

Spectrum and Retention Time Prediction for N-Glycopeptides Using Deep Learning

by

Shuyang Zhang

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

Waterloo, Ontario, Canada, 2023

© Shuyang Zhang 2023

Author's Declaration

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

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

Abstract

Sequencing proteins and glycans have important clinical applications, as glycosylation is shown to play a significant role in cellular communication and immune response. Certain glycans are linked to the diagnosis of cancer as well as targeted immunotherapy. Mass spectrometry is a powerful tool that helps us gain insight into peptide sequences and glycan structures, by using database search, spectral library, or *de novo* sequencing. Spectrum and retention time prediction using deep learning has gained popularity with studies on non-glycosylated peptides and has been shown to improve database search results via rescoring. This thesis proposes deep learning models to predict spectrum and retention time for N-glycopeptides and then discusses the applications of these models with respect to glycopeptide sequencing.

Chapter 3 presents a graph deep learning model to predict fragment ion intensities of observed spectrums and define a spectrum representation for glycan fragments with up to three cleavages. The spectrum prediction model has a median cosine similarity of 0.921, which is 20% higher than previous attempts at glycopeptide spectrum prediction.

For retention time prediction in Chapter 4, we propose a model with two parallel encoders for both peptide and glycan input and apply transfer learning for the sequence encoder. The retention time prediction model has a Pearson correlation of 1.0, which is higher than the previous 0.98 and 0.96 attempts. We also introduce the 95 percentile delta as an evaluation metric, as well as discuss the interpretability of our model.

Finally in Chapter 5, we apply our spectrum and retention time prediction models in glycopeptide sequencing pipelines, including database search and *de novo* search. We show that our model improves identification by rescoring and has the potential to be used as a filter for false positives. We also demonstrate that our model improves *de novo* identification when used in the scoring function.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Ming Li, who opened the door to bioinformatics for me and provided so many invaluable feedback on my progress. You pointed me in important directions for my research during this thesis, and I appreciate your patience and guidance throughout.

I would like to thank Dr. Lei Xin for guiding me through the process of this thesis and answering so many of my questions on mass spectrometry, glycans, and so on. I also appreciate the researchers at Bioinformatics Solutions Inc.: Dr. Weiping Sun, Dr. Xin Chen, Dr. Baozhen Shan, and Zihao Wang, whose expertise in bioinformatics helped my research in crucial areas.

My thanks also go to fellow students at the University of Waterloo: Qianqiu Zhang, Zeping Mao, Ruixue Zhang, Yonghan Yu, and Zhang Ma. You have provided me with useful insight during our research meetings and conversations.

Last, but certainly not least, I thank my parents and friends, who supported me emotionally during this thesis.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 The Study of Proteins and Glycans	1
1.2 Mass Spectrometry	2
1.3 Database Search for Peptides and Glycopeptides	4
1.4 Other Searching Methods	6
1.5 Overview and Structure of This Thesis	7
2 Previous Works and Background	8
2.1 Spectrum Prediction on Glycopeptides	8
2.1.1 Kinetic Model for Peak Intensity Prediction	8
2.1.2 Probabilistic Model for Peak Intensity Prediction	9
2.2 Retention Time Prediction on Glycopeptides	9

2.3	Deep Learning Models on Non-Glycosylated Peptides	10
2.4	Graphormer	10
3	Spectrum Prediction	12
3.1	Model Design and Structure	12
3.1.1	Glycan’s Tree Structure as Inputs	12
3.1.2	Modeling Spectrum Information as Outputs	13
3.1.3	Model Structure for Spectrum Prediction	16
3.2	Training	17
3.2.1	Data Acquisition	17
3.2.2	Training and Loss Functions	19
3.3	Model Evaluation	19
3.4	Discussion	20
3.4.1	Case Study on Charge	20
3.4.2	Conclusion and Future Research Directions	23
4	Retention Time Prediction	24
4.1	Model Design and Structure	24
4.1.1	Pre-training on Peptide Inputs	24
4.2	Training	25
4.2.1	Regression to Normalized Retention Time with iRT	25
4.2.2	Using Feature RT Instead of Spectrum RT	27
4.2.3	Data Acquisition	27
4.2.4	Training and Loss Functions	27
4.3	Model Evaluation	28
4.3.1	95 Percent Delta and Ablation Studies	28
4.3.2	Interpretation of Results	29
4.4	Discussion	32

5	Using Prediction Models in Glycopeptide Search	34
5.1	Database Search	34
5.1.1	Rescoring	34
5.1.2	Filtering	37
5.2	<i>De Novo</i> Search	41
5.3	Discussion	42
6	Conclusion	43
6.1	Contributions of This Thesis	43
6.2	Issues and Direction for Future Research	44
	References	45

List of Figures

1.1	N-glycan core structure	2
1.2	Tandem mass spectrometry illustration	4
1.3	Example of Glycopeptide Spectrum Match by PEAKS Glycan Annotation	6
2.1	Graphormer’s Illustration of Centrality, Edge and Spatial Encodings	11
3.1	Example of possible glycan fragments with the presence of Fuc attached to the root node.	16
3.2	Illustration of Spectrum Prediction Model	18
3.3	Example mirror plot of experimental versus predicted spectra for peptide AEPPLNASAGDQEEK and glycan id 23. The top graph shows the experimental spectrum and the bottom the predicted spectrum.	21
3.4	Examples of experimental and predicted spectrums. In each subfigure, the top panel is the experimental spectrum, and the bottom is the predicted. z in the plots refers to the charge of the precursor. The pair of (a) and (b) show that different charges on the same glycopeptide will result in different spectrum intensities. So does the pair of (c) and (d). However, we observe that (a) and (d), although having the same charge, show different spectrum intensities, while (a) and (c) are similar in the general trend of the spectrum.	22
4.1	Illustration of RT prediction models, transferring the sequence encoder from peptide RT pretraining	25
4.2	Correlation plots for iRT prediction models	30
4.3	Correlation plots for fixed glycan or peptide	31
4.4	Correlation between the sialic acid to HexNAc ratio and predicted iRT	32

5.1	Glycan score versus iRT difference plots for fission yeast data with different search engines	38
5.2	Glycan score versus iRT difference plot for mouse brain data	40
5.3	Choosing the best candidate by lowest iRT difference for each PTM reduces outliers and the identification of glycans with NeuGc and multiple Fucose .	40

List of Tables

1.1	5 common monosaccharides in glycans, their symbols, abbreviations, and residue masses	2
3.1	Categories of fragment ions resulting from at most two cuts.	15
4.1	Statistics on RT and iRT variance for the same glycopeptide across samples	26
4.2	Median and 95 percentile delta (in minutes) for predicted versus experimental iRT across different experiments	29
5.1	Number of PSMs outputted by PEAKS Glycan and pGlyco3	35
5.2	Number of PSMs, glycans, and glycopeptides after 1% FDR, according to different scoring and rescoring methods for PEAKS Glycan and pGlyco3	36
5.3	Statistics on the iRT difference with fission yeast PSMs from different search engines	39
5.4	When discarding PSMs with $\Delta_{iRT} > 30$, the percentage discarded in total as well as the percentage of false positives (excluding Fucose) discarded for each software	39
5.5	Glycan identification rates from <i>de novo</i> model using different scoring functions and counting strategies	42

Chapter 1

Introduction

1.1 The Study of Proteins and Glycans

Proteomics is the study of proteins, which is one of the most important yet varied building blocks of life. Proteins are responsible for all kinds of inter- and intra-cellular functions, and knowing a protein's structure helps the understanding of the functions and properties of said protein. Protein sequencing is the first step in discovering protein structures.

Glycosylation is one of the post-translational modifications (PTMs) on proteins. A PTM is adding a modifying group to an amino acid in the protein sequence, thereby altering the functions and properties of the modified protein. Glycosylation is a common PTM that happens to more than 50% of proteins in humans, but correctly identifying the glycan that is added to the protein remains a difficult task. It has been shown that glycosylation is vastly different in cancerous cells compared to normal cells, and the glycans on cancerous cells play important roles in the growth and metastasis of cancer [25]. Glycans can be used in the diagnosis of cancer as well as targeted immunotherapy [28].

In glycosylation, a glycan, which is a tree-like structure composed of monosaccharides, is added to an amino acid. Monosaccharides are diverse in structure, but for the course of this thesis, we only consider five main categories of monosaccharides: Hexose (Hex), N-Acetylhexosamine (HexNAc), Fucose (Fuc), N-Glycolylneuraminic acid (NeuGc) and N-Acetylneuraminic acid (NeuAc). Other monosaccharides are not in our particular concern for the makeup of glycans. Table 1.1 lists the five monosaccharides, their symbols, and their residue masses.

There are two types of glycosylation: N-linked glycosylation attaches a glycan to the

Symbol	Name	Abbr.	Residue Mass
●	Hexose	Hex	162.0528
■	N-Acetylhexosamine	HexNAc	203.0794
▲	Fucose	Fuc	146.0579
◆	N-Glycoylneuraminic acid	NeuGc	307.0903
◆	N-Acetylneuraminic acid	NeuAc	291.0954

Table 1.1: 5 common monosaccharides in glycans, their symbols, abbreviations, and residue masses

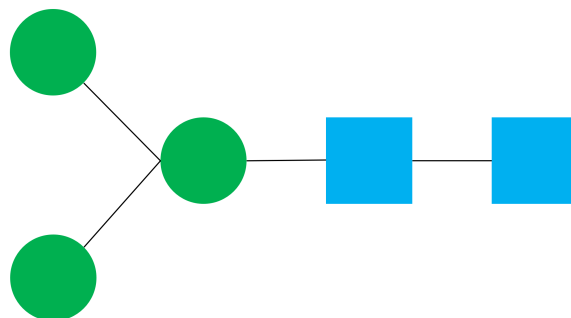


Figure 1.1: N-glycan core structure

nitrogen atom on the amino acid Asparagine (denoted as N), whereas O-linked glycosylation happens when a glycan is linked to the oxygen atom in Serine (S) or Threonine (T). N-glycans all have a common core structure composed of two HexNAc and three Hex monosaccharides, whereas O-glycans do not have a common core structure. In this thesis, we will focus on N-glycans only. Figure 1.1 illustrates the core structure of N-glycans.

1.2 Mass Spectrometry

Tandem Mass Spectrometry (MS/MS) [21] is a powerful tool that can be used to study proteins and glycans, among many other applications. Given a sample of proteins, one first digests it into shorter sequences, called peptides, using enzymes such as Trypsin. Then, one feeds the sample with peptides into a mass spectrometer, which has five stages. In the

first stage, the ion source produces gas phase ions from the peptide sample, then in the second stage, the mass analyzer separates the ions according to their mass to charge (m/z) ratio. The relative abundance, or intensity, for each m/z value is recorded, generating an MS1 spectrum. Each pair of m/z values and its intensity in the spectrum is called a *peak*, and in the MS1 spectrum, each m/z value is also called a precursor ion, often associated with a peptide in the sample. During the third phase, some precursor ions are selected and further fragmented into fragment ions by different dissociation methods such as collision-induced dissociation (CID) or higher energy collision dissociation (HCD). These fragment ions are again separated by m/z in the fourth stage and the peaks are recorded in the fifth stage, resulting in an MS2 spectrum for each selection of precursor ions.

Figure 1.2 illustrates a brief diagram of the process of tandem mass spectrometry and examples of MS1 and MS2 spectrums.

Liquid Chromatography (LC) [38] is a technology to separate a mixture by its physical and chemical properties. A sample in a solvent is carried through a column that is fixed with a certain material called the stationary phase. Different components in the mixture may interact with the stationary phase (in other words, retained in the column) for a different amount of time, resulting in their separation by time, so the output particles are each associated with a retention time value. In proteomics, LC is often used in conjunction with Tandem MS, and we call the process LC-MS/MS.

During stage three of MS/MS, there are two main types of strategies for selecting the precursor ions to proceed: data-dependent acquisition (DDA) or data-independent acquisition (DIA). DIA proceeds to fragment all the precursor ions within a certain range of m/z and retention time, whereas DDA's goal is to only select one precursor ion at a time. The resulting MS2 spectrums for DIA include all the fragment peaks for all the peptides included, while each MS2 spectrum for DDA only shows the fragment peaks for each precursor ion. Practically, DIA tools produce higher accuracy in generating spectrums, but sequencing with DIA data is more complicated.

Glycosylated proteins are processed similarly to normal proteins. The samples for the mass spectrometer may include peptides with glycans, and the mass of the precursor ion is the mass of the peptide plus the mass of the glycan. During the fragmentation, not only can links between amino acids be cleaved, but so can the links between the peptide and the glycan, as well as links between monosaccharides. Depending on the experiment, researchers may opt to use an additional step after enzyme digestion, called enrichment, where chemical processes are used to prioritize the selection of glycopeptides over regular peptides, allowing a larger percentage of MS2 scans to correlate with glycopeptides [22].

Sequencing peptides from DIA data is already met with challenges due to the highly

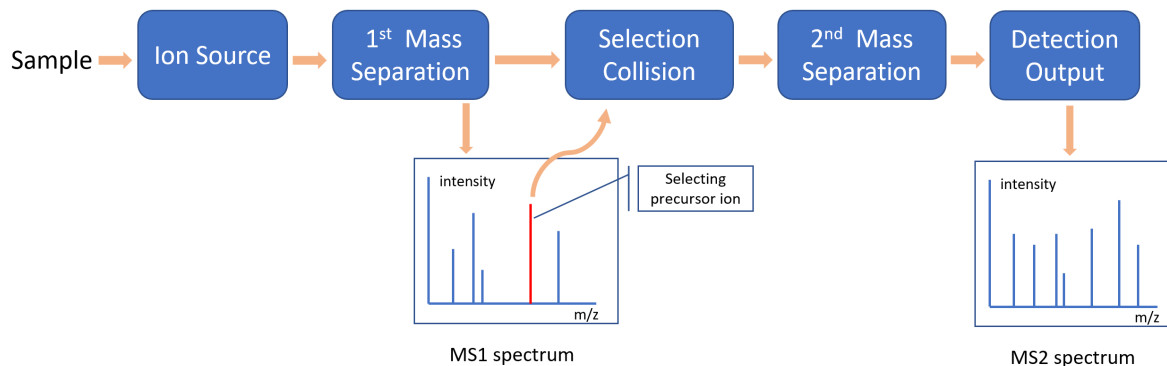


Figure 1.2: Tandem mass spectrometry illustration

complex spectrums, but it is even more complex with glycopeptides, as fragmentation with glycopeptides introduces many more peaks. Currently, mass spectrometry data on glycopeptides mostly use the DDA method, so this thesis will focus on the sequencing strategies for DDA.

1.3 Database Search for Peptides and Glycopeptides

When MS/MS produces the MS2 spectrums, each scan only contains peak intensities resulting from fragmentation from a precursor ion. Since each precursor ion is associated with a peptide, we wish to find out the exact peptide sequence matched to each MS2 scan. If we have a database of all proteins, and therefore peptides, that the sample is taken from, then theoretically we can search through the database to find the best candidate that matches the scan. This process is called database search, in opposition to the other method *de novo*, which builds the peptide from scratch, without information from the protein database.

The precursor mass is the mass of the peptide, which is the sum of the residue mass of each amino acid, plus the mass of one water molecule. This is because one Hydrogen (H) is attached to one end of the peptide, and one hydroxide (OH) is on the other end. When a precursor ion is fragmented in MS/MS, one link between amino acids is broken, resulting in two fragments. The one fragment with the hydrogen molecule on one end is called the b-ion, whereas the other is the y-ion. Consequently, the mass of the b-ion is the sum of the residue mass of the amino acids in the corresponding fragment plus 1.0078 Da, the mass

of Hydrogen, and vice versa for the y-ion. Given a peptide and a table of residue masses for each amino acid, one can calculate all the theoretical fragment masses (and therefore also m/z) that can be produced from fragmentation.

From the MS2 scan, we are given all the peaks, which are pairs of m/z values and their intensities. When the theoretical m/z of an ion fragment is within the tolerance threshold of a peak on the spectrum, we say that the peak is a match. Experimentally, we set the tolerance to be between 0.02 Da and 0.05 Da. From here, one can design a metric for how well a peptide matches to a spectrum, based on the theoretical fragment ions and peaks in the spectrum.

In database search, we look at one MS2 at a time, and first filter out all the peptides in the database whose precursor mass is within a tolerance level. That is to say, we only keep the peptides whose mass is close to the precursor mass, and go from there. Again, the precursor tolerance can be set depending on the experiment. Then for each filtered candidate, one can use the previously mentioned metric to score how well the peptide matches the scan and select the best candidate. The final candidate peptide and the spectrum form a peptide-spectrum match (PSM).

The target decoy strategy [7] ensures that we only output the PSMs that are correct with high confidence. Given that we do not know whether a PSM is correct, one makes the assumption that the probability of matching an incorrect peptide to a spectrum is close to that of matching a random peptide with the same precursor mass. We call the database of all possible peptides the *target* database and generate a *decoy* database by either reversing or randomly shuffling the protein sequences. The assumption is that the probability of selecting an incorrect peptide from the target database is equal to the probability of selecting a peptide from the decoy database. Therefore, the false discovery rate (FDR) can be defined as the number of decoys divided by the number of targets being produced. In most sequencing pipelines we opt to use $FDR \leq 1\%$ and use the number of PSMs as a criterion for the performance of a search engine. Many search engines and software are proposed to perform database searches, such as SEQUEST [24], Mascot[8], MaxQuant[5], and PEAKS[32].

Database search for glycopeptides is similar to that of peptides. Given a glycan and peptide pair, one can also calculate the theoretical fragment m/z values and match them to the peaks in a spectrum. In this case, a fragment may be a partial peptide, a partial glycan, or a partial glycan attached to the peptide. A glycopeptide matched to an MS2 scan is also called a GPSM (glycopeptide spectrum match). For the rest of this thesis, since we are mainly concerned with glycopeptides, we may use both PSM and GPSM to refer spectrum matches to glycopeptides.

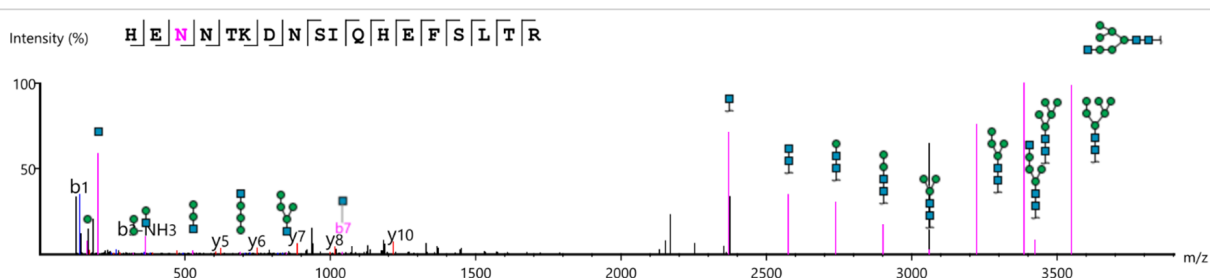


Figure 1.3: Example of Glycopeptide Spectrum Match by PEAKS Glycan Annotation

Search engines for glycopeptides include MSFragger [15], pGlyco3 [35], and PEAKS Glycan [31]. It is worth noting that MSFragger produces Glycopeptide PSMs that only have the glycan composition, that is the number of monosaccharides in each of the five categories, whereas pGlyco3 and PEAKS Glycan both output PSMs containing the matched glycan structure.

Figure 1.3 shows an example of matching a spectrum to a glycopeptide. The highlighted peak around $m/z = 2371.08$ is matched to a HexNAc molecule attached to the peptide, where the short wavy line attached to the blue square represents the entire peptide sequence. We arrive at this m/z value by adding all the residue mass of amino acids in the peptide sequence, plus the residue mass of the HexNAc, and the mass of a hydroxide molecule. The peak is highlighted because it is within the tolerance of 30 ppm of the theoretical m/z . This peak is a y-ion because it is attached to a hydroxide molecule. As another example, the highlighted peak at around $m/z = 204.087$ is matched to the b-ion of just a HexNAc. The HexNAc monosaccharide at the leaf of the glycan is attached to a Hydrogen molecule and when fragmented, results in a b-ion.

1.4 Other Searching Methods

In addition to database search, spectral library search is another method for matching spectrums to peptides. If one can build a library of peptides with their expected spectrums, one can compare the experimental spectrum with the spectrums in the library using a similarity metric. This can be advantageous over simple database searches because the similarity of relative peak intensity is taken into consideration. However, the main hurdle is to create an accurate yet comprehensive spectral library. For spectral library search, it is also proposed that instead of the decoy protein database, one permutes the spectrum peaks to generate decoy spectrums, and then proceeds with the FDR strategy.

Generating a spectral library with peak intensities can be difficult, so efforts are made to predict spectrums using deep learning. Once the database search has generated the top N candidates for each spectrum, one can run spectrum prediction for each candidate peptide, and obtain a similarity score between the experimental spectrum and predicted spectrum. From there, one can rescore the candidates based on previous database searching criteria and spectrum similarity. PROSIT [10] and pDeep [40] are examples of spectrum prediction models for rescoring. Besides spectrum prediction, PROSIT also proposes retention time prediction, where retention time (RT) is the recorded time that the precursor takes during the chromatography column. By comparing the predicted RT with the experimental RT, one can add yet another scoring feature to the rescoring process.

Besides database search, spectral library search, and rescoring, another prominent area for peptide sequencing is called *de novo* sequencing. As the name suggests, the sequencing means to start from scratch. Without the use of a protein database or spectral library, one may attempt to build the peptide just based on information of the spectrum. The advantage of *de novo* sequencing is its ability to discover peptides that are not in the database. The main obstacle for *de novo* sequencing is the abundant noise in the spectrum, so when finding the next probable amino acid, the search space may be large. Attempts at *de novo* algorithms include spectrum graph algorithms [6, 20], dynamic programming [4, 19], hidden markov model [9], and deep learning models [33, 27, 20].

With respect to Glycans, there have been non-deep-learning-based attempts at spectrum library search [39], as well as spectrum prediction [12] and RT prediction [14, 1]. There are also *de novo* methods for glycopeptide identification such as StrucGP [30]. There will be more in-depth discussions on glycan search methods in Chapter 2.

1.5 Overview and Structure of This Thesis

The main contributions of this thesis are designing deep learning models to predict glycopeptide spectrum and retention time, as well as applying these models in glycopeptide search. Our spectrum prediction model (Chapter 3) outperforms previous models on glycopeptides by at least 20%. As for the retention time prediction model in Chapter 4, the prediction accuracy of our RT prediction model improves upon other RT prediction models from $R^2 = 0.98$ to 1.0, and additionally, we will introduce another metric to evaluate RT prediction that previous works on glycopeptides did not use. In Chapter 5, we will demonstrate how our spectrum and RT prediction models help in glycopeptide search, when used in the filtering and rescoring of database search, as well as in *de novo* search.

Chapter 2

Previous Works and Background

2.1 Spectrum Prediction on Glycopeptides

2.1.1 Kinetic Model for Peak Intensity Prediction

Zhang *et al.* [39] proposed a kinetic model for predicting the peak intensity of a fragment on glycans. They use the mobile proton hypothesis [3] as a kinetic model of fragmentation and mathematically define the intensities of each fragment. Given a precursor ion P with possible fragments F_1, F_2, \dots, F_N , each with kinetic rate constants k_1, k_2, \dots, k_n , let $[P]_t$ be the abundance of the precursor ion P at time t . Then we can define

$$[P]_t = [P]_0 e^{-k_{total}t} \quad (2.1)$$

where k_{total} is the sum of rate constants $\sum_{i=1}^N k_i$. Therefore, the relative intensity of the fragment F_i at time t can be expressed as

$$\begin{aligned} [F_i]_t &= k_i [P]_0 \int_0^t e^{-k_{total}t} dt \\ &= \frac{k_i [P]_0}{k_{total}} (1 - e^{-k_{total}t}) \end{aligned} \quad (2.2)$$

By calculating the rate constants k_1, \dots, k_N , one can calculate the predicted fragment intensities. They build a mathematical model containing hundreds of parameters and train on 1831 spectra to obtain the prediction model. When testing the model on a testing set with glycans not seen in the training set, the median similarity between predicted and experimental spectrums (using cosine similarity) achieves 0.71 ± 0.08 .

2.1.2 Probabilistic Model for Peak Intensity Prediction

Klein *et al.* [12] considers the kinetic model by partitioning precursors into three categories: mobile, partially mobile, and immobile. Together with a list of features from the glycopeptide, such as glycan and peptide composition, charge, and so on, they model the intensity of a fragment as a probability drawn from a multinomial distribution parameterized by these features. They train their model on a mouse tissue dataset, withholding the mouse brain subset. When testing on the mouse brain dataset alone, they achieve a median similarity score (using cosine similarity) of 0.76.

2.2 Retention Time Prediction on Glycopeptides

Ang *et al.* [1] introduce the idea of retention time shifts by glycans. In a sample with both glycosylated and deglycosylated peptides, one can then calculate the shift in retention time for a glycan g : the retention time of the glycosylated peptide P_{glyco} minus the retention time of its deglycosylated counterpart $P_{deglyco}$.

$$\Delta RT_g = RT(P_{glyco}) - RT(P_{deglyco}) \quad (2.3)$$

Using a database to obtain experimental retention time shifts, and using an accurate RT prediction tool for peptides SSRCalc [16], the predicted retention time for a glycopeptide P_g can be calculated as

$$RT(P_g) = SSRCalc[RT](P_{deglyco}) + \Delta RT_g \quad (2.4)$$

The accuracy of this model achieves $R^2 = 0.967$.

Klein *et al.* [14] also build upon the RT prediction for peptides, and train a linear model with the peptide RT, normalized abundance of Hex, HexNAc, Fuc, NeuAc, and sulfate. They train both peptide-specific models and a cross-peptide model. The peptide-specific models achieve $R^2 = 0.98$ for a select few peptides, whereas the cross-peptide model achieves $R^2 = 0.897$.

Park *et al.* [23] propose a novel metric of Ln/Nn, which is a weighted ratio between HexNAc intensities and NeuAc intensities. They observed that for the same precursor mass, isomers with sialic acids (NeuAc and NeuGc) tend to shift the retention time early, whereas the addition of HexNAc results in a later retention time. For a sample glycan, let

n_A, n_B denote the number of HexNAc and NeuAc respectively, and let S_A, S_B denote the sum of peak intensities from HexNAc and NeuAc ions respectively. Then

$$Ln/Nn = \frac{S_A}{S_B} \times \frac{n_B}{n_A}. \quad (2.5)$$

They find that for the same precursor mass, the Ln/Nn value and the relative retention time are positively correlated, although they do not propose a predictive model for the retention time.

2.3 Deep Learning Models on Non-Glycosylated Peptides

The works described above all employ a relatively small regression model with at most hundreds of parameters. Deep learning exploits computational power and by using models of a much larger scale, has been shown to achieve extremely high accuracy in regression and prediction tasks. To the day of writing this thesis, we have not been made aware of attempts at predicting the peak intensities of glycopeptide spectrums or retention time using deep learning. Despite this, deep learning models have achieved success in spectrum and retention time prediction for regular peptides.

PROSIT [10] and pDeep [40] are examples of deep learning models on fragment ion intensity prediction. While pDeep employs an LSTM model that takes the peptide as one-hot vector input and predicts the relative intensity for the fragment corresponding to each link between amino acids, PROSIT uses an encoder-decoder model with GRU to achieve the same task. PDeep achieves cosine similarities of over 0.90 for many testing datasets, while PROSIT manages to arrive at a median cosine similarity of 0.99. There have also been attempts at full spectrum prediction such as by Liu *et al.* [17]. Instead of only predicting the intensities of fragment ions, they also predict all other peaks in the spectrum, arguing that the non-backbone peaks are also crucial in peptide spectrum matches. In this approach, they have a cosine similarity of 0.820 ± 0.088 for full spectrum prediction.

2.4 Graphormer

Graphormer proposed by Ying *et al.* [34] revolutionalized the area of deep learning on graphs. Based on a transformer architecture, Graphormer uses four different encodings

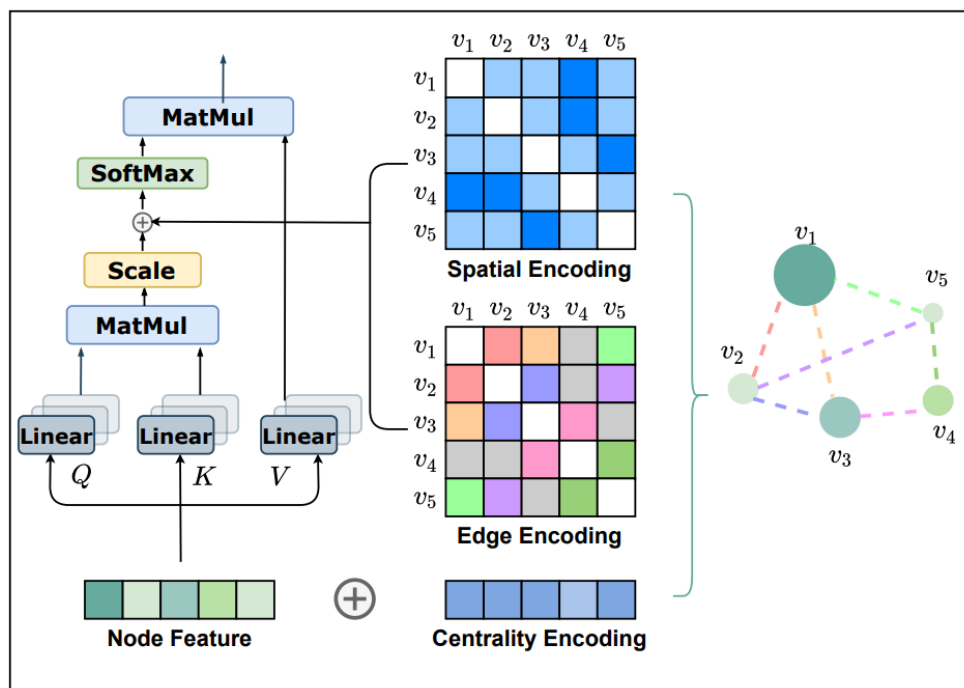


Figure 2.1: Graphormer’s Illustration of Centrality, Edge and Spatial Encodings

to translate graph information and structure into the deep learning network. First, the node feature intuitively encodes the information at each node of the graph. The centrality encoding embeds the indegree and outdegree of each node. The edge encoding keeps track of graph information along the edges. Finally, the spatial encoding records the relative spatial relationships between nodes, with one example being the shortest path distance between two nodes. Figure 2.1 shows Graphormer’s illustration of its encodings.

Graphormer performs both experimentally and provably better than its predecessor GNN, manifesting that the leap of performance achieved by transformers in sequential deep learning can be replicated in graphs. Given that glycans and glycopeptides are no longer sequential, but actually trees, a deep learning model like Graphormer shows good potential in processing glycopeptide data.

Chapter 3

Spectrum Prediction

In this chapter, we focus on the task of spectrum prediction for glycopeptides. The goal is to train a deep-learning model that takes a glycopeptide as input and predicts the corresponding spectrum as output. In particular, we do not predict the full spectrum but instead, output the predicted relative intensities of each glycopeptide fragment ion.

3.1 Model Design and Structure

3.1.1 Glycan’s Tree Structure as Inputs

Graphormer [34] is a graph-based deep learning network that takes in graph-based inputs, which is a good foundation for our model, whose inputs are tree structures. Given a glycopeptide, we can define a graph input for Graphormer as follows. Each amino acid and each monosaccharide is considered a node. The node feature is of $L \times d$, where L is the maximum number of amino acids and monosaccharides in the glycopeptide, and d is the node embedding dimension. Between each pair of amino acids, is a directed edge pointing in the direction of the peptide, and from the Asparagine (N) amino acid to the glycan, edges point to the leaves of the tree. We distinguish all the edges into three categories: between two amino acids, between an amino acid and the glycan, and between monosaccharides in the glycan. The node feature, therefore, is the embedded vector of each node, and the edge encoding is the embedding of each edge type in the adjacency matrix. Centrality encoding is defined as in Graphormer, and spatial encoding is taken to be the shortest path distance between each two nodes. It is noteworthy, that because a glycopeptide can

be considered a tree, the shortest path distance can be calculated very easily. The node feature, centrality encoding, edge encoding, and spatial encoding are all padded to the maximum glycopeptide size at $L = 64$.

3.1.2 Modeling Spectrum Information as Outputs

In PROSIT [10], the model outputs an $(N - 1) \times 6$ matrix, where N is the length of the peptide sequence, and 6 represents the 1+, 2+, and 3+ charged y-ions and b-ions for each fragment. Each value in the matrix represents the predicted relative peak intensity for the fragment ion. Given an experimental spectrum, one simply finds the matching peaks for each fragment ion m/z , and records the relative intensities in a matrix of the same size. Thus, one can use any similarity metric, such as cosine similarity, Pearson Correlation Coefficient (PCC), or spectral angle to determine the loss between the predicted spectrum and experimental spectrum.

For our task with graph input, we can adopt a similar method, by recording the y and b ions resulting from cutting each edge into a $(N + M - 1) \times 2$ matrix, where N is the peptide length, M is the number of monosaccharides in the glycan, $(N + M - 1)$ being the number of edges that can be fragmented, and 2 denoting y and b ions. Here, we make no attempts at predicting peak intensities for fragment ions with multiple charges, because experiments with glycan search data show that the other peaks have insignificant intensities.

Although the method described above is intuitive and works, upon further investigation into glycopeptide spectrum match data (Figure 1.3 shows an example), we find the following properties:

1. Although theoretically possible, there are no fragments where a full or partial glycan is attached to a partial peptide. If a partial peptide fragment is detected, the glycan has already been removed.
2. Fragments that result from cleavages on the glycan have significantly higher peak intensities, whereas fragments from peptides alone show very low intensities.
3. Among the glycan fragment ions, those attached to the peptide, namely y-ions, have higher peak intensities, in contrast to their b-ion counterparts, with the exception of a single HexNAc or Hex b-ion.
4. Some glycan fragment y-ions with high intensities result from not one fragment along an edge, but two or in rare cases, more fragments. In the example in Figure 1.3,

the matched peak right before $m/z = 3000$ cannot be the product of one single fragmentation.

In light of these properties, we can modify our spectrum representation as follows:

1. We can focus on fragments from glycans only, which can greatly reduce the prediction space.
2. We can also overlook the peak intensities of b-ions, and only predict the spectrum intensities for glycan y-ions that are attached to the peptide.
3. We need to allow for at least two fragments on a glycan.

With respect to modification 2, we experiment with removing glycan b-ions and or glycans fragments that are not attached to the peptide.

From modification 3 above, we propose a spectrum representation as a $M \times M \times k$ matrix, where M is the number of monosaccharides in a glycan. k varies depending on our decision on the previous two modifications. In the simplest case of only predicting glycan y-ion peaks attached to the peptide, we set $k = 1$. Before filling in the values of the matrix, we now need to define the types of fragments below. Using the N-glycan core in Figure 1.1 as an example, the following table 3.1 shows the types of fragments and corresponding examples. Among the four categories of fragment ions, our observation shows that the Y and YY fragments have high peak intensities, whereas B and BY fragments in most cases do not.

Now we attempt to fill in an $M \times M \times 1$ matrix A , considering only Y and Y-Y type glycans that are attached to the peptide. We number the monosaccharides in the glycan from 1 to M , with 1 being the root HexNAc connected to the peptide, traversing in breadth-first search order, and M being a leaf node. Due to the tree-like property of a glycan, we can therefore label each edge e_i to be the unique edge connecting node i to its parent.

When there is a single cut at edge e_i , we record the intensity of the resulting Y fragment at $A_{1,i}$. Note that at $A_{1,1}$, the corresponding peak is the deglycosylated peptide. When two cuts at edges $e_i, e_j, i < j$ form a YY type fragment, we record the intensity of said fragment at $A_{i,j}$. In order to form a YY-type fragment, there cannot be a cut at edge e_1 , so the values of YY fragments do not conflict with Y fragments. Note that this matrix is an upper triangular matrix.

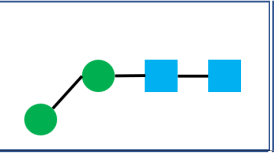
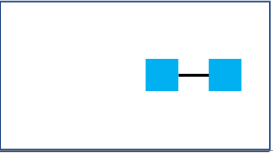

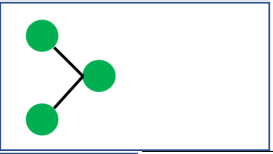
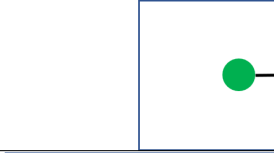
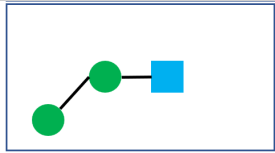
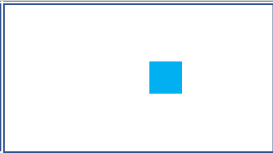
Type	Examples		Explanation
Y			The fragment from one cut attached to the peptide
B			The fragment from one cut attached to the Hydrogen on a leaf node
Y-Y			The fragment from two cuts attached to the peptide, resulting from the y-ends of both cuts
B-Y			The fragment from two cuts that is not a y-ion, resulting from the b-end of one cut and the y-end of another

Table 3.1: Categories of fragment ions resulting from at most two cuts.

If we consider other fragment types from modification 2, we set $k = 3$. The first channel is the same as when $k = 1$. The second channel records B and BY type fragments, whereas the third channel records Y and YY glycan fragments while removing the peptide. A B-type fragment ion resulting from a cut at edge e_i is filled in $A_{i,1,2}$. A BY-type fragment from edges $e_i, e_j, i < j$ must result from the b-end of cut e_i and the y-end of cut e_j due to the BFS order, and therefore is filled in $A_{j,i,2}$. Note that the second channel is a lower triangular matrix. For the third channel, for each Y and YY fragment, we remove the mass of the peptide and record the relative intensity of the matching peak in the same way as in the first channel.

A final consideration is a case where a Fucose is attached to the root HexNAc as in Figure 3.1a. We observe in the PSMs for these types of glycans, that the Fuc is easily fragmented, and could often result in high-intensity peaks for fragments such as in Figure 3.1c. In this case, the fragment is a result of three cleavages. If we were to include all three-cut fragments in glycans, our model would greatly multiply in scale and complexity, and given that this situation is only abundant in cases with a Fuc attached to the root, we can make the following modification.

For glycans with a Fuc at the root node, we add two more channels by setting $k = 5$:

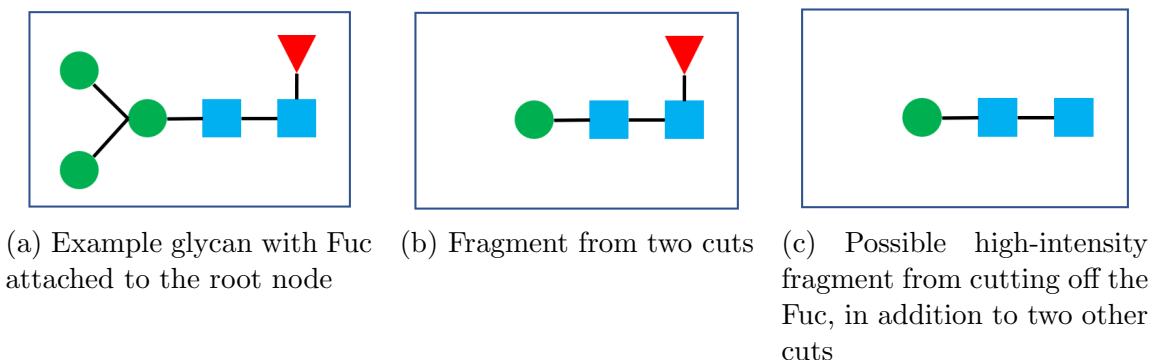


Figure 3.1: Example of possible glycan fragments with the presence of Fuc attached to the root node.

copies of the first and third channels while removing the Fucose molecule on the fragments. We should also note that this situation only occurs on Y and YY fragments, with or without the peptide.

Eventually, our model has the output dimension of $M \times M \times k$, with $k = 1, 3, 5$. In the later Section 3.3, the performance for each k value is discussed. Given an experimental spectrum as the truth label for training, we find all matching peaks for the identified fragments and fill in the matrix. In this way, we can use mean squared error (MSE loss) to fit our model. It is important to note, that more than half of the matrix entries do not have a matching intensity, either because the fragment is undefined, or because the experimental spectrum does not have a matching peak. In these cases, the values in the entries are set to 0. Later we describe a modified MSE loss to deal with the sparsity of the output matrix.

3.1.3 Model Structure for Spectrum Prediction

The model for spectrum prediction takes the general form of an encoder-decoder structure. The encoder is modeled after Graphormer, where the node feature and centrality encoding are fed to a multi-head attention (MHA) network, where the spatial encoding and edge encoding are added as bias terms for the attention. Afterward, layer normalization (LN) and feed-forward network (FFN) are applied as in a standard transformer. One layer of the Graphormer encoder includes the MHA and FFN, each with layer normalization applied and added afterward. In our spectrum prediction model, 32 layers of Graphormer encoder are stacked and eventually added to a learnable charge embedding. From our observation,

the same glycopeptide with different precursor charges can have very different spectrums, so the charge embedding is also added as a trainable part of the encoding. The Graphormer encoder outputs an encoding of dimension $L \times D$, where L is the maximum input length of the node feature, and D is the hidden embedding dimension. After adding the charge embedding and an FFN layer, the final output of the Graphormer is of dimension $L \times 1$. During each Graphormer layer, a graph representation of dimension $L \times L$ is kept during MHA, and the graph representation in the final layer is kept as another output of the model encoder.

The decoder takes in the $L \times 1$ Graphormer encoding, the $L \times L$ graph representation, and a pre-annotated edge mask of size $L \times L$ as input. The edge mask is a binary masking of possible peak predictions because we only care about the predicted intensities at possible fragments. We take the Graphormer encoding and multiply it with its transpose to obtain an $L \times L$ matrix. Then, we stack the three matrices to form a $L \times L \times 3$ input for the decoder. We use a convolutional neural network (CNN) as the decoder whose output is a $M' \times M' \times k$ matrix, where M' is the maximum size of a glycan, taken to be 32, and k is the hyperparameter for the number of channels of prediction.

Figure 3.2 demonstrates the structure of my spectrum prediction model for glycopeptides.

3.2 Training

3.2.1 Data Acquisition

Our training and testing data were first published by Liu *et al.* for pGlyco2.0 [18], which include enriched glycopeptides from mouse brain (PXD005411), kidney (PXD005412), heart (PXD005413), liver (PXD005553), and lung (PXD005555) tissues. PEAKS Studio with the Glycan Module [31] was used to analyze the data and produce glycopeptide spectrum matches. The protein database with UniProt Reference Mouse Proteome UP000000589 and the built-in mouse-yeast-N-glycans dataset from PEAKS were used respectively for the database search. The precursor tolerance was set at 10 ppm, and fragment tolerance at 30 ppm. The output PSMs were selected to be above 1% FDR.

Out of the five datasets, a total of 148812 PSMs were produced. The training, validation, and testing datasets are randomly selected with a fixed seed, to not include any glycans that appear in the other two datasets. The training dataset is further restricted by setting a minimum PEAKS glycan score of 29.0, selected manually. The validation dataset

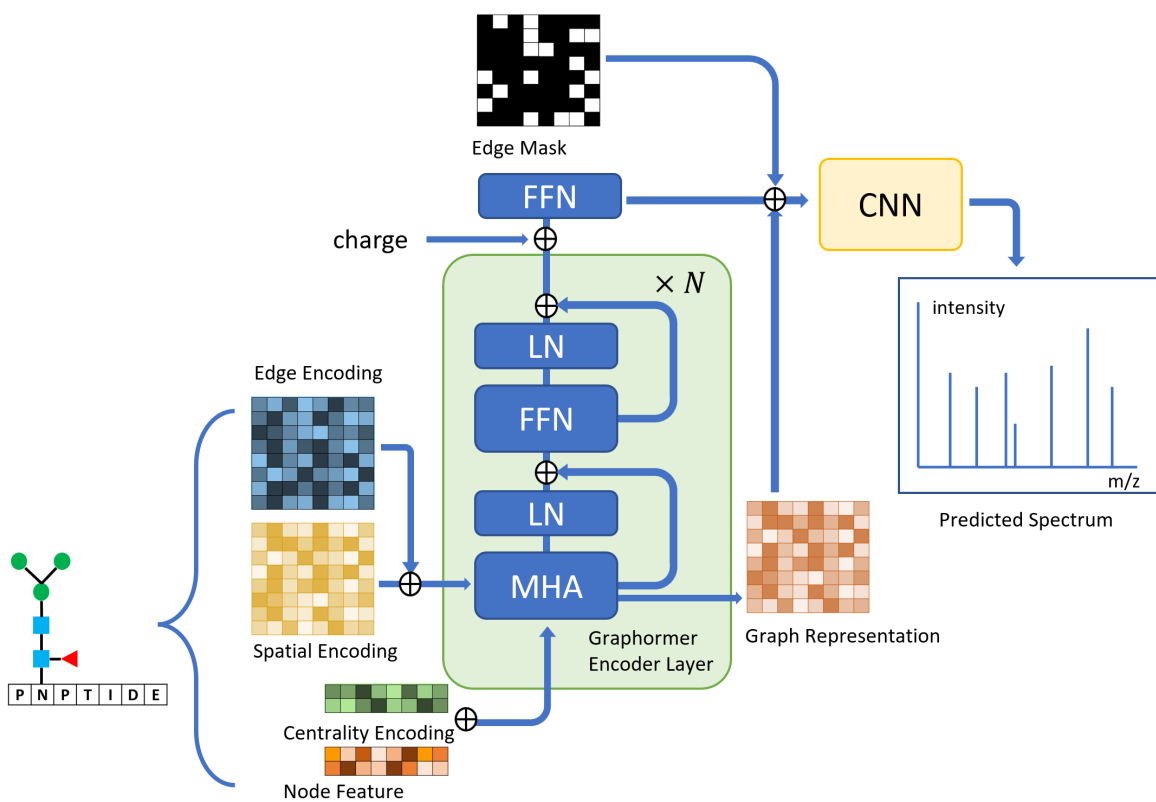


Figure 3.2: Illustration of Spectrum Prediction Model

is used to monitor overfitting during training, and the testing dataset is used for the final evaluation of the model.

3.2.2 Training and Loss Functions

A masked MSE loss is used as a loss function for training. Let I^P and I^E be the predicted and experimental spectrums in the form discussed in Section 3.1.2. Let B be a matrix of the same size with binary values, where 1 is filled if there is a theoretical fragmentation at the corresponding position, and 0 otherwise. For ease of documentation, we can flatten these matrices into 1-dimensional vectors of size n . We can then define our loss function

$$L(I^P, I^E) = \frac{\sum_{i=1}^n (I_i^P - I_i^E)^2 \cdot B_i}{\sum_{i=1}^n B_i} \quad (3.1)$$

Using the Adam optimizer with a learning rate of 0.001 and a batch size of 64, the model is trained on an NVIDIA GeForce RTX 3060 GPU for 30 epochs. Additionally, the Python package Glypy [13] was used for data processing.

3.3 Model Evaluation

We evaluate the similarity of each predicted spectrum to its experimental counterpart in the testing dataset with cosine similarity. We define the similarity metric differently from the masked MSE function during training. In the predicted matrix of peak intensities, it is likely that two entries in the matrix, although representing different fragments, are correlated to the same m/z value, and therefore are matched to the same peak intensity. This phenomenon may result in a bias for the true similarity of spectrums. Thus, from the predicted spectrum matrix, we can calculate the predicted relative intensity for each m/z value, taking the mean if there are multiple entries. From here, we apply cosine similarity to the processed spectrums. For each spectrum in the testing dataset, if the number of matched peaks is less than 3, it is discarded and not counted towards later evaluations, as the cosine similarities for these spectrum matches do not reflect the accuracy of the intensity predictions.

We observe an increase in median cosine similarity if we include the two Fucose channels in contrast to the three channels, with the median going from 0.893 when $k = 3$ to 0.909 when $k = 5$. However, the case of $k = 1$ outperforms the others, with a median cosine

similarity of 0.921. One conjecture is that there are significantly fewer predicted peaks, sometimes less than 50%, which leads to a higher cosine similarity score.

We compare our spectrum prediction results against previous attempts for glycopeptide data, namely Zhang *et al.* [39] and Klein *et al.* [12]. Zhang *et al.*'s work arrives at a median cosine similarity of 0.71, using their own generated dataset of 196 spectra in the testing set. Klein *et al.*'s work arrives at a median similarity score of 0.76 when testing on the mouse brain dataset. Although Klein *et al.*'s work used the same mouse tissues dataset as our experiment, their probabilistic regression model did not separate the training and testing dataset by glycans. Both Zhang *et al.* and Klein *et al.*'s works define features in the models manually, which makes it difficult to replicate their methods on our dataset. To make a meaningful comparison with Klein *et al.*'s work, our model was trained on the mouse tissues dataset other than the mouse brain, and tested on the mouse brain dataset. In this experiment, our median cosine similarity becomes 0.943, which is 24% higher than Klein *et al.*'s results. However, it is important to note that this number of 0.943 is only used to show our improvement over previous work, and cannot be taken as the true testing performance, as the deep learning model may overfit on previously seen data.

Figure 3.3 shows an example of predicted versus experimental spectra. In the mirror plot, the observed, or experimental spectrum is shown above the x -axis, without unmatched peaks, and the predicted spectrum is shown below the x -axis.

3.4 Discussion

3.4.1 Case Study on Charge

We examine the poorer-performing spectrum prediction results with the hopes of gaining insight into the reason for inaccuracy. One observation is that charge greatly affects the peak intensities of a spectrum.

Figure 3.4 shows an example of how the charge affects the spectrum of a glycopeptide. Observing the top portions of each plot (the experimental spectrum), we see that different charges may drastically affect the peak intensities of the same glycopeptide, but different glycopeptides are not equally affected by the charge of the precursor. Although we have added a charge embedding into the spectrum prediction model, for the case of different charges on the same glycopeptide, we do not see a significant difference in the predicted spectrum. Our conjecture for our model's failure at distinguishing charge is that examples like in Figure 3.4 are rare, and the majority of our training data display a general trend for spectrum intensities, which is what our model picks up in the end.

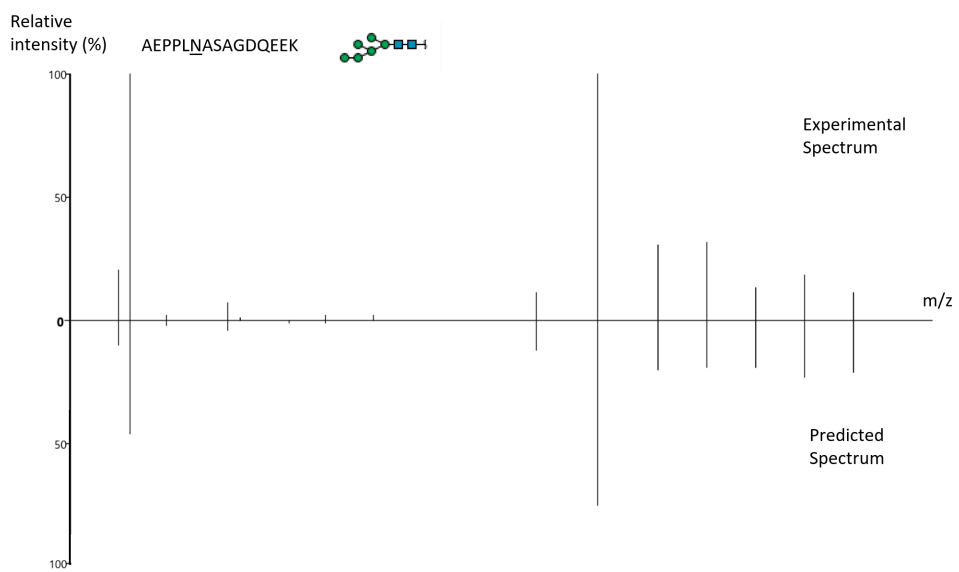


Figure 3.3: Example mirror plot of experimental versus predicted spectra for peptide AEPPLNASAGDQEEK and glycan id 23. The top graph shows the experimental spectrum and the bottom the predicted spectrum.

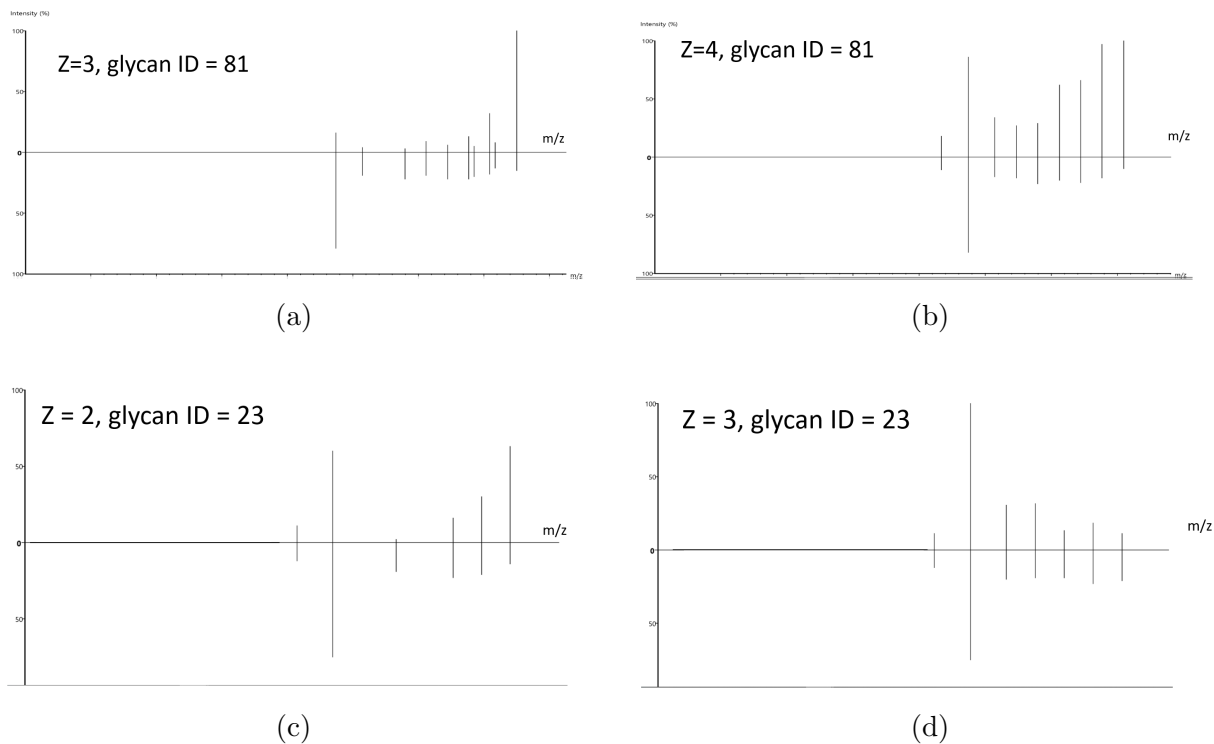


Figure 3.4: Examples of experimental and predicted spectra. In each subfigure, the top panel is the experimental spectrum, and the bottom is the predicted. z in the plots refers to the charge of the precursor. The pair of (a) and (b) show that different charges on the same glycopeptide will result in different spectrum intensities. So does the pair of (c) and (d). However, we observe that (a) and (d), although having the same charge, show different spectrum intensities, while (a) and (c) are similar in the general trend of the spectrum.

3.4.2 Conclusion and Future Research Directions

To conclude the chapter on spectrum prediction, we have described our deep learning model and presented its testing results, which are shown to have significantly improved over previous works on spectrum prediction for glycopeptides. The application of this model to the practical glycopeptide search will be investigated and discussed in Chapter 5.

As mentioned earlier, our model does not learn well how charge affects the spectrum, due to an overwhelming abundance of “normal” spectrums that show a similar trend. One possible way to correct this is by augmenting data that represent the effects of charge.

Another direction of future work is to investigate other deep learning models. For example, diffusion models have shown promising results in image generation lately and may be applied in the decoding stage of our model to generate the predicted spectrum. For our work, however, we decided against the use of diffusion models because they are massive deep-learning architectures that require a large dataset to train, which we do not have for glycopeptides. With more high-quality glycopeptide data, either from experiments or from data augmentation, diffusion models may become more possible.

Chapter 4

Retention Time Prediction

4.1 Model Design and Structure

4.1.1 Pre-training on Peptide Inputs

In contrast to spectrum prediction, the prediction of retention time requires an output of only a floating point value. The encoder model of a glycopeptide remains the same as the spectrum prediction model (see Chapter 3), but with 3 layers in contrast to 32 layers. The decoder takes the Graphormer encoder output and applies a fully connected network to output one single value.

From previous works on glycopeptide RT prediction, we notice that many only predict the shift in RT when a glycan is added. However, we attempt to predict the RT for a glycopeptide directly. Unlike spectrum prediction, where the peptide sequence may play a rather small role in the model, we recognize that RT prediction highly depends on peptide information.

Besides the Graphormer encoder on glycopeptide input, we decided to emphasize the peptide sequence by adding a sequence encoder, taking the peptide sequence as its input. We combine both encoders before feeding the latent representation into a decoder. In addition to the added attention to the peptide sequence, this proposed model structure has the advantage that the sequence model can be trained separately on non-glycosylated data.

Therefore, we decide to pre-train a peptide RT prediction model on peptide data, the model structured like one from PROSIT, but using Transformer layers instead of GRU. The

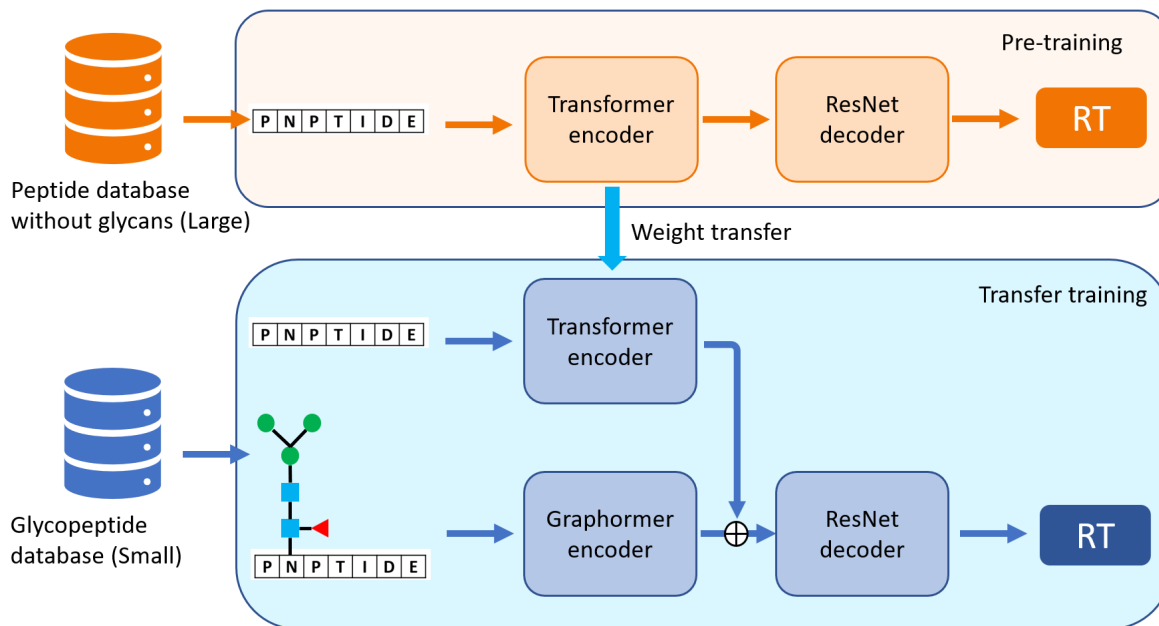


Figure 4.1: Illustration of RT prediction models, transferring the sequence encoder from peptide RT pretraining

peptide RT prediction model includes a Transformer-based sequence encoder and a decoder comprised of ResNet blocks and two fully connected nets. Then, the sequence encoder's learned weights are transferred to our sequence encoder in the glycopeptide prediction model. After the transfer, we train the entire model together on glycopeptide data to finetune the RT predictions.

Figure 4.1 shows a diagram of the RT prediction procedure for glycopeptides, as described above.

4.2 Training

4.2.1 Regression to Normalized Retention Time with iRT

Retention time records the time that a particle stays in the liquid chromatograph column. Depending on different experiment setups, the retention time for the same peptide may vary

Maximum variance for the same glycopeptide (in minutes)	iRT	RT
Mean	2.5	4.9
Median	0.29	0.41
95 Percentile	12.8	32.5

Table 4.1: Statistics on RT and iRT variance for the same glycopeptide across samples

greatly. The variation in RT introduces noise to the training process. Here, we introduce the idea of an indexed retention time (iRT), which is a way to normalize retention time across different samples. Given a library that records the iRT for peptides, one can use regression to fit the retention times of peptides from any sample. Isolating peptides in the sample that are in the library, one can use their sample retention time and the recorded iRT to fit a regression. Using the iRT for peptides across different samples ensures that sample variance does not change the final training result.

There is not an iRT library for glycopeptides, but our data samples (from [18]) contain both glycosylated and non-glycosylated peptides. Using the regular peptides as anchors and finding their iRT in a peptide iRT library, we can still map the retention times for glycopeptides into iRTs.

To show the significance of converting to iRT, we record the maximum difference in RT and iRT for each glycopeptide across all samples. For example, one glycopeptide has a retention time of 101.3 minutes from Mouse Kidney Sample 1 and another retention time of 145.6 from Mouse Liver Sample 5. After regression to iRT, the respective iRTs become 54.4 and 53.4 minutes. The mean, median, and 95 percentile statistics of these differences are shown in Table 4.1.

In light of the advantages of iRT, later discussions on “RT prediction” will generally refer to iRT prediction unless specifically stated otherwise. We preprocess all the label retention time to be mapped into iRT and have our models trained to predict iRT. We further make the assumption that with a good RT to iRT regression, if our model performs well on iRT prediction, it does so too on RT prediction.

To prepare our data for each sample, we find anchor peptides from each sample that are in the iRT library and use them in a polynomial regression for other glycopeptides in the sample. We perform a separate regression for each sample, thereby eliminating the sample variance problem.

4.2.2 Using Feature RT Instead of Spectrum RT

While calibrating each sample to iRT solves the problem of variance across samples, there are situations where different retention times are reported for the same glycopeptide within the same sample. We eliminate this problem by using the feature RT. During the database search by PEAKS, a feature detection step is performed, grouping several MS2 scans associated with the same peptide into the same “feature”. Each feature spans a range of retention times. So for each MS2 scan, instead of using the retention time of its precursor, we use the mean retention time of its feature. This new feature RT is then used in the regression above and calibrated to iRT.

Replacing precursor RT with feature RT reduces variance within each sample, and our experiments find that it improves the model accuracy.

4.2.3 Data Acquisition

The peptide database with iRT information is taken from the Westlake Spectral Library with human peptides [36] with 8175517 peptides. This dataset is used both for peptide iRT prediction pretraining and as an iRT library for sample calibration.

Glycopeptide data for training and testing are the same as that for spectrum prediction, using the mouse tissues dataset from pGlyco 2.0 [18]. We conduct an evaluation on an additional dataset of fission yeast, also taken from the pGlyco 2.0 paper. We ensure that the testing datasets do not contain glycans or peptides seen in the training database. The additional constraint for distinct peptides in the testing dataset is due to the high influence of peptide sequence on retention time. The glycan database is PEAK’s [31] mouse yeast glycan database, and the protein database is the combination of Uniprot Protein databases with species of *S. pombe* (yeast) and *Mus musculus* (mouse). Running the yeast samples on PEAK Glycan, we set the precursor tolerance at 10 ppm and fragment ion tolerance at 30 ppm. We then output the PSMs under 1% FDR.

4.2.4 Training and Loss Functions

As mentioned in the model design and structure, as well as illustrated in Figure 4.1, we first train a peptide model, and then transfer the sequence encoder and train the glycopeptide iRT model. Both models are trained with the Adam optimizer with a learning rate of 0.0001 and a batch size of 64. The peptide model is trained for 5 epochs, and the glycopeptide

model is trained for an additional 50 epochs, both on the same Nvidia GeForce RTX 3060 GPU.

With the output being a single floating point value, we use L1 loss over each batch. Later we show during evaluation, that the model trained on L1 loss outperforms that with L2 as well as Huber loss.

4.3 Model Evaluation

4.3.1 95 Percent Delta and Ablation Studies

Previous RT prediction attempts on glycopeptide use Pearson Correlation R as a criterion for prediction accuracy, with Ang *et al.* [1] getting at 0.967 on a small dataset and Klein *et al.* [14] getting at 0.98 and 0.897 respectively for fixed peptide and cross peptide predictions. We experiment with different training options such as using feature RT, different model hidden dimensions, and different loss functions, as well as evaluate both the testing dataset from mouse tissues and fission yeast. For all these experiments, the resulting Pearson R for iRT prediction is at least 0.98. Although this number is on par with previously reported results (and actually higher, because we predict on different peptides), it is neither convincing as an evaluation metric, nor demonstrative of the differences across our experiments.

PROSIT [10] predicts iRT for peptides and proposes the evaluation metric of 95 percentile delta $\Delta t_{95\%}$, which is the 95 percentile of the absolute difference in predicted and target iRT. For example, they report a $\Delta t_{95\%} = 85$ seconds on a testing dataset. Although we cannot compare our results with peptide iRT prediction models, we can use this metric to compare our experiments and optimize for the best hyperparameters.

We also implement two other iRT prediction methods to compare against our main model: a k-nearest neighbor method, and a composition-based model. For the k-nearest neighbor method, we take $k = 5$, and for each glycopeptide in the testing dataset, we find five glycopeptides in the training dataset that are the “closest” to it and use the mean iRT of the five examples as the predicted iRT. To define closeness, we use the Graphormer encoder to project a glycopeptide onto its latent embedding, and then use cosine similarity to measure distance in the latent space.

As for the composition-based model, we wanted to investigate whether different glycan structures with the same composition affect retention time. In previous works such as Klein *et al.* [14] and Park *et al.* [23], they only use the number of Hex, HexNAc, Fuc, *etc.* as

Experiment	$\Delta t_{50\%}$ (median)	$\Delta t_{95\%}$
Precursor RT instead of feature RT	2.14	16.02
L2 loss instead of L1 loss	1.89	10.33
Composition model	2.21	14.66
5-nearest neighbor	1.73	13.95
Final model	1.15	9.24

Table 4.2: Median and 95 percentile delta (in minutes) for predicted versus experimental iRT across different experiments

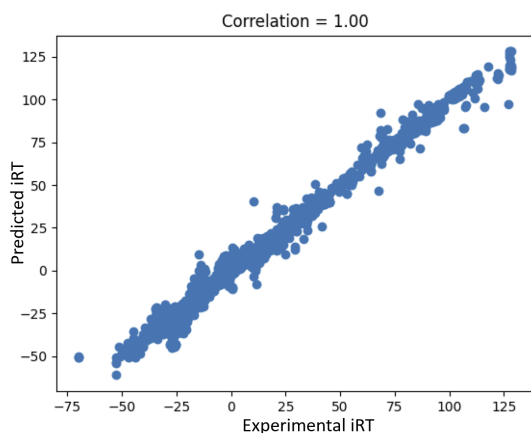
features in their prediction. That is to say, they only consider the *composition* of a glycan rather than its structure. Also for the glycopeptide search engine MSFragger [15], they only produce PSM matches with glycan composition. In Chapter 5, we apply our models to rescore results from different search engines. In order to comply with MSFragger, we decided to train a model that takes the glycan composition as input. The model contains a sequence encoder transferred from peptide iRT prediction, and the glycan encoder portion takes an input as a vector of five integers, representing the number of monosaccharides of each of the five types in the glycan. The encoder includes a trainable embedding and fully connected nets, and its results are concatenated with that of the sequence encoder, before being fed into the decoder portion kept the same as the glycan structure model.

Table 4.2 shows our ablation study that compares the performance between several modifications described above. The final model with feature RT and L1 loss outperforms the others in both median and 95 percentile delta. In Figure 4.2, we plot the correlation of observed and predicted retention times. By contrasting the two plots generated from the final glycopeptide model and the composition model, one interesting observation is the horizontal line of dots in the lower right quadrant in the composition plot 4.2b. This is the case when PSMs with different observed iRT are given the same predicted iRT. One conjecture on the reason for this phenomenon is that structural information is not fed to the composition model.

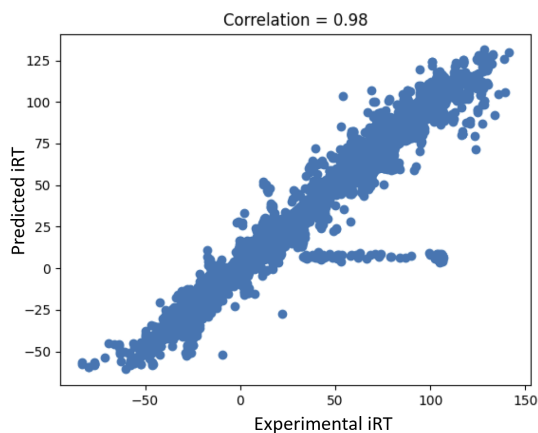
4.3.2 Interpretation of Results

With the Pearson R and with the 95 percentile delta, we still need to answer the following questions:

1. Does the model learn peptide information?



(a) Correlation plot of observed versus predicted iRT for final model



(b) Correlation plot of observed versus predicted iRT for composition model

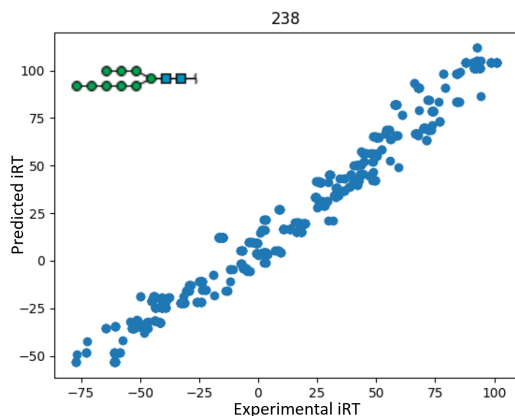
Figure 4.2: Correlation plots for iRT prediction models

2. What about glycan information?
3. Does glycan structure affect the retention time, or is it just the composition that matters?

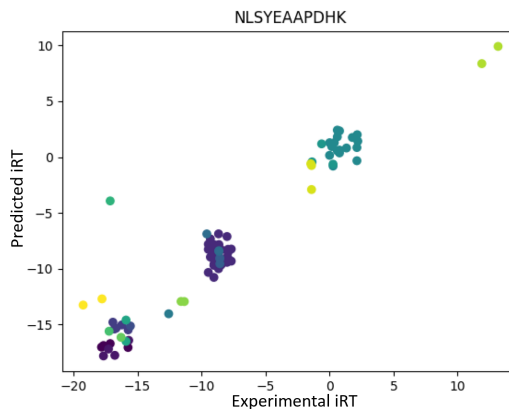
To further interpret our model’s understanding of peptide and glycan information, we conduct two small experiments on our iRT prediction model: plotting the predicted versus observed iRT in the testing data with the same glycan but different peptides (Figure 4.3a), and plotting the reverse: with the same peptide but different glycans (Figure 4.3b).

PSMs with the same glycan cover a wide range of iRT and show a highly correlated linear trend, so we can be confident in saying that the model understands peptide information. PSMs with the same peptide generally cover a smaller range of iRT, which is to be expected from earlier work on retention time shifts caused by glycosylation [1, 23]. In the example in Figure 4.3b, with the exception of one PSM, the data points show a highly correlated linear trend as well. This indicates that our model is able to pick up nuances in glycans that affect the retention time of the glycopeptide. The interesting phenomenon is that the PSMs are separated into clusters in contrast to the case with fixed glycans.

Further investigation into the clusters and the composition of each PSM for the fixed peptide example NLSYEAAPDHK, we observe that each composition is confined to a cluster, indicating that variance in observed and predicted iRT for the same peptide-composition pair is generally small. In fact, the 95 percentile variance of peptide-composition



(a) Correlation plot of observed versus predicted iRT for fixed glycan



(b) Correlation plot of observed versus predicted iRT for fixed peptide, where PSMs with the same glycan composition are given the same color

Figure 4.3: Correlation plots for fixed glycan or peptide

pairs in the testing dataset is 9.10 minutes, with the median being 0.02 minutes. From this observation, one can say that the composition of the glycan is the major factor in retention time difference, while different structures within the same composition have little variance in observed and predicted iRT. However, the glycopeptide structure model reduces the 95 percentile delta from the composition model by almost 37%. Increasing the scale of the composition model does not improve its performance, so we can only conjecture that there is structural information learned by our Graphormer-based model that composition-only input fails to include.

To gain insight into the rules or patterns in the structures and compositions that are grouped in the same cluster, we observe that the clusters with higher predicted and experimental iRT values often have a higher number of sialic acids (NeuAc and NeuGc), and therefore a higher ratio of sialic acids to HexNAc's. For the same peptide NLSYEAAPDHK, we plot the correlation between the sialic acid to HexNAc ratio and the predicted iRT in Figure 4.4. The Pearson correlation is 0.87, and we can say that in general, higher iRT values are only observed when the ratio is high. Similar trends are seen with experimental iRT in contrast to predicted iRT, with a Pearson correlation of 0.80. We also calculate the Pearson correlations for the ratio of NeuAc to HexNAc and the iRT values and discover similar results. This result corroborates that of Park *et al.*'s findings [23], where the Ln/Nn metric, which is the ratio of NeuAc to HexNAc, combined with

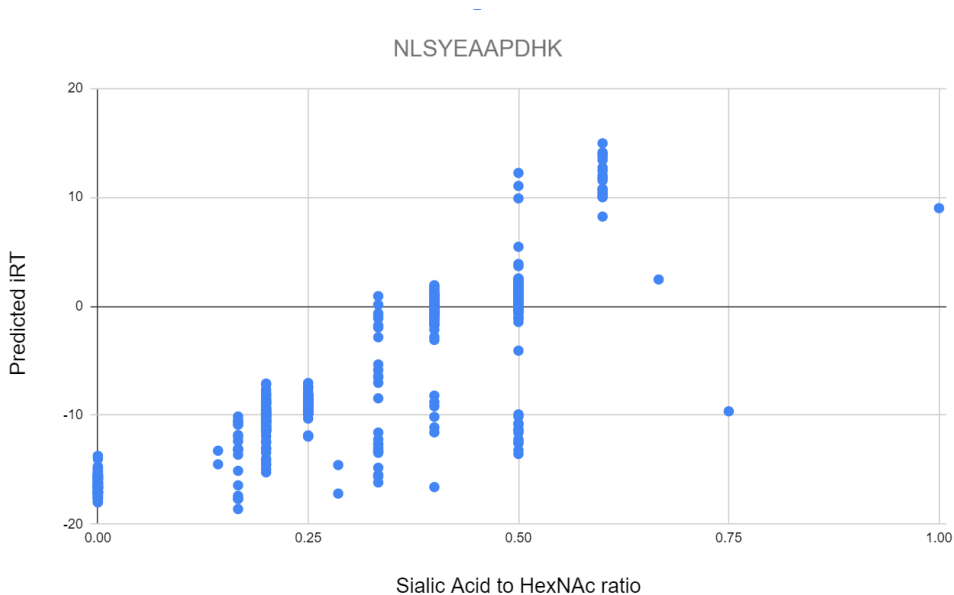


Figure 4.4: Correlation between the sialic acid to HexNAc ratio and predicted iRT

corresponding peak intensities, is also positively correlated to relative retention time.

4.4 Discussion

In this chapter, we introduce our model for retention time (iRT) prediction, joining a Transformer sequence encoder with a Graphormer glycopeptide encoder. In our ablation studies, we construct the best model using iRT regression, feature RT, and L1 loss and compare our model with two other models: k nearest neighbor and a composition model. Comparing our model against previous RT prediction models on glycopeptide data, we achieve a higher Pearson correlation than the earlier works. We also inherit the evaluation criteria of 95 percentile delta from PROSIT and achieve $\Delta t_{95\%} = 9.24$ minutes on the mouse tissues testing dataset. We attempt to interpret our model’s understanding of both peptide and glycan information, and the experiments with fixed glycan and peptide show that the model learns well on both peptide and glycan differences.

One area of concern in our approach is that earlier on, we make the assumption that the regression from RT to iRT is well-fitted. This is the case for most samples, especially larger ones, but the worst case in a smaller Mouse Heart sample reveals a 95 percentile

regression loss of 12.5 minutes, with the median being 1.2 minutes. This then introduces loss during regression from RT to iRT, but this loss is not manifested in the performance evaluation of our model, as we only compared the predicted and experimental iRTs. We now face the problem that our experimental iRTs, used as training targets, may include a large noise.

To fix this issue, larger data samples with more anchor peptides will provide more accurate regression. We circle back to the same issue we had with spectrum prediction: We are in need of more large-scale and high-accuracy data to train and evaluate our model.

Chapter 5

Using Prediction Models in Glycopeptide Search

5.1 Database Search

5.1.1 Rescoring

Our spectrum prediction and iRT prediction models show high performance in cosine similarity and 95 percentile delta respectively. However, we need to investigate whether the prediction models actually help glycopeptide search. We first take inspiration from PROSIT [10] to use our prediction models for database search rescoring, the process of which is described below.

We use the mouse tissues dataset as our target dataset, and generate an equally sized decoy dataset, by permuting the spectrum peaks for each MS2 scan. We then combine the target and decoy datasets as the input for glycopeptide search engines. We ask the database search to produce all PSMs, without limiting FDR. In the output, each PSM includes the scan number (and consequently whether it comes from the target or decoy database), the matched peptide, the matched glycan, and the match scores. The match scores generally include a glycan score, a peptide score, and a combined total.

For each output PSM, we run spectrum and iRT prediction models and obtain the spectrum match score with cosine similarity, and the iRT match score, which is the absolute difference between predicted and observed iRT. Combining the search engine PTM scores and our prediction scores, we can fit an SVM to fit a final score for each PTM, with 1 being

Search Engine	# Target PSMs	# Decoy PSMs	# PSMs with 1% FDR (before rescoring)
PEAKS Glycan [31]	19235	4001	11023
pGlyco3 [35]	20050	933	19539

Table 5.1: Number of PSMs outputted by PEAKS Glycan and pGlyco3

the maximum score for the best possible match, and 0 being the lowest score. We then sort all PSMs by their final score and record the number of PSMs, number of identified glycopeptides, and number of identified glycans with $FDR \leq 1\%$.

It is worth noting, that although our generated target and decoy databases have the same number of MS2 scans as input to the search engines, not all MS2 scans will be matched with a PSM for output. The search engines will filter out MS2 scans that do not match any glycopeptide first. Table 5.1 records the number of output scans from PEAKS glycan [31] and pGlyco3 [35], with the same target decoy input generated from the Mouse Brain Sample 1 dataset. There are 110004 MS2 scans in total in the generated target decoy input. We see that the numbers of identified PSMs in the target dataset are similar for both search engines, whereas PEAKS glycan outputs about three times as many decoy PSMs as pGlyco3. We also observe that there are many more target PSMs than there are decoy PSMs. This is to be expected, as the randomly shifted peaks in the decoy spectrums are more likely to be filtered out by the search engines.

From here, we proceed with rescoring as described above. Table 5.2 shows the numbers of PSMs, glycans, and glycopeptides after 1% FDR, by using different rescoring methods, that is, including different scores in the SVM classifier, for PEAKS Glycan and pGlyco3. For PEAKS Glycan, significant increases from the search engine results are shown in bold-face, while the best rescoring results are colored in red, whereas the rescoring attempts for pGlyco3 show no significant increase or decrease. The reason that pGlyco3 results are not affected by rescoring is likely due to its highly unbalanced target-to-decoy ratio, and also because their 1% FDR is already very high from the search engine scores. This makes it more difficult to fit the SVM otherwise, even though we use a weighted SVM model to account for the class imbalance already. In contrast to pGlyco3, rescoring on PEAKS Glycan seems much more promising. When using the spectrum score in conjunction with the search engine scores, we increase the PSM, glycan, and glycopeptide identification rates by 5.2%, 5.5%, and 5.6% respectively. The numbers drop when combining spectrum and iRT scores into the rescoring, which is likely because another dimension is introduced to the SVM.

Scoring Methods	PEAKS Glycan			pGlyco3		
	PSMs	Glycans	Gly-pep	PSMs	Glycans	Gly-pep
Search engine	11023	1168	6886	19539	1934	11606
+ iRT score	11021	1207	6865	10537	1935	11605
+ Spec score	11596	1230	7273	10538	1935	11607
All scores	11235	1222	7016	10535	1934	11607

Table 5.2: Number of PSMs, glycans, and glycopeptides after 1% FDR, according to different scoring and rescoring methods for PEAKS Glycan and pGlyco3

We have demonstrated above, that using spectrum and iRT prediction models can be used in rescoring to increase identification from the database search. However, we need to point out the flaws of this approach. The numbers of decoy PSMs from search engine outputs are much lower than those of target PSMs when we prefer a more balanced output. To increase the number of decoy PSMs, the decoy spectrums need to pass the filtering, which means that they need to have characteristics similar to a target spectrum. However, if the decoy spectrums are too close to the target spectrums, the purpose of the target decoy approach is lost, as a decoy spectrum may actually represent a real spectrum. Hence, there is a trade-off of how random the decoy spectrums are, and it is difficult to design a perfect decoy database.

PROSIT [10] is not faced with this issue in their rescoring process, because they use decoy proteins (and consequently peptides) instead of decoy spectrums. For each protein in the search database, the decoy protein is generated by reversing or randomly shuffling the protein sequence, and the peptides are generated following the same digestion rules. In this way, the search engine does not filter out the decoy entries and cannot distinguish between target and decoy entries. When it comes to glycopeptides, however, a decoy protein database is not enough and we would need a decoy glycan database as well. The generation of a decoy glycan database such as in [29] again faces the trade-off like decoy spectrums. Generating a decoy glycan by randomly building a tree of monosaccharides is too different from the real N-glycans and the decoy glycans can be easily told apart from real glycans, making the FDR estimation unreliable. On the other hand, if a decoy glycan is similar to a real glycan in structure and composition, our current knowledge of existing glycans is not enough to say for certain whether the generated decoy glycan cannot be a real one.

In any case, generating a convincing decoy database is the key to obtaining conclusive results on our rescoring attempt. During the database search process in search engines, they also use the target and decoy method to compute the 1% FDR for their final outputs,

and currently, different glycopeptide search engines use different decoy databases that are not openly disclosed. This can be an issue because simply comparing the number of PSMs produced after 1% FDR between search engines is not convincing as their FDR criteria are different. Building a universal decoy database that can be used across search engines can also unify an evaluation metric and enable us to compare search engine performance.

5.1.2 Filtering

Besides rescoring we can use the spectrum and iRT prediction models as filters for database search: if a PSM has a high glycan score produced by the search engine, but has low cosine similarity with the predicted spectrum, or has a high iRT difference, then we can say that it is likely to be an incorrect match. In this section, we show that the iRT prediction model has the potential in filtering out incorrect PSMs. The spectrum prediction model, however, does not show to be useful in filtering, so this section will be focused on the iRT model.

Fission Yeast Data

Zeng *et al.* [35] introduce two criteria on the correctness of glycopeptide identification for fission yeast data (PXD005565 [18]):

1. If the identified glycan contains Fuc, NeuAc, or NeuGc, the PSM is a false positive. This is because the glycans in fission yeast samples are highly mannose, which means to only contain Hex and HexNAc monosaccharides.
2. If the identified peptide is from the mouse protein database, then the PSM is a false positive.

We can then evaluate the iRT difference for PSMs that are false positives, and examine if the iRT difference can be used to filter the false positives.

We evaluate the iRT prediction model on the fission yeast data, with outputted PSMs from PEAKS Glycan [31], pGlyco3 [35], and MSFragger [15]. Specifically, we use the composition model because MSFragger does not provide a glycan structure in the output PSMs. For each search engine, we plot a scattered graph to see the correlation between the glycan score given by the search engine and the iRT difference calculated from our prediction model. We wish to see high glycan scores correlated to low iRT differences,

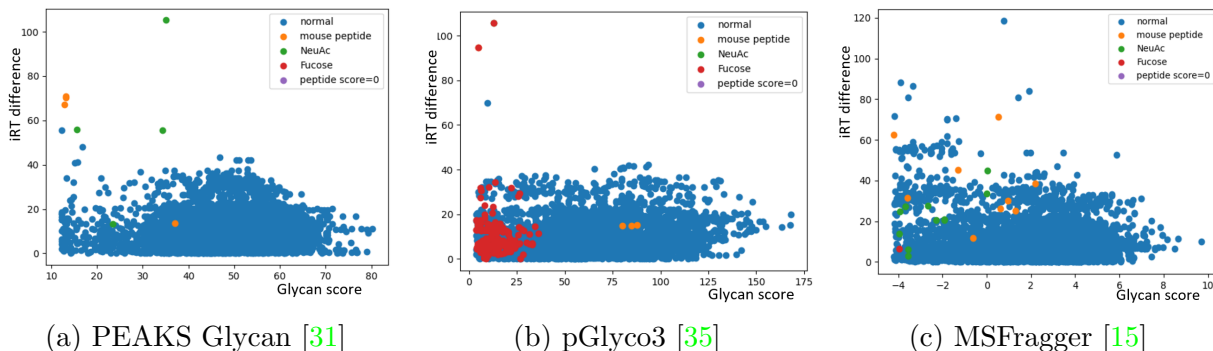


Figure 5.1: Glycan score versus iRT difference plots for fission yeast data with different search engines

and vice versa. Figure 5.1 show the plots for each search engine. In each plot, possible false positive PSMs as introduced above are highlighted in different colors. For example in the PEAKS Glycan plot, we see that three out of four PSMs with mouse peptide and three out of four PSMs with NeuAc are out of distribution with very high iRT differences. These outliers can be filtered out by iRT difference and can further increase the search engine precision. For pGlyco3, only two PSMs with Fucose are shown as outliers, whereas a majority of others are within the distribution of other normal PSMs. The distribution for MSFragger is more scattered, and we see outliers from both false positive PSMs and normal PSMs. Note that even if a PSM is labeled as normal, it could still be an incorrect match, but we do not have enough information to tell.

In Table 5.3, we calculate the mean, median, and 95 percentile of iRT differences (in minutes) from three search engines, with the lowest differences in boldface. We also want to see the effects of filtering by setting the iRT difference to be below 30 minutes and discarding the PSMs with high iRT differences. From the table, we observe that the percentage of discarded PSMs is twice as much for MSFragger as for the other two search engines, which is manifested in Figure 5.1, where there seem to be more outlier PSMs for MSFragger. In Table 5.4, we further investigate the percentage of PSMs and the percentage of false positives discarded by the filtering of retention time prediction. For each search engine, the percentage of false positives discarded varies, but are all significantly higher than that of total PSMs. It is worth noting that PSMs that contain Fucose (red dots in Figure 5.1) are not counted towards false positives, because when dealing with Yeast data, the user can set the allowed number of fucose to zero and these PSMs would not be outputted.

iRT Δ	Mean	Median	$\Delta t_{95\%}$	# PSM	# PSM after iRT $\Delta \leq 30$	% Discarded
MSFragger	11.59	9.05	32.38	4720	4386	7.08 %
pGlyco3	10.54	8.94	27.64	3553	3425	3.60%
PEAKS	10.17	8.55	27.67	4035	3890	3.59%

Table 5.3: Statistics on the iRT difference with fission yeast PSMs from different search engines

Software	# PSM	% Discarded	# FP (no Fuc)	% FP Discarded
MSFragger	4720	7.08 %	19	42.1%
pGlyco3	3553	3.60%	5	20%
PEAKS	4035	3.59%	8	75%

Table 5.4: When discarding PSMs with $\Delta_{iRT} > 30$, the percentage discarded in total as well as the percentage of false positives (excluding Fucose) discarded for each software

Mouse Tissues Data

Besides the false positive criteria on fission yeast by Zeng *et al.* [35], another collective study by Kawahara *et al.* [11] reveals a criterion on mouse tissues data. They state that search engines that report a low percentage of PSMs with NeuGc or multiple Fucose (> 2) tend to have low actual mass error. Therefore, we can examine how iRT difference correlates with PSMs with NeuGc or multiple Fucose. Using the PTMs generated from PEAKS Glycan (without the 1% FDR limit), we plot the glycan score versus iRT difference scatter plot, highlighting PSMs with NeuGc or at least three Fucose (Figure 5.2).

PTMs with high glycan scores and high iRT differences are considered outliers, as they could be incorrect. From Figure 5.2, the two outliers with glycan scores above 10.0 are both highlighted with color: one with NeuGc, and the other with multiple Fucose.

In another experiment, we have PEAKS Glycan produce the top 10 glycopeptide candidates for each spectrum and select the one with the lowest iRT difference. From Figure 5.3, we observe a general reduction in outliers, which is to be expected, since we choose based on low iRT difference. Additionally, another useful phenomenon is that the number of data points with NeuGc and multiple Fucose are reduced in Figure 5.3b. Therefore, we can make the important conclusion that using iRT difference to choose the search engine candidates can result in reduced identification of NeuGc and multiple Fucose PTMs.

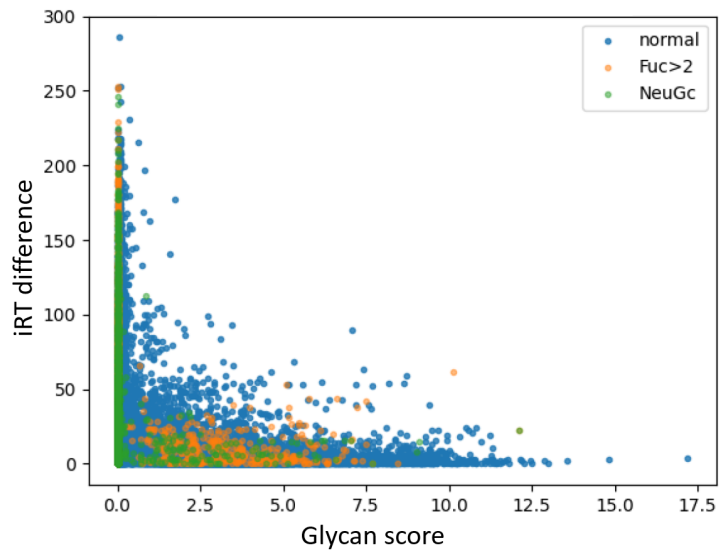
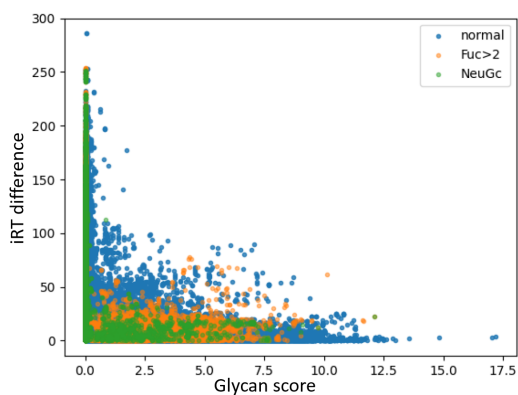
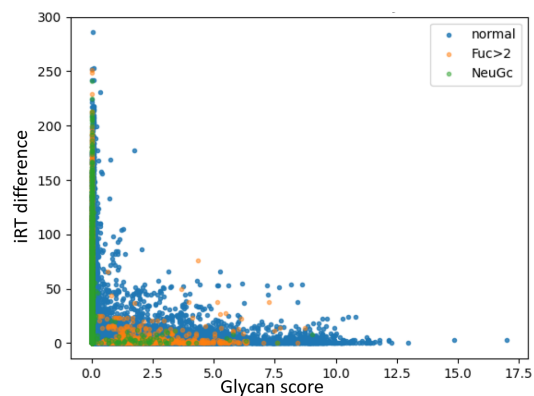


Figure 5.2: Glycan score versus iRT difference plot for mouse brain data



(a) All candidate PTMs for mouse brain data



(b) Choosing the best candidate by iRT difference

Figure 5.3: Choosing the best candidate by lowest iRT difference for each PTM reduces outliers and the identification of glycans with NeuGc and multiple Fucose

5.2 *De Novo* Search

Apart from the database search, we wish to investigate if our spectrum and iRT prediction models can be used during *de novo* glycopeptide search. The *de novo* glycopeptide search program written by Q. Zhang [37] in personal communication (later modified and published in [31]) is used. For each MS2 scan with an identified peptide, the *de novo* program produces two glycan structure candidates, at least one of which is an exact match to the target glycan. We incorporate spectrum and iRT prediction models to score the candidates. There are 834 sets of targets and candidates for this evaluation. To evaluate the performance of the *de novo* model in conjunction with the scoring, we have two strategies:

1. For any scoring scheme, if the candidate with a **strictly** better score matches the target glycan, we count it as a successful identification.
2. For any scoring scheme, we select the better-scoring candidate and in the case of a tie, we randomly select one. We then count the number of matches of selected glycans with target glycans as the number of successful identifications.

We look at five scoring functions:

- A. The number of peaks from theoretical fragments matching the experimental spectrum
- B. The average relative intensity of the matched peaks
- C. Cosine similarity between predicted and experimental spectra
- D. Difference between predicted and observed iRT (inversed)
- E. Linear combination of C and D

Using the five scoring functions with the two counting strategies, we generate Table 5.5. We observe that using strategy 1 with strictly higher scores, the iRT model has the highest rate of identification at 64.1%, whereas using strategy 2, which is a looser count allowing ties, the combination model achieves 70.0% identification. For either strategy, the identifications using the deep learning models in this thesis significantly improve upon those without deep learning.

Scoring Function	% identification by strategy 1	% identification by strategy 2
A # Peaks	26.7	61.6
B Avg. Intensity	31.4	60.6
C Cosine Sim	52.6	62.7
D iRT Diff	64.1	64.1
E Comb.	55.3	70.9

Table 5.5: Glycan identification rates from *de novo* model using different scoring functions and counting strategies

5.3 Discussion

In this chapter, we apply our spectrum and iRT prediction models to glycopeptide searching pipelines including database search and *de novo* search. In rescoring, we see that the spectrum model increases the identification of PTMs, glycans, and glycopeptides for PEAKS Glycan. However, we also uncover the flaw of decoy databases and therefore the persuasiveness of the rescoring results. When the iRT model is used in filtering, we see that it successfully reports the false positive PSMs for fission yeast data as outliers. For mouse data, our model reduces the number of PTMs with NeuGc and multiple Fucose by selecting the best candidate according to iRT difference. Hence, the iRT model shows potential to be used as a filter for database search. With respect to *de novo* searching, higher identification rates are achieved when the spectrum and iRT models are used in the scoring function, also indicating that future research in *de novo* searching could incorporate spectrum and iRT prediction results.

Chapter 6

Conclusion

6.1 Contributions of This Thesis

In Chapter 3, we propose a spectrum prediction model for glycopeptides using a Graphormer-based deep learning model. We define the inputs of the model based on the tree-like structure of glycopeptides. We further define the output of the model to match with the target spectrum, by designing a matrix to record theoretical fragments resulting from up to three cleavages. The model achieves a median cosine similarity of 0.92, which is more than 20% higher than any previous spectrum prediction attempts.

In Chapter 4, we describe an iRT prediction model for glycopeptides, using transfer learning from a peptide iRT prediction model. We discuss the importance of using iRT, feature RT, and conduct ablation studies to optimize the iRT model. Eventually, our glycan structure iRT prediction model achieves a Pearson R correlation of 1.0, which is higher than previous works. We also evaluate with 95 percentile delta and our best-performing model achieves that for 9.24 minutes. We also show that our model can accurately understand both peptide and glycan information by fixing either peptide or glycan.

In Chapter 5, we employ our models in both database search and *de novo* search procedures, and show that our model is applicable in glycopeptide sequencing. We show that our spectrum prediction model improves identification during rescoring. We apply the iRT prediction model on both fission yeast and mouse data to show that it can be used as a filter during database search. Finally, we run a small experiment with a relatively premature *de novo* model to show that when used as a scoring function, our spectrum and iRT prediction models increase identification rates.

6.2 Issues and Direction for Future Research

We discuss in Chapters 3 and 4, that more high-quality glycopeptide data would greatly boost our models' performance. With respect to spectrum prediction, we are in need of more balanced data showing the effects of precursor charge on spectrum intensities, and for iRT prediction, we raise the issue of RT to iRT regression loss, which can be mitigated by higher-quality data with less RT variance. Research on generating a large volume of high-quality mass spectrometry data can be very useful in this regard. On the other hand, data augmentation techniques may be considered, in order to create more balanced data.

In Chapter 5, we uncover the urgent issue of decoy database generation for glycopeptide database search. We argue that a well-crafted decoy database is very difficult to generate and that FDR and rescoring results would be more convincing if one existed. Future research should focus on finding a universal decoy database, or a coherent decoy generation algorithm so that results from different search engines are more comparable.

Throughout this thesis, we only consider the spectrum and iRT prediction for N-glycopeptides and not O-glycopeptides, because N-glycans are more regularized, and therefore more easily and accurately identified by search engines. Thus, there are more high-quality glycan PSM data for our training. Future work may extend our methods to O-glycopeptides, and given the fact that O-glycan structures are less predictable, spectrum and iRT prediction may show significant improvement in identification.

We have also limited our research on DDA data for glycopeptides, while spectrum prediction has been shown to aid in DIA sequencing as well [10]. The high accuracy produced by DIA technology is appealing in glycopeptide identification. When DIA data is more available for glycopeptides, and when the DIA sequencing pipeline is more mature, spectrum prediction shows great promise in improving DIA identification for glycopeptides.

References

- [1] Evelyn Ang, Haley Neustaeter, Vic Spicer, Helene Perreault, and Oleg Krokhin. Retention time prediction for glycopeptides in reversed-phase chromatography for glycoproteomic applications. *Analytical chemistry*, 91(21):13360–13366, 2019.
- [2] Bioinformatics Solutions Inc. Peaks, 2023.
- [3] Robert Boyd and Árpád Somogyi. The mobile proton hypothesis in fragmentation of protonated peptides: a perspective. *Journal of the American Society for Mass Spectrometry*, 21(8):1275–1278, 2010.
- [4] 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.
- [5] Jurgen Cox, Nadin Neuhauser, Annette Michalski, Richard A Scheltema, Jesper V Olsen, and Matthias Mann. Andromeda: a peptide search engine integrated into the maxquant environment. *Journal of proteome research*, 10(4):1794–1805, 2011.
- [6] Vlado Dančák, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of computational biology*, 6(3-4):327–342, 1999.
- [7] Joshua E Elias and Steven P Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods*, 4(3):207–214, 2007.
- [8] 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.

- [9] 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.
- [10] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard De-langhe, Andreas Huhmer, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods*, 16(6):509–518, 2019.
- [11] Rebeca Kawahara, Anastasia Chernykh, Kathirvel Alagesan, Marshall Bern, Weiqian Cao, Robert J Chalkley, Kai Cheng, Matthew S Choo, Nathan Edwards, Radoslav Goldman, et al. Community evaluation of glycoproteomics informatics solutions reveals high-performance search strategies for serum glycopeptide analysis. *Nature methods*, 18(11):1304–1316, 2021.
- [12] Joshua Klein, Luis Carvalho, and Joseph Zaia. Expanding n-glycopeptide identifications by fragmentation prediction and glycome network smoothing. *bioRxiv*, pages 2021–02, 2021.
- [13] Joshua Klein and Joseph Zaia. Glypy: an open source glycoinformatics library. *Journal of proteome research*, 18(9):3532–3537, 2019.
- [14] Joshua Klein and Joseph Zaia. Relative retention time estimation improves n-glycopeptide identifications by lc–ms/ms. *Journal of proteome research*, 19(5):2113–2121, 2020.
- [15] Andy T Kong, Felipe V Leprevost, Dmitry M Avtonomov, Dattatreya Mellacheruvu, and Alexey I Nesvizhskii. Msfragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature methods*, 14(5):513–520, 2017.
- [16] Oleg V Krokhin. Sequence-specific retention calculator. algorithm for peptide retention prediction in ion-pair rp-hplc: application to 300- and 100-Å pore size c18 sorbents. *Analytical chemistry*, 78(22):7785–7795, 2006.
- [17] Kaiyuan Liu, Sujun Li, Lei Wang, Yuzhen Ye, and Haixu Tang. Full-spectrum prediction of peptides tandem mass spectra using deep neural network. *Analytical chemistry*, 92(6):4275–4283, 2020.
- [18] Ming-Qi Liu, Wen-Feng Zeng, Pan Fang, Wei-Qian Cao, Chao Liu, Guo-Quan Yan, Yang Zhang, Chao Peng, Jian-Qiang Wu, Xiao-Jin Zhang, et al. pglyco 2.0 enables

- precision n-glycoproteomics with comprehensive quality control and one-step mass spectrometry for intact glycopeptide identification. *Nature communications*, 8(1):438, 2017.
- [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] Zeping Mao, Ruixue Zhang, Lei Xin, and Ming Li. Mitigating the missing fragmentation problem in de novo peptide sequencing with a two stage graph-based deep learning model. 2023. under review.
- [21] Fred W McLafferty. Tandem mass spectrometry. *Science*, 214(4518):280–287, 1981.
- [22] Jonas Nilsson, Ulla Rüetschi, Adnan Halim, Camilla Hesse, Elisabet Carlsohn, Gunnar Brinkmalm, and Göran Larson. Enrichment of glycopeptides for glycan structure and attachment site identification. *Nature methods*, 6(11):809–811, 2009.
- [23] Chi Soo Park, Minju Kang, Ahyeon Kim, Chulmin Moon, Mirae Kim, Jieun Kim, Subin Yang, Leeseul Jang, Ji Yeon Jang, and Ha Hyung Kim. Fragmentation stability and retention time-shift obtained by lc-ms/ms to distinguish sialylated n-glycan linkage isomers in therapeutic glycoproteins. *Journal of Pharmaceutical Analysis*, 2023.
- [24] 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.
- [25] Salomé S Pinho and Celso A Reis. Glycosylation in cancer: mechanisms and clinical implications. *Nature Reviews Cancer*, 15(9):540–555, 2015.
- [26] Rui Qiao. *Peptide Sequencing with Deep Learning*. PhD thesis, University of Waterloo, 2020.
- [27] Rui Qiao, Ngoc Hieu Tran, Lei Xin, Xin Chen, Ming Li, Baozhen Shan, and Ali Ghodsi. Computationally instrument-resolution-independent de novo peptide sequencing for high-resolution devices. *Nature Machine Intelligence*, 3(5):420–425, 2021.
- [28] Ernesto Rodríguez, Sjoerd TT Schetters, and Yvette van Kooyk. The tumour glycode as a novel immune checkpoint for immunotherapy. *Nature Reviews Immunology*, 18(3):204–211, 2018.

- [29] Shun Saito. False discovery rate analysis for glycopeptide identification. *Electronic Thesis and Dissertation Repository*, 8871, 2022. MA thesis.
- [30] Jiechen Shen, Li Jia, Liuyi Dang, Yuanjie Su, Jie Zhang, Yintai Xu, Bojing Zhu, Zexuan Chen, Jingyu Wu, Rongxia Lan, et al. Strucgcp: de novo structural sequencing of site-specific n-glycan on glycoproteins using a modularization strategy. *Nature Methods*, 18(8):921–929, 2021.
- [31] Weiping Sun, Qianqiu Zhang, Xiyue Zhang, Ngoc Hieu Tran, M Ziaur Rahman, Zheng Chen, Chao Peng, Jun Ma, Ming Li, Lei Xin, et al. Glycopeptide database search and de novo sequencing with peaks glycanfinder enable highly sensitive glycoproteomics. *Nature Communications*, 14(1):4046, 2023.
- [32] Ngoc Hieu Tran, Rui Qiao, Lei Xin, Xin Chen, Chuyi Liu, Xianglilan Zhang, Baozhen Shan, Ali Ghodsi, and Ming Li. Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry. *Nature methods*, 16(1):63–66, 2019.
- [33] 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.
- [34] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.
- [35] Wen-Feng Zeng, Wei-Qian Cao, Ming-Qi Liu, Si-Min He, and Peng-Yuan Yang. Precise, fast and comprehensive analysis of intact glycopeptides and modified glycans with pglyco3. *Nature Methods*, 18(12):1515–1523, 2021.
- [36] Fangfei Zhang, Weigang Ge, Guan Ruan, Xue Cai, and Tiannan Guo. Data-independent acquisition mass spectrometry-based proteomics and software tools: a glimpse in 2020. *Proteomics*, 20(17-18):1900276, 2020.
- [37] Qianqiu Zhang. personal communication, 2022.
- [38] Yaoyang Zhang, Bryan R Fonslow, Bing Shan, Moon-Chang Baek, and John R Yates III. Protein analysis by shotgun/bottom-up proteomics. *Chemical reviews*, 113(4):2343–2394, 2013.

- [39] Zhongqi Zhang and Bhavana Shah. Prediction of collision-induced dissociation spectra of common n-glycopeptides for glycoform identification. *Analytical chemistry*, 82(24):10194–10202, 2010.
- [40] Xie-Xuan Zhou, Wen-Feng Zeng, Hao Chi, Chunjie Luo, Chao Liu, Jianfeng Zhan, Si-Min He, and Zhifei Zhang. pdeep: predicting ms/ms spectra of peptides with deep learning. *Analytical chemistry*, 89(23):12690–12697, 2017.