Enhanced protein identification through automated search parameter selection

Henrik Øvrebø Søgaard

Master Thesis
Department of Informatics
University of Bergen

June 1st, 2023

Table of Contents

1. Abstract **4**

2. Aims **5**

3. Background **6**

3.1 Proteomics *6*
3.2 Protein cleavage into peptides *7*
3.3 Mass spectrometry *8*
3.4 Peptide fragmentation *10*
3.5 Protein databases *12*
3.6 Proteomics search engines *12*
3.7 Post-translational modifications *15*
3.8 SearchGUI and PeptideShaker *15*
3.9 Search engine parameters – general and advanced *16*
3.10 Software for search parameter selection *19*

4. Methods **21**

4.1 Spectrum file type selection *21*
4.2 Spectrum quality *22*
4.3 Selected dataset *23*
4.4 Categorizing the PSMs *23*
4.5 Generation of subsets *25*
4.6 Code for subset generation *26*

5. Results **27**

5.1 Parameter selection *27*
5.2 Generating subsets *30*
5.3 Random selection *33*
5.4 Picking every n-th spectrum *34*
5.5 Testing different general parameters *36*
5.6 Testing advanced parameters *38*
5.7 Testing the n-th spectra selection on other datasets *40*
5.8 Manual optimization of parameters *41*

6. Discussion **45**

6.1 Number of PSMs as a test metric 45
6.2 Possible overfitting 45
6.3 Spectrum quality 46
6.4 Expanding to other proteomics search engines 46
6.5 Potential consequences of randomizing 46
6.6 Choosing an appropriate value of n 47
6.7 Testing combinations of parameters 47
6.8 Expanding to more data 47

7. Future work **49**

8. Conclusion **50**

9. Acknowledgements **51**

10. References **52**

Table of Figures

Figure 1: Proteomics Overview. 7
Figure 2: A Simplified Depiction Of Proteolytic Enzymes Cleaving A Protein Into Peptides. 8
Figure 3: Simple Figure Showing How A Mass Spectrometer Works. 9
Figure 4: Example Of An MS/MS Spectrum. 10
Figure 5: Proteomics Workflow. 11
Figure 6: Number Of Citations Per Year Of The Ten Most Cited Proteomics Search Engines. 14
Figure 7: Number Of Citations Per Year For The Proteomics Search Engines In The "Other" Category 14
Figure 8: Example Of (Part Of) An MGF File.. 22
Figure 9: Example Chromatogram 23
Figure 10: The Number Of PSMs From The Original Dataset Are Distributed Across The Five Categories In PeptideShaker. 24
Figure 11: Screenshot Of SearchGUI. 28
Figure 12: Overview Of The Common Default Parameters Used In This Project. 30
Figure 13: The Three Approaches Used When Generating A Subset Dataset File. 31
Figure 14: What Categories The Potential PSMs Are Put In For Three Datasets From Different Parts Of The File, Compared To The Complete File. 32
Figure 15: ID Rate Of Ten Randomized Subsets. 34
Figure 16: Testing Different N-Values. 35
Figure 17: Comparing The Chronological Approach To Every N-Th Spectrum Approach.. 36
Figure 18: Identification Rates In Percent Of The Complete File And The Subset File Generated With N=10 When Changing The Cleavage Enzyme. 37
Figure 19: The Different Fragment Ion Type Parameter Combinations Resulting In Almost No Change In The Identification Rate. 38
Figure 20: The Amount Of Confident PSMs, In Percent, For Different Parameter Settings For MS-GF+. 39
Figure 21: The Number Of PSMs, In Percent, When Changing The Same Parameters On The Subset Containing 10 Percent Of The Spectra From The Original Dataset.. 40
Figure 22: Testing Other Datasets. 41
Figure 23: The Amount Of PSMs After Searching With MS-GF+ With Default Parameters, Our Own Manual Optimization, And With The Parameters Used In The Experiment.. 43

1. Abstract

Mass spectrometry-based proteomics plays a critical role in identifying and quantifying proteins. Proteomics search engines, integral to this process, require meticulous parameter selection to achieve accurate results. However, the large number of available search parameters makes it challenging to manually choose the optimal combinations. This thesis focuses on optimizing and automating the parameter selection process. The main idea is to select and search a subset of the data that preserves the properties of the complete dataset, enabling efficient parameter exploration while minimizing both computational resources and time requirements. The approach has been validated using various mass spectrometry datasets from PRIDE analyzed via the SearchGUI and PeptideShaker framework. Easy support for testing multiple parameter combinations ultimately leads to overall better parameter selection, thus enhancing protein identification accuracy and workflow efficiency, which in turn contributes to getting the most out of valuable biological samples.

2. Aims

This project aims to optimize the parameter selection for proteomics search engines, which is crucial for identifying proteins in mass spectrometry-based proteomics. Most search engines have numerous parameters that can be essential for the outcome and to some degree have to be tailored to the data being searched. The work builds on the SearchGUI [1] framework for executing multiple search engines, and a corresponding tool called PeptideShaker [2] to compare the results and explore how they are affected by the search parameters. The main goal is to optimize the parameters without having to search all of the mass spectrometry dataset, but instead selecting and searching a subset of the data that mimics the properties and structure of the complete dataset. A potential benefit would be a reduction of time and computational resources, which in turn would make it possible to test for all combinations of parameters at once. Ultimately, the goal would be to identify the best combinations of parameters for the given dataset.

3. Background

For this background section, I will introduce and talk about topics that are important for both proteomics and for this thesis. Firstly, I will talk about proteomics in general. Thereafter, protein cleavage and peptide fragmentation. Then I will introduce the high-throughput technology mass spectrometry and different spectrum file types, before moving on to databases, and proteomics search engines and their corresponding parameters. Next up is peptide identification and the SearchGUI framework. Lastly, I will talk about existing software on the topic.

3.1 Proteomics

Proteomics is a rapidly growing field of life science that aims to identify, quantify, and study the functions of proteins expressed by either a cell, tissue, or organism. It involves the large-scale analysis of proteins, which carries out vital functions such as catalysis, regulation, and structural support [3]. A proteome is defined as a set of proteins produced in an organism, system, or biological context. One may refer to, for instance, the proteome of a species (*e.g.*, homo sapiens) or an organ (*e.g.*, the liver). Proteomics research relies heavily on high-throughput technologies, which enables the identification and quantification of thousands of proteins simultaneously. These techniques provide valuable information about protein expression, post-translational modifications, protein-protein interaction, and protein localization, contributing to our understanding of biological processes and disease mechanisms [3]. An overview of the proteomics field is shown in **Figure 1**.

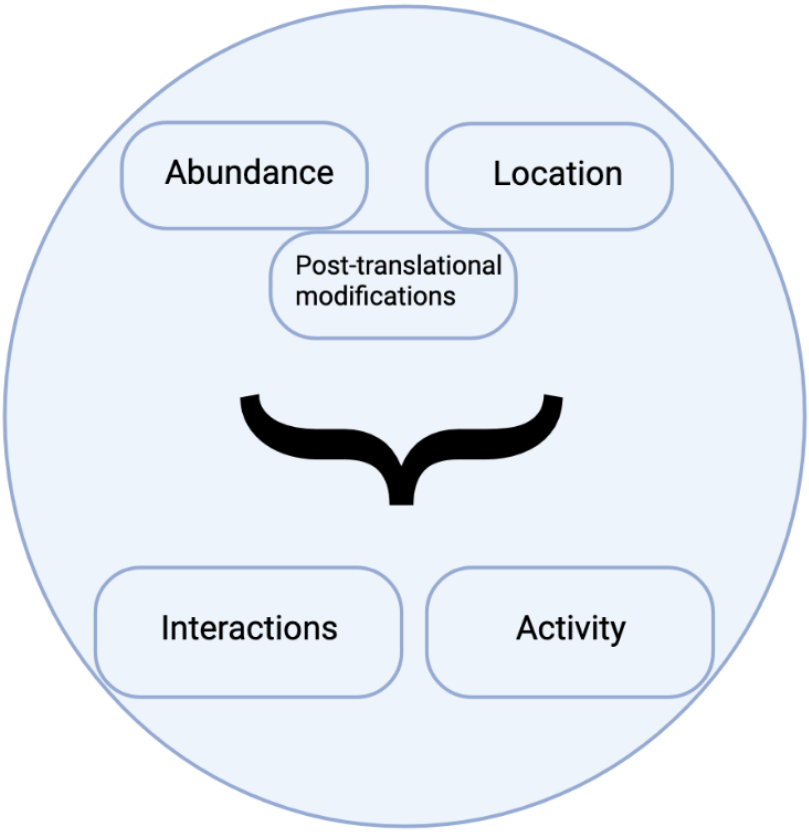

Figure 1: Proteomics overview. Experiments generally collect data on three properties of proteins in a sample: location, abundance/turnover and post-translational modifications. The aim is to infer protein activity and interactions.

3.2 Protein cleavage into peptides

Peptides are short chains of amino acids that play an important in proteomics. In many ways, a peptide can be thought of as a small protein fragment. These fragments are most often analyzed in so-called bottom-up proteomics. Peptides are typically generated in samples by enzymatically digesting proteins into smaller parts. This process is called enzymatic cleavage, and it is done by proteolytic enzymes, also known as proteases. The process is shown in **Figure 2** below.

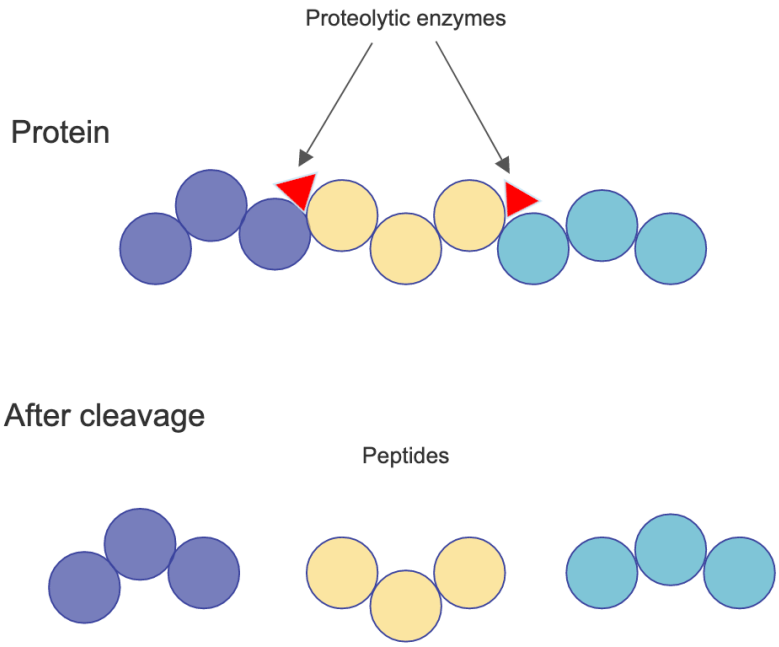

Figure 2: A simplified depiction of proteolytic enzymes cleaving a protein into peptides.

Proteases can be classified based on their mechanism of action, specificity, and optimal conditions for activity. For example, the most frequently used protease, trypsin, is a protease that cleaves peptide bonds on the carboxyl side of basic amino acids, arginine and lysine. Chymotrypsin, on the other hand, cleaves on the carboxyl side of hydrophobic amino acids, phenylalanine, tryptophan, tyrosine and leucine. Other commonly used proteases include Lys-C, Asp-N, and Glu-C [4].

Enzymatic cleavage is a critical step in proteomics, as it creates a mixture of peptides that can be analyzed to identify proteins and their post-translational modifications.

3.3 Mass spectrometry

Mass spectrometry has been widely used to analyze biological samples and has evolved into an indispensable tool for proteomics research [5]. It has become the main technique for protein identification and characterization [6]. The method determines the mass and chemical composition of molecules by ionizing them and measuring their mass-to-charge (m/z) ratio. Since the molecules are ionized in order for them to be detected, they will always have a charge, commonly known as the precursor charge.

In mass spectrometry, a sample is first ionized, usually by an electron beam or a laser, and then separated based on their mass-to-charge ratio using an electric or magnetic field. The resulting ions are detected and measured, allowing for the identification and quantification of the sample's constituents. **Figure 3** shows a mass spectrometer and the most important elements of it. Measuring the masses of the ions in the MS scans are done by an analyzer, and the accuracy of this analyzer is often referred to as peptide tolerance, peptide mass error, or precursor tolerance [7]. The resulting output is provided as a mass spectrum, and it consists of the m/z values plotted against the intensity values. Depending on the software and instrument used, the output format can vary. Since the mass spectrometer distinguishes the molecules based on their mass, isotopes play a key role because they have different masses. Each isotope will show up in the mass spectrum as its own line.

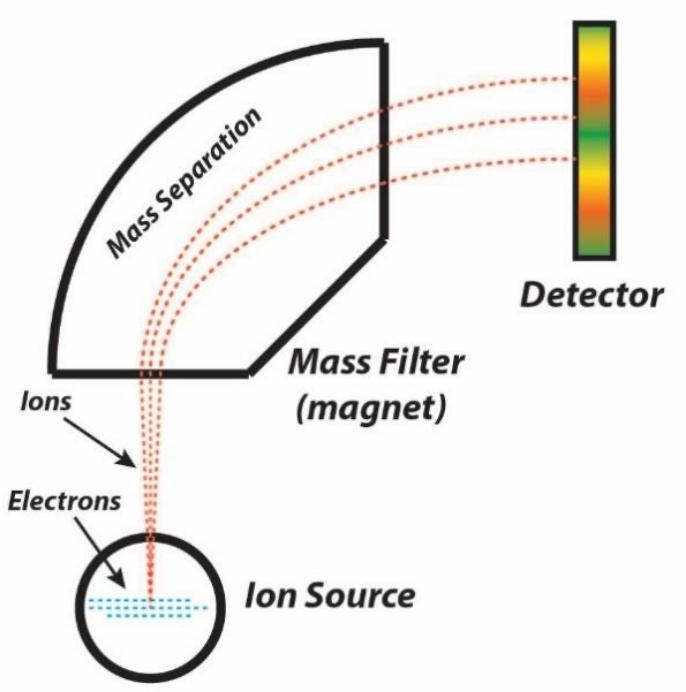

Figure 3: Simple figure showing how a mass spectrometer works. (From https://www.researchgate.net/figure/Schematic-diagram-of-a-mass-spectrometer_fig14_312664731)

It is also worth mentioning that there are different forms of mass spectrometry. There is mass spectrometry, as described above, referred to as simply MS or MS1, and tandem mass spectrometry, commonly referred to as MS/MS or MS2. MS/MS is a two-step technique used

to analyze a sample using a single mass spectrometer with several analyzers arranged one after another. The analyzers measure the mass of the fragments, and the mass accuracy of the analyzer is referred to as fragment tolerance, fragment mass error, or product ion tolerance [7]. Tandem mass spectrometry increases the ability to analyze chemical samples and also increases specificity [8]. The resulting output from a MS/MS differs somewhat from the MS output since in the MS mass spectrum each peak represents a peptide. However, in a MS/MS spectrum, each peak represents a fragment of a given peptide, and all of the peaks in the spectrum are from the same peptide, only different fragments of it. **Figure 4** below shows a typical MS/MS mass spectrum.

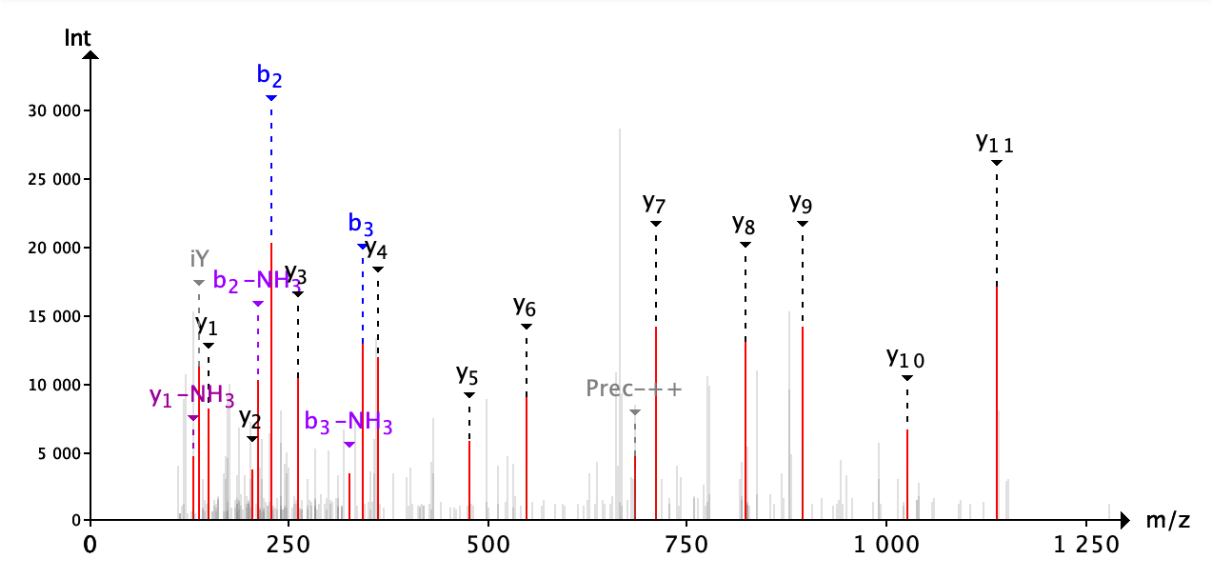

Figure 4: Example of an MS/MS spectrum. With m/z ratios along the x-axis and intensity along the y-axis. Each peak represents a component of unique m/z in the sample, and the height of the peaks show the relative abundance of the given component in the sample. (Figure from PeptideShaker [2]*)*

3.4 Peptide fragmentation

An important step in mass spectrometry-based proteomics is peptide fragmentation. It is a process in mass spectrometry-based proteomics where peptides are broken down into smaller fragments or ions. This fragmentation is essential for the identification of peptides in complex biological samples [9]. Fragmentation occurs through different mechanisms such as collision-

induced dissociation (CID), electron-transfer dissociation (ETD), and higher-energy collisional dissociation (HCD). During CID, the peptides are fragmented by colliding with a gas molecule in a collision cell. ETD involves transferring electrons to the peptides to induce fragmentation [10]. In HCD, the process is similar to that of the CID, however the collision with the gas molecule in the collision cell is high-energy. The resulting fragments are then detected by the mass spectrometer, and their masses are used to determine the amino acid sequence of the peptide, which is in turn used to identify the protein from which the peptide originated [11]. This process is shown in **Figure 5**, which depicts a typical proteomics workflow.

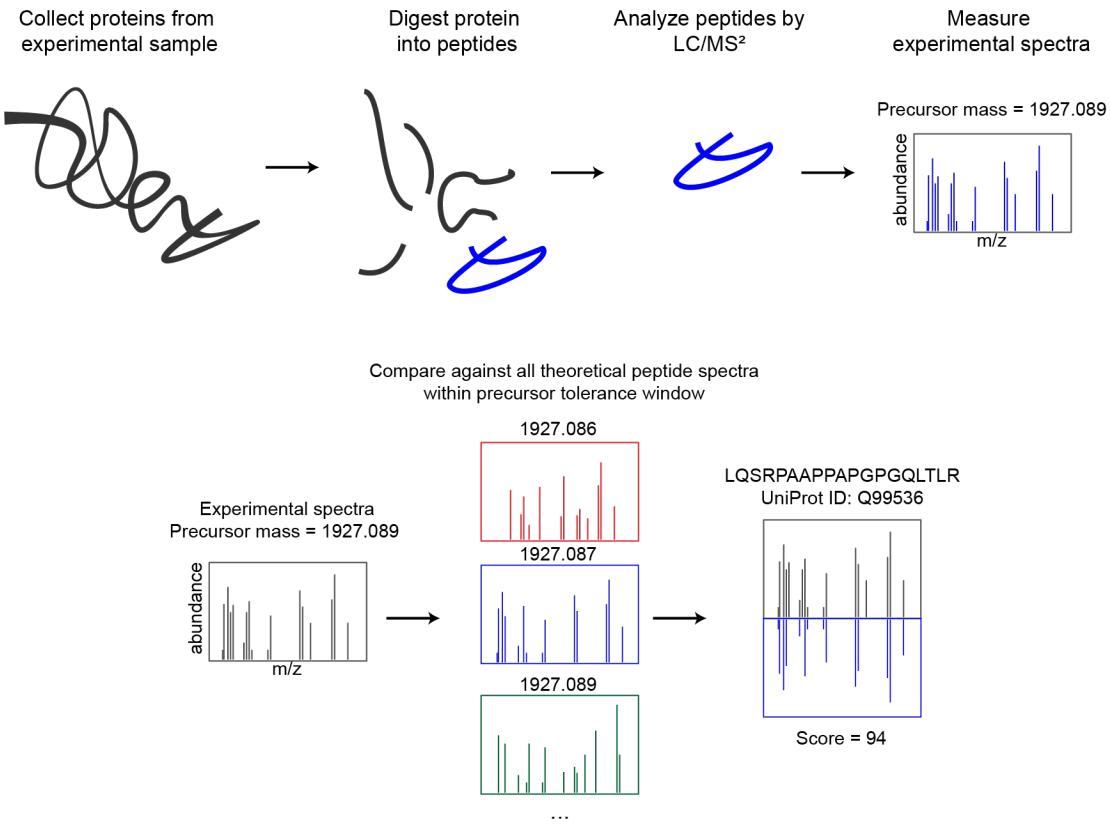

Figure 5: Proteomics workflow. (From: https://lazear.github.io/sage)

Depending on the fragmentation mechanism, one gets different fragment ion types. For CID and HCD fragmentation, the most commonly observed ions are b-ions and y-ions. While ETD fragmentation usually generate c- and z-type ions [12].

3.5 Protein databases

Protein databases in proteomics are essential resources that store information about the protein sequences of various organisms. These databases contain protein sequences obtained from various sources, such as experimental studies, computational predictions, and from literature. They are used as a reference for peptide identification in mass spectrometry-based proteomics studies. A variety of protein sequence databases exist, ranging from simple sequence repositories, which store data with little or no manual intervention in the creation of the records, to expertly curated universal databases that cover all species [13]. One of the most common is The Universal Protein Resource [14], or UniProt for short, a comprehensive resource for protein sequence and annotation data. Another popular database is the PRoteomics IDEntifications (PRIDE) Archive [15]. It is a public data repository for mass spectrometry proteomics data, including protein and peptide identifications and the corresponding expression values, post-translational modifications and supporting mass spectra.

3.6 Proteomics search engines

The data stored in spectrum files can be used to identify peptides by importing them and running a search in a proteomics search engine. Proteomics search engines are software tools that are widely used in mass spectrometry-based proteomics to identify peptides from complex mixtures [9]. These tools use databases of protein sequences to match with experimental spectra generated by mass spectrometry. It is also possible to do this without a database and deduce the sequence of peptides directly from the experimental MS/MS spectra. This method is called “de novo” sequencing. However, we will be sticking to database matching in this thesis.

Peptide identification using search engines involves several steps. One of the steps is database searching, and it involves the matching of the experimental spectra against protein sequences in a database, using algorithms that consider a range of factors such as mass accuracy, peptide length, and the presence of post-translational modifications. During the database searching step, the proteomics search engines iterate through a protein database.

Post-processing involves filtering and scoring the search results to identify the most confident peptide identifications. This step typically involves the use of statistical models to estimate the probability of correct identification. The scoring itself is done by comparing experimental spectra (the actual mass-to-charge ratio values and intensities observed in a mass spectrometry experiment) to theoretical spectra derived from protein databases. The search engine evaluates

how well a theoretical spectrum matches an experimental spectrum based on various factors, such as the number and intensity of matching peaks, the mass error, the charge state of the peptide, and the likelihood of the peptide sequence given the enzyme used for protein digestion [16].

Once the search engine has evaluated each potential peptide match, it assigns a score to each peptide-spectrum match (PSM). This score represents the likelihood that the peptide sequence is correct and is reported either as a standalone score or alongside a statistical measure, such as *p*-value or *e*-value [16]. Different search engines use different scoring algorithms, but they all aim to balance the need for sensitivity (*i.e.*, finding true positive PSMs) and specificity (*i.e.*, avoiding false positive PSMs) in peptide identification.

There are numerous proteomics search engines available, for example, MS-GF+ [17], Mascot [18], SEQUEST [19], Comet [20] and X! Tandem [21], to name a few. Some of these are more popular and frequently used than others, as shown in **Figure 6** and **Figure 7** below. The figures are an updated version of the supplementary figures from the original paper on the anatomy and evolution of proteomics search engines [9]. From the figures it becomes clear that older search engines see less and less usage, and that the newer ones, like MaxQuant [22], have gained more popularity.

In this thesis, the main focus will be on the search engines featured in SearchGUI [1], an open-source software tool that provides a graphical user interface (GUI) for configuring and running multiple peptide search engines in a single pipeline. SearchGUI also includes various output formats for easy integration with downstream data analysis tools. This, combined with the ability to run multiple peptide search engines in a single pipeline, allows the user to compare and combine the results of different search algorithms.

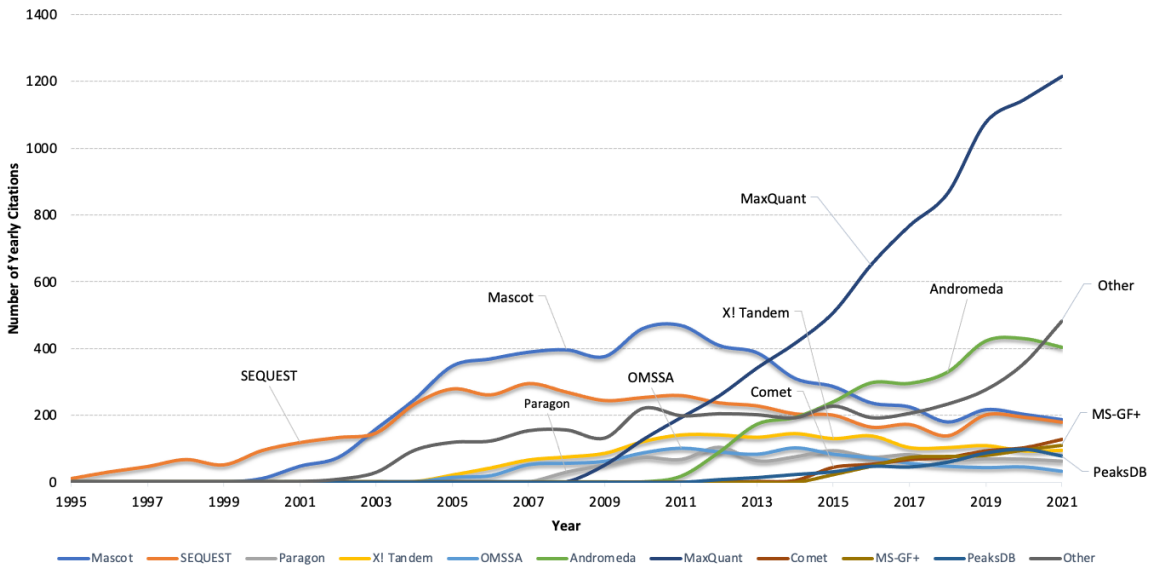

Figure 6: Number of citations per year of the ten most cited proteomics search engines. (Data collected from Clarivate Web of Science. © Copyright Clarivate 2023. All rights reserved.)

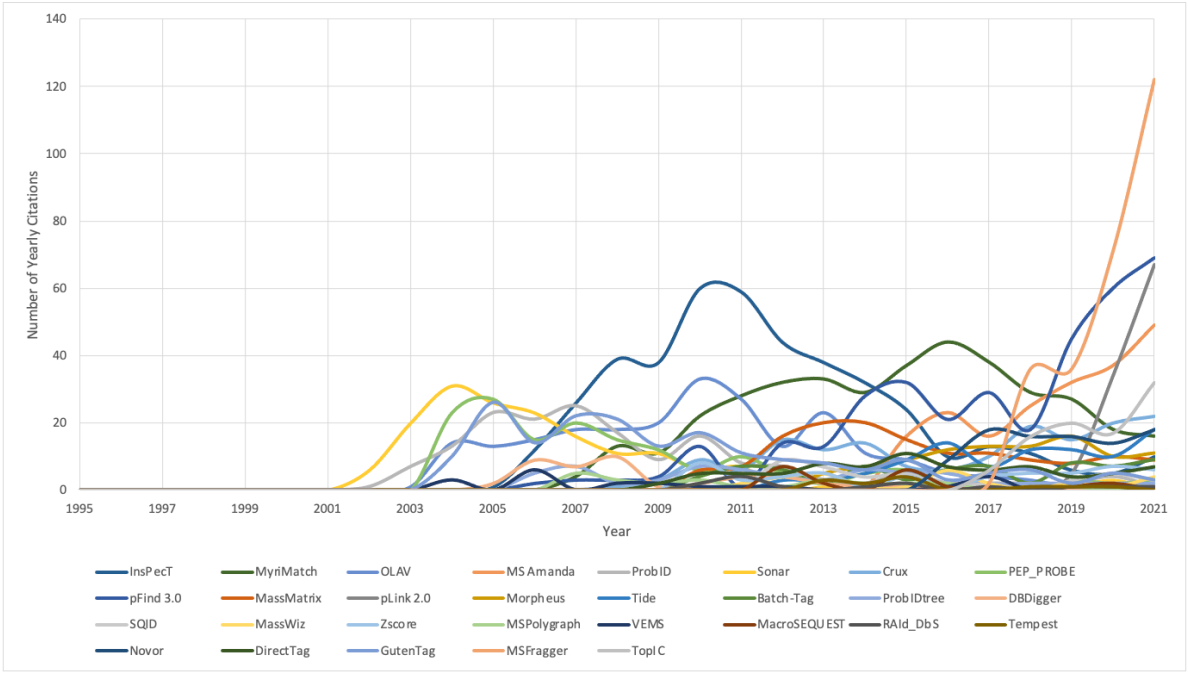

Figure 7: Number of citations per year for the proteomics search engines in the "Other" category in Figure 6. (Data collected from Clarivate Web of Science. © Copyright Clarivate 2023. All rights reserved.)

3.7 Post-translational modifications

A protein is at any stage of its life cycle susceptible to post-translational modifications (PTMs). PTMs refer to the covalent modification of proteins after they have been synthesized from their corresponding genes. PTMs are essential for regulating protein activity, stability, localization, and function, and are involved in various cellular processes. Proteins can undergo a wide range of PTMs, and each PTM can alter the structure and function of the protein in a unique way, allowing for precise control of protein activity and function [23]. Search engines generally use protein sequence databases to match experimental spectra to theoretical spectra. The protein sequences in the databases can be annotated with PTMs, which can result in a large number of potential matches for a given experimental spectrum. Searching all possible combinations of PTMs causes both the run time and the search space to grow exponentially [24]. However, there are methods for addressing this, namely Open Modification Search (OMS) which considers all modifications and can handle the complexity. A popular software for performing OMS is for example SpecOMS [25].

To improve the sensitivity and accuracy of peptide identification, search engines typically allow for the inclusion of specific PTMs in the search parameters. This can greatly reduce the search space and increase the specificity of peptide identification. However, PTMs are a research topic on their own, and outside the scope of this master project.

3.8 SearchGUI and PeptideShaker

The Proteomics Unit at UiB has implemented a software tool for the comprehensive interpretation, processing, and inspection of the output from the peptide search engines, called PeptideShaker [2]. PeptideShaker provides a user-friendly interface for peptide and protein identification and validation, quantification, and visualization of the results from the search engines. It assigns a confidence level to each identified peptide and protein using a combination of scoring algorithms and statistical tests.

PeptideShaker is also capable of automatically validating and filtering the peptide identification results based on various criteria, such as precursor and fragment mass tolerance, and post-translational modifications, which are parameters the user specifies in SearchGUI before running a search. This allows the user to focus on the most reliable and relevant results to avoid false positives. PeptideShaker also enables users to export the data in several different file formats for further analysis and integration with other tools.

3.9 Search engine parameters – general and advanced

Peptide identification in mass spectrometry-based proteomics is a crucial task that enables the characterization of proteins in complex biological samples. The database identification process relies on search engines that compare experimental spectra with theoretical spectra derived from protein databases. This task is implemented in various search engines. All of these have lots of parameters that are vital for the outcome and that, to some extent, have to be tailored to the data searched. For example, one has to set which enzyme/protease was used to cleave the proteins into peptides, the accuracies for the used mass spectrometer, and the number of charges to consider. In addition to these common parameters, there are also numerous custom advanced parameters for each search engine that can potentially have a major impact on the results.

Furthermore, there are also different types of parameters. One parameter may take integer values as input, while for another parameter one has to choose a value from a list. There are also parameters which are just Boolean input.

The different common parameters, and an example selection of custom advanced parameters, and their corresponding parameter types are listed in **Table 1** and in **Table 2**.

Parameters	Type	Description
Fixed modifications	List	Fixed modifications to consider.
Variable modifications	List	Variable modifications to consider.
Digestion	List	Type of digestion. Enzyme, unspecific or whole protein digestion.
Enzyme	List	Proteolytic enzyme used to cleave the protein into peptides.
Specificity	List	Specificity of the enzymatic cleavage.
Max missed cleavages	Integer	Maximum missed cleavages allowed.
Fragment ion types	List	Classification of the N-terminal charged fragment ions and the C-terminal charged fragment ions.
Precursor tolerance	Number	Precursor mass tolerance.
Fragment tolerance	Number	Fragment mass tolerance.
Precursor charge	Range	A range of peptide charges to consider.
Isotopes	Integer	Number of isotopes to consider.

Table 1: Overview of the most common search parameters that can be set for all search engines in SearchGUI.

Parameters	Parameter type	Description
Search decoy database	Boolean	Indicates whether to search normal (forward only) protein sequences, or a decoy file where the reversed protein sequences are appended to the normal protein sequences.
MS/MS Detector	List	Identifier of the instrument used to generate MS/MS spectra.
Fragmentation method	List	Fragmentation method identifier (used to determine the scoring model).
Protocol	List	Protocol identifier. Protocols are used to enable scoring parameters for enriched and/or labeled samples.
Enzymatic terminals	List	Number of tolerable termini (aka tryptic termini). This parameter is used to apply the enzyme cleavage specificity rule when searching the database.
Peptide length	Range	Min and max length of peptides to be considered.
Max variable PTMs per peptide	Integer	Maximum number of dynamic (variable) modifications per peptide.
Number of spectrum matches	Integer	Number of peptide matches per spectrum to report.
Additional output	Boolean	Changes the verbosity of the output.
Number of tasks	Integer	Manually set the number of tasks to create for the search.

Table 2: Overview of the advanced parameters that can be set in the MS-GF+ search engine.

3.10 Software for search parameter selection

Search parameter selection in proteomics plays a crucial role in the accurate identification and characterization of peptides from mass spectrometry data. These parameters include enzyme specificity, mass tolerances, variable modifications, and database selection. By fine-tuning these parameters, researchers can enhance the confidence and depth of their peptide identifications, leading to valuable insights into biological systems.

To our knowledge there are only two software tools that provide parameter optimization: Preview [26] and Param-Medic [27]. Preview is a program that is part of the Byonic™ [28] software package. It analyzes a set of mass spectra for mass errors, digestion specificity, and known and unknown modifications, thereby facilitating parameter selection. After making a simplifying assumption that the 100 most detectable proteins represent the entire sample, Preview is able to run a full database search for digestion-specific peptides in a fraction of the time of a standard search program, only performing a single pass over the full protein database. Preview then performs all subsequent searches on representative proteins and likely peptides from said representative proteins. The program is very much focused on post-translational modifications.

The simplifying assumption made by Preview causes some sensitivity loss. Preview assumes, as mentioned, that the most detectable proteins represent the entire dataset for the full menu of search parameters. This is true for smaller samples with less than 100 proteins. However, it is less true for more complex samples [26].

Preview also supports recalibration of m/z measurements, *i.e.,* the process of correcting systematic errors that may have occurred during the mass spectrometry measurement. There are several causes for this, including instrument drift, mass spectrometer calibration issues, or signal processing artifacts. The process involves using a set of reference masses (*e.g.*, peptides) to adjust the mass accuracy of the instrument. Recalibration of m/z measurements is, however, outside the scope of this master thesis and will not be detailed further.

The second option available is Param-Medic. It is an open source and cross-platform program, available as a standalone tool and integrated into the Crux proteomics toolkit [29], where it provides parameter selection for the Comet [20] and Tide [30] search engines. Param-Medic focuses on two of the most important parameters; precursor mass tolerance and fragment mass

tolerance, also known as bin size. Precursor mass tolerance defines the peptide candidates considered for each spectrum, and the fragment mass tolerance determines how close the observed and theoretical fragments must be in order to be considered a match.

For either of these two parameters, too wide a setting yields randomly high-scoring false peptide spectrum matches, whereas a too narrow setting erroneously excludes true peptide spectrum matches, in either case lowering the yield of peptides detected at a given false discovery rate [27].

To summarize, Param-Medic examines the spectra in a file to best estimate precursor and fragment mass tolerance, and Preview heavily targets post-translational modifications. In other words, neither of the two focus on the long list of additional adjustable parameters.

4. Methods

4.1 Spectrum file type selection

In nearly all high-throughput proteomics workflows, extensive analysis with custom software is required to translate the mass spectra into peptide identifications and perform abundance measurements. As a result, a wide variety of data formats have emerged. Two common data formats, mzData and mzXML, were developed around the same time, and stores more or less the same information. They both store data on what MS instrument was used, what mass analyzer was used, and what the detection method is, to name some. Therefore, the developers behind the two formats came together, and the mzML file format was created, including the best features from both mzData and mzXML [31].

However, all of these three formats have quite complex overheads containing a lot of information, as mentioned above. This makes the file size and access time significantly higher than for pure text-based formats [31]. A typical file contains upwards to 1 million spectra, making the file size about 1-3 GB. There are however projects working towards bettering this, for example the mz5 project [32] and the introduction of mzMLb [33].

The simpler Mascot Generic Files (MGF) format came before the already mentioned file types, and it is similar to those formats in that it encodes for multiple MS/MS spectra in a single file via m/z intensity pairings. The data stored in MGF files include the spectra of peptide ions and their associated data, such as the precursor mass, charge state, and retention time. **Figure 8** shows how an MGF file is structured. The MGF format is likely the most common text format in mass spectrometry-based proteomics, probably because of its simplicity [31], and is the chosen format used throughout this thesis.

```
BEGIN IONS
TITLE=controllerType=0 controllerNumber=1 scan=4
PEPMASS=429.088836669922
RTINSECONDS=1.09895904
SCANS=4
324.9841 1282.247
338.89993 1063.086
341.01318 7826.125
341.37024 971.024
342.99326 3999.271
344.45935 964.455
359.02533 104021.117
360.02563 8910.044
364.32349 1065.102
375.88959 1580.365
392.86929 1730.834
393.85831 1254.68
410.89307 1562.968
411.84525 9868.051
428.84695 978.383
429.08633 11400.894
END IONS

BEGIN IONS
TITLE=controllerType=0 controllerNumber=1 scan=104
PEPMASS=327.077911376953
RTINSECONDS=30.6152742
SCANS=104
110.07941 362.851
113.05914 428.749
```

Figure 8: Example of (part of) an MGF file. All the information between "BEGIN IONS" and "END IONS" represents one spectrum.

4.2 Spectrum quality

When running a search with a spectrum file, it is commonly known that the start and the end of the file contains poor-quality spectra. This can be due to several factors, including instrument instability, ion suppression, and contamination. The ion source may not be fully stabilized, or the sample may not be completely ionized at the beginning of the analysis, leading to poor quality spectra. Similarly, at the end of the analysis, the ion source may be depleted, resulting in a decrease in ion signal and poor-quality spectra.

Another factor that may contribute to poor spectra at the start and end of the data is contamination. Contaminants, such as residual matrix or sample carryover, can interfere with ionization and affect the quality of spectra. There may also be a low abundance of peptides at the start and end of the file, as seen in **Figure 9**, but because of mass spectrometers high sensitivity, they may pick up and return a hit on what is generally known as "bad spectra." Although the quality of spectra at the start and end of the file may be poor, it is still important to include this data in the analysis to ensure that all detectable peptides are identified.

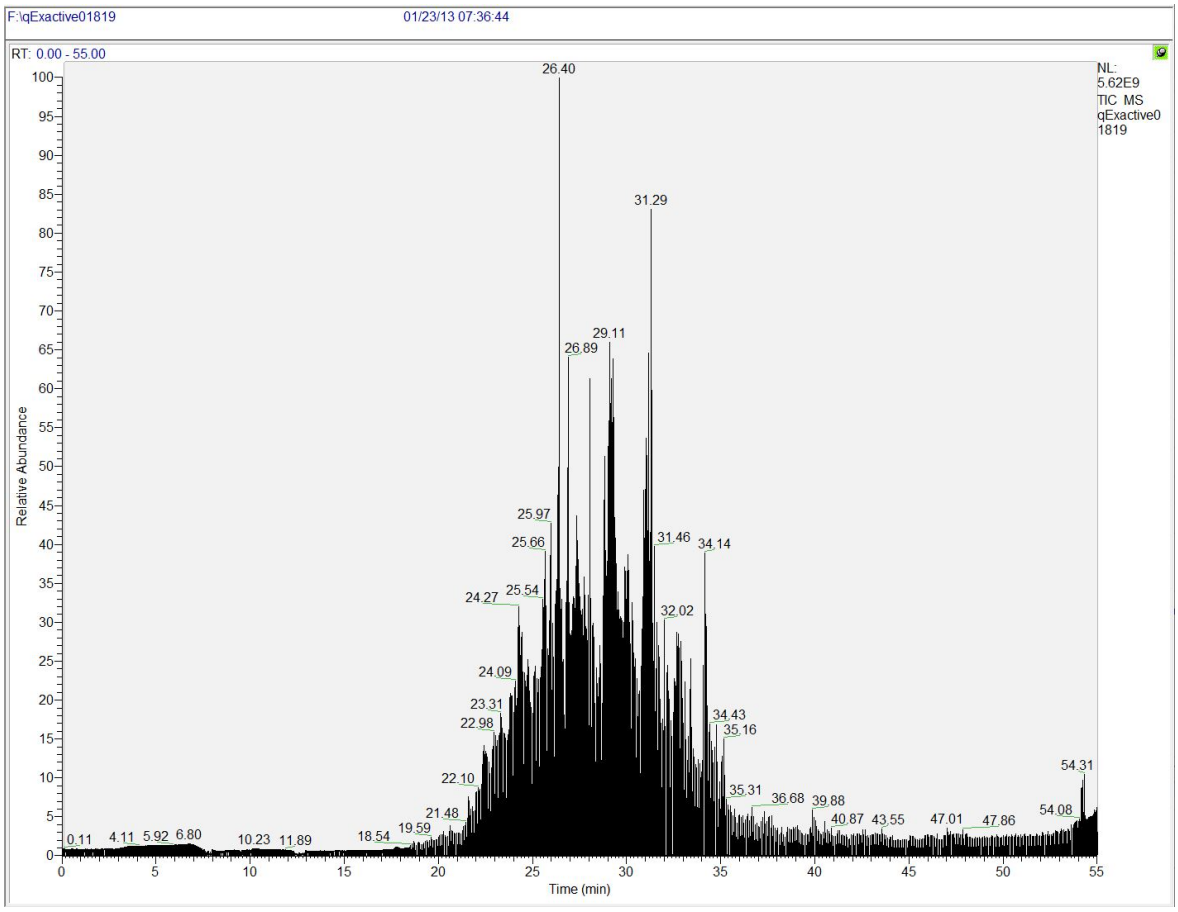

Figure 9: Example chromatogram. Indicating that both the start and the end of the file contains low abundance spectra, while the middle part contains relatively high abundance spectra. (Image generated by Xcalibur™, Thermo Fisher Scientific™)

4.3 Selected dataset

When conducting testing later on, we used an example dataset file generated by a Q Exactive (*Thermo Fisher Scientific™)* mass spectrometer. The Q Exactive file contains 11,332 spectra. The properties of the dataset include the number of PSMs and the distribution of these. In terms of differentiating and measuring the results from testing, we have chosen the number of PSMs compared to the total amount of spectra, also called identification rate, as our test metric.

4.4 Categorizing the PSMs

In SearchGUI, in the search engine(s) used, every potential peptide match is given a score after evaluation. This score represents the likelihood that the peptide sequence is correct and is reported either as a standalone score or paired with a statistical metric. In addition to this, each

potential PSM is placed in a category by PeptideShaker. Which category the potential PSM is put in depends on their score from the search engine. PeptideShaker operates with five categories in total. These are No Value, No Validation, Not Validated, Doubtful and Confident. When we export the information from PeptideShaker, these categories become numbered accordingly: -2, -1, 0, 1, and 2.

If we take the full QExactive dataset mentioned earlier in the Methods chapter as an example, **Figure 10** shows how the potential PSMs are categorized after a search with MS-GF+ and default parameters.

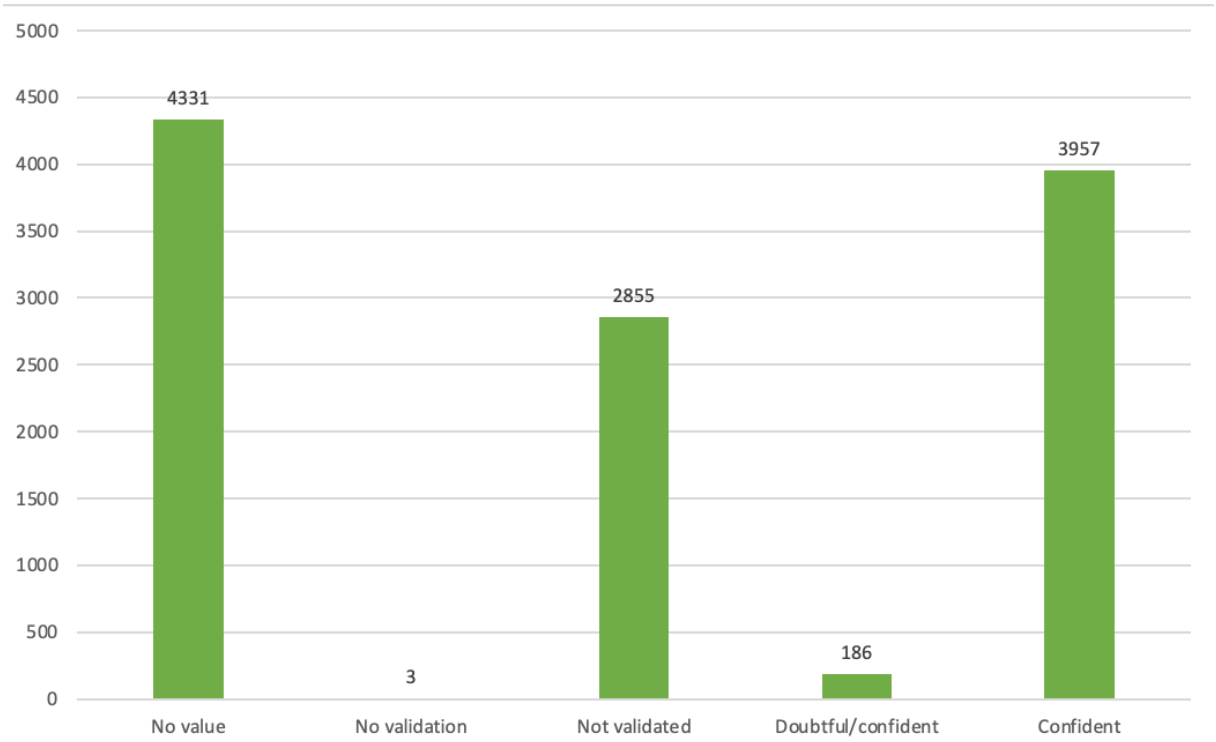

Figure 10: The number of PSMs from the original dataset are distributed across the five categories in PeptideShaker.

As we can see, almost no potential PSMs are put in the No Validation category, and very few are put in the Doubtful category. For the rest of the testing procedures, our decision was to merge categories. We decided to merge category -2 and -1, since they practically mean the same, and category 1 and 2, because a significant part of the potential PSMs in category 1 have

a high enough score to be considered validated by PeptideShaker overall. Subsequently it also becomes easier to handle the categories when there are only three instead of five.
Table 3 shows the distribution across the new categories.

CATEGORY	FREQUENCY
NO VALIDATION	4,334
NOT VALIDATED	2,855
CONFIDENT	4,143
TOTAL:	**11,332**

Table 3: The number of PSMs from the original dataset distributed across the three new categories we will be working with.

4.5 Generation of subsets

Subset generation is done in three different ways later in this thesis. For the first approach, subsets were generated using SearchGUI's built-in function of splitting datasets into smaller parts. This built-in function splits the file chronologically and does so by iterating over the file and putting a pre-selected number of spectra in each file.

In the two other approaches, the generation of subsets were done by manually coding the subset generation itself in Python. This was done by utilizing Pyteomics [34, 35], a collection of handy tools and software for handling various proteomics data. Especially the collection's mgfRead and mgfWrite functions came in handy.

For the randomized approach, the following was done. The process includes importing the original dataset file, indexing each spectrum, and then creating a random list of indexes, where the random indexes indicate which spectra to add to the subset file. This creates a file of spectra with random indexes, and the desired number of spectra in the file can easily be decided by setting the length of the index list to your desired number of spectra to include in the file.

For the n-th approach, the procedure of creating the subsets was again done in Python, with the help of Pyteomics. The MGF file is first imported and indexed accordingly by the mgfRead function from Pyteomics. From there, the length of the entire file is counted, followed by a process where every n-th spectra gets written to the new subset file. The size of the new subset file depends on the n-value selected.

4.6 Code for subset generation

The Python scripts and code used to generate subsets, both by random indexing and by selecting a specific *n*-value, are made available at GitHub (https://github.com/barsnes-group/automatic-subset-selection).

5. Results

In this project, the approach is slightly different from the existing providers of parameter optimization. We will start by extract a subset of spectra from the original spectrum file, greatly reducing the runtime and computational resources necessary for a search, and then check possible combinations of parameters, hopefully returning a selection of parameters which can then be used to search the entire spectrum file to achieve the best possible result for the given sample. This can help to improve the accuracy and reliability of the protein identification results.

To our knowledge there is no open source and freely available implementation that allows a user to carry out such an optimization of their own proteomics searches, especially not for up to ten search engines at the same time, and across all of the parameters considered here.

5.1 Parameter selection

Throughout this thesis, SearchGUI and PeptideShaker were the most frequently used tools. A screenshot of SearchGUI is provided in **Figure 11**. At the top of the user interface, the user can adjust the search settings, import spectrum and database files, and decide where the output should be placed. Under that one has the option to convert raw files, and beneath that again is the list of proteomics search engines available in the tool. Depending on the operating system of the computer in use, some of the search engines may not be supported, as shown by the operating system logos highlighting which OS the search engine supports.

Figure 11: Screenshot of SearchGUI. The greyed-out proteomics search engines are search engines that cannot be used on the current operating system.

The next step is choosing the common parameters. For the post-translational modifications, it was decided to go for the most common ones: carbamidomethylation of cysteine (C) as a fixed modification, and oxidation of methionine (M) as a variable modification. Carbamidomethylation of C is often considered a fixed modifications because it is introduced to prevent the cysteine residues from (re-)forming disulphide bonds with one another, while the oxidation of M is commonly considered a variable modification because it can occur spontaneously during sample preparation, *e.g.*, due to the methionine residues frequently reacting with the oxygen in the ionization source environment.

The next parameters are digestion, enzyme, and specificity. These parameters should match what was done in the sample preparation stage of the experiment. Max missed cleavages is set to 2. Increasing this parameter makes the search space increase as well, and since proteases sometimes do not cleave the proteins perfectly, it is common to set this parameter to either 1 or 2.

Precursor and fragment tolerance are two highly customizable parameters. We have set them to 10 parts per million (ppm) precursor tolerance and 0.02 Dalton (Da) fragment tolerance. Precursor tolerance determines the search space, which should be stringent, but broad enough to have several entries per search space (*e.g.*, for *e*-value calculation). 5-10 ppm is commonly used for data acquired on well-calibrated MS instruments. Fragment tolerance is the distance we allow between the theoretical and the experimental fragment masses. Fragment tolerance should also be stringent but provide enough flexibility for statistical assessment.

The precursor charge is set to 2 to 4. If one has knowledge of what charge states are included in your data, it is possible to save time by defining a range of just those. The fragment ion types, and isotope range are kept at their default values. **Figure 12** illustrates the resulting default parameter setup.

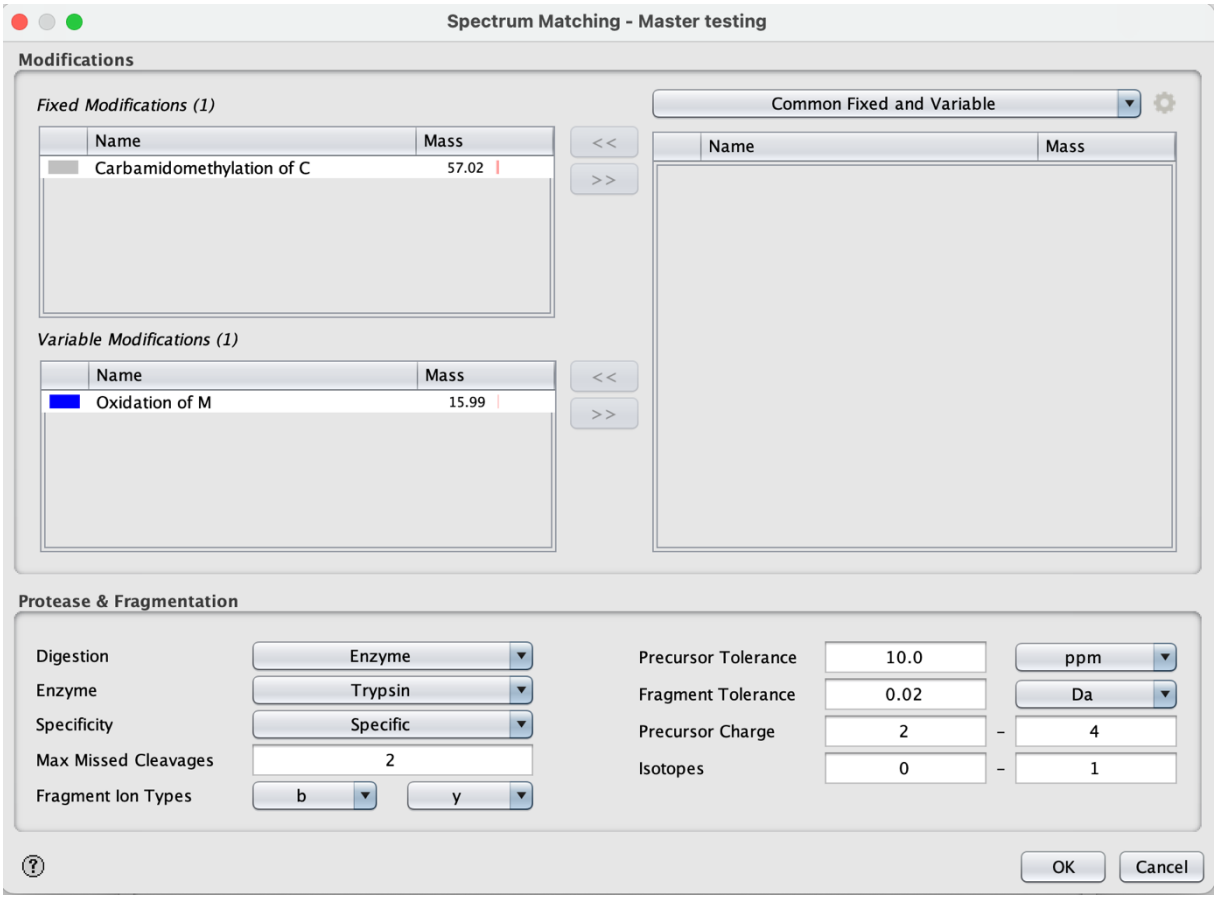

Figure 12: Overview of the common default parameters used in this project.

5.2 Generating subsets

In the background section it was mentioned that proteomics datasets are rapidly growing in size. Optimizing parameters on such large datasets takes a substantial amount of time and computational resources. To address these limitations, the concept of generating a subset from the original file is investigated, aiming to mitigate these constraints while preserving the essential properties of the initial dataset.

Three approaches were considered for extracting a subset of the original spectrum file: i) split the original file chronologically into smaller files and select one of these, ii) randomly pick x spectra from the original file, and ii) pick every n-th spectra from the original file, creating a file of desired size depending on the n value. The three approaches are illustrated in **Figure 13**.

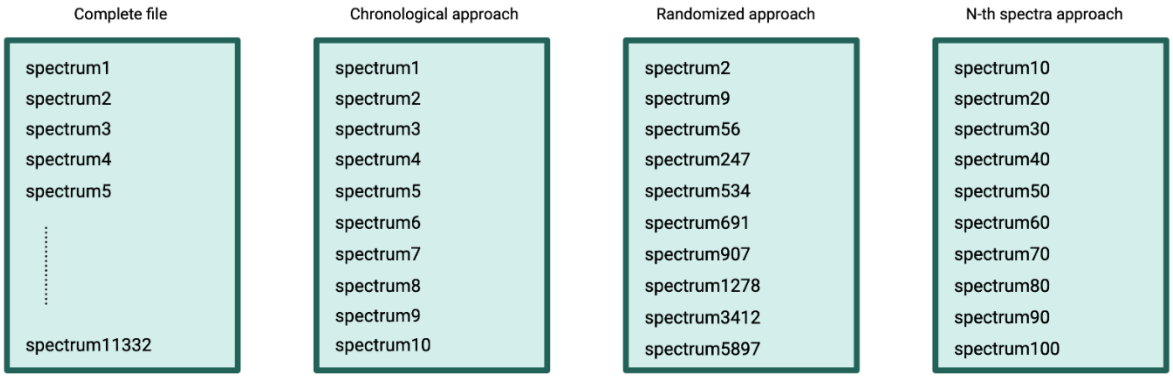

Figure 13: The three approaches used when generating a subset dataset file.

Due to the structure of MGF files, splitting the file into multiple smaller files is a straightforward process. The spectra are organized one at a time, making it easy to extract one or multiple spectra without affecting the rest of the file or the other spectra.

SearchGUI's built-in feature for splitting large MGF files in a chronological order was utilized for the first approach. The decision was to divide the dataset containing 11,332 spectra into 11 smaller files, each file containing 1000 spectra, except the last one, which includes the remaining 132 spectra in addition. Consequently, the last file was slightly larger than the others, containing 1132 spectra in total.

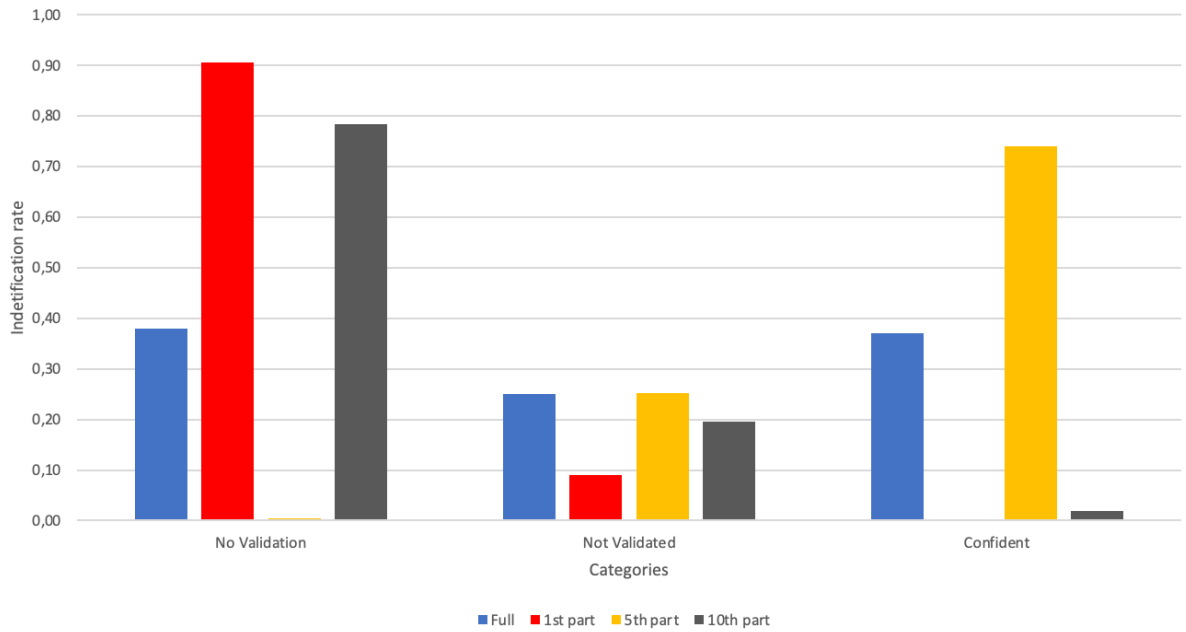

Figure 14: What categories the potential PSMs are put in for three datasets from different parts of the file, compared to the complete file.

As mentioned earlier, the quality and abundance of spectra at the start and at the end of mass spectrometry data are considered low. Therefore, when running searches on the 11 files created, the first file gave an exceptionally low number of confident PSMs, as illustrated in **Figure 14**. The file containing the spectra at the end of the original file gave the same result, a very low number of confident PSMs. However, for the file containing the middle section of the original file, the number of confident PSMs were high. Almost 3/4 of the spectra present in the file were an identified match to a peptide.

This becomes an issue because the files that only get a handful of PSMs out of 1000, obviously cannot be used. And the files in the middle that are close to a 75% identification rate are too good. The numbers are artificially high in terms of representing the original dataset. This led to the next approach: randomizing which spectra to put in the subset file.

5.3 Random selection

Random selection is a widely employed statistical approach. Given the limitations associated with the chronological approach, randomization was implemented to select spectra at random from the complete file.

Randomizing and selecting which spectra to put in the new file were done in Python, with the help of Pyteomics [34, 35]. The process includes importing the original dataset file, indexing each spectrum, and then creating a random list of indexes, where the random indexes indicate which spectra to add to the subset file. This creates a file of randomly selected spectra from the complete file, and the desired number of spectra in the file can easily be decided by setting the length of the index list to the desired number of spectra.

During the testing of the randomized approach, ten subsets of 10% of the size of the original dataset were created. This percentage was used because it sped up the search time significantly, while still including a solid number of spectra, namely 1134.

The randomized subset performed better than the chronological ones in terms of mimicking the original dataset, **Figure 15** shows the identification rate of ten separate randomized subsets containing 1134 spectra. It did not perform better than the subset containing spectra from the middle part of the original dataset, as it had an identification rate of almost 75%. However, in terms of keeping the properties of the original dataset, it is a close match. Randomizing the subset gave roughly the same identification rate as the original dataset, with a few deviations. The original dataset had an ID rate of 36,4%.

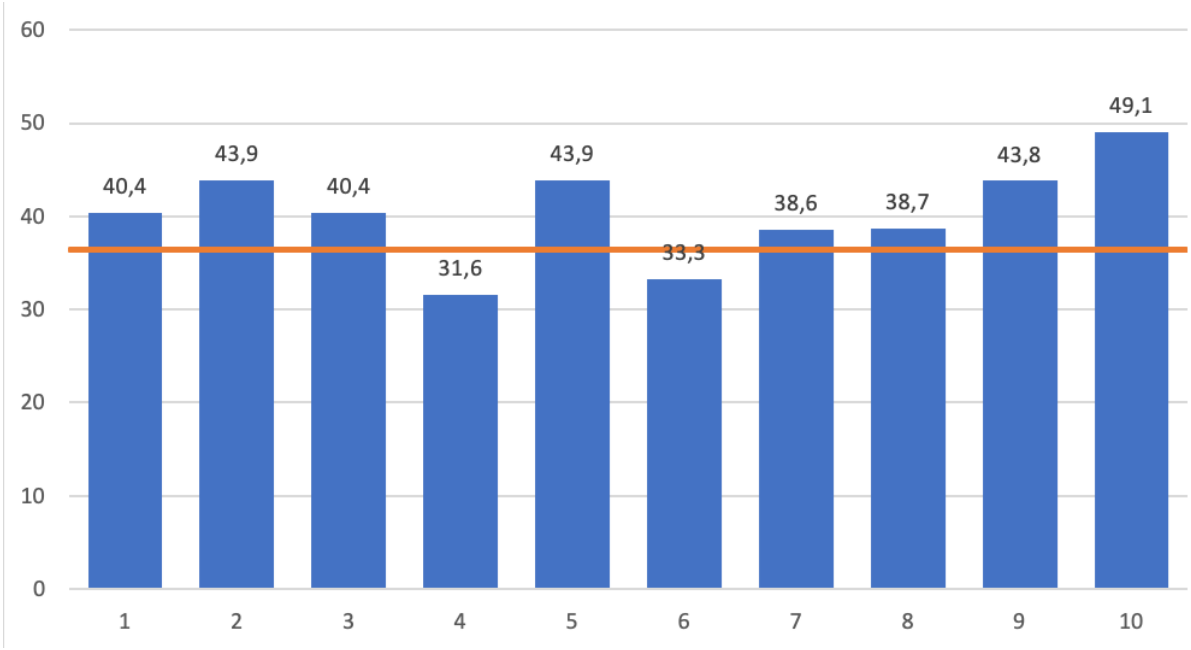

Figure 15: ID rate of ten randomized subsets. The identification rate of ten separate randomized subsets with 1134 spectra in each. The orange line indicates the identification rate of the original dataset, i.e., 36,4%.

As we can see, the randomized subset has more or less similar ID rates as the original datasets, with some smaller deviations, and a few larger ones. The last subset for example, gives an identification rate of nearly 50%, indicating that it probably contains more spectra from the middle of the file compared to, for example, Subset 4, which has an identification rate of only 31,6%.

5.4 Picking every n-th spectrum

In order to deal with the complications of randomization, a third option was explored. Taking only a set percentage of the file and picking the spectra by extracting every *n*-th spectrum from the file. This ensures a consistent number of spectra from every section of the file. In this approach, the main task is selecting an appropriate value for *n*. Different values were evaluated, namely 10, 50, 100 and 200. The results can be seen in **Figure 16.**

The procedure of creating the subsets was again done in Python, with the help of Pyteomics. The MGF file is first imported and indexed by the mgfRead function from Pyteomics. From

there, the length of the entire file is counted, followed by a process where every *n*-th spectra gets written to the new subset file. The size of the new subset file depends on the selected value of *n*.

After testing the different values of *n*, a universal percentage of 10 was chosen, meaning that ten percent of the original file will be written to the subset file, in every tenth spectra fashion. Selecting a smaller *n* reduces the runtime further, but not by a great amount compared to a subset consisting of ten percent of the spectra. For instance, running a search on the complete dataset takes 2 minutes and 28 seconds with MS-GF+, and running it on datasets generated with n-values of 10 and 200 takes 25 and 18 seconds, respectively. **Figure 17** shows the results of the different values of *n* compared to the chronological approach.

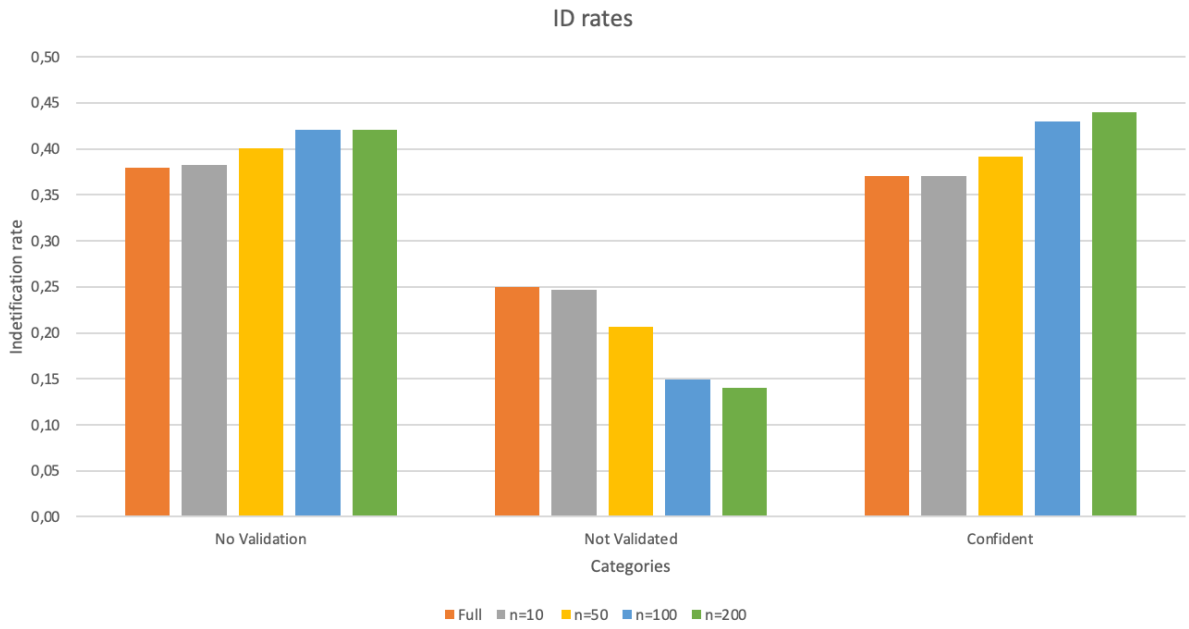

Figure 16: Testing different n-values. The percentage of spectra in each file in each category with different values of n. The orange column represents the complete original dataset. The other colors each represent a different value of n, as seen in the figure.

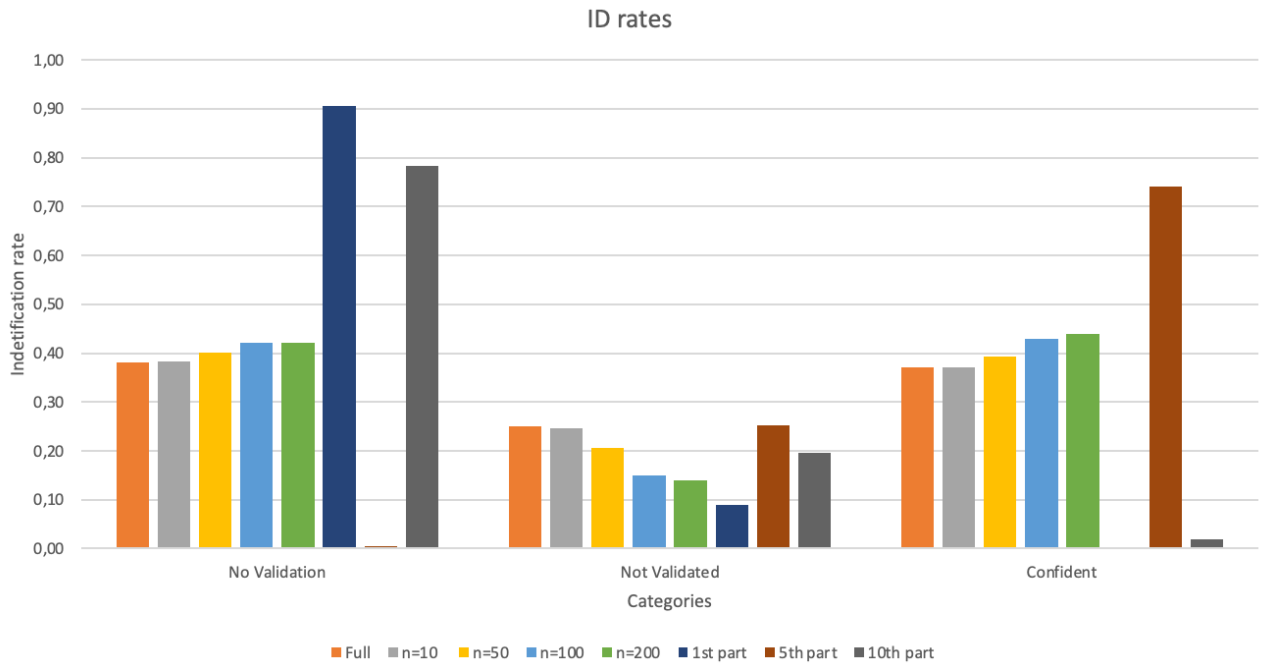

Figure 17: Comparing the chronological approach to every n-th spectrum approach. The percentage of spectra in each file ending up in each category. The "1st part", "5th part" and "10th part" indicate the subsets with spectra from the start, middle and the end of the original file. The orange column represents the complete original dataset.

The figure clearly shows, as stated earlier, that the chronological approach has limitations in terms of the low abundance spectra at the start. It also shows that the larger the value of *n* (*i.e.*, the smaller the percentage selected from the original file), the further away the results are from the properties of the original file. In addition, it backs up our choice to stick with selected percentage of 10, which corresponds to a value of $n = 10$, because it is the subset that is closest to the original in terms of identification rate and how many PSMs were placed in each category.

5.5 Testing different general parameters

The more common parameters of the search engines, namely the ones already implemented for all of them in SearchGUI were tested next. This was conducted on multiple parameters, like precursor charge, max missed cleavages, fragment charge, cleavage enzyme, and digestion. An example of the findings is shown in **Figure 18**.

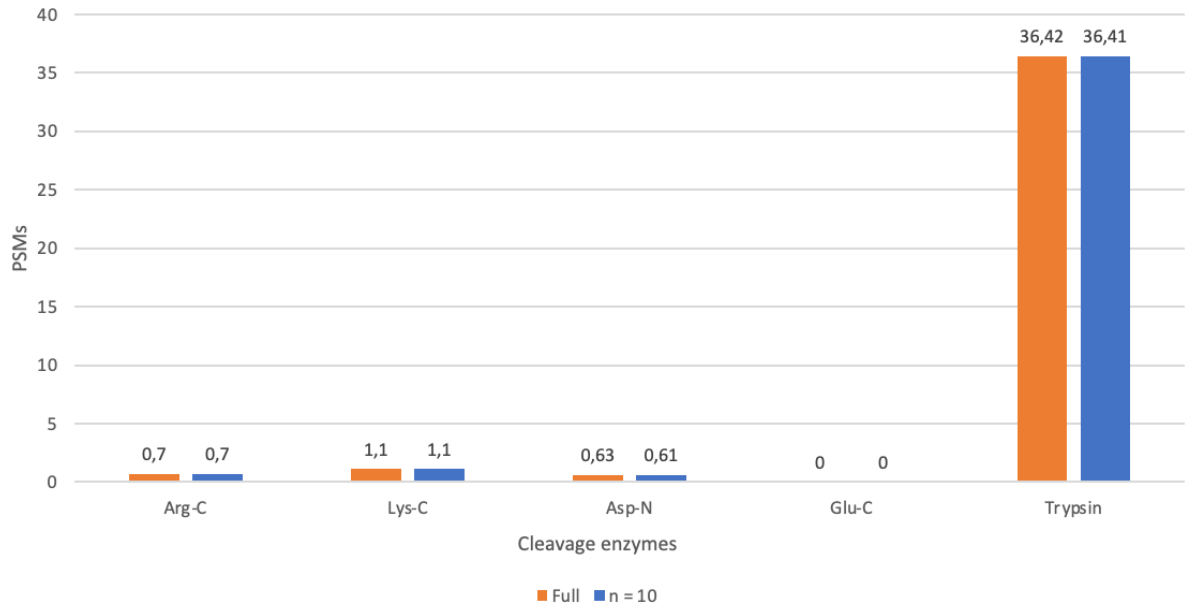

Figure 18: Identification rates in percent of the complete file and the subset file generated with n = 10 when changing the cleavage enzyme.

The results tell us that changing the cleavage parameter on the smaller subset has an effect, both on the smaller subset and on the complete file. It becomes clear that running a search with the wrong cleavage enzyme can have severe consequences. For the other parameters mentioned in the paragraph above, the same trend is present. In other words, changing the parameter on the small subset has the same effect when changed on the complete file and some of the parameters have a larger impact on the result than others. For example, changing the fragment ion types parameter, barely changes the rate of identification, as seen in **Figure 19**.

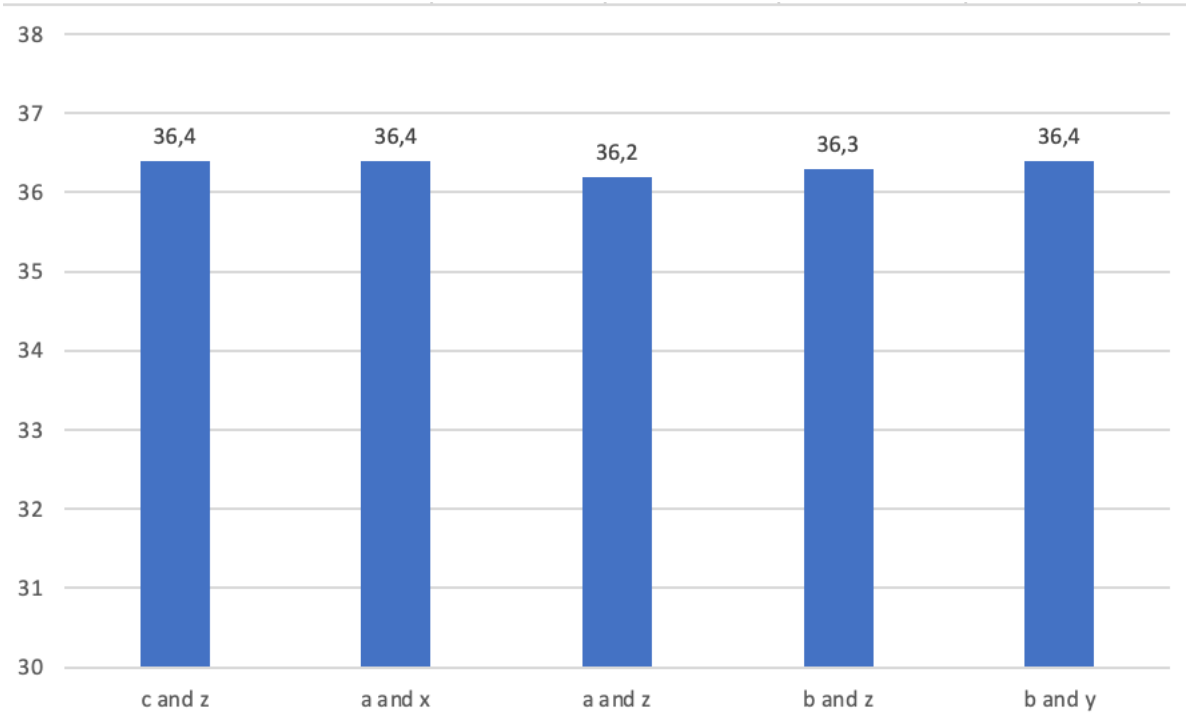

Figure 19: The different fragment ion type parameter combinations resulting in almost no change in the identification rate.

5.6 Testing advanced parameters

When testing the advanced parameters in the different proteomics search engines we used the MS-GF+ search engine, mainly because many of the search engines' lists of parameters are very long and it would make the initial testing process very time consuming. In MS-GF+, the parameter list however seemed manageable. The parameters of the MS-GF+ search engine is listed in **Table 2** in the Background section.

From the overview of the MS-GF+ parameters it became clear that some of these will not have any effect on the result at all. For example, parameters such as Number of Tasks only affects the time and computational resources used when executing the search, and it will not influence the resulting output. On the other hand, parameters such as Peptide Length clearly influences the result. For example, if this parameter is set to an unusually low number, *e.g.*, from 5 to 10, the MS-GF+ search engine will not consider peptides of lengths longer than 10, whereas most peptides detected by MS will be in the 6 to 30 amino acids range.

The remaining MS-GF+ advanced parameters are the MS/MS Detector, the Fragmentation Method, and the Protocol. All of these are limited in the sense that they have a finite number of choices. However, they might still affect the result.

First, tests on the three parameters MS/MS Detector, Fragmentation Method, Protocol were conducted, where all of the choices for the three parameters were tested on the complete dataset one at the time. The results can be seen in **Figure 20**. All of the common parameters were set to the values mentioned earlier in this section, and the other search engine specific parameters were kept at their default values.

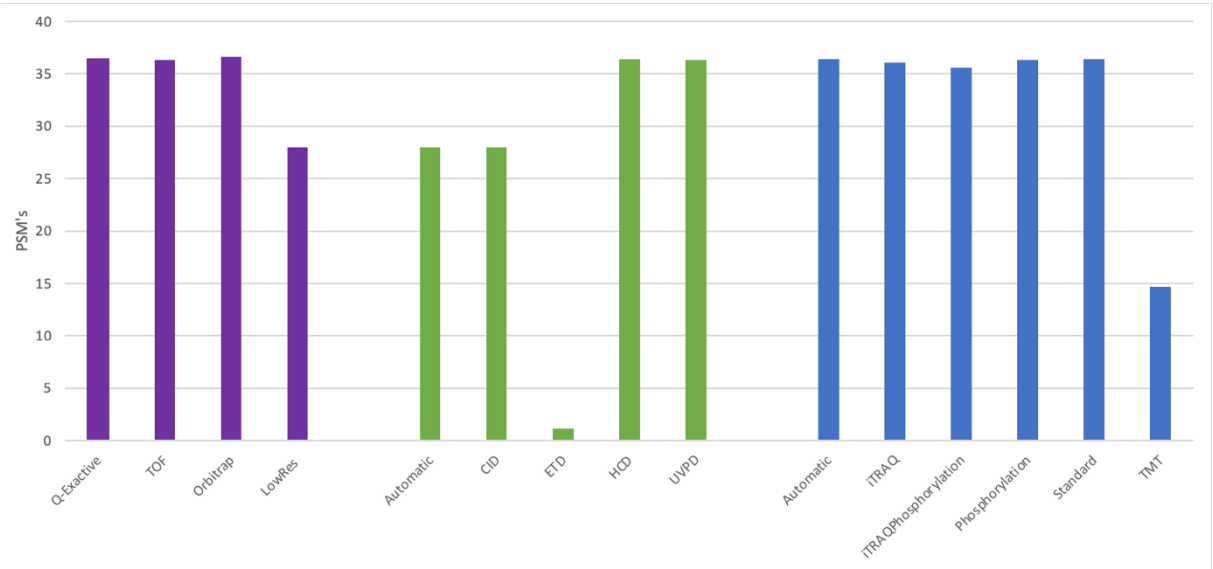

Figure 20: The amount of confident PSMs, in percent, for different parameter settings for MS-GF+. Purple columns: the MS/MS Detector, green columns: the Fragmentation Method, and blue columns: the Protocol. The leftmost purple column named Q-Exactive is the default MS/MS Detector in MS-GF+, the green CID column is the default Fragmentation Method, and the blue Automatic column is the default Protocol.

From **Figure 20** it becomes clear that changing the parameters has a small effect in most of the cases, however, if Fragmentation Method is wrongly set to ETD, we can see a massive drop in PSMs. It becomes clear from the test results that Fragmentation Method is the parameter that overall suffers the most from incorrect parameter setting out of the three. In terms of PSMs, the most stable parameter seems to be the Protocol, however, when the Protocol parameter is incorrectly set to TMT, the number of PSMs decreases by almost 60% compared to the number of PSMs when using the default parameters.

From these results, one can obviously tell that the parameter setting for these three have an effect on the result, as will some of the other parameters such as peptide length, as mentioned earlier. Next, it was tested whether the same effect could be seen for the complete dataset.

This was tested by changing the same three parameters on the subset containing 10 percent of the spectra from the original file, the subset created by using a value of $n = 10$. The common parameters were kept the same, and only Fragmentation Method, MS/MS Detector and Protocol were changed as for the entire dataset. The results are shown in **Figure 21**.

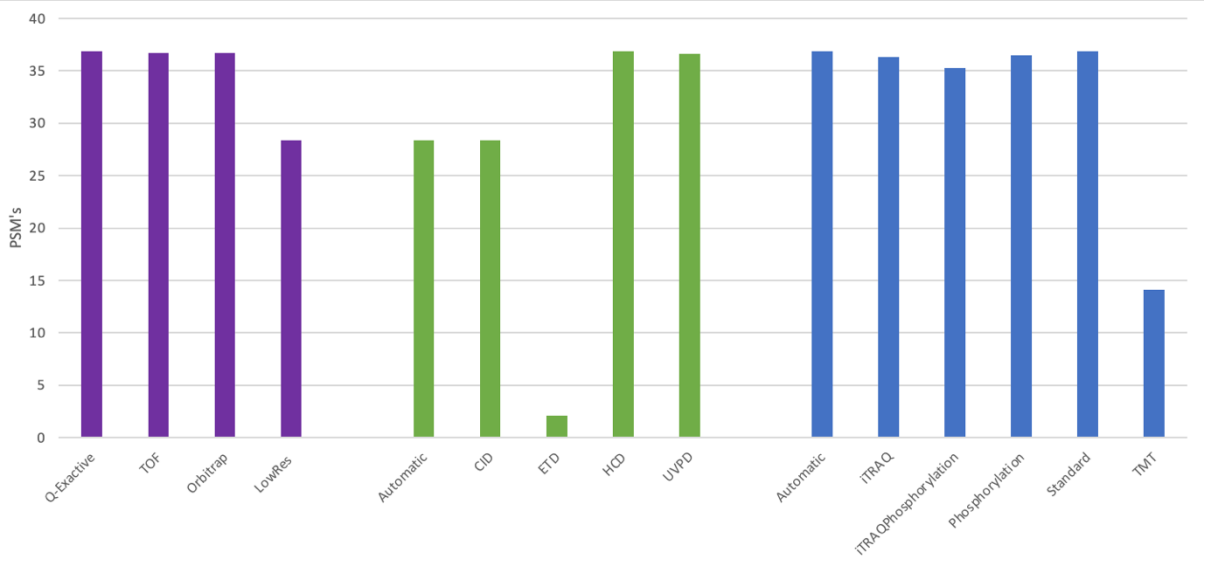

Figure 21: The number of PSMs, in percent, when changing the same parameters on the subset containing 10 percent of the spectra from the original dataset. Purple columns: MS/MS Detector, green columns: Fragmentation Method, and blue columns: Protocol.

Figure 21 shows that the number of PSMs indeed follows the same trend in the subset as in the original dataset. It also shows that the subset with a value of $n = 10$ mimics the original dataset well, and that changes in these parameters seem to affect both datasets in the same way.

5.7 Testing the n-th spectra selection on other datasets

Further testing of the *n*-th spectra selection with a value of $n = 10$ was done on three random datasets taken from the PRIDE archive. This testing was done to ensure that the results from the selected dataset were not a one-time case. It is worth mentioning that in this case, the

parameters were set to match that of the experiment, and in other words not run with the default parameters. The results can be found in **Figure 22**.

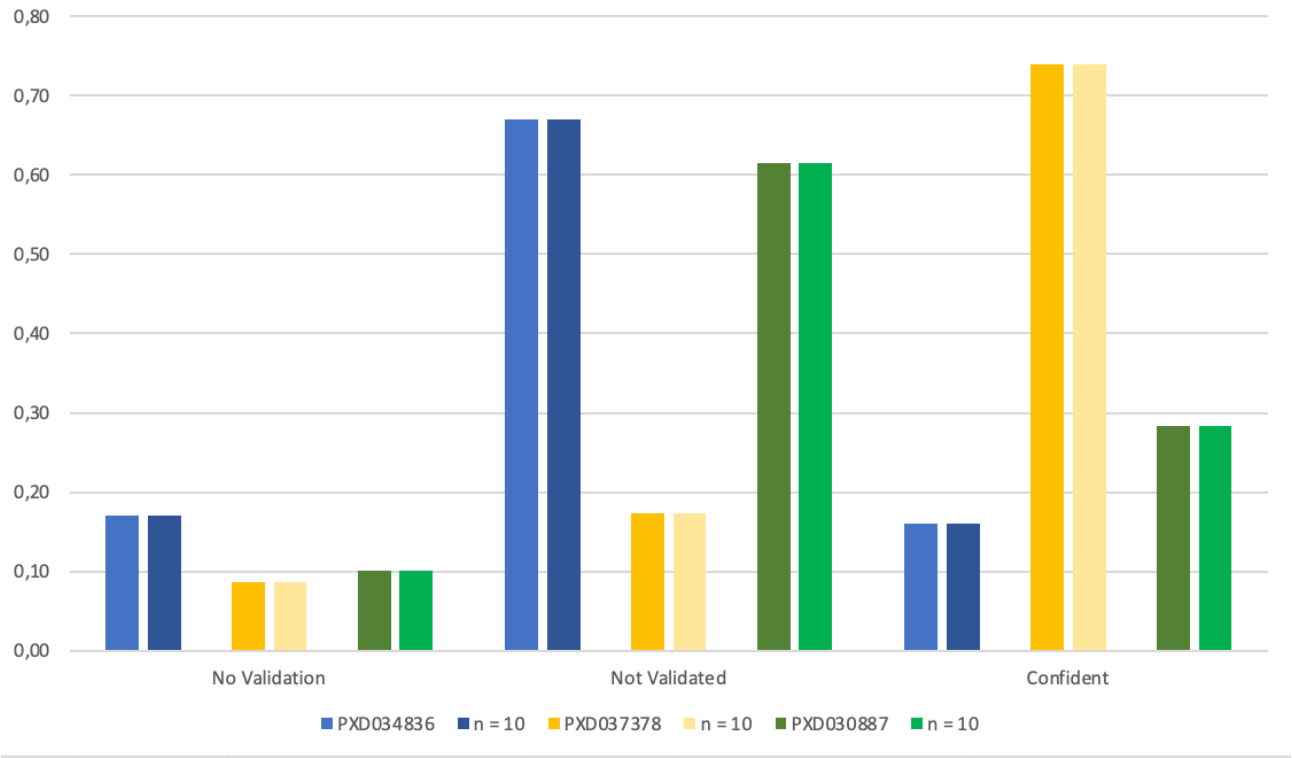

Figure 22: Testing other datasets. The identification rates in terms of which spectra are placed in which category. The different colors represent the different experiments from PRIDE, and the shade represents the subset of the dataset generated with a value of n = 10.

As **Figure 22** clearly shows, generating a subset of the original datasets by picking every tenth spectra so that you get 10 percent of the original dataset in the subset, still represents the original dataset well. There are only minor differences in terms of identification rates and categorization between the original datasets and their corresponding subsets, meaning that selecting $n = 10$ when generating subsets seems to retain the properties of the original dataset.

5.8 Manual optimization of parameters

Next, the concept was tested blindly on a dataset. In other words, a dataset from PRIDE was selected where the parameters used in the experiments are not looked at, and from there try to find the best parameters by testing different combinations. The acquired dataset contained

66,444 spectra(PXD030822) and was one of several spectrum files from a study on liver cells [36].

The process started with the default parameters of both the common parameters and the MS-GF+ search engine specific parameters. The advanced search engine specific parameters were at first left unchanged, only trying to customize the common parameters.

Firstly, the dataset was downloaded, and a non-random subset with an *n*-value of 10 was generated using the same Python code as earlier. The complete dataset gave 18,248 PSMs when run with SearchGUI's default parameters, with an identification rate of 27,5%. Subsequently, testing was performed on different parameters selections where the values were systematically both increased and decreased from their default values in order to observe the effect. Thereafter, the result was compared to the actual parameter values used in the PRIDE dataset. The number of PSMs from the testing is shown in **Figure 23**.

It is worth mentioning that when observing the parameters used in the experiment, there were more post-translational modifications included compared to the standard two PTMs used in this thesis. The experiment parameters were run without the extra PTMs included as modification parameters, to ensure that the PTMs did not affect the outcome of the testing.

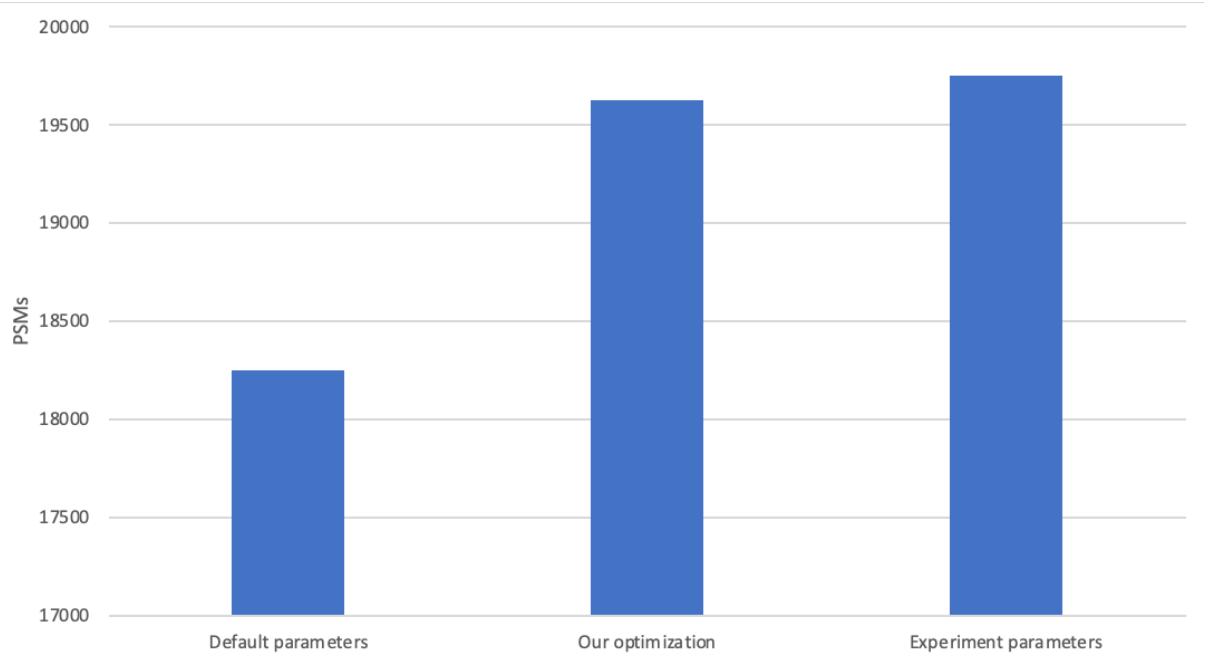

Figure 23: The amount of PSMs after searching with MS-GF+ with default parameters, our own manual optimization, and with the parameters used in the experiment. The experiment parameters were run without the additional non-standard PTMs.

The figure shows that testing enough parameters can indeed yield a better result than using the default parameters. The best parameter selection resulted in an increase of almost 1500 PSMs. Obviously, the testing also yielded a lower number of PSMs for certain parameter values, however, this is of course not the result aimed for.

Compared to the parameter values used in the experiment from which the dataset was acquired from, our parameter values are different. However, they do not deviate a whole lot from the experiment parameters, as shown in **Table 4**.

Parameter	Default value	Optimized value	Experiment value
Fixed modification	Carbamidomethylation of C	Carbamidomethylation of C	Carbamidomethylation of C
Variable modification	Oxidation of M	Oxidation of M	Oxidation of M
Digestion	Enzyme	Enzyme	Enzyme
Enzyme	Trypsin	Trypsin	Trypsin
Specificity	Specific	Specific	Specific
Max missed cleavages	2	1	2
Fragment ion types	*b* and *y*	*b* and *y*	*b* and *y*
Precursor tolerance	10.0 ppm	5.0 ppm	4.5 ppm
Fragment tolerance	0.02 Dalton	0.01 Dalton	20 ppm
Precursor charge	2 to 4	2 to 4	N/A
Isotopes	0 to 1	0 to 1	N/A

Table 4: Table showing the default parameter values, compared to the manual optimization and the parameter values used in the experiment itself.

6. Discussion

In proteomics today, only a limited number of software tools focus on optimizing parameters, and they only explore a small subset of the possible parameter combinations. Exploring all possible parameter combinations would require a lot of time and computational resources. Testing all parameter combinations is only an option if the datasets are very small, however, proteomics datasets are rather continuing to grow in size [37].

The results presented in this thesis indicate that selecting a representative subset of the complete dataset, while also keeping the properties of the original dataset, can be an effective approach for testing and optimizing multiple parameters at the same time. However, there are some remaining challenges that has to be considered.

6.1 Number of PSMs as a test metric

In this thesis, it was decided to measure and compare the result based on how many PSMs were found out of the total number of spectra in the original file, *i.e.*, the identification rate. This came from the fact that usually the desired goal is to get as much as possible out of the valuable biological data from the spectrum files. Hence, it is preferable to obtain a greater number of spectrum matches, particularly those of good quality, in order to determine the presence of proteins in the samples.

There is however uncertainty regarding whether this is the correct test metric or not. There may be other ways of testing the quality of the results other than PSMs, for example, looking at the distribution of scores, or looking at how the PSMs are distributed into each category. Furthermore, it could also be possible to evaluate the quality of each peptide-spectrum match and employ this information as a metric for testing purposes.

6.2 Possible overfitting

In terms of optimizing the parameters on a smaller subset, one has to be careful with overfitting. Since we tailor the parameters on a smaller subset, there is a chance that the parameters get optimized on the small subset provided, and therefore fail to recognize possible PSMs on the complete dataset when the parameters have been tailored.

6.3 Spectrum quality

In terms of the quality of the spectra produced by mass spectrometry, there is an ongoing discussion on what to label as a good or bad spectrum. As already mentioned, the start and the end of the spectrum file contains low number of spectra, however there is uncertainty regarding whether these spectra should be considered as bad spectra, because they only contain a relatively low number of peaks. Even though these spectra contain a very low number of peaks, mass spectrometers today are sensitive enough to detect these and return a hit, even though they may be irrelevant.

6.4 Expanding to other proteomics search engines

In SearchGUI, and in general, there are a lot of proteomics search engines to choose from when facilitating peptide identification. Throughout this thesis, testing was done using the MS-GF+ search engine, mainly due to its manageable list of advanced parameters. However, the concept of subset generation and parameter optimization is not done by the search engines themselves, and the results should therefore easily be transferable to other search engines.

For example, since one of the aims is to reduce the search time, one could perform testing in one of the reportedly faster proteomics search engines, like Sage [38]. However, one limitation is that Sage exclusively supports mzML, which presents a challenge in the subset generation process given the specific structure of mzML files.

6.5 Potential consequences of randomizing

The generation of subsets by randomizing the selection of spectra proved to be a better alternative than doing it chronologically. However, there is a risk connected to randomizing. One can be unlucky and pick many spectra from either the start or end of the original file, or both, resulting in a file which would yield a low amount of PSMs. In turn, one can also end up only picking spectra from the middle part of the file, yielding an artificially high number of PSMs. It is of course unlikely that such a situation will occur, but there is still a possibility of this happening when randomizing. Hence, considering the potential consequences, an alternative approach was deemed more suitable.

6.6 Choosing an appropriate value of n

When generating subsets, the *n*-th approach showed the best results in terms of representing the original dataset. The decision was to pick a value of 10, meaning every tenth spectra from the file. Selecting this number was logical due to the significantly reduced search time and because every tenth spectrum is the same amount as 10% of the complete spectrum file. Selecting 10% of the file also appears to be a reasonable choice, because it still maintains the main properties from the original dataset. The test results showed that the dataset generated with an *n*-value of 10 best represented the original file, while still reducing the time enough such that testing multiple parameters was feasible.

It may however be possible to select an even greater *n*-value, *i.e.*, a smaller percentage of the file, on larger datasets, while still keeping the main properties of the complete dataset. Selecting 10% worked well on the datasets used in this thesis but a lower value could be just as viable on other datasets. Furthermore, there are possibilities to reduce the runtime and computational resources even further, by increasing the *n*-value, subsequently selecting a much lower percentage of spectra from the file. However, as shown in the results, doing so has its limitations because a larger *n*-value will represent the original dataset more vaguely than what a subset generated with a lower *n*-value would do.

6.7 Testing combinations of parameters

When conducting testing of the parameters themselves in this thesis, the testing was done manually, and by changing one parameter at a time. This approach is the easiest to test, however, there might be interactions between parameters that has an effect on the result. Hence, there might be some overlooked optimizations of the parameters since combinations of them have not been tested.

6.8 Expanding to more data

On majority of the testing conducted in this thesis , a single example dataset was used. There is therefore a possibility that the approach only works on this particular dataset, or on datasets with certain properties. The three randomly selected datasets from PRIDE however provided further support that picking every tenth spectra seems to mimic the original dataset in a sufficient way.

7. Future work

Due to the limited time frame of the thesis, various tests and experiments have been deferred for future investigation.

MS-GF+ was chosen due to its simpler set of customizable parameters compared to other search engines. The extensive parameter lists of other engines would have prolonged the testing process. However, this doesn't mean that the concept shouldn't be tested on other proteomics search engines. Further investigation is necessary to explore parameter optimization by generating subsets with n-th spectra.

It is highly advisable to conduct a comprehensive evaluation of the n-th spectra selection concept on a wider scale. A recommended approach would be to perform extensive testing by obtaining a diverse range of datasets and systematically applying the concept to test each of them.

Automation of the testing procedure should be considered, for example as a command line pipeline, to make testing much faster and make it possible to get even more accurate results and parameter recommendations. Ideally, the entire process should be implemented into an existing framework such that the user can generate subsets, optimize parameters on them, and apply those parameters to the complete dataset, all on the same platform.

8. Conclusion

This master's thesis aimed to investigate the optimization of proteomics search engine parameters by focusing on subsets of the data instead of the entire spectrum file. Through comprehensive analysis and experimentation, the study demonstrated promising results and shed light on the potential benefits of the approach.

The main findings highlight the effectiveness of optimizing search engine parameters on subsets of data. By dividing the spectrum files into manageable subsets that still maintain the main properties of the complete dataset, it is possible to optimize search engine parameters specifically tailored to each dataset. This strategy not only improved the overall efficiency of the search process but also enhanced the quality and accuracy of protein identifications.

By reducing the computational burden and streamlining the search process, subset-based optimization can significantly enhance the efficiency and scalability of proteomics analyses. This is particularly relevant in large-scale studies involving vast amounts of data, where traditional approaches may become prohibitively time-consuming and computationally demanding.

While the findings are encouraging, there are still avenues for further exploration. Future research could focus on refining the subset selection criteria, exploring different ways to divide the data, and investigating the generalizability of the optimized parameters across diverse datasets. Additionally, the integration of machine learning algorithms and advanced statistical techniques could provide valuable insights into the optimization process and enable more sophisticated parameter tuning strategies.

9. Acknowledgements

I extend my heartfelt gratitude to my supervisors, Prof. Harald Barsnes and PhD student Yehia M. Farag for their invaluable guidance and support throughout this thesis. Their expertise and unwavering commitment to my academic growth, their constructive feedback, and their encouragement have been instrumental in shaping this research.

To my girlfriend Kristin, your constant support, patience, and love have been the pillars of my success. Your presence in my life has brought joy, love, and balance to every aspect of my journey. Your belief in me, even during moments of self-doubt, has pushed me to overcome obstacles and strive for excellence.

I am deeply grateful to my family and friends for their constant support and understanding. Their belief in me and their encouragement have been a source of strength during this challenging journey.

Finally, I would also like to extend my gratitude to the staff at the Proteomics Unit at the University of Bergen (PROBE) for providing an enriching academic environment and for providing the necessary resources for my research.

10. References

1. Barsnes, H. and M. Vaudel, *SearchGUI: A Highly Adaptable Common Interface for Proteomics Search and de Novo Engines.* J Proteome Res, 2018. **17**(7): p. 2552-2555.
2. Vaudel, M., et al., *PeptideShaker enables reanalysis of MS-derived proteomics data sets.* Nat Biotechnol, 2015. **33**(1): p. 22-4.
3. Aslam, B., et al., *Proteomics: Technologies and Their Applications.* J Chromatogr Sci, 2017. **55**(2): p. 182-196.
4. Dau, T., G. Bartolomucci, and J. Rappsilber, *Proteomics Using Protease Alternatives to Trypsin Benefits from Sequential Digestion with Trypsin.* Anal Chem, 2020. **92**(14): p. 9523-9527.
5. Han, X., A. Aslanian, and J.R. Yates, 3rd, *Mass spectrometry for proteomics.* Curr Opin Chem Biol, 2008. **12**(5): p. 483-90.
6. Aebersold, R. and D.R. Goodlett, *Mass spectrometry in proteomics.* Chem Rev, 2001. **101**(2): p. 269-95.
7. Eng, J.K., et al., *A face in the crowd: recognizing peptides through database search.* Mol Cell Proteomics, 2011. **10**(11): p. R111 009522.
8. Barh, D. and V. Azevedo, *Omics technologies and bio-engineering : towards improving quality of life*. 2017, Amsterdam: Academic Press.
9. Verheggen, K., et al., *Anatomy and evolution of database search engines-a central component of mass spectrometry based proteomic workflows.* Mass Spectrom Rev, 2020. **39**(3): p. 292-306.
10. Dass, C., *Fundamentals of Contemporary Mass Spectrometry*. 2007: Wiley.
11. Frank, A.M., *Predicting intensity ranks of peptide fragment ions.* J Proteome Res, 2009. **8**(5): p. 2226-40.
12. Macias, L.A., I.C. Santos, and J.S. Brodbelt, *Ion Activation Methods for Peptides and Proteins.* Anal Chem, 2020. **92**(1): p. 227-251.
13. Apweiler, R., A. Bairoch, and C.H. Wu, *Protein sequence databases.* Curr Opin Chem Biol, 2004. **8**(1): p. 76-80.
14. UniProt, C., *UniProt: the Universal Protein Knowledgebase in 2023.* Nucleic Acids Res, 2023. **51**(D1): p. D523-D531.

15. Perez-Riverol, Y., et al., *The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences.* Nucleic Acids Res, 2022. **50**(D1): p. D543-D552.
16. Choi, H. and A.I. Nesvizhskii, *False discovery rates and related statistical concepts in mass spectrometry-based proteomics.* J Proteome Res, 2008. **7**(1): p. 47-50.
17. Kim, S. and P.A. Pevzner, *MS-GF+ makes progress towards a universal database search tool for proteomics.* Nat Commun, 2014. **5**: p. 5277.
18. Perkins, D.N., et al., *Probability-based protein identification by searching sequence databases using mass spectrometry data.* Electrophoresis, 1999. **20**(18): p. 3551-67.
19. Eng, J.K., A.L. McCormack, and J.R. Yates, *An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database.* J Am Soc Mass Spectrom, 1994. **5**(11): p. 976-89.
20. Eng, J.K., T.A. Jahan, and M.R. Hoopmann, *Comet: an open-source MS/MS sequence database search tool.* Proteomics, 2013. **13**(1): p. 22-4.
21. Craig, R. and R.C. Beavis, *TANDEM: matching proteins with tandem mass spectra.* Bioinformatics, 2004. **20**(9): p. 1466-7.
22. Cox, J. and M. Mann, *MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification.* Nat Biotechnol, 2008. **26**(12): p. 1367-72.
23. Duan, G. and D. Walther, *The roles of post-translational modifications in the context of protein interaction networks.* PLoS Comput Biol, 2015. **11**(2): p. e1004049.
24. Renard, B.Y., et al., *Overcoming species boundaries in peptide identification with Bayesian information criterion-driven error-tolerant peptide search (BICEPS).* Mol Cell Proteomics, 2012. **11**(7): p. M111 014167.
25. David, M., et al., *SpecOMS: A Full Open Modification Search Method Performing All-to-All Spectra Comparisons within Minutes.* J Proteome Res, 2017. **16**(8): p. 3030-3038.
26. Kil, Y.J., et al., *Preview: a program for surveying shotgun proteomics tandem mass spectrometry data.* Anal Chem, 2011. **83**(13): p. 5259-67.
27. May, D.H., K. Tamura, and W.S. Noble, *Param-Medic: A Tool for Improving MS/MS Database Search Yield by Optimizing Parameter Settings.* J Proteome Res, 2017. **16**(4): p. 1817-1824.
28. Bern, M., Y.J. Kil, and C. Becker, *Byonic: advanced peptide and protein identification software.* Curr Protoc Bioinformatics, 2012. **Chapter 13**: p. 13 20 1-13 20 14.

29. McIlwain, S., et al., *Crux: rapid open source protein tandem mass spectrometry analysis.* J Proteome Res, 2014. **13**(10): p. 4488-91.
30. Diament, B.J. and W.S. Noble, *Faster SEQUEST searching for peptide identification from tandem mass spectra.* J Proteome Res, 2011. **10**(9): p. 3871-9.
31. Deutsch, E.W., *File formats commonly used in mass spectrometry proteomics.* Mol Cell Proteomics, 2012. **11**(12): p. 1612-21.
32. Wilhelm, M., et al., *mz5: space- and time-efficient storage of mass spectrometry data sets.* Mol Cell Proteomics, 2012. **11**(1): p. O111 011379.
33. Bhamber, R.S., et al., *mzMLb: A Future-Proof Raw Mass Spectrometry Data Format Based on Standards-Compliant mzML and Optimized for Speed and Storage Requirements.* J Proteome Res, 2021. **20**(1): p. 172-183.
34. Goloborodko, A.A., et al., *Pyteomics--a Python framework for exploratory data analysis and rapid software prototyping in proteomics.* J Am Soc Mass Spectrom, 2013. **24**(2): p. 301-4.
35. Levitsky, L.I., et al., *Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework.* J Proteome Res, 2019. **18**(2): p. 709-714.
36. Leger, T., et al., *Fate and PPARgamma and STATs-driven effects of the mitochondrial complex I inhibitor tebufenpyrad in liver cells revealed with multi-omics.* J Hazard Mater, 2023. **442**: p. 130083.
37. Reiter, L., et al., *Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry.* Mol Cell Proteomics, 2009. **8**(11): p. 2405-17.
38. Lazear, M. *Introducing Sage: a new cross-platform, extremely performant, open source proteomics search engine; written in Rust*. 2022 January 23, 2023]; Available from: https://lazear.github.io/sage/.