904.8 "? 11/11"
143.1177	1824.1 "? 8/11"
147.1127	9650.2 "y1/-0.7ppm 11/11"
155.0813	873.2  "? 8/11"
155.1176	668.7  "? 7/11"
156.1018	4907.9 "? 11/11"
157.0970	433.9  "y1-H2O+CO/-1.0ppm 5/11"
157.1335	7595.9 "? 10/11"
159.0770	789.2  "? 8/11"
159.0914	3020.5 "? 9/11"
166.0861	639.5  "? 7/11"
169.1336	2069.0 "? 7/11"
170.0600	607.5  "? 5/11"
171.0765	419.1  "? 6/11"
171.1127	723.7  "? 6/11"
173.0921	653.2  "? 5/11"
173.1283	5541.5 "? 10/11"
175.1190	4337.8 "IKF/0.3ppm 10/11"
183.1128	5556.8 "? 11/11"
183.1492	553.7  "? 5/11"
184.0968	906.2  "? 10/11"
185.1288	4060.7 "? 8/11"
185.1647	11693.1	      "? 5/11"
187.0714	400.4  "? 5/11"
187.1076	764.9  "? 6/11"
187.1439	1068.5 "? 7/11"
191.1177	918.4  "? 5/11"
197.1284	3025.2 "? 7/11"
199.1073	647.7  "? 6/11"
199.1803	2941.1 "? 10/11"
200.1394	487.6  "? 6/11"
201.1232	101753.8      "a2/-0.8ppm 11/11"
202.1266	6056.6 "a2+i/1.2ppm 11/11"
207.1123	767.1  "? 7/11"
211.1077	431.9  "b2-H2O/-0.1ppm 5/11"
213.1597	6177.0 "? 6/11"
```

Figure 2.5: Example of the NIST Orbitrap_HCD spectrum.

One major factor that affects the accuracy is the quality of the spectra data. Ideally, the mass spectra should contain all the signal peaks, and the abundance of the signal peaks should be much higher than that of noise peaks. However, in practical situations, it is common that some signal peaks are missing, which leads to difficulties in determining the amino acid in that interval. Moreover, many noise peaks' intensities are larger than the intensities of signal peaks, which leads to the misidentification of signal peak and noise peak. Figure 2.6 shows examples of an ideal spectrum and a realistic spectrum.

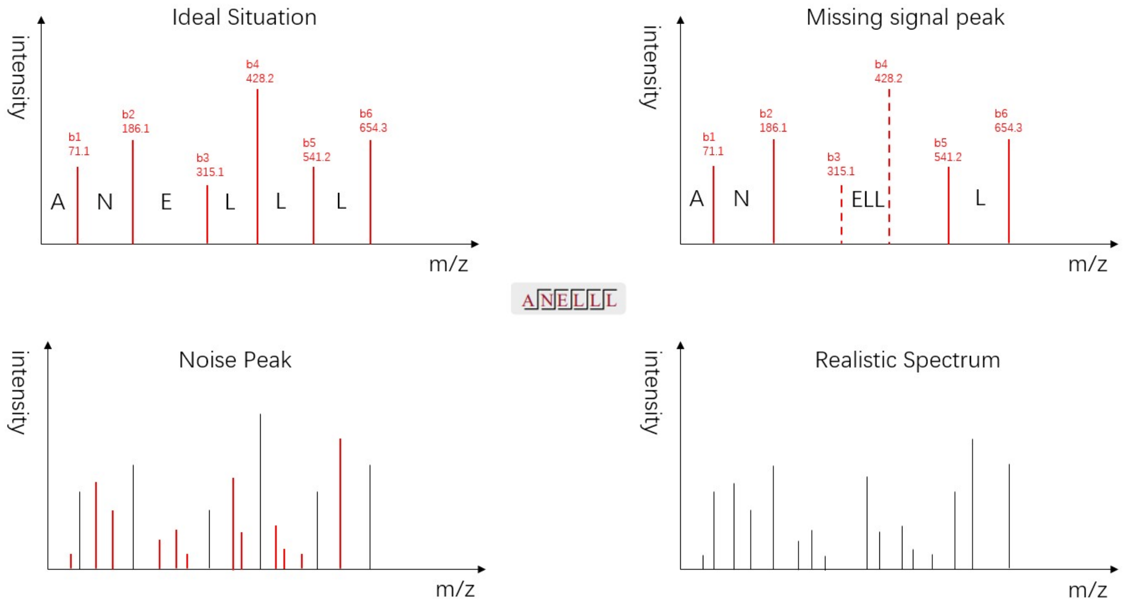


Figure 2.6: Tandem mass spectrum in different situations. The ideal situation presents the *b*-ions of the peptide 'ANELLL'. The missing signal peak situation presents an example that the b3 and b4 peaks are missing. And some noise peaks' intensities are higher than signal peaks' intensities.

In general, *De Novo* sequencing from a spectrum involves identifying a path containing solely y-ion peaks or *b*-ion peaks, starting from the beginning and ending at the precursor ion real mass. To create a path with indexes, the spectrum is treated as a one-dimensional histogram, referred to as a "mass map". The index of the mass map corresponds to the peaks' masses, while the values represent the intensities of the peaks. The length of a mass map depends on the peptide's mass and its resolution. Higher resolution leads to more accurate results. In this study, we consider a resolution of 0.01 Da. For example, if the maximum mass of the peptide is 1500 Da and the resolution is 0.01 Da, the length of the mass map is $1500 * 100 = 150,000$. The value stored in each bin of the mass map corresponds to the representative value of the m/z value at that specific position. If no peak is present at a particular position, the value stored in the corresponding bin will be zero.

2.1.4 Mass Deviation

In tandem mass spectrometry, the accuracy of the spectra would influence the accuracy. A common issue in this analysis is the deviations between the experimental mass and the theoretical (monoisotopic) mass of fragment ions. And the mass deviations of fragment ions can be attributed to the deviation of the experimental mass and the theoretical (monoisotopic) amino acids' masses. In the feature-extracting process in the deep learning model as well as determining the positions in the dynamic programming algorithm, the amino acid mass is utilized to identify the potential positions of the preceding and following signal peaks when provided with a signal peak. If directly using the monoisotopic mass to determine the position, some peaks that have some large mass deviation may not be extracted. Therefore, it is necessary to find the distribution of the mass deviations and set up a mass tolerance interval to ensure that all possible signal peaks even with some acceptable mass deviations would be correctly extracted in the process.

Then the experiment of the mass deviation is done on the above-mentioned dataset. We compute the mass differences of every two consecutive signal peaks in the spectra, which correspond to certain amino acid masses, and compared them with their monoisotopic masses. We counted 7438 mass deviations between the experimental and the theoretical mass values. It can be observed in Figure 2.7 that most of the deviation is less than ±0.01, to be more precise, 99.34% of the deviations are within ±0.01. Therefore, given a signal peak and an amino acid mass, the mass value difference between the calculated next signal peak and the real next signal peak will be within ±0.01. In other words, if peaks are found outside the mass tolerance interval, it is highly likely that they are not signal peaks of the corresponding ion.

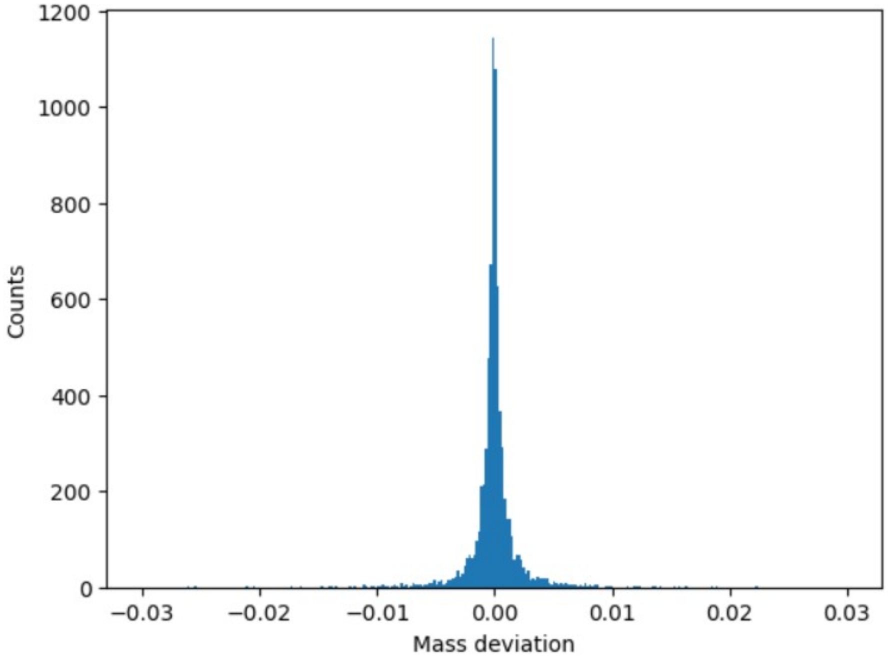


Figure 2.7: The distribution of the mass deviations between theoretical amino acid mass and experimental amino acid mass.

2.2 Single Position Probability Model

The intensity of a peak in the spectrum represents the abundance of the fragment. Typically, the intensities of signal peaks are usually higher than that of noise peaks. However, there are always some exceptions the signal peaks have lower intensity and noise peaks have higher intensity. Furthermore, the magnitude of the intensities in different spectra can vary significantly, sometimes differing by a factor of more than one thousand. Therefore, directly using the original intensity to train the network is not feasible. Some methods utilize nature logarithms to normalize the intensity. However, it is deemed necessary to obtain a more significant and standardized numerical value to provide a more meaningful and reliable representation of the data. The concept in this model involves utilizing a convolutional neural network (CNN) to capture spectral features. The spectrum can be approximated as an image, allowing the CNN model to effectively learn and extract these features. In this model, we are aiming to convert the original intensity to a more accurate value for better training results.

2.2.1 Feature Extraction

In *De Novo* sequencing, the primary information for determining an amino acid is the mass difference between two peaks that matches the mass of an amino acid. There are a total of 20 amino acids, with masses ranging from 57.02146 Da to 186.07931 Da, as well as three types of modifications. The first modification is "*Acm*" occurring on Cysteine, the chemical formula is C_3H_5NO, and the monoisotopic mass is 71.03711. The second modification is "*Acetyl*" occurring on N-terminus, Serine, and Threonine, the chemical formula is C_2H_2O, and the monoisotopic mass is 42.01056. The third modification is "*Oxidation*" occurring on Methionine and Tryptophan, and the monoisotopic mass is 15.99491. As a result, a total of 23 different ions are considered. Consequently, given a peak, referred to as the 'target peak', the positions of any amino acid that serves as either a preceding or following amino acid can be determined. If these two positions contain peaks, the probability of the target peak being a signal peak would increase. These features are extracted at the sequence level, and the length of the sequence feature is thus $20 + 3 = 23$. Both the preceding position and the following position need to be considered, resulting in a final sequence-level feature size of 46.

A *b* or *y*-ion may experience some neutral losses or isotopic ions during the fragmentation process. For each candidate fragment ion, we also consider these ions to extract the feature at the ion level. If a fragment ion is denoted as F, and the complementary ion of F is denoted as CF, then the ion-level features can be extracted at the following positions: F, CF, F-1 (-1 isotopic ion), F+1 ($+1$ isotopic ion), F^2 (+2 charges), F-H_2O, F-NH_3, CF-H_2O, CF-NH_3. The length of the ion-level feature is 9. Figure 2.8 is an example of the ion-level features and sequence-level features of a signal peak.

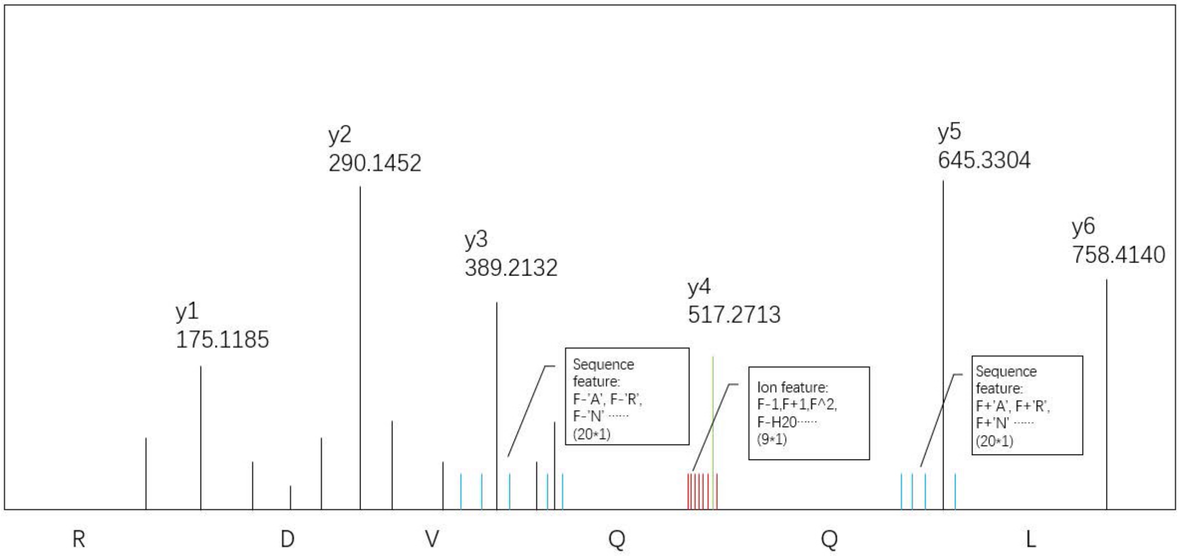


Figure 2.8: The feature map of the Single Position Probability Model. Take y4 as an example, the ion-level feature is represented in red lines, and the sequence-level feature is represented in blue lines. The blue lines and red lines in the figure only represent some of the features.

As previously mentioned, directly using the intensities to be the values of features is not appropriate. Therefore, after testing, it was found that using “intensity rank” to be the values yields good performance. To calculate the intensity rank, the peaks are first sorted in ascending order, and each peak is assigned a rank. Then, the intensity rank is calculated by dividing the assigned rank of each peak by the total number of peaks. If two peaks have exactly the same intensities, their ranks are the same. However, the situation that two peaks have the same intensities is very rare. Thus, these scores will be evenly distributed from zero to one. The intensity rank ensures that the value for each peak is between zero and one, regardless of the magnitude of the spectrum, and the peak with higher intensity would receive a relativity higher intensity rank. Then the intensity rank is substituted for the original mass value in the mass map.

During the feature extracting process of one peak, the feature is extracted based on both the ion-level feature position and the sequence-level feature position. By concatenating the ion-level feature with the two sequence-level features in the prescribed order, namely the ion-level feature followed by the two sequence-level features, one can obtain the final feature map for a single peak. Thus, the feature map should be a vector of length 55(9+23+23). Unlike DeepNovo, which extracts an intensity window of size 10 around the ion location [12], Dp-Novo only considers the windows of size 3. It is mentioned in Section 2.1.4 that most of the deviations of amino acid mass are less than ±0.01 in the HCD spectrum, For example, given a signal peak and an amino acid, the mass difference between the real next signal peak position and the calculated next signal peak position is less than ±0.01. Therefore, the real next signal peak should be either in the previous bin or in the next bin, which means that three bins (0.03 Da) are more than enough to cover all the possible deviations. Larger windows would include more noise peaks and lead to inaccuracy. If there are peaks outside the three bins, they are less likely to become the signal peaks of the corresponding ions. Therefore, extracting three bins around the mass value is precise enough for almost all the cases, and can reduce the probability that extracts noise peaks. It is worth mentioning that there may not be a large difference between the calculated position and real position for neutral losses or isotopic ions, but a size three is also extracted for these features to keep the size consistent. Therefore, the size of one concatenated feature map is (55,3).

2.2.2 Model Structure and Training

The model used here is a convolutional neural network. The first layer is a 1D convolution layer, and the size of the kernel is 3. The purpose of using the convolution layer here is to enable the kernel to discern the appropriate penalty to assign to different deviations. We want the model gives relatively more penalties to the peaks that have larger mass deviations and fewer penalties to the peaks that have deviations less than 0.01 Da. Then the output of the convolutional layer is a feature map with shape (55,1). It is worth noting that the distribution of deviations in the ion-level and sequence-level features may differ, and it may be more appropriate to use distinct kernels for each type of feature. However, this model only employs one kernel for convenience. Then the extracted features will be passed into a fully connected neural network to do the classification. The deep learning model used in Single Position Probability Model is a four-layer fully connected network. Each fully connected layer is followed by a *ReLU* activation function, and Dropout is also utilized to reduce overfitting. In the last dense layer, a *sigmoid* function is used to map the variable between 0 and 1. The number of neurons in the first fully connected layer is 32, and the number of neurons in the last layer is 1. With regard to the predicted results, a peak with a probability larger than 0.5 is considered a signal peak, and a peak with a probability less than or equal to 0.5 is considered a noise peak. Figure 2.9 shows the structure of the Single Position Probability model.

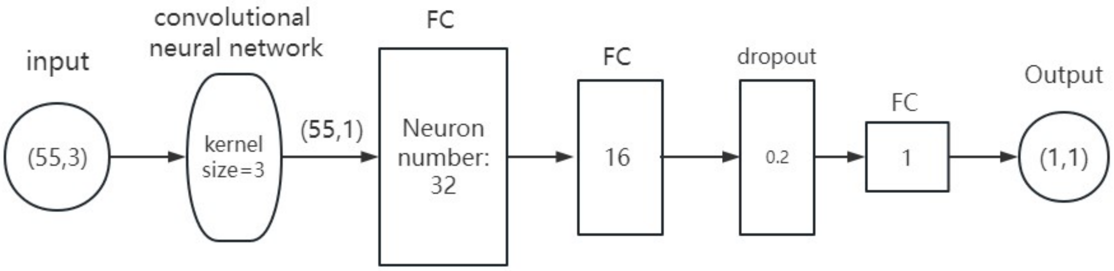


Figure 2.9: The structure of the Single Position Probability model.

The training set, validation set, and testing set are acquired from the NIST H. sapiens Orbitrap_HCD spectra. The training set is mainly used for training, the validation set is primarily used for evaluating the model's performance during the training process, and the testing set is used to evaluate the model's performance after the whole training process has been completed. We only consider the spectra in charge two and charge three for the training set, validation set, and testing set. The reason for only choosing charge-two and charge-three spectra will be explained in Section 2.3. By traversing the spectra in the training set, the feature of the noise peaks and the signal peaks can be obtained, and the label set can be acquired from the annotation of the peaks. Then the question is essentially a binary classification problem, the training data is the extracted (55,3) feature map, and the label is the type of the peak.

The TensorFlow Keras library is used to train the neural network model. The 'BinaryCrossentropy' function is used as the loss function, and the optimizer is Adam. The batch size is set to 128. And Early Stopping callback is used to avoid overfitting. The hyperparameters employed in this model have been carefully selected through multiple rounds of testing. It has been observed that variations in the hyperparameters do not have a substantial impact on the overall performance. The result of the training process can be seen in Supplementary 3. After training, this model is capable of assigning a probability value to each peak within a given spectrum, indicating the likelihood of that peak being a signal peak.

It is worth mentioning that we only extract the b-ion features and y-ion features for the training set of signal peaks, and regard a-ion as a noise peak. Because the number of consecutive a-ions is relatively small compared with the number of b-ion and y-ion, and there is no x-ion in the spectra. Additionally, as the dynamic programming process takes into account only a single type of ion path, y-ions are selected as the primary ions for consideration. As a result, the positions corresponding to 19 Da and the sum of the residue masses + 19 Da are assigned high initial values to ensure the complete features for the '$y1$' peak and the last signal peak.

2.2.3 Model Performance Evaluation

After training, the testing process is done on the testing set of NIST H. sapiens Orbitrap_HCD mass spectra. The testing set consists of 128 randomly chosen spectra, containing 12943 peaks. These spectra are all charge-two and charge-three. And Figure 2.10 shows the result of the predictions for signal peaks and noise peaks.

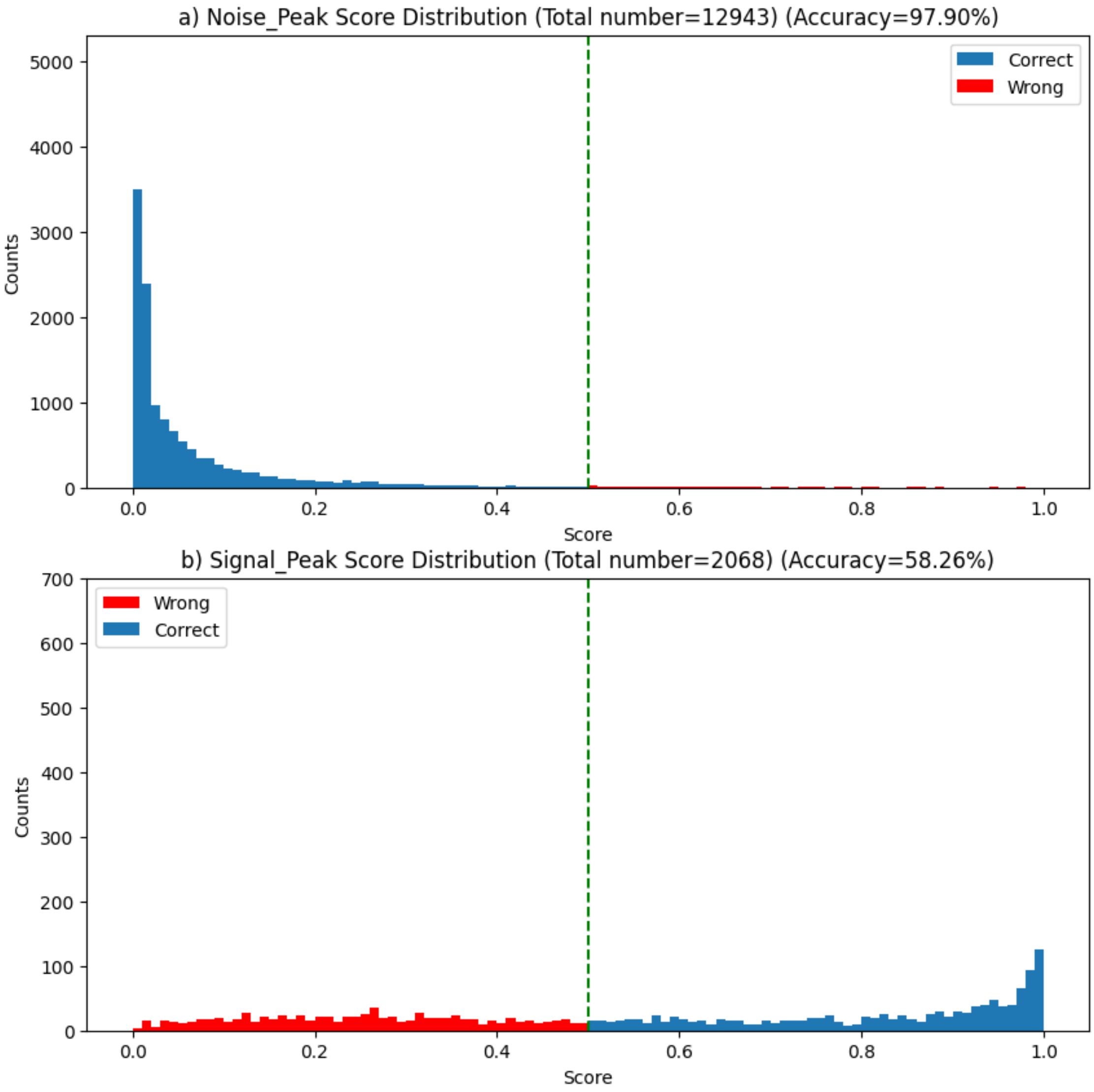


Figure 2.10: a) The Single Position Probability Model scores for noise peaks. b) The Single Position Probability Model scores for signal peaks.

In Figure 2.10 a) and b), the accuracy is defined as the number of correctly identified peaks divided by the total number of peaks of a certain type. The x-axis represents the score and the y-axis represents the number of peaks falling into certain intervals. In this task, any peak with a score larger than 0.5 is regarded as a signal peak, and any peak with a score less than or equal to 0.5 is regarded as a noise peak. The blue bars represent the correctly predicted peaks. Figure a) represents the distribution of scores of 12943 noise peaks, with an accuracy of 97.90%, and 81.5% of the noise peak scores are smaller than 0.05. And Figure b) represents the scores of 2068 signal peaks, with an accuracy of 58.26%.

Directly using the result from the Single Position Probability Model may not be good enough to find the final path. Since 41% of signal peaks are incorrectly identified as noise peaks. But if we compare the distribution of the original intensities, the ranks, and the probabilities, we can find that the distribution of peak probability improved a lot concerning distinguishing signal peaks and noise peaks.

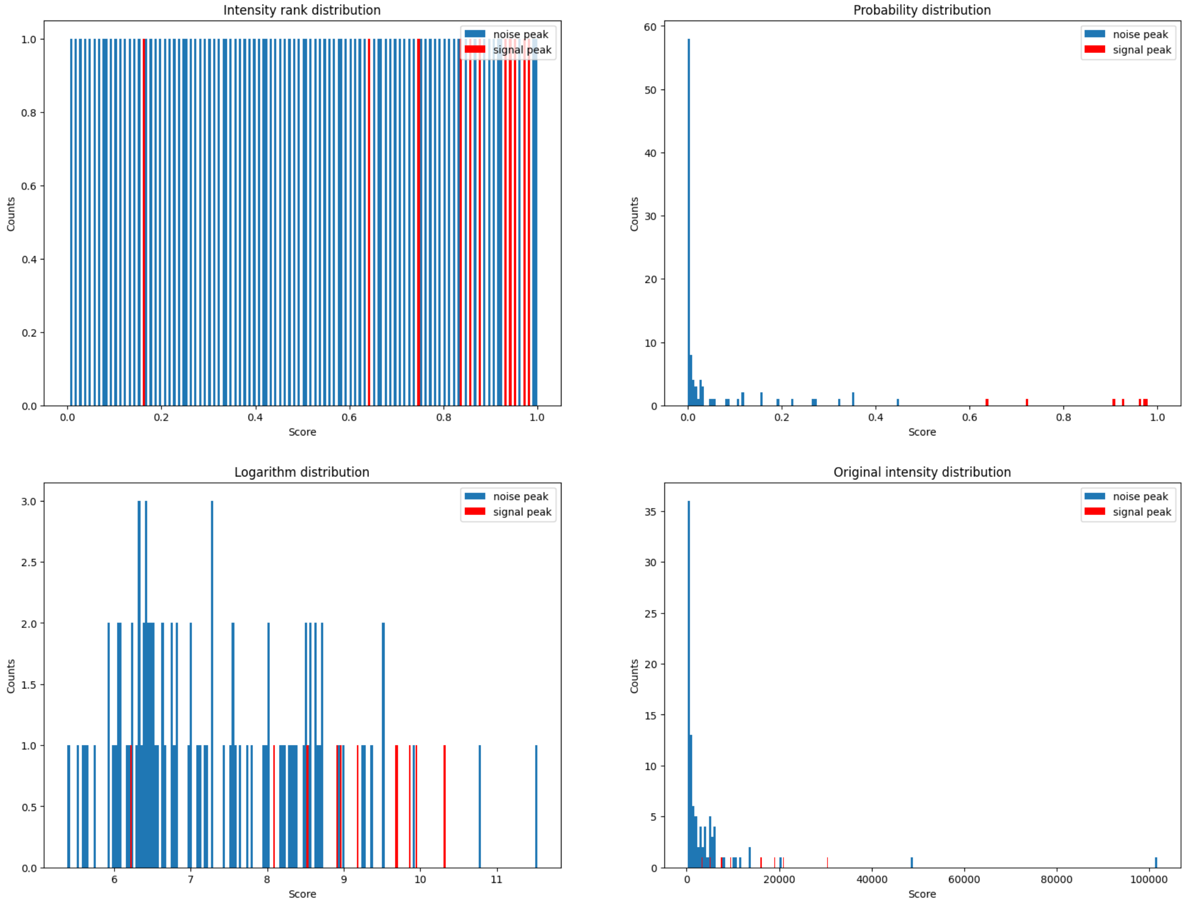


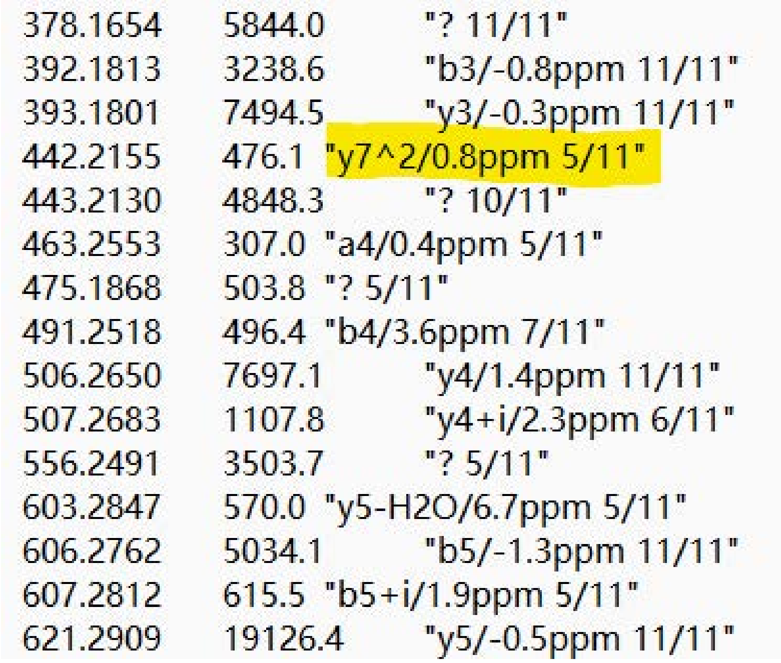
Figure 2.11: The figures presented in this study illustrate the distribution of scores for both signal and noise peaks, using four different methods: intensity rank, transformed intensity (probability), logarithm, and original intensity for one testing spectrum. Each figure includes two types of lines, red indicating signal peaks and blue indicating noise peaks. The *x*-axis of each figure represents the respective scores, while the *y*-axis represents the number of peaks within a given interval.

As illustrated in Figure 2.11, the intensity rank can restrict the value from 0 to 1, however, the values of a large number of noise peaks are higher than that of signal peaks. Although logarithmic transformation can mitigate the disparities in intensities, it cannot confine the values within a fixed interval for different spectra. Compared to other intensity representations such as Intensity Rank, original intensity, and logarithmically transformed intensity, the probability distribution has been found to produce more accurate representations for distinguishing the type of peaks. The probability scores displayed in the top-right figure distinctly separate the values for noise peaks and signal peaks. In summary, this model preprocesses intensities to generate more meaningful values, and in the next model, the intensity values of a peak in the mass map are replaced by the probability of that peak being a signal peak, thereby enhancing the accuracy of the features.

2.3 Reconstruction of High-charge Peaks

In MS/MS spectrometry, a spectrum can be in different charge states. When the precursor ion mass is large or the charge of the precursor ion is large, the fragment ions are more likely to get additional charges. However, when a fragment ion acquires more than one charge, the position of the corresponding peak may not be consecutive with the peaks of charge-one ions. The m/z value of charge-two ions can be calculated using their definition. Furthermore, the feature extraction process in deep learning models identifies the positions of the features based on the charge-one mass value, implying that the model may not accurately extract the features of charge-two ions. As a result, it is essential to reconstruct the charge-one peaks for those charge-two ions.

To address this issue, high-charge peaks must be repositioned to their charge-one positions, forming a consecutive ladder. This thesis focuses primarily on charge-two and charge-three spectra. When the charge of the precursor ion is two or three, more than 95% of the high-charged(more than one) fragmentation is charge-two. While some algorithms can transform high-charge spectra into charge-two spectra, this approach can also accommodate this scenario.



```
378.1654    5844.0        "? 11/11"
392.1813    3238.6        "b3/-0.8ppm 11/11"
393.1801    7494.5        "y3/-0.3ppm 11/11"
442.2155    476.1  "y7^2/0.8ppm 5/11"
443.2130    4848.3        "? 10/11"
463.2553    307.0  "a4/0.4ppm 5/11"
475.1868    503.8  "? 5/11"
491.2518    496.4  "b4/3.6ppm 7/11"
506.2650    7697.1        "y4/1.4ppm 11/11"
507.2683    1107.8        "y4+i/2.3ppm 6/11"
556.2491    3503.7        "? 5/11"
603.2847    570.0  "y5-H2O/6.7ppm 5/11"
606.2762    5034.1        "b5/-1.3ppm 11/11"
607.2812    615.5  "b5+i/1.9ppm 5/11"
621.2909    19126.4       "y5/-0.5ppm 11/11"
```

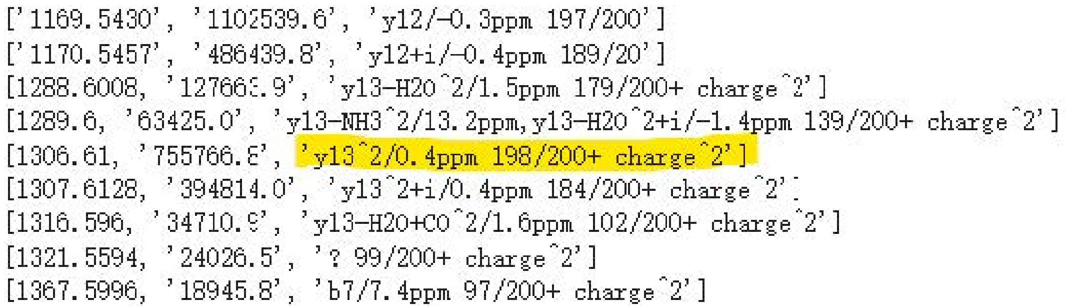
Figure 2.12: The charge-two peak in a spectrum.

It is not appropriate to arbitrarily determine which peaks are charge-two peaks; instead, it is crucial to devise a method for identifying peaks more likely to become charge-two ion peaks. Observing various peaks reveals that as the length of an amino acid fragment chain increases, it is more likely to retain additional charges. Furthermore, it has been found that for long peptides, the charge-one ladder is interrupted at three-quarters of the sequence, with the remaining portion generating another charge-two ladder. Consequently, the position where the charge-one ladder terminates is usually the charge-one signal peak with the largest m/z value. By utilizing the probability calculated by the Single Position Probability Model, the largest

m/z value with a probability greater than 0.5 on the original spectrum can be computed. As a result, it can be assumed that when the m/z value of an ion exceeds this threshold, the ion is more likely to obtain more than one charge and form a charge-two peak.

If a peak is considered to be a charge-two peak, the charge-one m/z value can be calculated according to the definition of mass-to-charge value. To reconstruct the charge-one peaks for the possible charge-two peaks in the spectrum, all the peaks are assumed as charge-two peaks first. Through the definition, the theoretical charge-one m/z can be computed. For all the assumed charge-two peaks if their computed charge-one m/z is exceeding the largest m/z value of a predicted signal peak, then the model will reconstruct their corresponding charge-one peaks. All peaks in the original spectrum will be retained, and the reconstructed peaks will be added additionally to the original spectrum.

Although it is acknowledged that several noise peaks may be incorrectly reconstructed, it can be deemed justifiable when taking into account the potential drawbacks of overlooking all signal peaks of charge-two, in comparison to the cost incurred. And with the help of scores, the reconstructed interval can be limited to a very narrow range. It is worth mentioning that the intensities of these reconstructed peaks are equal to their original charge-two peaks' intensities. And if there is already a peak in the reconstructed position, the intensity will be accumulated.



```
['1169.5430', '1102539.6', 'y12/-0.3ppm 197/200']
['1170.5457', '486439.8', 'y12+i/-0.4ppm 189/20']
[1288.6008, '127663.9', 'y13-H2O^2/1.5ppm 179/200+ charge^2']
[1289.6, '63425.0', 'y13-NH3^2/13.2ppm,y13-H2O^2+i/-1.4ppm 139/200+ charge^2']
[1306.61, '755766.8', 'y13^2/0.4ppm 198/200+ charge^2']
[1307.6128, '394814.0', 'y13^2+i/0.4ppm 184/200+ charge^2']
[1316.596, '34710.9', 'y13-H2O+CO^2/1.6ppm 102/200+ charge^2']
[1321.5594, '24026.5', '? 99/200+ charge^2']
[1367.5996, '18945.8', 'b7/7.4ppm 97/200+ charge^2']
```

Figure 2.13: The reconstructed charge-one peak of charge-two ion. The annotation marked with "+ charge^2" represents the reconstructed peaks.

Last, by implementing the same feature extracting and predicting process on the newly reconstructed spectrum, each peak in the reconstructed spectrum will be assigned a probability, and the final mass map can be calculated. The value in the mass map represents the probability that a peak is a signal peak, and this mass map will be used in the following model.

2.4 Dual Position Scoring Model

The Single Position Probability Model successfully updates the values in the mas map into more accurate probabilities. Although the accuracy of the prediction is still not very high, the updated numbers would be extremely beneficial to further model training. To a certain extent, using the probability as the value of the mas map is like the step of reducing the noise pixel in the image processing, which lets the noise peaks have smaller scores and signal peaks have higher scores.

In this section, we proposed a new concept: the probability of a certain amino acid being located between two positions. By deducting the mass of a given amino acid from a target position, the preceding position can be calculated. Then the probability is the likelihood that the amino acid is located between the target position and the preceding position, in other words, the target position and preceding position contain consecutive signal peaks. For example, given a target peak, the preceding position of each amino acid can be computed. Then the probability that each amino acid serves as the preceding amino acid can be computed. The preceding amino acid of a target peak means that the target peak is a signal peak and the amino acid sequence fragment of the target peak ends with that amino acid.

2.4.1 Feature Extraction

The approach to determining the probability is similar to the previous Single Position Probability Model. For convenience, we call the position we prepare to predict: the 'target position', and the possible position for each preceding amino acid: the 'preceding position'. Instead of just extracting one position's ion feature and sequence feature, this model considers the target position and the preceding position at the same time, which greatly improves the accuracy. The probability is relatively high when both positions contain peaks, and low when there is only one or no peak in these two positions. Given a position, if we calculate the probabilities for all 23 ions, the largest probability among these 23 probabilities can be considered the probability of the peak being a signal peak.

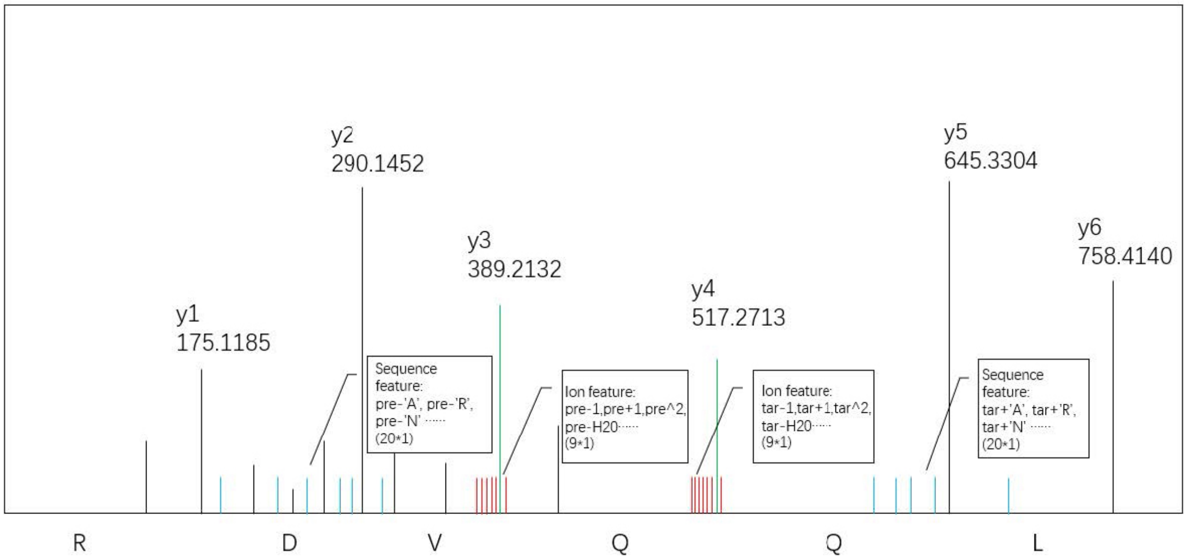

y2
290.1452
y5
645.3304
y3
389.2132
y6
758.4140
y4
517.2713
y1
175.1185
Sequence feature: pre-'A', pre-'R', pre-'N' ······ (20*1)
Ion feature: pre-1,pre+1,pre^2, pre-H20······ (9*1)
Ion feature: tar-1,tar+1,tar^2, tar-H20······ (9*1)
Sequence feature: tar+'A', tar+'R', tar+'N' ······ (20*1)
R
D
V
Q
Q
L

Figure 2.14: The feature map of the Dual Position Scoring model. Take y4 as an example, when we need to calculate the probability that glutamine is located between y3 and y4, the ion features of y3 and y4 are represented by the red lines. And the preceding sequence feature of y3 and the following sequence feature of y4 are represented in blue lines.

The ion-level feature contains the ion features for the target position and the ion feature for the preceding position. And a size three window around the mass value will be extracted. The ion feature size for each position is (9,3), and after combining them, the total ion feature is a vector of size (18,3) with the predicted probabilities as values. The sequence feature considers the preceding amino acid of the preceding position and the following amino acid of the target position. Similar to the Single Position Probability model, each sequence feature is a (23,3) vector, and after concatenation, it becomes a (46,3) vector. After combining the ion-level feature and sequence-level feature, the final feature map for a given target position and one of the amino acids is a vector of shape (64,3). If the calculated positions of some features are less than 0 or larger than the length of the mass map, 0 will be the value. Figure 2.14 shows an example of the sequence feature and ion feature of a target position y4.

2.4.2 Model Structure and Training

The structure of the Dual Position Scoring model is similar to the previous one and is shown in Figure 2.15. The only change is adding the neuron number in the first dense layer to 64. A convolution layer is located at first to learn the penalty given to the feature in the extracted window. And a deep neural network is used to learn the features, consisting of three fully connected layers. Still, the *Sigmoid* function is used in the last dense layer to map the result from 0 to 1. The training process is similar to the previous model. The ‘BinaryCrossentropy’ function is used as the loss function, and the optimizer is Adam. The batch size is set to 128.

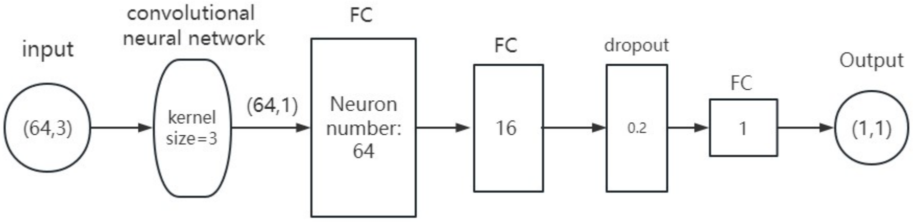


Figure 2.15: The structure of the Dual Position Scoring model.

The next step involves acquiring the training data. Unlike the previous model, the Dual Position Scoring model requires generating and simulating all possible features that could potentially be present in the actual experiment to be used as training data. For better understanding, the two positions’ situation is described using a pair (A, B), where B represents the target position, and A is the preceding position. In this context, S refers to signal peaks, N refers to noise peaks, and ”+” and ”-” denote containing peaks and not containing peaks, respectively. For example, if there is a signal peak in the preceding position and no peak in the target position, the situation can be described as $(S+, -)$. We use labels 1 and 0 to represent if there is an amino acid located between this interval, and it can still be regarded as a binary classification task. It is noteworthy to mention that the phenomenon of certain signal peaks being absent is a commonly observed occurrence. However, the likelihood of two consecutive peaks being absent is exceedingly low. For the possible positive label, the possible situation pairs are $(S+, S+)$, $(-, S+)$. The possible negative label pairs include: $(N+, N+)$, $(-, N+)$, $(-, S+)$, $(S+, N+)$, $(N+, S+)$. The training data should include all these types of features. And the training set, testing set, and validation set are still acquired from NIST H. sapiens Orbitrap_HCD spectra. The situation that $(S+,-)$ is not considered here for a positive label, reason is that we only extract features and do the prediction where there is a peak in the target position to reduce the time for the predicting process. Therefore, there will not be any situation like $(S+,-)$ or $(N+,-)$.

2.4.3 Model Performance Evaluation

After training, we conducted tests on the same testing set in Figure 2.10, containing 128 spectra. Given a peak, if we calculate the probability that each amino acid serves as the preceding amino acid, and get the max value among these probabilities, we can say that this max probability is the probability that this peak is a signal peak. For each peak in the spectrum, we extracted 23 different features and predicted the result for each feature separately. Therefore, the same as the previous result, every peak would have a probability of being a signal peak.

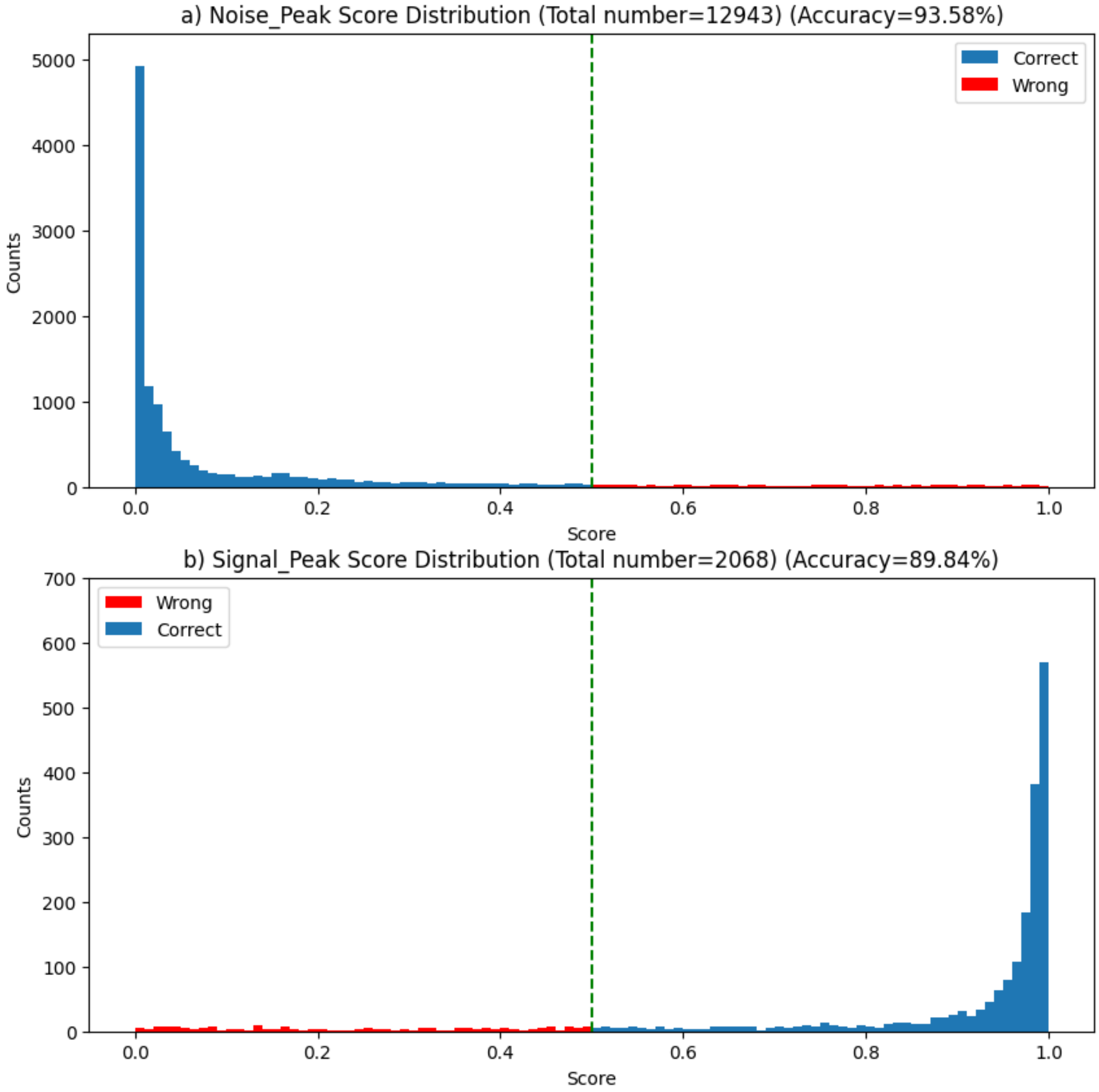


Figure 2.16: The result of the Dual Position Scoring model for noise peaks and signal peaks.

From Figure 2.16, we can find that the accuracy of the prediction of noise peaks is 93.58% which experienced a small decrease compared with the previous model. However, the accuracy of the prediction of signal peak significantly increases from 58% to 89.84%. Compared with the Single Position Probability model, more signal peaks got scores larger than 0.9, to be more specific, 73.5% signal peaks scores are within the interval from 0.9-1. The mass distribution of the wrong prediction of the signal peak can be seen in Supplementary 4. With regard to the result of noise peak, 78.4% noise peaks' scores are less than 0.1. It must be acknowledged that more noise peaks get scores larger than 0.5, and it is the cost of improving the precision of signal peak prediction. Nevertheless, it is important to note that the accuracy of this model does not necessarily reflect the accuracy of the final prediction. For instance, even if the accuracy of detecting signal peaks and noise peaks individually reaches 100%, the final prediction may still be incorrect. This is due to the possibility of missing signal peaks in certain situations. It can be found that the result of the Dual Position Scoring model is much better than the result of the Single Position Probability model. It would be more proper to utilize these scores for the dynamic programming algorithms.

However, one problem that remains is that it cannot distinguish the signal peak of *y*-ion, *b*-ion, which is one of the problems that will lead to wrong predictions in the dynamic programming process. As Figure 2.17 shows, both *b*-ion peak and *y*-ion peak receive high scores, it is hard to distinguish the *b*-ion and *y*-ion through the score; while *a*-ions receive extremely low scores, mainly because there are not too many consecutive *a*-ions. It is worth mentioning that the low accuracy for predicting *a*-ion is beneficial because only *y*-ions are considered the main ions to find the path in dynamic programming.

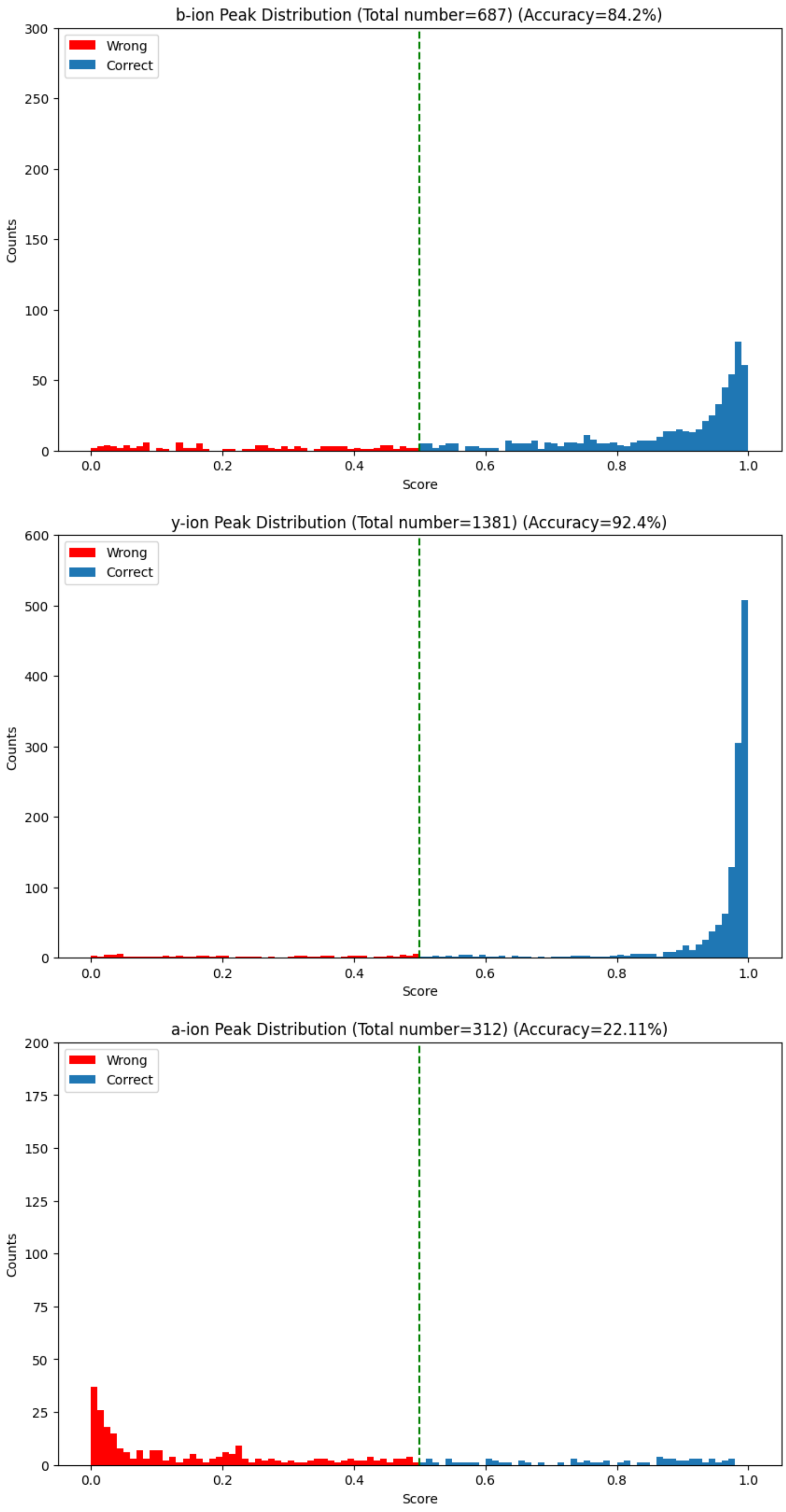


Figure 2.17: The distribution of scores of *y*-ion, *b*-ion and *a*-ion.

2.5 Dynamic programming

In *De Novo* sequencing, the objective is to predict the peptide sequence from a spectrum. After the above-mentioned process, for each peak, the probability that each amino acid is the preceding amino acid is computed, and the probabilities can be regarded as scores. By considering the scores of each peak, a final prediction is made by selecting the path that has the highest cumulative score of peaks. The main idea of dynamic programming in DpNovo is computing and saving the largest accumulated scores up to each mass value until the precursor ion mass. Finally, backtracking from the precursor ion mass can be performed to determine the final prediction amino acid path.

In dynamic programming, a table is used to store the intermediate result of subproblems. By constructing a dynamic programming matrix dp that has the same length as the mass map, the accumulated scores on the previous path ending at this position can be represented by the value in each cell in the matrix dp. Once all values in the dynamic programming matrix have been calculated, backtracking can be performed from the mass of the precursor ion.

In the Orbitrap_HCD spectra, most signal peaks are the y-ion peaks and the b-ion peaks, and there are a very small number of a-ions. As illustrated in Figure 2.4, it is evident that the y-ion constitutes the predominant ion species present within the spectra. Therefore, in this thesis, the dynamic programming algorithm is aiming to find the y-ion path.

Based on the aforementioned model, the likelihood of each peak serving as the endpoint for a particular amino acid can be computed. As a result, for every peak, a (23, 1) vector can be computed, indicating the probability that a certain preceding amino acid terminates at that position. Then these probability vectors will be saved in the corresponding position in a new array called Θ. And the length of Θ is equal to the length of dp. Θ is only for saving the result from the Dual Position Scoring model.

Given a fragment P, the residue mass of the fragment can be denoted as $\mid P \mid$. For a m/z value M, and the 0.01 Da as the resolution γ, the corresponding position in the mass map can be denoted as M_γ. Considering the final path only contains the charge-one peaks, from the above calculation, the offset of a y-ion is 19. Given a precursor ion P, the starting position of this path is 19_γ, and the ending position of this path is $(\ 19 + \mid P \mid\ )_\gamma$. The 23 amino acids alphabetic Ω is sorted in alphabetical order $\{A, R, N, D, B, C, Q, E, Z, G, H, I, L, K, M, F, P, S, T, W, Y, V, A*, C*, M*\}$. With the Dual Position Scoring model and the reconstructed mass map with values, given a mass i, its position i_γ, the probability of each peak serving as the endpoint for a certain amino acid can be computed, and the result is an array of size (23,1), saved in $\Theta[i_\gamma]$. If there is no peak of mass i, $\Theta[i_\gamma]$ is a zero vector with a size of 23. Given a $\lambda \in [0, 22]$, $\Theta[i_\gamma][\lambda]$ denotes the probability that the amino acid $\Omega[\lambda]$ is the preceding amino acid. To be more specific, there is an amino acid $\Omega[\lambda]$, starting at position $(i - \mid \Omega[\lambda] \mid)_\gamma$ and ending at position i_γ, where $\mid \Omega[\lambda] \mid$ is the corresponding amino acid mass. The maximum index of dp and Θ is both $(19 + \mid P \mid)_\gamma$.

Given a dynamic programming matrix dp and a position, each cell represents the maximum accumulated scores ending at this position. For every position i_γ in dp, $dp[i_\gamma]$ can be calculated as:

$$dp[i_\gamma] = \max_{\lambda \in [0,22]} (\ dp[(i - \mid \Omega[\lambda] \mid)_\gamma] + \Theta[i_\gamma][\lambda]\ ) + penalty + 0.01 * \max(\Theta[i_{C\gamma}]) \qquad (2.3)$$

Each cell in dp also records which preceding amino acid contributes the largest score, making it easier for the algorithm to backtrack. A penalty is applied in cases where no peak is present in i_γ, with the assigned penalty set at -1.0. The result of other penalties can be seen in supplementary 5. $i_{C\gamma}$ presents the complementary position of i_γ, and can be computed by Equation 2.1. The last term presents the maximum probability score in the complementary position in Θ, multiplied by a minor factor of 0.01. This is because when certain y-ion peaks are absent, the situation in the complementary b-ion position can be assessed. However, it is intended for the dynamic programming to primarily identify paths based on y-ions, with b-ions only influencing the prediction when y-ion peaks are missing.

What is noteworthy is that the majority of HCD spectra end with arginine or lysine in the C-terminal, because trypsin cleaves peptide bonds on the carboxyl side of lysine (K) or arginine (R) amino acid residues. So $dp[(19 + \mid arginine \mid)_\gamma]$, $dp[(19 + \mid lysine \mid)_\gamma]$ would be given relatively high initial value. And the starting position $dp[19_\gamma]$ should also be given a high initial value to make sure the y-ion path ends at 19_γ instead of the offset of b-ion 1_γ. Then from the ending position $(19 + \mid P \mid)_\gamma$, the backtracking can be done in dp by finding which preceding amino acid contribute the largest score, and the preceding amino acid can be decided. Recursively do the backtracking until the position is 19_γ, and the final path can be found.

In tandem mass spectra, it is a common situation that a signal peak and its complementary-ion peak are both missing. According to the algorithm given above, if both peaks are missing and we are calculating the missing peak's dp value, it is possible that several paths have the same score. For example, given an amino acid sequence "DIYVDMK", if the peak y3 is missing and the complementary-ion peak b5 is missing, then the two sequences "DIYVDMK" and "DIYDVMK" will have the same score. With the help of dynamic programming, we can know the mass of the uncertain mass interval, for example, "DIY(214.09)MK", and find all possible amino acid sequences that satisfied that mass value. But in further experiments, only one peptide is sequence chosen to be the final prediction, which is obtained by selecting the amino acids according to their ranking within the amino acid alphabetic Ω.

However, one drawback of this dynamic programming algorithm is that when γ is set to 0.01 Da and the mass of the precursor ion is, for example, 1500 Da, then a total of 150,000 dp values need to be computed. This results in a significant amount of computing time.

Chapter 3

Result

3.1 Dataset

The dataset for training is NIST *H.sapiens* Orbitrap_HCD, Library 1 (best): human_hcd_tryp _best (high-quality spectra), containing 398373 spectra. And the dataset can be found and downloaded at https://chemdata. nist.gov/ [41]. The training set contains about 250 randomly chosen spectra. All these training spectra are charge-two and charge-three.

The testing process was carried out using two databases: the NIST database and the ProteomeXchange database. For the NIST database, only charge-two and charge-three spectra were considered, whereas all spectra were included in the ProteomeXchange database testing. The experiments were conducted on three species (*H.sapiens*, *M.musculus*, and *C.griseus*) from the NIST database, while the testing was only carried out on *H.sapiens* in the ProteomeXchange database.

Due to the relatively low computing speed, testing all the spectra in one dataset would cost an extremely large time. Therefore in this chapter, the test of one dataset was carried out only on a subset of the original dataset. It is expected that the results obtained from this subset will provide an approximation of the entire dataset.

3.2 Evaluation Criteria

To check the performance of the model and compare it with other models, we use the same evaluation metrics proposed in DeepNovo [12]. There are three evaluation metrics: Amino Acid Recall (AAR), Amino Acid Precision (AAP), and Peptide Recall (PR). And these metrics can be calculated as follow:

$$
\begin{aligned}
AAR &= (AA\_recall)/(AA\_target)\\
AAP &= (AA\_recall)/(AA\_predicted)\\
PR &= (peptide\_recall)/(peptide\_target)
\end{aligned}
$$

Where the AA_recall presents the number of correctly predicted amino acids, AA_target presents the number of target amino acids and AA_predecited presents the number of predicted amino acids. For a predicted amino acid to be classified as a ”match” with a real amino acid, its mass must differ by no more than 0.1 Da, and the masses of the prefix preceding it must differ by no more than 0.5 Da. Peptide_recall represents the number of correctly predicted peptides, and peptide_target presents the number of target peptides. The peptide_recall is the fraction of real peptide sequences that were fully correctly predicted [12].

3.3 Experiment Result

3.3.1 Result of different peptide Lengths

When comparing the effect of peptide lengths, we tested a subset containing 500 randomly chosen *H.sapiens* Orbitrap_HCD spectra from the NIST database. The result of different lengths results is listed in Table 3.1.

Table 3.1: Experimental results of different lengths of peptides

	$X = 6$	$X = 7$	$X = 8$	$X = 9$	$X = 10$	$X = 11$	$X = 12$	$X = 13$	$X = 14$	$X = 15$	$X = 16$	$X = 17$	$X = 18$	$X = 19$	$X = 20$	$X = 21$	$X = 22$	$X = 23$
AAR	0.944	0.896	0.812	0.962	0.888	0.852	0.916	0.783	0.807	0.721	0.627	0.638	0.768	0.621	0.579	0.768	0.732	0.689
AAP	0.918	0.884	0.78	0.971	0.872	0.845	0.925	0.772	0.812	0.73	0.627	0.632	0.793	0.643	0.583	0.795	0.718	0.694
PR	0.865	0.727	0.636	0.65	0.588	0.51	0.557	0.565	0.4	0.273	0.106	0.087	0.1	0	0	0	0	0

It can be found that the amino acid recall and precision do not exhibit a significant change when the length is larger than 15, however, the peptide recalls decrease significantly with the increasing length. The low peptide recall is mainly because when the length of the peptide is getting longer, the situation that a certain ion's peak and its complementary ion's peak are both missing appears more frequently. This situation will cause many paths to have the same scores when calculating the dp matrix. And the algorithm just randomly chooses one path to be the final prediction. Therefore, if there are many missing peaks, the peptide recall would be considerably low. It is worth noting that a significant proportion of the incorrect prediction can be attributed to the selection of a random path on the missing peak interval.

3.3.2 Result on different species

On the NIST website, different species' spectra can be downloaded. We had tests on subsets of *H.sapiens*, *C.griseus*, and *M.musculus* respectively, and each subset contains 500 randomly chosen spectra from that dataset. And use the evaluation criteria to evaluate the performance of the model. All the spectra in the tests are charge-two or charge-three. The testing result is shown in Figure 3.1.

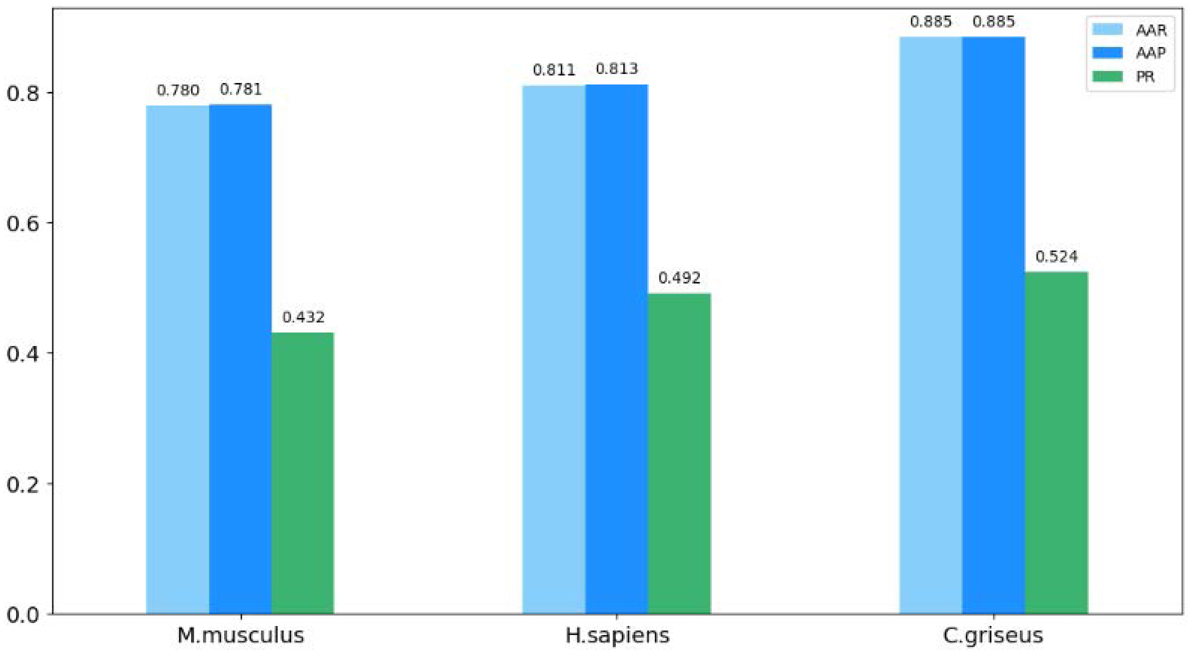


Figure 3.1: The Amino Acid recall, Amino Acid precision, and Peptide Recall of three species: *M.musculus*, *H.sapiens*, and *C.griseus*. They all use Orbitrap-HCD as the instrument.

Only these three species have the Oribitrap-HCD spectrum on the website, other species only have the Ion Trap spectrum. The Ion Trap spectrum cannot reach the resolution of 0.01, so these species are not counted. As the result shows, the model is trained on the *H.sapiens* spectrum dataset but gets even higher amino acid recall and accuracy on the *C.griseus* dataset, indicating that this model can be applied to different species. The close similarity between the amino acid recall and amino acid precision values can be attributed to the fact that the predicted amino acid sequences are of similar length or equal to the target sequence.

3.4 Comparison of the result with other *De Novo* sequencing models

In this chapter, the performance of this model will be compared with other *De Novo* sequencing models, for example, PEAKS [19], Novor [45], PepNovo [18], and DeepNovo [12]. And the result of these models is referenced from "*De Novo* peptide sequencing by deep learning" [12]. The testing dataset used here is from the ProteomeXchange database [42]. And data was acquired from the Thermo Scientific Orbitrap Fusion with the higher-energy collisional dissociation (HCD) technique. We only compare the result of one specie *H.sapiens*, and the accession no. is PXD004424.

We randomly selected a subset of 500 spectra to be the testing set for comparison with the other four models listed in Figure 3.2. It should be noted that [18] did not specify the number of spectra in the testing set, and we assumed that the results in [18] were obtained using the entire dataset. Although the test on DpNovo was conducted on only 500 spectra, the results can still be considered an approximation.

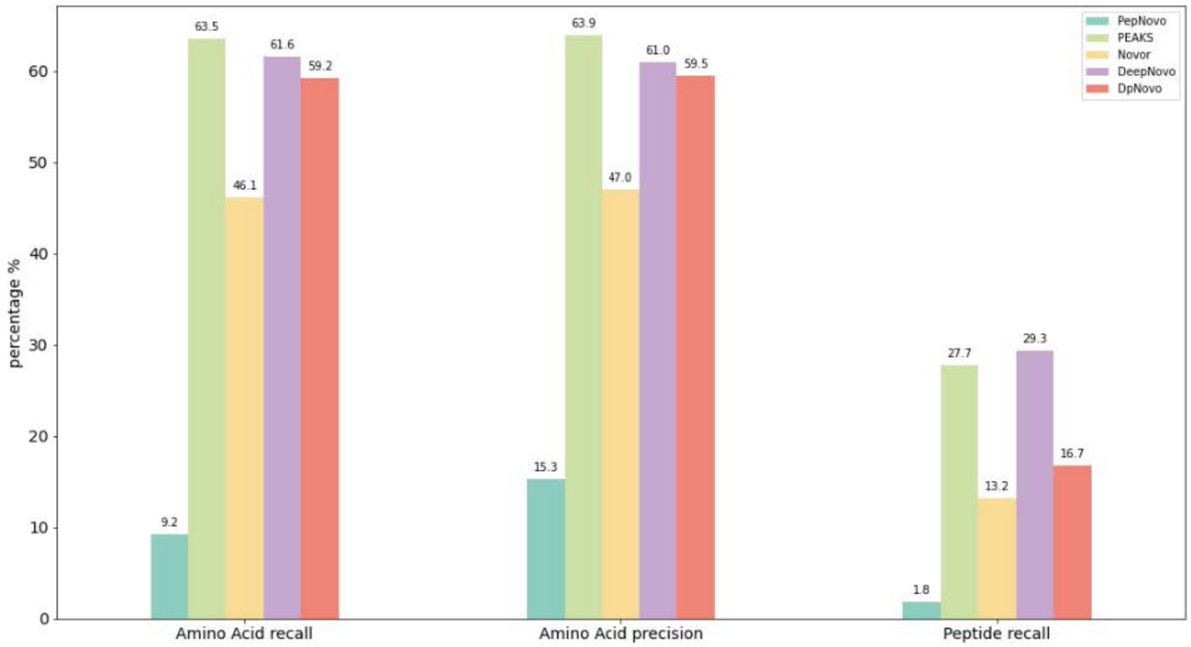
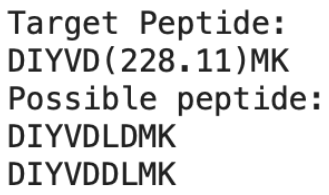


Figure 3.2: The Amino Acid recall, Amino Acid precision, and Peptide Recall of a subset, and compared with the other 4 models on *H.sapiens* HCD Ms/Ms spectra from the ProteomeXchange database.

From the result, it can be observed that DpNovo performs comparably with PEAKS in both 'Amino Acid recall' and 'Amino Acid precision' metrics, reaching 59.2% and 59.5% respectively, with a difference of approximately 4.3% and 4.4%. Concerning 'Peptide recall', DpNovo reached 16.7%. The low peptide recall is mainly because in the spectra in the ProteomeXchange database, the number of missing signal peaks is increasing, compared with the spectra in the NIST database. Therefore, when a signal peak and its complementary peak are both missing, simply randomly selecting one possible sequence to be the final prediction will lead to the low peptide recall. In our model, there is no additional post-processing step when multiple candidates are present. However, it is worth noting that certain commercial software, such as PEAKS, often employ post-processing steps that utilize biological approaches to refine the selection and identify the most optimal candidate from the available options. If our model combines those post-procession steps, the result must be better. Therefore, in the situation that we only randomly select one possible sequence to be the final prediction, the result is relatively satisfactory.

3.5 Replace Unsure Mass Interval with Mass Value

In the MS/MS spectrum, when a signal peak and the complementary-ion peak are both missing, using the dynamic program algorithm would lead to a situation that multiple paths have the same scores. However, in the dynamic programming process, all possible amino acid sequence paths which have the same highest scores can be acquired. To be more rigorous and accurate, the unsure mass interval can be replaced with the corresponding mass value.



```
Target Peptide:
DIYVD(228.11)MK
Possible peptide:
DIYVDLDMK
DIYVDDLMK
```

Figure 3.3: The peptide sequence prediction with a mass value and all the possible sequences.

3.6 Conclusion and Future Research

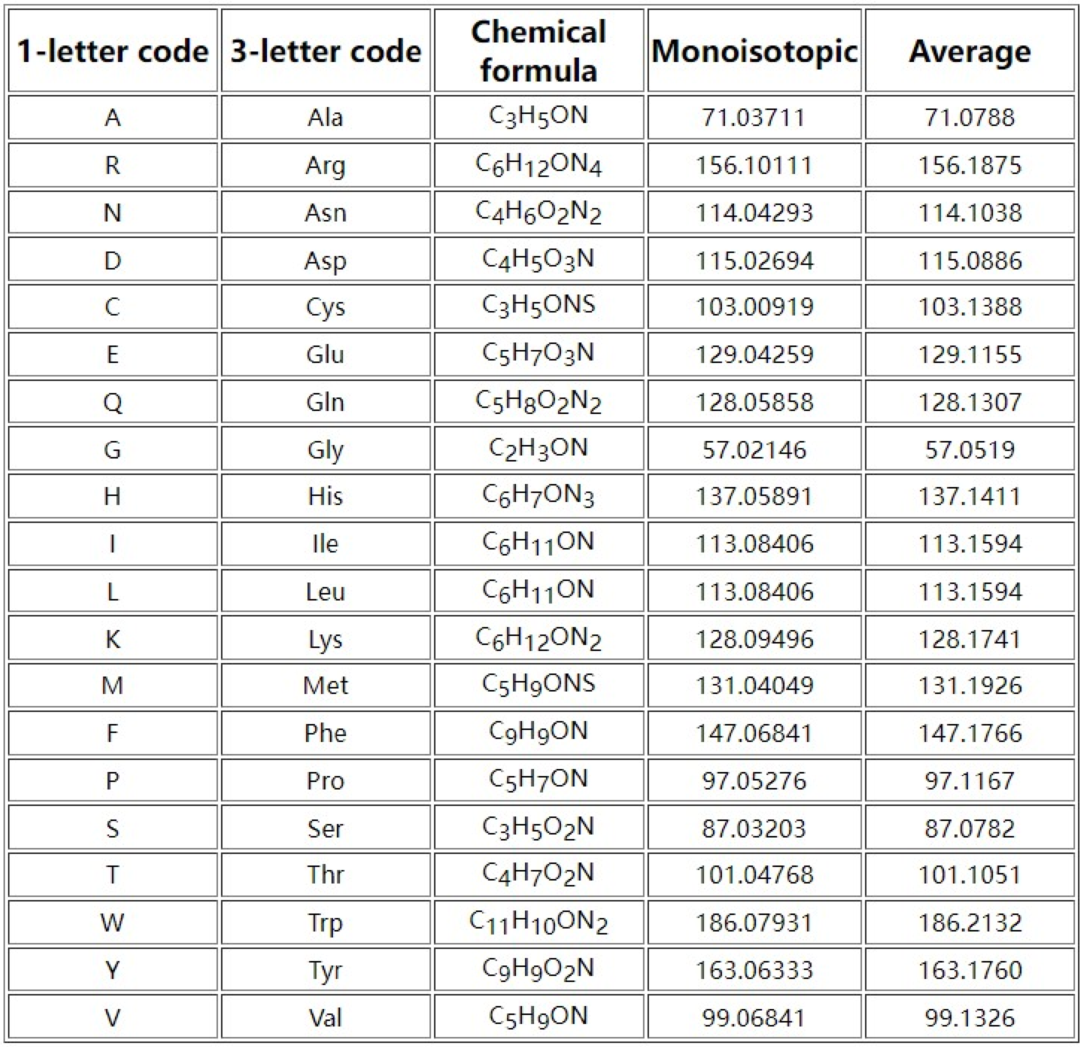
This thesis proposed a new *De Novo* sequencing model for tandem MS spectrometry, combining deep learning and dynamic programming. With regard to the deep learning models, there are two convolutional neural network models working together. The first model(Single Position Probability model) converts the original mass intensities to more meaningful probabilities, and charge-two peaks can be reconstructed in their charge-one position based on the probabilities. The second model(Dual Position Scoring model) utilizes the converted values to calculate the probability that each amino acid serves as the preceding amino acid for each peak. With the results, dynamic programming can be done, and use backtracking to find the final predicted sequence. Finally, the unsure mass interval can be replaced by the corresponding mass value and all possible amino acid sequences will be given.

A large number of tests have been conducted on different datasets and different species. The dataset we used as a training set is NIST *H. sapiens* Orbitrap_HCD mass spectra. We use three evaluation criteria to measure the performance, amino acid recall, amino acid precision, and peptide recall. The amino acid recall and precision reached 81.3%, and peptide recall reached 49.5% on the testing set of *H.sapiens*. Moreover, the amino acid recall and precision reached 88.5% on *C.griseus*, and peptide recall reached 52.9%, which means that this model can be applied to different species' spectra. In the comparison with other models, we used the spectra data from the ProteomeXchange database. The amino acid recall and precision reached 59.2% and 59.5%, which are comparable to those of PEAKS and DeepNovo. The peptide recall is 16.7%.

The peptide recall is relatively low, and there are a few reasons that lead to the low peptide recall. The first is that the dynamic programming algorithm is relatively simple, for example, the penalty is only given on the missing peak. Second, this model cannot distinguish *y*-ion peaks and *b*-ion peaks. Therefore, the y-ion peak path may contain some b-ion peaks, if some consecutive b-ion peaks have high scores. Third, there are some ions whose *y*-ion peak and b-ion peak are both missing, in the experiment we just randomly choose one possible amino acid sequence to be the final prediction, and do not apply any post-procession process like other commercial software. While the peptide recall may not be exceptionally high in this algorithm, the scores assigned to all the peaks can offer a more precise and detailed understanding of the spectrum. Consequently, these scores can be leveraged to enhance the spectrum, thereby improving the overall quality and accuracy of the predicting. And the scores could be used in other algorithms to be the initial values for the computation.

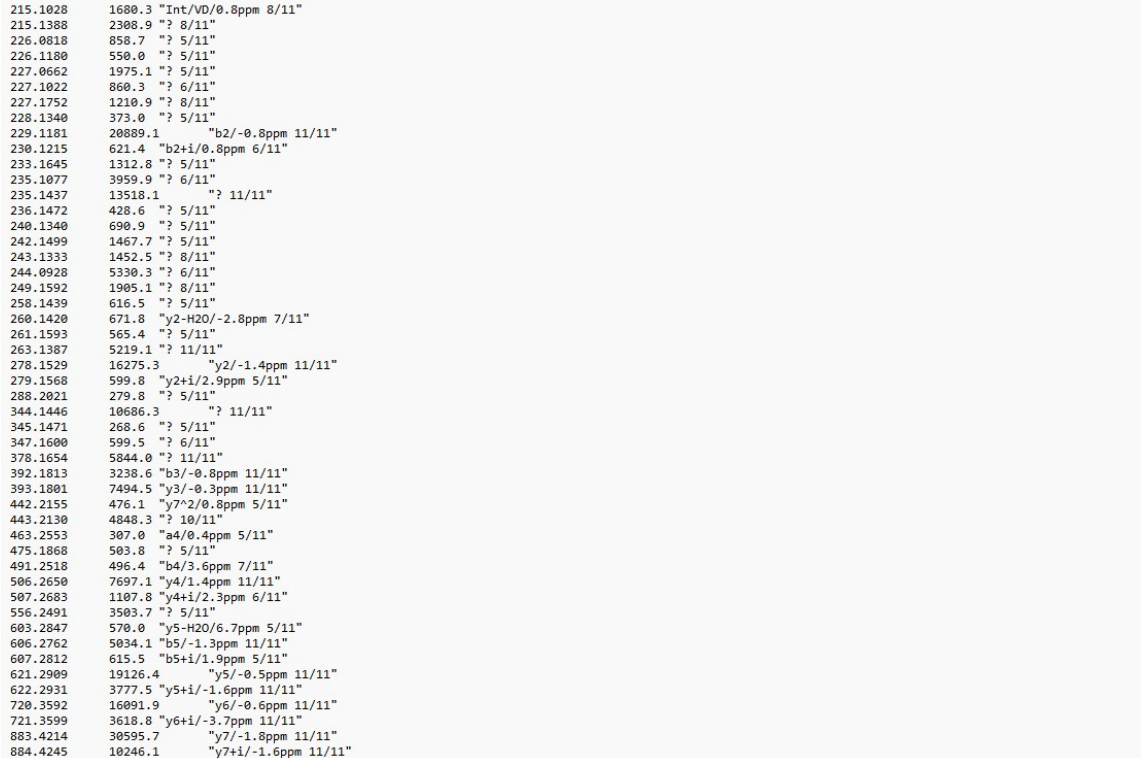
In the future, we will try to figure out a more sophisticated dynamic programming algorithm. For example, giving adjustable penalties to peaks with different score values. Besides, the dynamic programming algorithm can be optimized to improve the computational speed. Furthermore, we are actively exploring the integration of additional biological technologies to incorporate post-processing steps. This will enable a more comprehensive prediction and selection process among all the candidates derived from the dynamic programming methods.

Supplementary



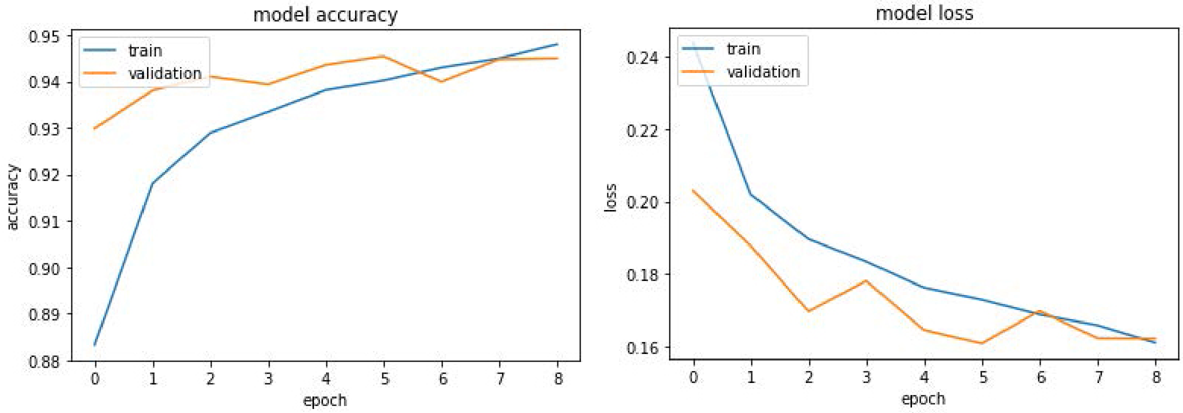
1-letter code	3-letter code	Chemical formula	Monoisotopic	Average
A	Ala	C_3H_5ON	71.03711	71.0788
R	Arg	$C_6H_{12}ON_4$	156.10111	156.1875
N	Asn	$C_4H_6O_2N_2$	114.04293	114.1038
D	Asp	$C_4H_5O_3N$	115.02694	115.0886
C	Cys	C_3H_5ONS	103.00919	103.1388
E	Glu	$C_5H_7O_3N$	129.04259	129.1155
Q	Gln	$C_5H_8O_2N_2$	128.05858	128.1307
G	Gly	C_2H_3ON	57.02146	57.0519
H	His	$C_6H_7ON_3$	137.05891	137.1411
I	Ile	$C_6H_{11}ON$	113.08406	113.1594
L	Leu	$C_6H_{11}ON$	113.08406	113.1594
K	Lys	$C_6H_{12}ON_2$	128.09496	128.1741
M	Met	C_5H_9ONS	131.04049	131.1926
F	Phe	C_9H_9ON	147.06841	147.1766
P	Pro	C_5H_7ON	97.05276	97.1167
S	Ser	$C_3H_5O_2N$	87.03203	87.0782
T	Thr	$C_4H_7O_2N$	101.04768	101.1051
W	Trp	$C_{11}H_{10}ON_2$	186.07931	186.2132
Y	Tyr	$C_9H_9O_2N$	163.06333	163.1760
V	Val	C_5H_9ON	99.06841	99.1326

Supplementary 1. The amino acid masses. Source: [3] .

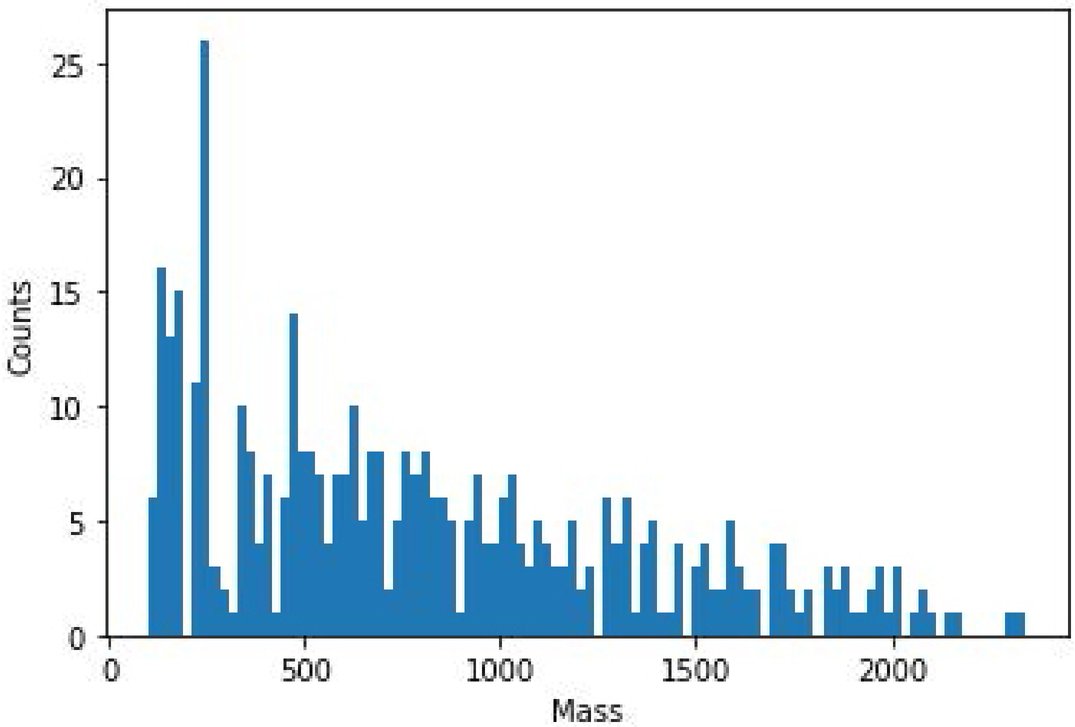


```
215.1028      1680.3 "Int/VD/0.8ppm 8/11"
215.1388      2308.9 "? 8/11"
226.0818      858.7  "? 5/11"
226.1180      550.0  "? 5/11"
227.0662      1975.1 "? 5/11"
227.1022      860.3  "? 6/11"
227.1752      1210.9 "? 8/11"
228.1340      373.0  "? 5/11"
229.1181      20889.1       "b2/-0.8ppm 11/11"
230.1215      621.4  "b2+i/0.8ppm 6/11"
233.1645      1312.8 "? 5/11"
235.1077      3959.9 "? 6/11"
235.1437      13518.1       "? 11/11"
236.1472      428.6  "? 5/11"
240.1340      690.9  "? 5/11"
242.1499      1467.7 "? 5/11"
243.1333      1452.5 "? 8/11"
244.0928      5330.3 "? 6/11"
249.1592      1905.1 "? 8/11"
258.1439      616.5  "? 5/11"
260.1420      671.8  "y2-H2O/-2.8ppm 7/11"
261.1593      565.4  "? 5/11"
263.1387      5219.1 "? 11/11"
278.1529      16275.3       "y2/-1.4ppm 11/11"
279.1568      599.8  "y2+i/2.9ppm 5/11"
288.2021      279.8  "? 5/11"
344.1446      10686.3       "? 11/11"
345.1471      268.6  "? 5/11"
347.1600      599.5  "? 6/11"
378.1654      5844.0 "? 11/11"
392.1813      3238.6 "b3/-0.8ppm 11/11"
393.1801      7494.5 "y3/-0.3ppm 11/11"
442.2155      476.1  "y7^2/0.8ppm 5/11"
443.2130      4848.3 "? 10/11"
463.2553      307.0  "a4/0.4ppm 5/11"
475.1868      503.8  "? 5/11"
491.2518      496.4  "b4/3.6ppm 7/11"
506.2650      7697.1 "y4/1.4ppm 11/11"
507.2683      1107.8 "y4+i/2.3ppm 6/11"
556.2491      3503.7 "? 5/11"
603.2847      570.0  "y5-H2O/6.7ppm 5/11"
606.2762      5034.1 "b5/-1.3ppm 11/11"
607.2812      615.5  "b5+i/1.9ppm 5/11"
621.2909      19126.4       "y5/-0.5ppm 11/11"
622.2931      3777.5 "y5+i/-1.6ppm 11/11"
720.3592      16091.9       "y6/-0.6ppm 11/11"
721.3599      3618.8 "y6+i/-3.7ppm 11/11"
883.4214      30595.7       "y7/-1.8ppm 11/11"
884.4245      10246.1       "y7+i/-1.6ppm 11/11"
```

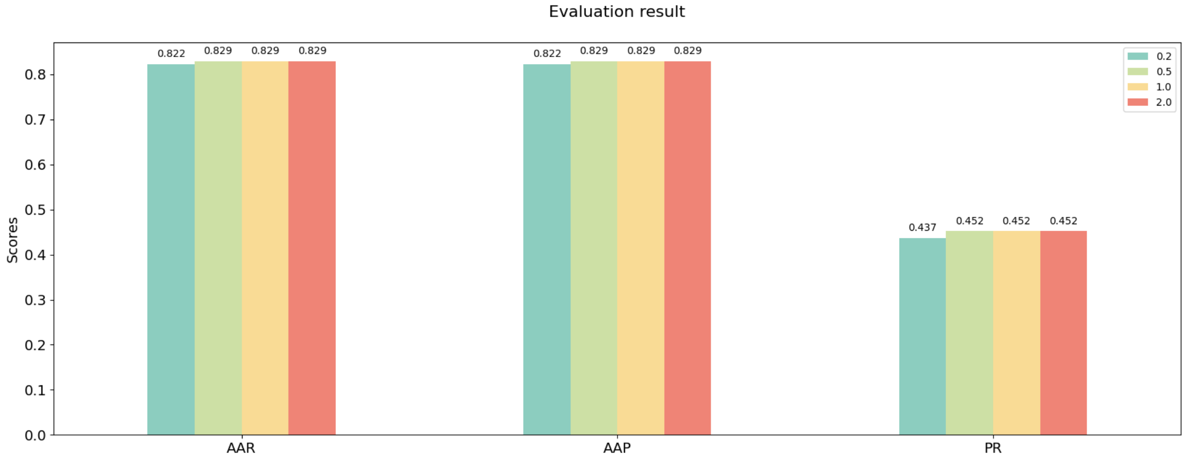
Supplementary 2. The full example of a NIST Orbitrap HCD spectrum. .



Supplementary 3. The training result of the Single Position Model .



Supplementary 4. The mass distribution of the wrong prediction of signal peaks in the Dual Position model.



Supplementary 5. Evaluation results using different penalties.

Bibliography

[1] Jimmy K Eng, Ashley L McCormack, and John R Yates. 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):976–989, 1994.

[2] Bin Ma, Kaizhong Zhang, and Chengzhi Liang. An effective algorithm for the peptide de novo sequencing from ms/ms spectrum. In *Combinatorial Pattern Matching: 14th Annual Symposium, CPM 2003 Morelia, Michoacán, Mexico, June 25–27, 2003 Proceedings 14*, pages 266–277. Springer, 2003.

[3] Ting Chen, Ming-Yang Kao, Matthew Tepel, John Rush, and George M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8(3):325–337, 2001.

[4] Ida Chiara Guerrera and Oliver Kleiner. Applications of tandem mass spectrometry in proteomics. *Bioscience Reports*, 25(1-2):71–93, 2005.

[5] Lekha Sleno and Dietrich A Volmer. Ion activation methods for tandem mass spectrometry. *Journal of mass spectrometry*, 39(10):1091–1112, 2004.

[6] Clifton K Fagerquist, Bertram G Lee, William J Zaragoza, Jaszemyn C Yambao, and Beatriz Quiñones. Software for top-down proteomic identification of a plasmid-borne factor (and other proteins) from genomically sequenced pathogenic bacteria using maldi-tof-tof-ms/ms and post-source decay. *International Journal of Mass Spectrometry*, 438:1–12, 2019.

[7] Ghazaleh Yassaghi, Zdeněk Kukačka, Jan Fiala, Daniel Kavan, Petr Halada, Michael Volný, and Petr Novák. Top-down detection of oxidative protein footprinting by collision-induced dissociation, electron-transfer dissociation, and electron-capture dissociation. *Analytical Chemistry*, 94(28):9993–10002, 2022.

[8] ON Chen, Susan Groh, Alison Liechty, and Douglas P Ridge. Binding of nitric oxide to iron (ii) porphyrins: Radiative association, blackbody infrared radiative dissociation, and gas-phase association equilibrium. *Journal of the American Chemical Society*, 121(50):11910–11911, 1999.

[9] Gun Wook Park, Ji Won Lee, Hyun Kyoung Lee, Jong Hwan Shin, Jin Young Kim, and Jong Shin Yoo. Classification of mucin-type o-glycopeptides using higher-energy collisional dissociation in mass spectrometry. *Analytical Chemistry*, 92(14):9772–9781, 2020.

[10] John R Yates, Ashley L McCormack, and Jimmy Eng. Peer reviewed: mining genomes with ms. *Analytical chemistry*, 68(17):534A–540A, 1996.

[11] Jesper V Olsen, Boris Macek, Oliver Lange, Alexander Makarov, Stevan Horning, and Matthias Mann. Higher-energy c-trap dissociation for peptide modification analysis. *Nature methods*, 4(9):709–712, 2007.

[12] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.

[13] A. H. El-Khatib, Y. He, D. Esteban-Fernández, and M. W. Linscheid. Application of higher energy collisional dissociation (hcd) to the fragmentation of new dota-based labels and n-termini dota-labeled peptides. *Journal of Mass Spectrometry*, 52(8):543–549, 2017.

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

[15] Dequan Li, Yan Fu, Ruixiang Sun, Charles X Ling, Yonggang Wei, Hu Zhou, Rong Zeng, Qiang Yang, Simin He, and Wen Gao. pfind: a novel database-searching software system for automated peptide and protein identification via tandem mass spectrometry. *Bioinformatics*, 21(13):3049–3050, 2005.

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

[17] Jing Zhang, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng Zhang, Gilles A Lajoie, and Bin Ma. Peaks db: de novo sequencing assisted database search for sensitive and accurate peptide identification. *Molecular & cellular proteomics*, 11(4), 2012.

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

[19] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 17(20):2337–2342, 2003.

[20] Hao Chi, Rui-Xiang Sun, Bing Yang, Chun-Qing Song, Le-Heng Wang, Chao Liu, Yan Fu, Zuo-Fei Yuan, Hai-Peng Wang, Si-Min He, et al. pnovo: de novo peptide sequencing and identification using hcd spectra. *Journal of proteome research*, 9(5):2713–2724, 2010.

[21] Kyowon Jeong, Sangtae Kim, and Pavel A Pevzner. Uninovo: a universal tool for de novo peptide sequencing. In *Research in Computational Molecular Biology: 17th Annual International Conference, RECOMB 2013, Beijing, China, April 7-10, 2013. Proceedings 17*, pages 100–117. Springer, 2013.

[22] Bernd Fischer, Volker Roth, Franz Roos, Jonas Grossmann, Sacha Baginsky, Peter Widmayer, Wilhelm Gruissem, and Joachim M Buhmann. Novohmm: a hidden markov model for de novo peptide sequencing. *Analytical chemistry*, 77(22):7265–7273, 2005.

[23] Lijuan Mo, Debojyoti Dutta, Yunhu Wan, and Ting Chen. Msnovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry. *Analytical chemistry*, 79(13):4870–4878, 2007.

[24] Yu-xin Peng, Wen-wu Zhu, Yao Zhao, Chang-sheng Xu, Qing-ming Huang, Han-qing Lu, Qing-hua Zheng, Tie-jun Huang, and Wen Gao. Cross-media analysis and reasoning: advances and directions. *Frontiers of Information Technology & Electronic Engineering*, 18(1):44–57, 2017.

[25] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010.

[29] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[30] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[34] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[35] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[37] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[39] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[40] Leslie N. Smith. Cyclical learning rates for training neural networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 464–472. IEEE, 2017.

[41] National institute of standards and technology.

[42] Proteomexchange: A public repository for mass spectrometry data.

[43] Sara M Shatat, Basma M Eltanany, Abeer A Mohamed, Medhat A Al-Ghobashy, Faten A Fathalla, and Samah S Abbas. Coupling of on-column trypsin digestion–peptide mapping and principal component analysis for stability and biosimilarity assessment of recombinant human growth hormone. *Journal of Chromatography B*, 1072:105–115, 2018.

[44] P ROEPSTORFFE. Proposal for a common nomenclature for sequence ions in mass spectra of peptides. *Biomed. Mass Spectrom.*, 11:601–605, 1984.

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