A Machine-Learning-Based Algorithm for Peptide Feature Detection from Protein Mass Spectrometry Data

by

Xiangyuan Zeng

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, 2021

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

Liquid chromatography with tandem mass spectrometry (LC-MS/MS) has been widely used in proteomics. Two types of data, MS and MS/MS data, are produced in an LC-MS/MS experiment. The MS data contains signal peaks corresponding to the intact peptides in the samples being analyzed. However, research on protein mass spectrometry data has focused more on extracting information from MS/MS data than on MS data. To effectively utilize MS information, we propose a novel software tool, MSTracer, to detect peptide features from MS data. Two machine-learning-combined scoring functions were incorporated in the implementation: one for detecting the peptide features and another for assigning a quality score to each detected peptide feature. The software was compared with several other tools and demonstrated significantly better performance.

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Bin Ma for his continuing guidance, remarks, and engagement through the learning process of this thesis. Without his assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished. Furthermore, I would like to thank the readers of my thesis, Professor Lila Kari and Professor Andrew Doxey for reviewing my work. I am gratefully indebted to them for their very valuable comments on this thesis. I would also like to thank Dr. Shenheng Guan for his assistance in my research with his expertise in the field of mass spectrometry. Moreover, I would like to thank all members in bioinformatics laboratory. I am sincerely grateful for their support and friendship. Finally, I would like to thank my parents, grandparents, and Benson for their encouragement and support through all these years of school.

Dedication

This is dedicated to my grandfather, Qingxi Zeng (1926 - 2019). Your life shall not be forgotten.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Protein is the fundamental functional macromolecule in living organisms and proteomics is the large scale study of proteins. Ever since the term "proteomics" was coined from "protein" and "genomics" in the 1990s [20], new protein analysis techniques have emerged, rapidly evolved, and contributed to large-scale proteomics studies. These techniques are widely used in cutting edge research ranging from fundamental protein research such as systems biology to drug discovery [9], biomarker discovery [8], and disease diagnostics [10]. One of the technologies at the forefront of this field, mass spectrometry (MS), is credited for advancing the development of proteomics on account of its sensitivity, speed, and versatility [1]. This has resulted in the rapid growth of protein MS data within the proteomics community, and the emergence of protein MS data platforms such as ProteomeXchange for standardized proteomics data submission and dissemination worldwide.

In proteomics analysis, proteins are digested by proteases, whereby they are broken down into pieces to produce a large number of peptides. Mass spectrometry accurately determines the mass and characteristics of peptides by measuring the mass-to-charge ratio (m/z) of the peptide ions present. Liquid chromatography (LC) is usually used before MS for peptide separation. Liquid chromatography with tandem mass spectrometry (LC-MS/MS) combines LC with multiple quadrupole mass spectrometers and is widely used to perform bottom-up (or shotgun) proteomics analysis, including protein identification and quantification. The LC column elutes different peptides at different retention times (RTs), and an ion source passes through these peptides. The mass analyzer then performs an MS scan of the coeluting peptide ions periodically, the output of which are the MS1

spectra. The peptide ions of interest are then selected and fragmented, and further MS/MS scans are performed by a second mass analyzer, the output of which are the MS2 spectra. These selected peptide ions are defined as precursor ions. The workflow of LC-MS/MS is presented in Figure 1.1.

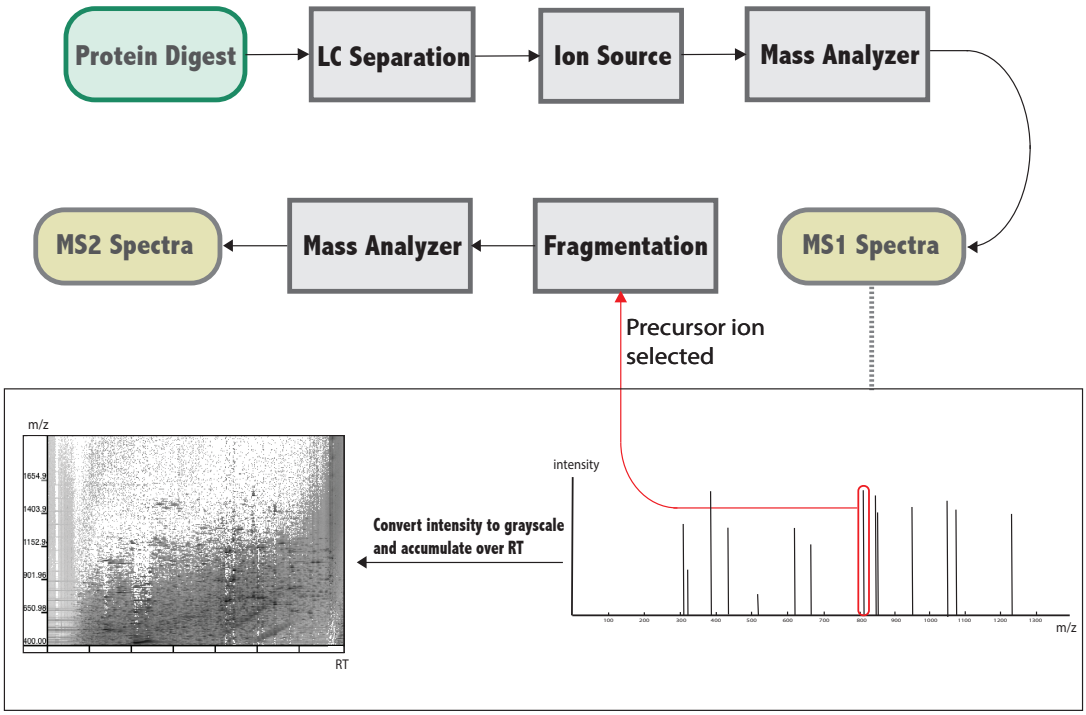


Figure 1.1: Workflow of LC-MS/MS.

The rich primary structural information of peptide precursors contained in MS2 spectra have enabled software tools to be optimized to identify and quantify peptides. This is done either by searching a sequence database (such as Mascot [16], SEQUEST [5], XTandem [3], PEAKS DB [22], MS-GF+ [12], or MaxQuant [2]) or by *de novo* sequencing (such as PEAKS [14],pNovo [21], Novor [13], or PepNovo [6]). However, little attention has been paid to MS1 spectra.

In MS1 spectra, the elution time window of a peptide usually lasts for several seconds. Peptide ions usually appear in several MS scans during the elution time window.

A heatmap of MS1 spectra can be plotted with respect to m/z, RT, and intensity (e.g., Figures 1.1 & 1.2). MS1 spectra can be used to detect peptide features, which offers several advantages for bottom-up proteomics analysis. First, they can accurately determine the charge state, RT, and m/z of the peptide. Such information can be added to the information contained in the MS2 spectra to improve the performance of peptide identification. Second, the peak areas of the peptide features can be used in a label-free quantification analysis to study changes in the quantity of peptides across multiple samples.

In this research, we aim to develop an algorithm that detects peptide features from mass spectrometry data with higher accuracy than the existing software applications. The purpose is to efficiently utilize mass spectrometry data to benefit peptide quantification and identification.

1.2 MS Peptide Detection and MS/MS Peptide Identification

1.2.1 Feasibility and Challenges of MS Peptide Detection

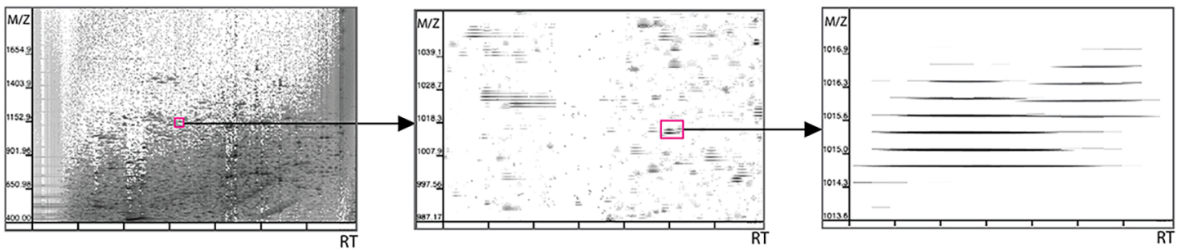


Figure 1.2: Heatmap view of MS scan with trails of peptide precursor ions. (A) The full heatmap view of the MS1 data. (B) An enlarged view of the square area in (A) shows many trail clusters. Each cluster is a peptide feature. (C) An enlarged view of the square area in (B) shows two peptide features that overlap slightly.

Peptide feature can be detected from MS1 spectra by searching for certain characteristics on the map. As the heatmap of MS1 spectra illustrates, peptide ions show parallel trails that are allowed to be detected - these trails are the presentation of isotopes. In nature, the same chemical elements with different masses are defined as isotopes. A monoisotope

is one whose elemental composition contains the most abundant isotopes. Each trail represents the elution profile of an isotope of a peptide over an RT. The trails from a single peptide, which include a monoisotopic ion and corresponding isotope ions, are collectively called a peptide feature and can be detected by mass difference. A peptide mainly contains elements such as carbon (C), hydrogen (H), oxygen (O), nitrogen (N), and sulfur (S), the lightest isotope of which is the most abundant and is called the monoisotope. Thus, the trail with the smallest m/z of the peptide feature is usually the monoisotope.

Peptide feature detection can be non-trivial. When peptide features do not overlap, their detection is rather straightforward. However, when two or more peptide features overlap (e.g., Figure 1.2 (C)) and when the signal-to-noise ratio of the peptide feature is low, detection is particularly challenging for computer software and even for human experts. Existing software applications, such as MaxQuant, Dinosaur, and OpenMS, do not take these factors into account. In addition, these tools do not provide a quality assessment system for the reported peptide features. Overall, existing software tools might be affected by intermingled information on peptide features might affect the performance, and quality score assignment is lacked in existing software tools. We aim to solve the aforementioned problems by proposing a tool that performs better than existing software tools and includes a quality assessment system incorporating machine learning (ML) models.

An ML model automatically learns the underlying rules of a training dataset to predict future data. Recently, ML has been used by the proteomics community for peptide identification analysis with MS. This includes analyzing MS2 data to improve the database search in the Distiller [16] software tool and *de novo* sequencing in the Novor [13] and DeepNovo [19] software applications. Machine learning has also facilitated data-independent acquisition scan extraction in data dependent-independent acquisition by predicting LC-MS/MS properties [7]. Since the proteomics community has accumulated a vast amount of mass spectrometry data, training new ML models is a promising approach for MS data analysis.

1.2.2 MS/MS Peptide Identification

Peptide identification from MS2 spectra is used in training and testing machine learning models in this research. Identifying peptide can be performed by searching MS2 spectra. In a typical LC-MS/MS setting, the precursor ions are selected for fragmentation by data-dependent acquisition (DDA). In DDA mode, high-abundant precursor ions are dynamically selected from the MS scans (Figure 1.3) to produce the MS2 spectra of the corresponding peptides. Then, the MS2 spectra can be used to identify the peptide with software. A peptide spectrum match (PSM) is the output of the peptide identification

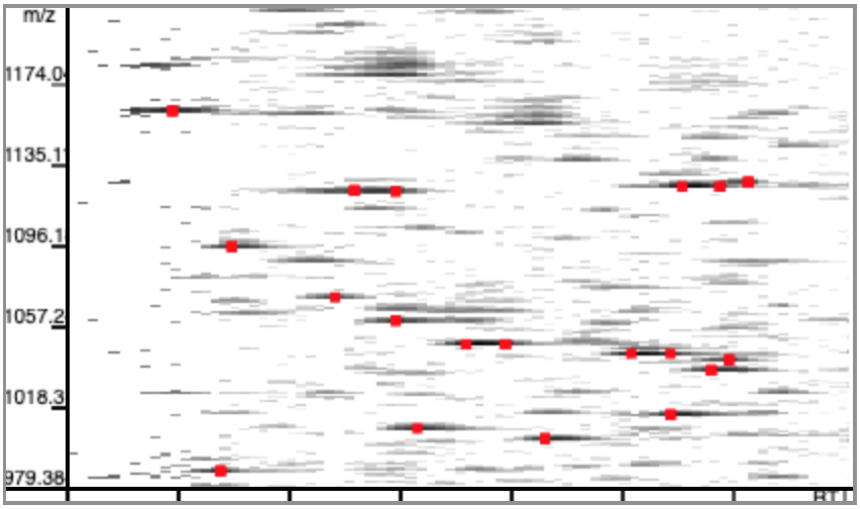


Figure 1.3: The enlarged heatmap view of a DDA dataset. The red points indicate which peptides are selected by instrument for MS/MS events.

tools. If the tools are implemented by database search, a decoy database can be used to determine the false discovery rate (FDR), which is widely used in quality control.

As mentioned, identifying peptides from MS2 spectra involves in two essential parts: to build the training dataset for the machine learning models, and to evaluate the sensitivity of the software tools. The identified peptides allow us to conduct a sensitivity test of our program output, since their precursor ions are a subset of peptide ions in MS1 spectra with high confidence. Similarly, the traits of these MS/MS-identified precursor ions contained in the MS scan can be learned by the ML models to identify high-confidence peptide features from MS1 spectra.

1.3 Protein MS Datasets

We downloaded protein MS datasets for training and testing purposes. As of format, a protein MS dataset is a raw MS file (".raw") that is produced by an MS instrument or a converted file, such as ".mzML", ".mzXML", ".ms1", or ".mgf", derived from a raw file. The datasets were retrieved from ProteomeXchange, a repository platform that provides free access to protein MS datasets uploaded by researchers worldwide. Each dataset comes with at least one raw MS file, and many are with additional search results produced by search engines given by the provider. The datasets were obtained from experiments on a

wide range of species, such as human, mouse, yeast, and *E. coli* with various MS instrument types, such as Q Exactive, Orbitrap Fusion, and LTQ Orbitrap Velos.

1.4 Contributions

Our main contribution in this thesis is a new and open-source software tool, MSTracer, for peptide feature detection from MS data. MSTracer is available at https://github.com/waterlooms/ms-tracer. The manuscript of this research has been submitted to *Journal of Proteome Research* at the time of the submission of this thesis, and is under review and revision. The rest of the thesis is organized as follows:

- In Chapter 2, we review related work, including current peptide detection software tools and the general method for detecting peptide features.
- In Chapter 3, we describe the training and testing datasets, as well as the procedure for data processing.
- In Chapter 4, we introduce MSTracer, an ML-combined software tool that detects peptide features from LC-MS data with enhanced performance and a novel quality ranking system.
- In Chapter 5, we first present ML models and explain their performance in MSTracer. We then compare MSTracer with current software tools, such as OpenMS, Dinosaur, and MaxQuant, in terms of sensitivity and reproducibility. Finally, we analyze the quality score and the joint use of MSTracer with other software.
- In Chapter 6, we summarize the thesis and discuss future work.

Chapter 2

Related Work

In this chapter, we first introduce the common method used in peptide detection. Then, we review the existing software tools for peptide detection. Lastly, we introduce the MS/MS search engines that are used in training and testing the ML models.

2.1 Basic Pipeline of Peptide Detection

The pipeline for detecting peptides from raw data generally includes three steps:

- First, centroid MS spectra. Mass spectrometric data produces a cluster of m/z values for a peptide feature at a certain RT, and the intensities corresponding to these m/z values fit a Gaussian shape [2]. To accurately identify m/z values, we centroid the data, which means selecting the peak of each Gaussian curve. In the image on the left in Figure 2.1(A), the red line represents the peak of the curve and the green dashed lines are the separation points of the Gaussian curves; the image on the right shows that all scans are centroided on the map.
- Second, construct the trails of isotopes. Centroided peaks of the same m/z (usually allowing for a small margin of error) can collectively form a trail on a continuous RT range.
- Third, cluster the trails into isotopic groups based on the m/z difference. A peptide produces signals that are identifiable on the MS scans due to the mass difference between the isotopes (including monoisotope). The mass difference is the mass of a

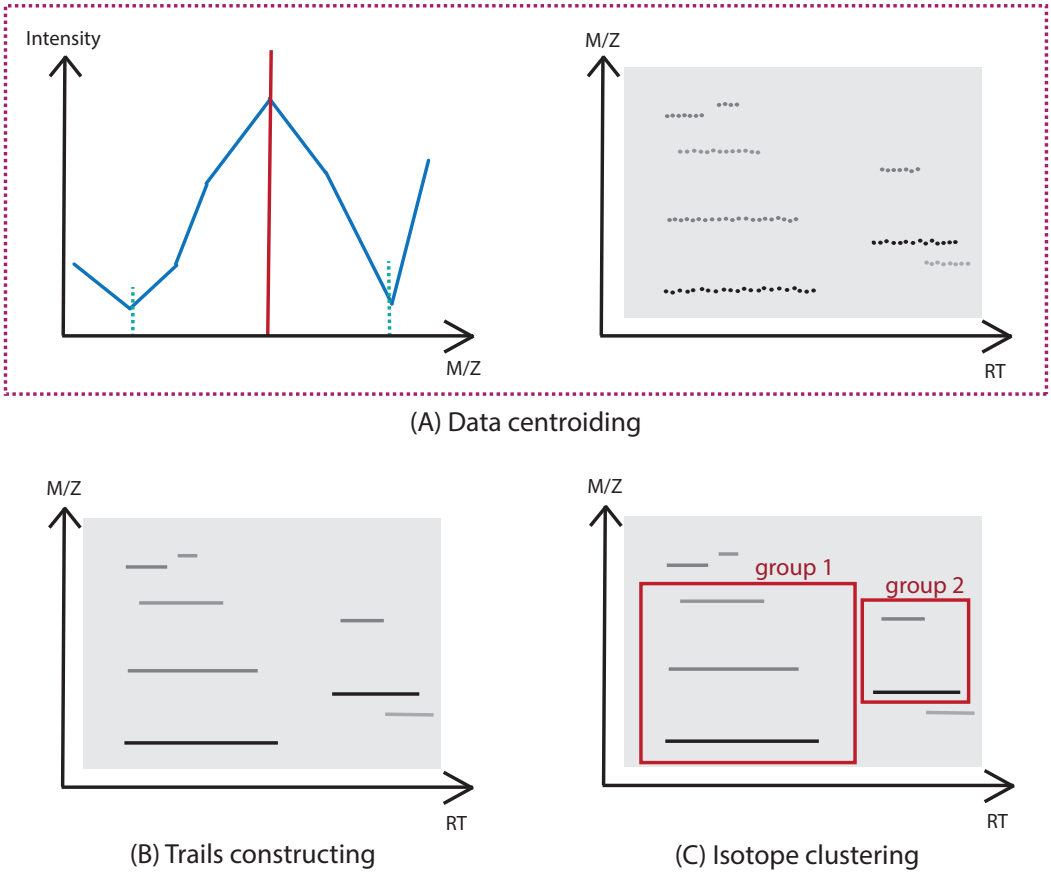


Figure 2.1: Workflow of basic peptide detection.

proton ($1Da$). Therefore, given m/z difference, di , the charge state can be determined by the following equation:

$$z = \frac{1Da}{di} \tag{2.1}$$

This step is referred as de-isotoping in MaxQuant [2] and as hill clustering and isotope deconvolution in Dinosaur [18]

2.2 Existing MS Peptide Detection Tools

We compare our algorithm with three software tools that detect peptide from MS1 spectra, OpenMS (pyOpenMS library v2.5.0), MaxQuant (v1.16.12.0), and Dinosaur (v1.1.3). The performance evaluation and comparison are described in Chapter 5. OpenMS was introduced in 2005, followed by the release of its Python library, pyOpenMS, which performs MS analysis, including peptide detection [17]. In 2008, MaxQuant was published by the Max-Planck-Institute of Biochemistry [2]. It is the most popular tool as it also supports peptide identification from MS/MS scans. More recently, in 2016, Dinosaur was developed not only to detect peaks but also to optimize performance across platforms by automatically auditing and tuning parameters [18]. These software tools and descriptions of their usage can be retrieved from the following resources:

- MaxQuant: https://www.maxquant.org
- Dinosaur: https://github.com/MSGFPlus/msgfplus
- OpenMS: https://pypi.org/project/pyopenms

MaxQuant detects peptides by following the three steps set out in Section 2.1, consuming the raw MS files as input, whereas Dinosaur and OpenMS take in the ".mzXML" or ".mzML" files as input, which should also be centroided. It should be noted that when using MaxQuant, we used all the features reported in "allPeptides.txt" for comparison. Overall, MaxQuant has the full control over the MS data analysis, while Dinosaur and OpenMS take the advantages of other tools to simplify their implementation.

Recently, DeepIso, a peptide detection tool based on convolutional neural network and recurrent neural network, was released in 2019 [23]. The work has contributed to the use of machine learning methods in the proteomics community. Since this software relies on GPU computation and our work computes light-weighted ML models on CPU, we did not compare with DeepIso.

2.3 MS/MS Search Engines

We previously mentioned the use of peptides identified from MS2 spectra to improve the performance of peptide detection. In this research, two popular MS/MS search engines, MS-GF+[12] and Comet[4], were used for peptide identification. MS-GF+ was used to

build the training dataset and for performance evaluation. Comet was used for an additional test to verify whether the performance was impacted by the search engines used to establish the benchmark data.

MS-GF+ is a database search tool for peptide identification by scoring MS/MS spectra. The results are the identified peptides with their information such as m/z, RT, and charge state. In addition to this scoring function, quality control factors such as FDR can be used to select highly confident identified peptides. In the MS-GF+ scoring function, an E-value is employed to evaluate individual PSMs, and the Q-value (labeled as $PepQValue$ in MS-GF+ format) is used as a proxy for FDR [12]. MS-GF+ is available at https://github.com/MSGFPlus/msgfplus/releases. Comet is another database search tool that is widely used for peptide identification and allows for decoy search to control the FDR. The software and a description of its usage can be retrieved from http://comet-ms.sourceforge.net.

Both software tools require the entire protein sequence information for the species of interest to perform peptide search. The sequence data can be retrieved from UniProt (https://www.uniprot.org).

Chapter 3

Data Acquisition and Processing

In this chapter, we first describe the datasets used in the research, then provide the usage and the parameter settings of the tools used in processing the datasets.

3.1 MS Data Acquisition

The most common species uploaded to ProteomeXchange are human, mouse, and yeast. Initially, we chose the human datasets for training and testing the ML models; then, to further test our models on the other species, we chose one mouse dataset and one yeast dataset. We used the following four MS datasets as the benchmark datasets for testing purposes:

- Mouse BV2: Trypsin digest of mouse BV-2 cell lysates [15] (ProteomeXchange: PXD012238; measured with a Q Exactive instrument).
- Yeast S288c: Trypsin digest of yeast S288c cell lysates [11]. (ProteomeXchange: PXD006631; measured with a Q Exactive instrument).
- Human HeLa-1: Trypsin digest of human HeLa cell lysates. (ProteomeXchange: PXD022287; measured with an Orbitrap Fusion instrument).
- Human HeLa-2: A replicate experiment of Human HeLa-1 under the same experimental conditions (instrument; temperature). (ProteomeXchange: PXD022287; measured with an Orbitrap Fusion instrument).

Another MS dataset was used as the training dataset to train the machine learning models in MSTracer:

- Human HeLa Training Data: Trypsin digest of human HeLa cell lysates (ProteomeXchange: PXD022287; measured with an Orbitrap Fusion instrument).

Human HeLa-1, Human HeLa-2, and Human HeLa Training Data are the datasets we have uploaded to ProteomeXchange.

For the MS/MS protein search programs to perform a database search from the MS2 spectra, the protein sequence files of three species, UP000000589 (Mouse), UP000002311 (Yeast), and UP000005640 (Human) were downloaded from UniProt.

3.2 Data Processing

The pipeline for processing the ".raw" file using software tools is illustrated in Figure 3.1. MSConvert, a general format conversion tool developed by ProteoWizard, is used to convert the ".raw" files to the ".mzML" or ".mzXML" format. Both MS spectra and MS/MS spectra are centroided during the conversion. The following command line is used:

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

The input file format of MSTracer, Dinosaur, and OpenMS is ".mzML" or ".mzXML" files; whereas the input of MaxQuant and MS-GF+ is ".raw" files.

The parameters for the MS-GF+ search are as follows: enzyme is set as trypsin; carbamidomethylating on Cysteine is used as fixed post translational modification; precursor mass tolerance is set to 5 ppm; and isotope error range is set to $[0, 1]$. The following command line is used for conversion:

```
[1]  java -Xmx2000M -jar MSGFPlus.jar -s filename.mzML -t 5ppm
-ti 0,1 -tda 1 -d proteome.fasta
[2]  java -Xmx2000M -cp MSGFPlus.jar edu.ucsd.msjava.ui.MzIDToTsv
-I filename.mzid -o filename.tsv
```

From the MS-GF+ output, the m/z, RT, and charge states of the identified PSMs with $PepQValue \leq 0.01$ (FDR≤ 0.01) are used as a benchmark to compare the performances of different feature detection software tools.

In the Comet search, parameters are modified in the file "comet.param". Precursor mass tolerance is 10ppm; decoy search is set to turn on; isotope error range is set as $[0, 1]$; and other parameters remain as default. The command line is:

```
./comet.exe filename.mzML
```

In the Comet output, the PSMs at FDR ≤ 0.01 are retained. The FDR is calculated by ranking the search results according to their E-values, as reported by Comet.

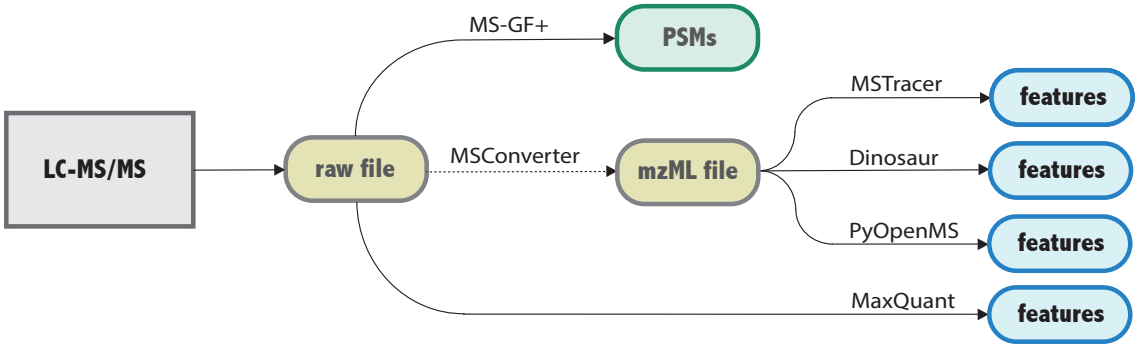


Figure 3.1: Pipeline of MS data processing.

Chapter 4

MSTracer

In this chapter, we first introduce the overall pipeline of the algorithm design. Then, we explicitly explain two scoring functions utilized as the variables in the Support Vector Regression (SVR) model and the Neural Networks (NN) model. Finally, we describe in details of the application and purpose of SVR and NN models in our algorithm.

4.1 Peptide Feature Detection

We discussed the general method for detecting peptide features in Section 2.1. Our program, MSTracer, implements the second and the third step of the general method (trails detection, isotope clustering), along with the application of ML models, SVR and NN. Since our input data has already been centroided by MSConvert, we omit the first step (data centroiding). In this section, we will describe the implementation of trails detection and isotope clustering in MSTracer; in the next section, we will describe the building and usage of the ML models.

The first step of MSTracer is to detect the trails (show in Figure 4.1 (A)). This is realized by locating local maxima from the MS data. A trail consists of the signal peaks of the same isotopic precursor ion that appear persistently at the same m/z (allowing for a mass error tolerance) for a period of time. The signal intensity over time on a trail can be represented as the function $h(t)$. By locating the most intensive peak signal on each trail, a precursor ion (either a monoisotope or an isotope) can be distinguished with confidence. Then, we define the local maximum: A trail with intensity function $h(t)$ is said to reach the local maximum at retention time t if there is a time window $[t_1, t_2]$ such that

(1) $t_1 < t < t_2$

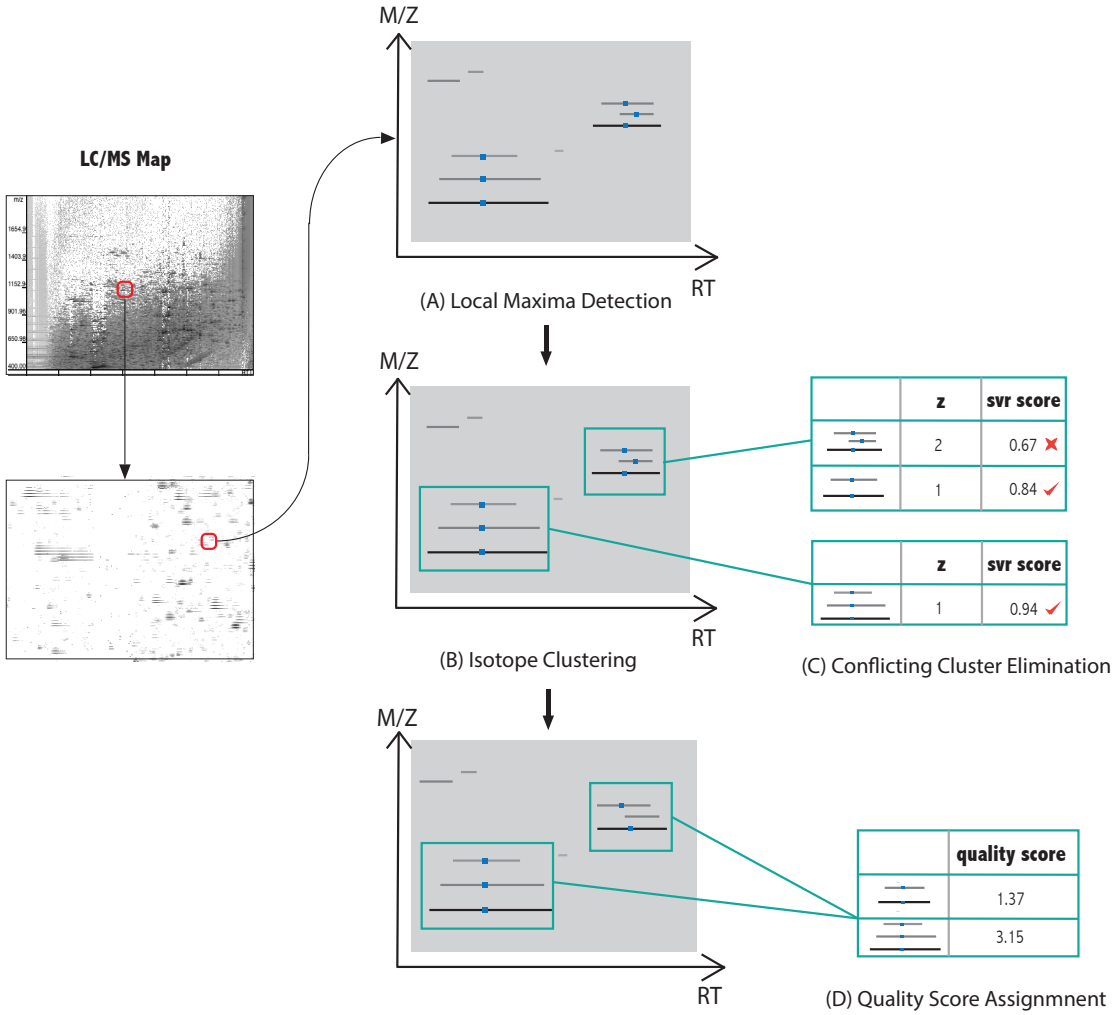


Figure 4.1: Workflow of MSTracer: (A) detection of local maxima on trails; (B) grouping trails into isotope groups by m/z difference; (C) eliminating conflicting isotope groups by the SVR score; and (D) the assignment of the final NN-based quality score.

(2) $h(t) > h(t^{'})$ for $t_1 < t' < t_2$, and
(3) $h(t_1) < \frac{h(t)}{2}$ and $h(t_2) < \frac{h(t)}{2}$.

However, trails from different precursors with the similar m/z and RT may overlap (see Figure 1.2 (C)). We strive to distinguish overlapping trails using the ML method described in the third step.

The second step is to cluster the trails (represented by local maxima) into isotope groups (shown in 4.1 (B)). We define an isotope group as a group of isotopes formed by clustering all the isotopes (represented by local maxima) for a monoisotope. A sequence of local maxima forms an isotope group if they have similar RT and the m/z of every pair of adjacent local maxima differ by $\frac{m(proton)}{k}$ for an integer k. Here, k is the charge state of the candidate, and $m(proton)$ is the atomic mass of a proton. We start searching for isotope groups from $k = 1, 2, ..., k_{hi}$, where k_{hi} is the largest charge state appearing in the MS/MS scans. For efficiency, we allow the user to set an upper bound for k_{hi}. It should be noted that the local maxima can only be clustered once for the same charge state; however, they can be repetitively clustered for a different charge state. Therefore, each local maximum may belong to multiple isotope groups with different charge states and/or monoisotopic mass. A set of isotope groups that share the same local maximum are called a conflicting set of isotope groups.

The third step is to eliminate conflicting isotope groups (shown in Figure 4.1 (C)). Isotope groups are candidates for peptide features. However, different isotope groups may conflict with each other. We have to determine the isotope group that the trail is most likely to belong to and remove the group in which the trail is falsely clustered. For example, the local maxima in each isotope group with charge $2k$ contain a subset of local maxima that form a charge-k isotope group. Allowing both charge states to exist will substantially increase the false positives. Therefore, a scoring function is used to remove the inferior isotope groups within a set of conflicting ones. After this filtration, the remaining isotope groups become the peptide features. The scoring function used in this step is trained with a support vector regression (SVR) model, described in Section 4.3.

The SVR scores are optimized to compare overlapping isotope groups. However, this may not be optimal for comparing peptide features that do not overlap. Therefore, in the fourth step, another scoring function is used to assess the quality of each peptide feature (shown in Figure 4.1 (D)). The scoring function is trained with a neural network (NN) model and is described in Section 4.4.

Finally, features with very close m/z and RT values and the same k are clustered to remove redundancies. In each clustering operation, only the feature with the highest NN quality score is maintained. The remaining peptide features and their quality scores are output as the final results.

4.2 Scoring Function

Two scoring functions, coelution coefficient and isotope shape coefficient are described in this section. These scoring functions take an essential part in the SVR and NN models, as both can represent important characteristics of peptide features.

4.2.1 Coelution Coefficient

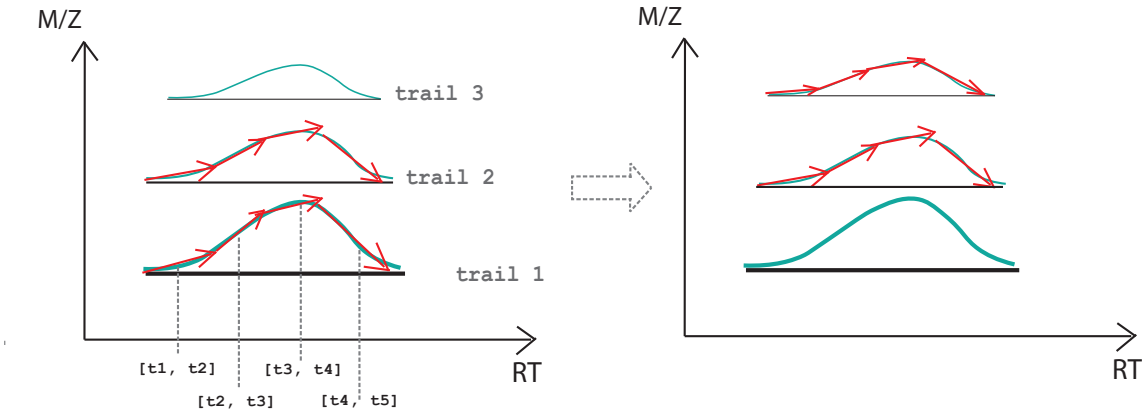


Figure 4.2: Demonstration of calculating coelution coefficient. The blue curves represent the intensity of the trails. The cosine similarity values of the adjacent trails (e.g., trail1 and trail2, trail2 and trail3) with respect to intensity should be calculated. Then, calculate the average of the sum of the cosine similarity values.

The coelution coefficient measures how similar the intensities of the trails in one isotope group change over RT. Different isotopes of the same peptide should coelute; therefore, their trails should correlate well. The coelution coefficient of trails is measured as follows: Given two trails $h(t)$ and $g(t)$, their coelution coefficient in time interval $[t_a, t_b]$ is defined as the cosine similarity of the two trails:

$$coelution(h, g) = \frac{\sum_{t=t_a}^{t_b} h(t)g(t)}{\sqrt{\sum_{t=t_a}^{t_b} (h(t))^2}\sqrt{\sum_{t=t_a}^{t_b} (g(t))^2}} \tag{4.1}$$

The coefficient for an isotope group with n local maxima is defined as:

$$\frac{1}{n-1}\sum_{i=1}^{n-1} coelution(h_i, h_{i+1}) \tag{4.2}$$

Here, h_i is the intensity function of the i-th isotopic trail in the isotope group.

Figure 4.2 shows an example of a coelution coefficient calculation. The time interval is composed of four sub-intervals with $t_a = t_1$, $t_b = t_5$. We first calculate $coelution(trail1, trail2)$ and $coelution(trail2, trail3)$ (Equation 4.1) respectively. Here $trail(t)$ is the intensity of the trail at time t. Then we calculate the coefficient using $\frac{1}{2}(coelution(trail1, trail2) + coelution(trail2, trail3))$ (Equation 4.2).

4.2.2 Isotope Shape Coefficient

The isotope shape coefficient represents the similarity of the theoretical isotope distribution and the isotope distribution from real data. If we take the sum of the intensities of a single isotope trail over a given RT for all the isotopes in a group, the intensity sums will form the isotope shape, or the isotopic distribution. The more the isotopic distribution resembles the theoretical isotope distribution, the higher the score assigned to the peptide feature. The theoretical isotopic distribution at a given mass value is calculated by using the average composition of elements in peptides. Similar to the coelution coefficient, the isotope shape coefficient is defined as the cosine similarity between its isotopic distribution, d, and its corresponding theoretical isotopic distribution, e. Given the time interval $[t_1, t_2]$, the isotopic distribution of an isotope group is:

$$d_i = \sum_{t=t_1}^{t_2} h_i \tag{4.3}$$

Here, h_i is the intensity function. Then, the isotope shape coefficient with n local maxima is defined as:

$$distribution(d, e) = \frac{\sum_{i=1}^{n} d_i e_i}{\sqrt{\sum_{i=1}^{n} d_i^2}\sqrt{\sum_{i=1}^{n} e_i^2}} \tag{4.4}$$

Here, e_i is the theoretical intensity sum of the i-th isotopic trail.

The theoretical isotopic distribution at a given mass value is calculated using the average composition of C, H, O, N, and S elements in peptides without using the peptide sequence information.

4.3 Support Vector Machine-Based Conflicting Isotope Groups Removal

As described in Step 3, we trained an SVR model to eliminate conflicting isotope groups since a local maximum can be clustered into multiple isotope groups of different charge states. We trained an SVR model to predict which isotope group most likely contains the local maxima. Each isotope group has one SVR score; if one local maximum appears in multiple isotope groups, only the isotope group with higher SVR score keeps the local maximum. The other isotope groups are discarded if they no longer contain more than one local maximum. Each remaining isotope group is a peptide feature.

The training dataset for the SVR model was obtained from Human HeLa Training Data (described in Section 3.1) according to the following procedure: First, the isotope groups obtained in Step 3 of the algorithm were compared to the PSMs identified by MS-GF+. Specifically, among a set of isotope groups that share a trail (represented by local maxima), the group that matches an identified PSM with respect to m/z, RT, and charge state is labeled 1. The unmatched ones are labeled 0. We obtained 58996 negative samples and 2587 positive samples. To balance the training data, we selected all the positive samples and randomly selected 2587 negative samples. The Python module sklearn v0.22.2.post1 was used to implement the model. The trained SVR model can then be used in a new dataset to calculate a score between 0 and 1 for each isotope group. This score is used to select the top-scoring isotope group from each conflicting set of isotope groups.

The SVR model was trained to predict the label by considering the following variables:

* the charge state.
* the number of local maxima.
* the coelution coefficient.
* the isotope shape coefficient.

We also considered other variables, such as the charge state and the intensity sum of the trails. Several models were trained with different combinations of these variables. Finally, the above variables were determined to build the final model based on its performance among all models. The variables used in each model and the test results are described in Section 5.1.1.

4.4 Neural Network-Based Quality Score

As described in Step 4, an NN model was trained to assign a quality score to each detected feature. The network was trained using the Human HeLa Training Data. The positive samples are the features that match the identified PSMs and are labeled 1; the negative samples are those that do not match and are labeled 0. To implement the NN model, we used Tensorflow (v1.14.0). We considered the following variables to build the NN model:

* the number of local maxima.
* the coelution coefficient.
* the isotope shape coefficient.
* the ratio between the trail intensity and the total peak intensity in the surrounding area. Suppose the trails can be fit into a rectangle area $[t_1, t_2] \times [m_1, m_2]$. Here $[t_1, t_2]$ is the RT interval and $[m_1, m_2]$ is the m/z interval. The surrounding area refers to the rectangle $[t_1, t_2] \times [m_1 - 1.5, m_2 + 1.5]$.

Following the same procedure as that for the SVR model, for the NN model, we tested different combinations of variables, including the charge state and the intensity sum of the trails. We also adjusted the NN construction in the model with respect to the number of layers, the number of nodes in each layer, and the number of epochs to achieve the best performance. The final model has three hidden layers, each of which has 100 nodes, and the layers are fully connected. The model was trained with 100 epochs. The results for each tested model are presented in Section 5.1.1.

The value predicted by the NN model is a probability indicating the likelihood of the feature corresponding to a peptide identifiable by MS/MS. This probability p is converted to a quality score using the following formula:

$$score = \log \frac{p}{1-p}.$$

Chapter 5

Evaluation and Discussion

In this chapter, we evaluate the performance of MSTracer as well as MaxQuant, Dinosaur, and OpenMS by conducting the sensitivity test, the quality score analysis, and the reproducibility test. We also illustrate the combine use of multiple software tools and the runtime. Finally, we discuss the results from the performance evaluation.

5.1 Performance Evaluation

In this section, we first describes the selection of the best-performed variables used in the SVR and NN models, and the sensitivity of MSTracer compared to the other software tools. Next, we analyze the distribution of the quality score on the testing datasets and introduce a cut-off score for high- and low-quality peptide features. Then, we illustrate the combine use of multiple software tools and its advantage in improving the quality of peptide features, where Venn diagrams are presented to summarize the results. Lastly, we demonstrate a reproducibility test on replicate datasets and report the runtime of all software tools on the datasets.

Four test datasets, Mouse BV2, Yeast S288c, Human HeLa-1, and Human HeLa-2, were used for assessing the performance. For each MS file, peptide features were detected from the MS scans using four software tools: MaxQuant, Dinosaur, OpenMS, and MSTracer.

5.1.1 Sensitivity

The sensitivity test measures how well an algorithm detects the features of the peptides identifiable by a database search. For a given peptide feature detection result, the sensitivity is defined as the percentage of PSMs that match one of the detected peptide features. The detected peptide features are compared to the PSMs identified from the MS/MS spectra by MS-GF+ and Comet, respectively. The precursor of a PSM and a detected feature are matched if the conditions below are satisfied:

* Their m/z match is within a ± 10 ppm error.
* Their RT match is within a ± 0.5 min error.
* Their charge states match.

To optimize the SVR and NN models, we constructed the model with several combinations of variables and tested the sensitivity. The model with the best performance was selected as the final model. The variables we considered are listed as below:

(1) The charge state
(2) The number of local maxima
(3) The coelution coefficient
(4) The isotope shape coefficient
(5) The scaled intensity sum of the trails, calculated as the real intensity sum divided by window size and the $100th$ largest intensity sum in the clusters.
(6) the ratio between the trail intensity and the total peak intensity in the surrounding area. Suppose the trails can be fit into a rectangular area $[t_1, t_2] \times [m_1, m_2]$. Here $[t_1, t_2]$ is the RT interval and $[m_1, m_2]$ is the m/z interval. The surrounding area refers to the rectangle $[t_1, t_2] \times [m_1 - 1.5, m_2 + 1.5]$.

An extra tree classifier was applied to decide the combinations of the variables. The classifier fits randomized decision trees and estimates feature importance. The importance of each variable is represented by weights, as shown in Table 5.1.

We selected the groups of variables according to the weights of the variables. Initially, we included all variables; next, we eliminated the least significant variables in the decision trees one at a time. The variable combinations for the SVR model were:

a. (1) (2) (3) (4) (5) (6).
b. (1) (2) (3) (4) (6).

c. (1) (2) (3) (4).
d. (1) (2) (4).
e. (2) (3) (4) (6).
f. (2) (3) (4).

Table 5.2 shows that SVR models *b*, *c*, and *d* significantly outperformed *a*, *e*, and *f* across the four datasets. Their sensitivity was higher even though they produced fewer results. We choose model *c* as the final model.

	(1)	(2)	(3)	(4)	(5)	(6)
Weight	0.504	0.141	0.088	0.140	0.061	0.066

Table 5.1: Weights of variables.

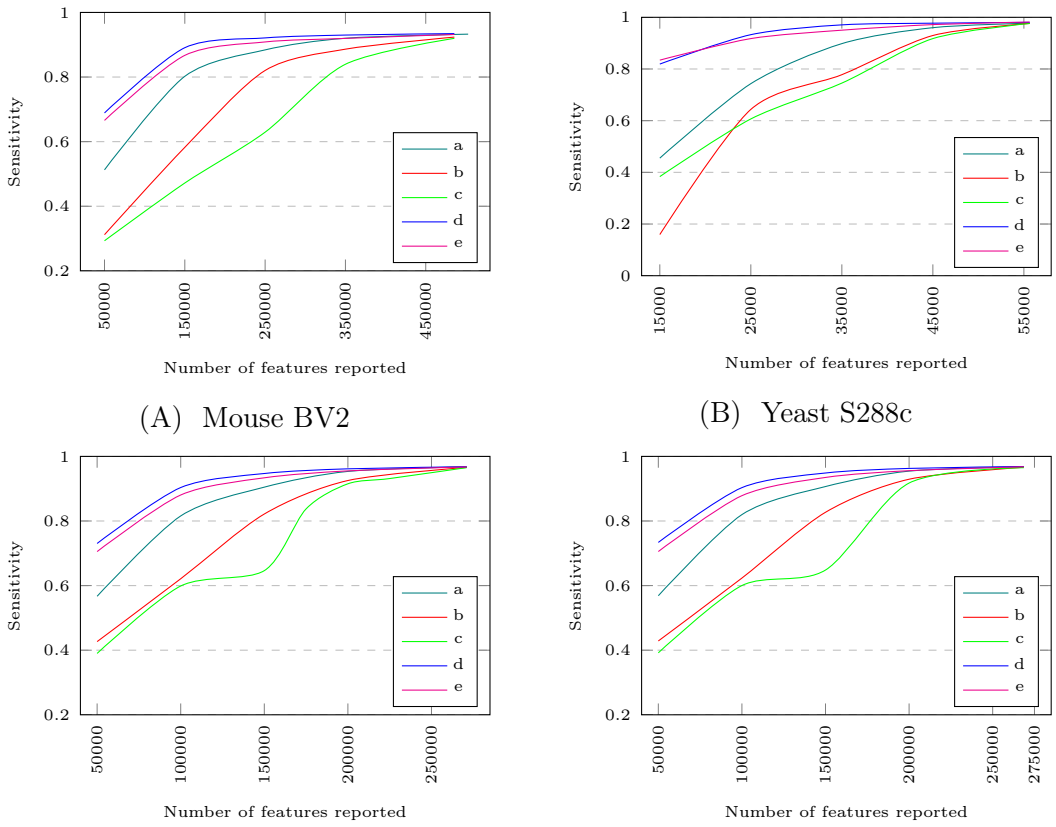
Sensitivity	a	b	c	d	e	f
Mouse BV2	61.29%	89.88%	91.01%	91.09%	88.36%	88.52%
	(1233309)	(1223827)	(1226329)	(1214624)	(1082062)	(108077)
Yeast S288c	71.93%	98.26%	98.27%	98.05%	94.69%	94.91%
	(110049)	(109278)	(109238)	(109227)	(105957)	(105939)
Human HeLa-1	76.55%	96.63%	97.19%	97.31%	95.76%	95.81%
	(481783)	(474695)	(474105)	(472708)	(430391)	(429677)
Human HeLa-2	76.26%	96.73%	97.19%	97.26%	95.74%	95.81%
	(463809)	(457607)	(457297)	(455728)	(413544)	(412949)

Table 5.2: The sensitivity of model *a*-*f* for the four datasets. Numbers in the brackets are the number of peptide features after eliminating conflicting isotope clusters.

Next, we built the NN models based on the same variables. We eliminated variable (5) since it showed a notable negative effect on the model. The variable combinations were:

a. (1) (2) (3) (4) (6).
b. (1) (2) (3) (4).
c. (1) (2) (4).
d. (2) (3) (4) (6).
e. (2) (3) (4).

We ranked the peptide features in descending order by the quality score that the NN model assigned and then tested the sensitivity of the top N peptide features. The results, shown in Figure 5.1, indicated that model d outperformed the other models on all datasets.



(A) Mouse BV2

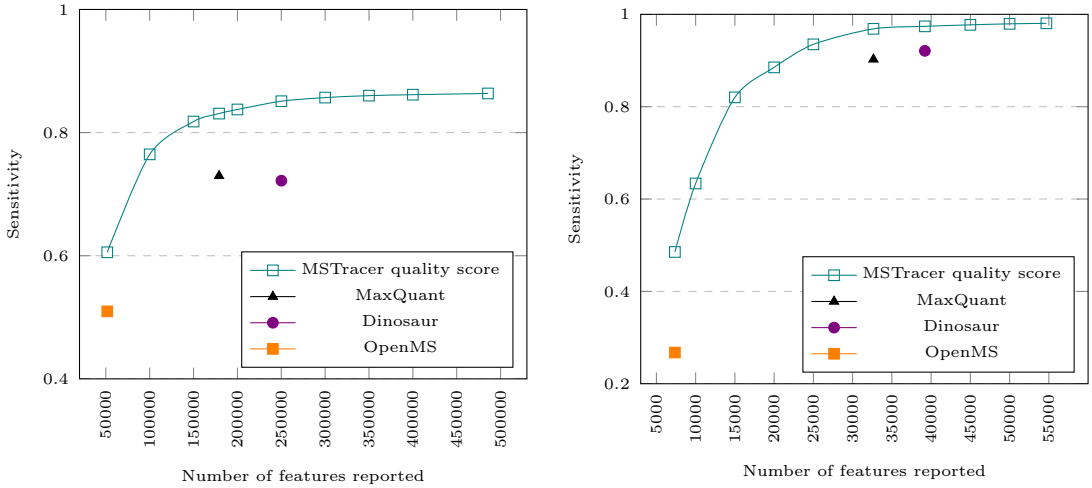
(B) Yeast S288c

(C) Human HeLa-1

(D) Human HeLa-2

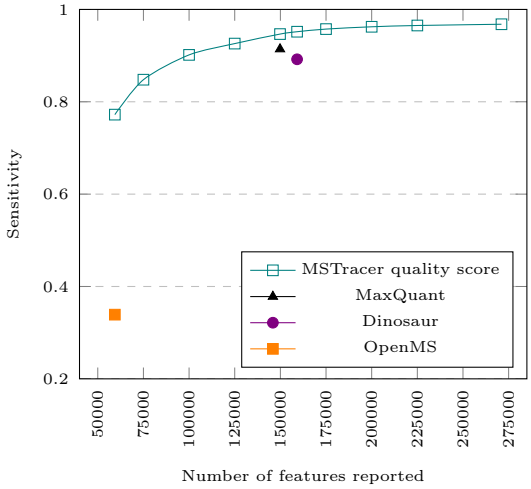
Figure 5.1: Sensitivity test on NN model a, b, c, d, and e for the four datasets. The x-axis is the top N peptide features reported by the model. The y-axis (sensitivity) is the percentage of the benchmark peptides that are included in the reported features.

Finally, SVR model c and NN model d were selected for our algorithm. Figure 5.2 shows the sensitivity of MSTracer and other software tools with respect to the number of reported features with PSMs identified by MS-GF+. When a similar number of features are reported, MSTracer is able to identify more benchmark features (features matching the identified PSMs) than any other tool. The PSMs in Figure 5.3 were identified using Comet, which show similar results as MS-GF+. This indicates that the relative performances of

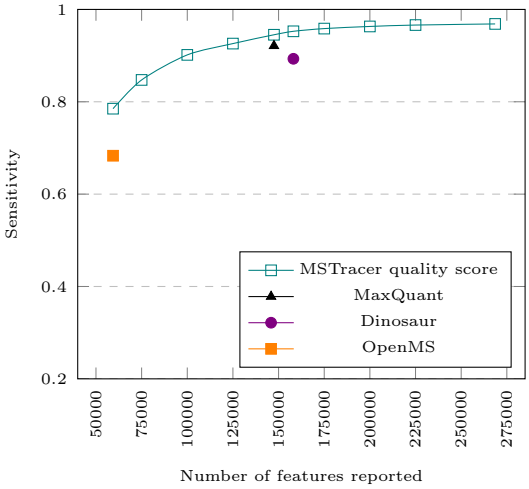


(A) Mouse BV2

(B) Yeast S288c

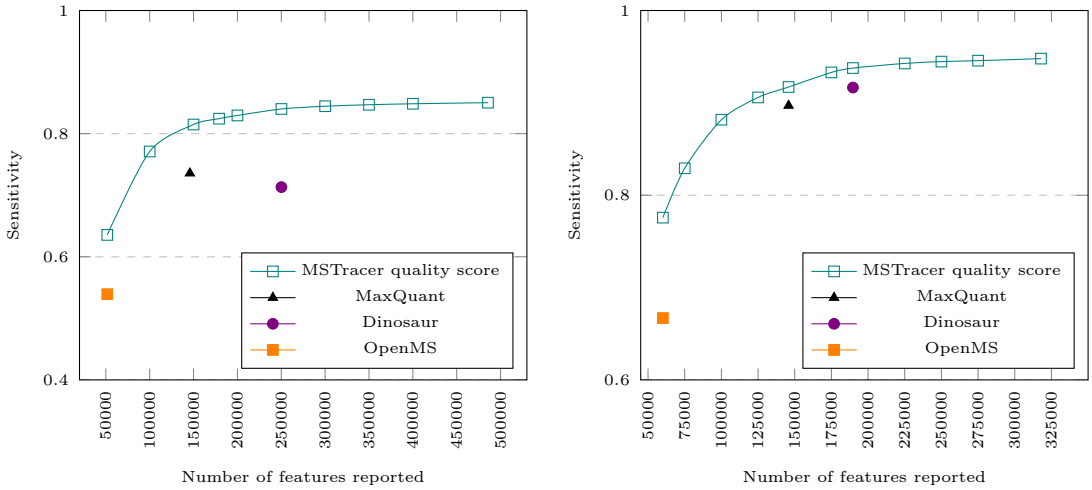


(C) Human HeLa-1



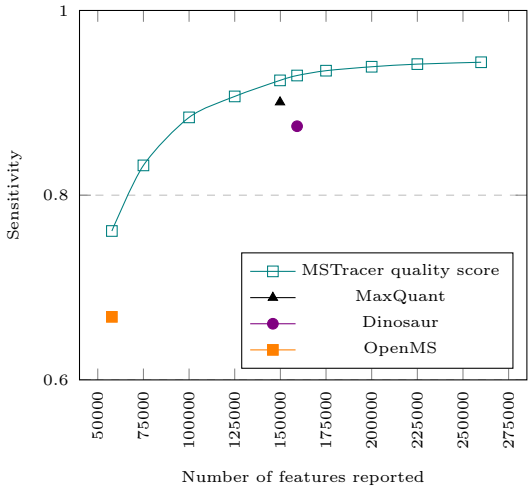
(D) Human HeLa-2

Figure 5.2: The sensitivity of MSTracer, MaxQuant, Dinosaur and OpenMS on the four datasets, where ground truth PSMs are matched with MS-GF+. The x-axis is the number of features reported by each software. The y-axis (the sensitivity) is the percentage of the benchmark peptides that are included in the reported features. By adjusting the score threshold, MSTracer can report different number of features. Therefore, its sensitivity is a curve, whereas the other software reports a fixed number of features.

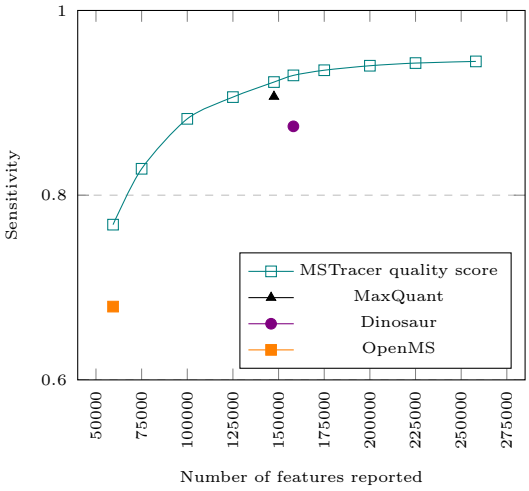


(A) Mouse BV2

(B) Yeast S288c



(C) Human HeLa-1



(D) Human HeLa-2

Figure 5.3: The sensitivity of MSTracer, MaxQuant, Dinosaur and OpenMS on the four datasets, where ground truth PSMs are matched with Comet at FDR ≤ 0.01

different feature detection programs are not greatly impacted by the search engines used to establish the benchmark data.

5.1.2 Quality Score Analysis

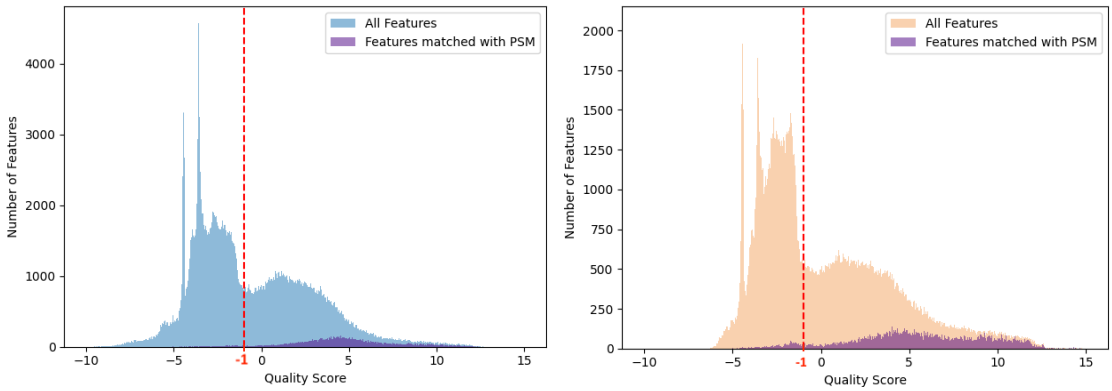
We designed the quality score to build a ranking system; we then identified a breaking point which separate "good" and "bad" peptide features. Figure 5.4 shows the quality score distribution of the features detected by MSTracer for the four test datasets, Mouse BV2, Yeast S288c, Human HeLa-1, and Human HeLa-2. The scores for PSMs are also plotted. The figures show that the quality score distribution is approximately an aggregation of two Gaussian distributions. Moreover, most of the features that match an identified PSM have a quality score above -1. This indicates that the score -1 can be used to separate high- and low-quality features and that it can be used as an empirical cut-off score to filter the detected features.

5.1.3 Intersection of the Results of MSTracer and Other Software

Prior to our research, peptide features are not assigned with scores by a ranking system. To generate high-quality peptide features, multiple software tools by taking their intersection. We wanted to investigate the use of MSTracer combined with other tools to see 1) how well the quality separating system in MSTracer works with the other software tools and 2) how combining software might affect the results. Figure 5.5 presents Venn diagrams showing the features detected by the three best-performing tools: MSTracer, Dinosaur, and MaxQuant. To obtain high-quality peptide features, a score threshold of -1 (found in Section 5.1.2) is used to filter the results from MSTracer.

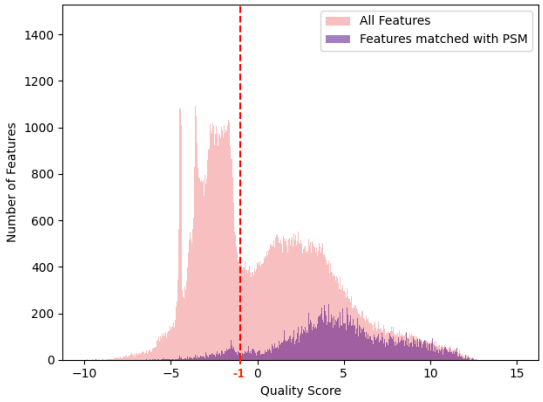
In the Venn diagrams, the intersections of the results from the three tools are illustrated. The parentheses indicated on the diagrams include a number and a percentage. The number is how many features from that intersection match the PSMs, and the percentage is simply the number divided by the number of all features from that intersection.

The diagrams indicate that the number of the features matched with PSMs identified by all three tools is higher than those who are only detected by two tools or less. For example, in the Mouse BV2 dataset, $26.00\%(17,348/66,717)$ of features detected by all three tools were benchmark features. In contrast, only 5.55%, 0.09%, and 0.88% of the features detected by only MSTracer, Dinosaur, and MaxQuant, respectively, were benchmark features.

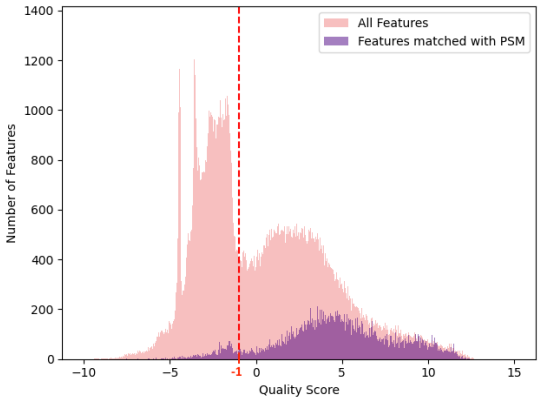


(A) Mouse BV2

(B) Yeast S288c

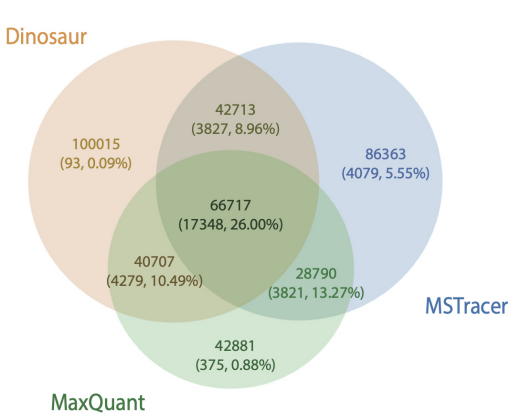


(C) Human HeLa-1

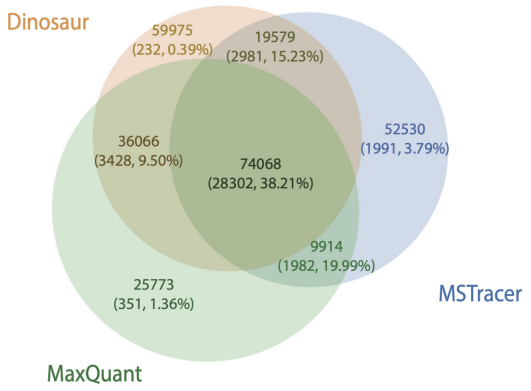


(D) Human HeLa-2

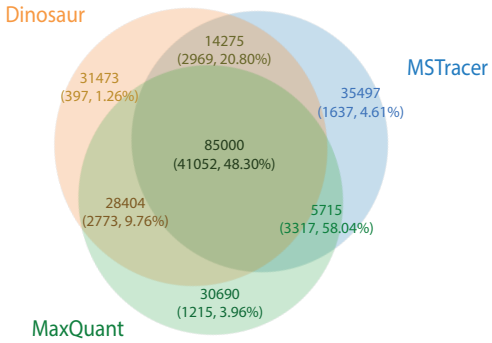
Figure 5.4: The quality score distribution of the features detected by MSTracer and of those that also matched an identified PSM for the four datasets. The dashed vertical lines illustrate a quality score, -1.



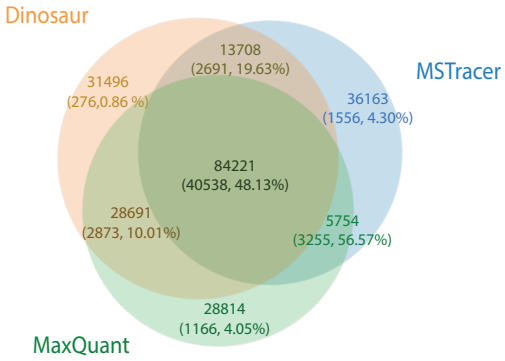
(A) Mouse BV2



(B) Yeast S288c



(C) Human HeLa-1



(D) Human HeLa-1

Figure 5.5: Venn diagrams showing the features detected by MSTracer, MaxQuant, and Dinosaur for the four datasets. The numbers in parentheses are the numbers of benchmark features and their corresponding percentages in the features reported.

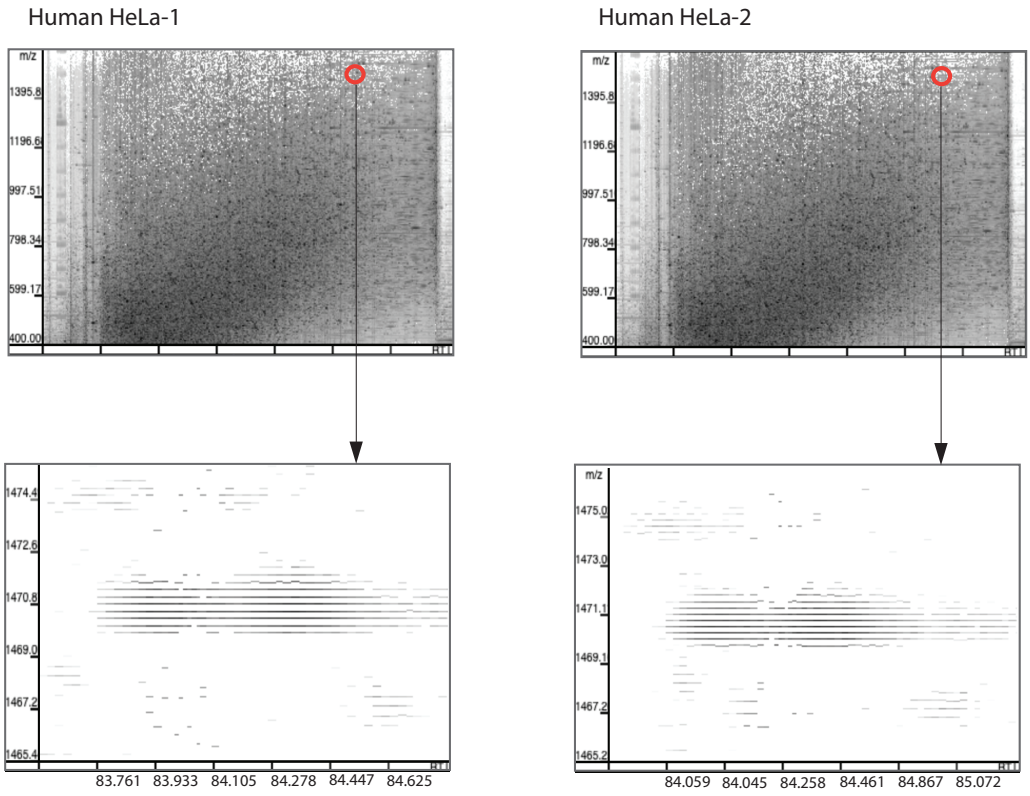


Figure 5.6: Heatmap view of Human HeLa-1 and Human HeLa-2. The bottom images show the patterns of identical peptide features from the two datasets.

We also found that combining MSTracer with other software tended to increase the percentage of benchmark peptides. For example, the datasets Yeast S228c, Dinosaur+MSTracer (15.23%) and MaxQuant+MSTracer (19.99%) reported a higher percentage of benchmark peptides than MaxQuant+Dinosaur (9.50%). The Venn Diagram for Human HeLa-1 and Human HeLa-2 show similar results.

5.1.4 Reproducibility

The reproducibility test evaluates whether the program is able to reproduce the same result on the datasets from replicate experiments. Human HeLa-1 and Human HeLa-2 are identical samples processed twice under the same protocol by mass spectrometer. Their

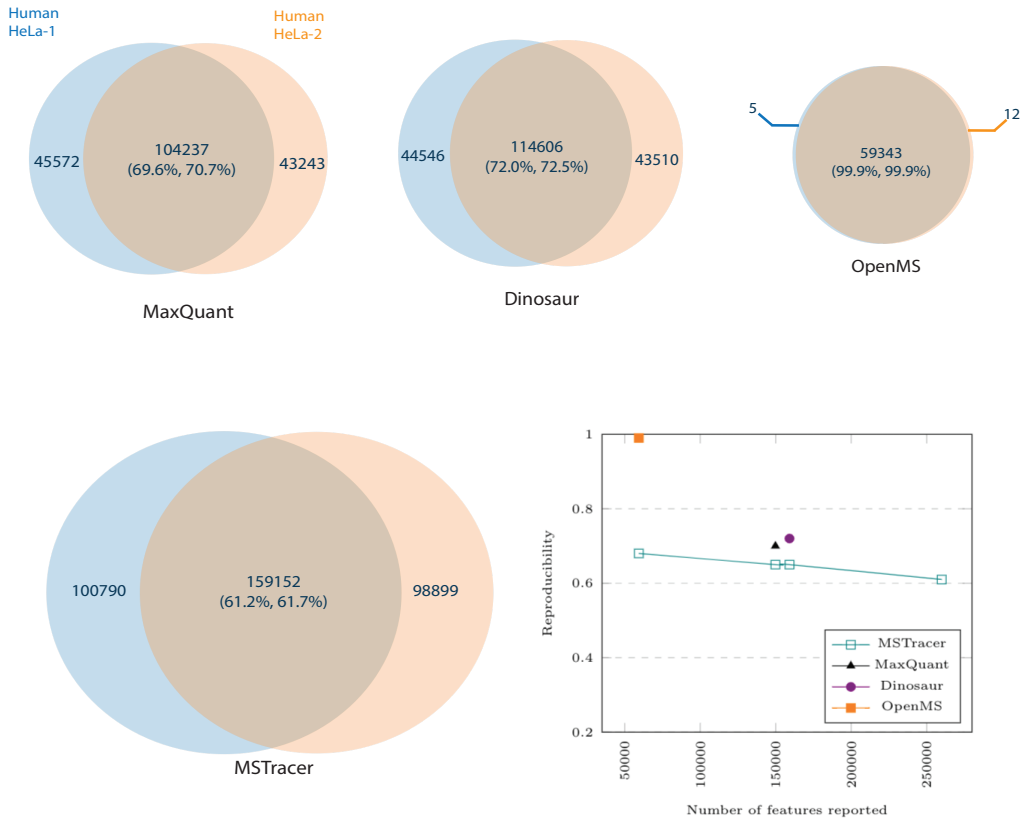


Figure 5.7: Venn diagrams and a plot demonstrating the reproducibility test results on Human HeLa-1 (blue in Venn diagram) and Human HeLa-2 (orange in Venn diagram) for MaxQuant, Dinosaur, OpenMS, and MSTracer. The percentage in the parentheses represents proportion of intersecting peptide features for Human HeLa-1 and Human HeLa-2, respectively. In the results of MSTracer, the Venn diagram illustrates the result of all reported peptide features; the plot illustrates the results of the reported peptide features.

MS scans should show similar patterns if the MS experiment is fully reproducible (shown in Figure 5.6) and the detected features from the two datasets are expected to have a large intersection. However, we know that LC-MS/MS is not fully reproducible. Therefore, the irreproducibility comes from both the software and the nature of the data. The images at the bottom of Figure 5.6 show that identical peptide features in the two datasets have the same m/z value, whereas the RT values may differ. Therefore, we allow higher tolerance in the RT matching condition than in the PSM matching condition. We consider that two features to match if

* their m/z values match within a ± 10 ppm error,
* their RT values match within a ± 1 min error, and
* their charge states match.

The reproducibility test result is shown in Figure 5.7. In the figure, the percentage indicates the proportion of intersecting features for the two datasets. For example, for Human HeLa-1, the proportion intersecting with Human HeLa-2 detected by MaxQuant is $104237/(104237 + 45572) = 69.6\%$. The reproducibility of MSTracer for features with low and high thresholds ranges from around 61% to 68%. OpenMS produces almost identical outputs on the two replicate datasets; however, the number of output is remarkably lower than that of the other software. Dinosaur reproduces the results better than MaxQuant and MSTracer. Above all, the reproducibility of MSTracer, MaxQuant, and Dinosaur is no higher than 75%.

5.1.5 Runtime

Table 5.3 shows the runtime of each software tool on the benchmark datasets. It should be noted that the runtime for MaxQuant includes both MS peptide detection and MS/MS peptide identification, since the program sequentially conducts MS/MS identification after MS peptide detection without pause. The runtime of the other programs is for the MS peptide detection only.

5.2 Discussion

In a bottom-up DDA mass spectrometry experiment, the instrument does not have sufficient duty cycles to produce MS/MS scans of each peptide in the sample. However, these

	Mouse BV2	Yeast S288c	Human HeLa-1	Human HeLa-2
MSTracer (MS1)	33	12	11	11
MaxQuant (MS1+MS2)	22	3	145	145
Dinosaur (MS1)	6	3	2	2
OpenMS (MS1)	65	50	18	18

Table 5.3: Runtime (min) of MSTracer, MaxQuant, Dinosaur, and OpenMS on the four datasets on a Linux machine (Intel Core i7-6770HQ CPU with 16 GB of memory). MS1 represents MS peptide detection from MS1 spectra, and MS2 represents MS/MS peptide identification from MS2 spectra.

peptides still produce traces in the MS scans. These traces are called peptide features and can provide important information about the corresponding peptides, such as m/z, charge state, RT, and relative quantity. Accurate detection of these features is the first step in utilizing such information. This paper presents a new software application, MSTracer, for peptide feature detection.

The built-in ML models are essential implementations in MSTracer. When building the models, we extracted several pieces of information from the intensity of peptide features: Based on the change of the intensity of trails over RT, we obtained the coelution coefficient. Based on the sum intensity of each isotope in a cluster, we obtained the isotope shape coefficient. We also obtained the total intensity of isotope clusters and the relative intensity of each isotope cluster in the surrounding area. We tested various combinations of these variables in the SVR and NN training and finalized a best-performing model.

A unique feature of MSTracer is that, unlike other software tools, it assigns a quality score to each detected peptide feature. The quality score can be used to rank the results and offers users the flexibility to consider trade-offs between result accuracy and sensitivity (Figure 5.5). This feature also facilitated a fair comparison between MSTracer and other software tools, as it allowed us to truncate the results from MSTracer to give the same number of peptide features as another tool, thereby limiting the comparison strictly to sensitivity (i.e., the percentage of identifiable PSMs detected by each tool). MSTracer outperformed every other software tool in these comparisons (Figure 5.2).

It is noteworthy that a true peptide feature may not necessarily be identified by the MS/MS scans. In fact, the peptide may not even be acquired by the mass spectrometer to produce an MS/MS scan. On one hand, this makes the feature detection from MS scans particularly useful. On the other hand, it makes it difficult to determine whether a

prediction is a false positive and making it hard to compare different prediction algorithms based on precision or ROC curves.

As Figure 5.4 shows, the quality score can be used to effectively separate high- and low-quality peptide features. Moreover, the score distributions on the four separate datasets are very similar, indicating highly robust scores. We determined an empirical score threshold of -1 to filter the results of MSTracer.

The potential of combinations of multiple software tools is examined in Figure 5.5, which shows that peptide features detected by two or more software tools have a higher chance of matching the identifiable PSMs than those detected by only one tool. This confirms the common assumption that results from multiple tools are more reliable than those from a single tool. It can also be seen that MSTracer contributed more to the overlapping peptide features than each of the other tools. Moreover, among the peptide features detected by only one tool, those detected by MSTracer have a higher chance of matching the PSMs identified by MS/MS. All these findings suggest that MSTracer can identify not only more but also higher quality peptide features than the other tools.

If we only regard peptide features detected by at least two of the three software tools as correct, then the combination of the three tools was able to detect 178,927, 139,627, 133,394, and 132,374 peptide features from the MS1 spectra in the mouse, yeast, HeLa-1, and HeLa-2 datasets, respectively. Among these detected peptide features, only 29,275 (16%), 36,693 (26%), 50,111 (38%), and 49,357 (37%), respectively, were identified by the MS/MS spectra. This lower identification rate achieved by MS/MS is the result of two factors: not every peptide feature is acquired for MS/MS and not every MS/MS spectra is identifiable by a database search. This suggests that MS scans are a rich source of information that can be utilized by future proteomics software tools. Furthermore, there is much room to improve the number of identified peptides with new and more intelligent MS/MS acquisitions.

A novel reproducibility test was examined and is shown in Figure 5.7. Reproducibility can be affected by both datasets and the peptide detection software. Specifically, noises caused by slight changes in experimental conditions and the accuracy of the instrument can largely affect the results. Given that none of the software except for OpenMS has fairly high reproducibility, further study is warranted to investigate the source of irreproducibility and to either reduce or exploit it.

The runtime of MSTracer and the other software tools is shown in Table 5.3. Considering producing an MS dataset takes hours to generate on the mass spectrometer, all software tools have acceptable runtime. Within this time range, the detection sensitivity and accuracy become more important to compare.

Chapter 6

Conclusions and Future Work

In this thesis, we introduced a software tool, MSTracer, for peptide feature detection from the MS1 spectra of LC-MS/MS map.

In the beginning, we described the importance of the efficient use of MS1 spectra data and the challenges of the accurate detection of peptide features from MS1 spectra. Next, we reviewed current software tools and their methods. Then, we proposed a novel algorithm that combines SVR and NN models.

The algorithm was established on the basic pipeline of peptide feature detection, where it implements the steps of trail construction and isotope clustering. During the process, all possible isotope groups were detected for being further analyzed by SVR and NN models. SVR selects the peptide features with high confidence from conflicting isotope groups, and NN assigns a quality score to each reported peptide feature. Variables used in the SVR and NN models were carefully selected to optimize the models.

During the performance evaluation, the sensitivity of MSTracer outperformed the existing software tools, MaxQuant, Dinosaur, and OpenMS for the test datasets from different species - human, mouse, and yeast. Furthermore, the results from using the PSMs from two MS/MS search tools revealed that the sensitivity test result of MSTracer is not affected by which search engine is used for the test. In addition, MSTracer assigns quality scores to the reported peptide features. From a histogram of the quality score distribution, we concluded that the value -1 can be used to separate high- and low-quality peptide features. We then applied this finding to MSTracer to evaluate the combine use of multiple software tools, where we took the features of quality score > -1. The evaluation showed that the joint use of multiple software increases the probability of the peptide features matching

PSMs. Lastly, we proposed a new concept, reproducibility, which measures how identical the peptide features that the algorithm detects for identical experimental samples.

For the future work, we propose the following avenues:

- One avenue for future research is to improve the performance of the ML models. In our current work, the ML models have been trained with a small human cell dataset only. The training dataset could be expanded to train the models for specific species and specific widely used instruments (e.g., Q Exactive, Orbitrap Fusion, LTQ). We anticipate that providing users with options regarding species and instruments will further improve the sensitivity and accuracy of MSTracer.

- Another avenue is to add functionalities to MSTracer, including monoisotope picking, that are comparable with MS/MS search tools. In addition, as demonstrated in the reproducibility test, the robustness of our program can be further improved.

- Last but not least, we plan to incorporate MSTracer into a full protein MS dataset analysis pipeline that includes MS peptide detection, MS/MS peptide identification, and peptide quantification.

References

[1] Steven A. Carr, Ronald S. Annan, Guoan Zhang, and Thomas A. Neubert. Overview of peptide and protein analysis by mass spectrometry. *Current protocols in protein science*, Chapter 16:Unit16.1, 2010.

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

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

[4] Jimmy K. Eng, Tahmina A. Jahan, and Michael R. Hoopmann. Comet: An open-source ms/ms sequence database search tool. *Proteomics*, 13(1):22–24, 2012.

[5] 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.

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

[7] Shenheng P. Guan, Paul F. Taylor, Ziwei Han, Michael F. Moran, and Bin Ma. Data dependent–independent acquisition (ddia) proteomics. *Journal of Proteome Research*, 19(8):3230–3237, 2020.

[8] Thomas J. Hedl, Rebecca San Gil, Flora Cheng, Stephanie L. Rayner, Jennilee M. Davidson, Alana De Luca, Maria D. Villalva, Heath Ecroyd, Adam K. Walker, and Albert Lee. Proteomics approaches for biomarker and drug target discovery in als and ftd. *Frontiers in Neuroscience*, 13:548, 2019.

[9] Yeakuty M. Jhanker, Mohammad F. Kadir, Rajibul I. Khan, and Rubaiyat Hasan. Proteomics in drug discovery. *Journal of Applied Pharmaceutical Science*, 2(8):1–12, 2012.

[10] Maria Kavallaris and Glenn M. Marshall. Proteomics and disease: opportunities and challenges. *The Medical Journal of Australia*, 182(11):575–579, 2005.

[11] Dong Kyu Kim, Dohyun Han, and et al. Deep proteome profiling of the hippocampus in the 5xfad mouse model reveals biological process alterations and a novel biomarker of alzheimer's disease. *Experimental Molecular Medicine*, 51(11):1–17, 2019.

[12] Sangtae Kim and Pavel A. Pevzner. Universal database search tool for proteomics. *Nature Communications*, 5:5277, 2014.

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

[14] Bin Ma and Richard Johnson. De novo sequencing and homology searching. *Molecular Cellular Proteomics*, 11(2):O111.014902, 2011.

[15] Chi N. I. Pang, Sara Ballouz, and et al. Analytical guidelines for co-fractionation mass spectrometry obtained through global profiling of gold standard saccharomyces cerevisiae protein complexes. *Molecular Cellular Proteomics*, 19(11):1876–1895, 2020.

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

[17] Hannes L. Röst, Timo Sachsenberg, and et al. Openms: a flexible open-source software platform for mass spectrometry data analysis. *Nature Methods*, 13:741–748, 2016.

[18] Johan Teleman, Aakash Chawade, Marianne Sandin, Fredrik Levander, and Johan Malmström. Dinosaur: A refined open-source peptide ms feature detector. *Journal of Proteome Research*, 15(7):2143–2151, Aug 2016.

[19] Ngoc H. 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, 2018.

[20] Valerie C. Wasinger, Stuart J. Cordwell, and et al. Progress with gene-product mapping of the mollicutes: Mycoplasma genitalium. *Electrophoresis*, 16(7):1090–1094, 1995.

[21] Hao Yang, Hao Chi, Wen-Feng Zeng, Wen-Jing Zhou, and Si-Min He. pnovo 3: precise de novo peptide sequencing using a learning-to-rank framework. *Bioinformatics*, 35(14):i183–i190, 2019.

[22] Jing Zhang, Lei Xin, and et al. Peaks db:de novo sequencing assisted database search for sensitive and accurate peptide identification. *Molecular Cellular Proteomics*, 11(4):M111.010587, 2011.

[23] Fatema T. Zohora, M. Z. Rahman, Ngoc H. Tran, Lei Xin, Baozhen Shan, and Ming Li. Deepiso: A deep learning model for peptide feature detection from lc-ms map. *Scientific Reports*, 9:17168, 2019.