Clustering Tandem Mass Spectra for Better Peptide Identification

by

Megha Panda

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Identifying the peptide sequence from a mass spectrum is done either by database search or De novo peptide sequencing. This thesis focuses on identification of peptides by using database search, which is a process where an MS/MS spectrum is searched against an entire database of spectra representing peptides of known proteins, in order to identify an exact match or a match with a spectrum of a homologous peptide. In a mass spectrometry experiment, a database search has two notable challenges: the large size of the sequence database (search space) and the volume of mass spectra generated during an experiment (millions of spectra), each of which has to be searched against the database. The output of a database search depends on the quality of the spectrum being searched and on whether the spectrum of the peptide sequence is in the target database.

One of the ways to address this problem is to use clustering as a preprocessing method. The past literature has shown that for a mass spectrometry experiment clustering decreases the time taken to perform a database search and increases the number of acceptable identifications for mass spectra. Clustering reduces the number of spectra undergoing database search by replacing a large amount of MS/MS spectra with a smaller number of cluster representatives. It boosts the signal-to-noise ratio (SNR), leading to the identification of one strong spectrum rather than many unidentified weak spectra.

In this dissertation, we apply various clustering techniques to data obtained from Tandem Mass Spectrometry and study how it affects the number

of acceptable peptide identifications. To improve peptide identification over previous work, we propose a new way to extract clusters from HDBSCAN* hierarchies. We experimentally show that this approach outperforms previous work in this area and performs comparably with other clustering techniques from the data mining literature.

We also study well-known cluster validation techniques to identify good parameter values for the different clustering algorithms and show that these approaches, unfortunately, do not work well in the context of peptide identification.

Preface

This thesis is an original work by Megha Panda. No part of this thesis has been previously published.

To my mother,

For always believing in me and encouraging me and most importantly for being the immense source of strength and support.

"The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a challenge and opportunity beyond imagination."

– Stuart G. Walesh.

"Life is about learning; when you stop learning, you die."

– Tom Clancy

Acknowledgements

I would first like to express my deepest and sincerest gratitude to my thesis advisor Professor Jörg Sander. He has helped me become a better researcher. He has always asked questions which have helped me gain a sound understanding of what I was doing. The door to Prof. Sander office was always open; whether I ran into a spot of trouble or had a question about my research or writing. He consistently allowed this thesis to be my own work but steered me in the right the direction whenever he thought I needed it. Thanks a lot for your mentorship and constant encouragement. I couldn't have asked for a better supervisor.

I am grateful to Dr. Zukui Li who first introduced me to the field of proteomics and the potential to explore this field from the viewpoint of a computer scientist.

I would like to thank my friends Nasim, Shrimati, Giovanna, Samreen, Sankalp, Sanket, Gautham, Roberto, Toni, Housam, Shubhayan and Talat for all the inspiring and amazing discussion on life and research.

Most importantly I would like to thank my parents, my brother and extended family for always being with me through thick and thin. Special thanks to wannabe scientist Navaneeth K. K.for everything.

Lastly, I would like to thank all the people who directly or indirectly were a part of my Masters journey, without whom this entire thesis would not have been possible.

Contents

List of Tables

List of Figures

Glossary

anions Ions result from atoms or molecules that have lost one or more valence electrons, giving them a negative charge. Those with a negative charge are called anions. 13

CID Collision-induced dissociation. 4

Da Dalton or the Unified atomic mass unit is a standard unit of mass which is approximately the mass of one nucleon (either a single proton or neutron) and is numerically equivalent to 1 g/mol. 40

dta dta format wherein each spectrum was written to a separate file containing one header line for the known or assumed charge and the mass of the precursor peptide ion, calculated from the measured m/z and the charge. This one line is followed by all the m/z, intensity pairs that represent the spectrum. 39

ETD Electron transfer dissociation. 4

FASTA FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. 50

HCD Higher-energy collisional dissociation. 4

MS/MS spectra Spectra that are generated by tandem mass spectrometry experiment. 2, 4

mzXML mzXML is a XML (eXtensible Markup Language) based common file format for proteomics mass spectrometric data. 39

SNR Signal-to-Noise Ratio. 5

Tandem mass spectra The data generated by a tandem mass spectrometry experiment. 2, 3, 5, 6

Chapter 1

Introduction

The past decade has seen many technological advances in almost all fields. The data explosion is one of the by-products of these advances; thus leaving behind an enormous amount of data available for analysis. Proteomics is one such field benefited by data deluge. Proteomic methodologies developed in recent times make it possible to identify, characterize, and comparatively quantify the relative level of expressions of hundreds of proteins that are co-expressed in a given cell type or tissue, or that are found in biological fluids such as serum. This is a result of all the interdisciplinary research in the field of molecular and cellular biology, protein/peptide chemistry, bio-informatics, analytical and bio-analytical chemistry, including the use of various instruments and software tools such as chromatographic separations and mass spectrometry.

Mass spectrometry instruments produce thousands of spectra representing peptides in each run, and one experiment may consist of multiple runs, thus, generating hundreds and thousands of spectra to be analyzed and identified. One way to analyze and identify these mass spectra is to search each of the spectra against an annotated database of spectra. Those spectra that do not get any hit in the database, further undergo more advanced and time-consuming investigation. More often the mass spectrometry data sets are redundant in nature as they contain multiple spectra representing peptides that have the same amino acid sequence. Many of these spectra, when searched against an annotated database of spectra, do not receive a hit in the database, as they have low signal-to-noise ratio. On the other hand, the size of the

database is growing rapidly with the addition of new protein sequences. This continuous expansion of the database leads to an increase in the search space. Thus, making the database search more challenging in terms of time and space.

This dissertation addresses these issues by taking advantage of the redundancy found in these large datasets. We cluster peptides (represented by spectra) that have the same amino acid sequences and replace each cluster by its cluster representative (consensus spectrum). These consensus spectra, when searched against an annotated database of spectra, result in fewer spurious hits to the database and increase the number of peptide identifications as compared to regular non-clustered searches. This increase in peptide identification is because, the formation of consensus spectra boosts the signal-to-noise ratio (the probability of two noise peaks appearing at the same point in multiple spectra that have the same amino acid sequence is low). Clustering replaces the duplicate spectra with their representative spectrum, thus reducing the number of points that undergo a database search. Thereby reducing the time taken to perform a database search. The cluster consensus spectra which do not get a hit in the database search, then undergo more advanced search with the aim to identify novel or post-transitionally modified peptides. Also, the increase in the number of the peptides identified improves the confidence of protein identification.

1.1 Related Work

The goal of clustering is to form groups of data that are extremely similar to one another. In the past, clustering was used on Tandem mass spectra for various applications. This section discusses at length how various applications are benefited by clustering mass spectra.

Bandeira et al., in the paper "Shotgun Protein sequencing by Tandem Mass Spectra Assembly" [6], defines a clustering method that follows a triangle condition to detect partial and complete overlaps between uninterpreted MS/MS spectra. In this context, the triangle rule means that, two spectra A and B, whose similarity is defined as a match score which is above a chosen thresh-

old, can be in the cluster, if there exists another spectrum C, such that the similarity between spectrum A and spectrum C, and the similarity between spectrum B and spectrum C is also above the same chosen threshold. These discovered clusters of overlapping spectra, when assembled and aligned into a graph, provide valuable information, which makes it possible to recover sequence information where virtually no MS/MS spectrum peaks are available. Thus, by peptide reconstruction from several partially overlapping peptides, this approach significantly improves the quality and extent of the de novo sequencing of an entire protein.

Pep-Miner [7] is a clustering algorithm that demonstrates how clustering improves analysis of Tandem mass spectra by reducing the runtime and generating additional peptide identifications. The clustering is performed in two stages; in the first stage, a transitive closure of MS/MS spectra is computed, where pairwise similarity is above 0.6 and parent masses differ by 2.5 Da. [1] In stage two the CAST clustering algorithm devised by Ben-Dor et al. [8] is applied on the members of each of the groups produced in stage I. Pep-Miner was developed by IBM, and unfortunately is not publicly available and little information is available on its clustering performance. Pep-Miner also relies on retention time prediction for clustering quality assurance, calibrating it can be a challenge when multiple MS runs are being clustered.

In 2008, Frank et al., [18] attempted to cluster large MS/MS datasets containing over 10 million spectra. They proposed MS-cluster, a new and efficient clustering approach to cluster large datasets of tandem mass spectrometry data. This clustering approach is discussed and explained in detail in Chapter 2. The method follows a greedy approach, by merging the first pair of clusters/points that it encounters having the similarity above a chosen threshold rather than merging first the clusters/points having the highest similarity. Clustering replaces the large volume of MS/MS spectra with cluster consensus spectra. These consensus spectra, when searched against an annotated database of spectra, lead to identification of more spectra as compared

[1] In this context a transitive closure means if spectrum A is similar to spectrum B and Spectrum B is similar to spectrum C, then spectrum C is similar to spectrum A.

to the number of identifications made with a standard database search of non-clustered data. Frank et al. show that MS-cluster can accelerate database search by reducing the number of spectra submitted by 10 folds in some cases.

Pride cluster [19] uses a modified version of MS-Cluster, to cluster all the MS/MS spectra that are submitted to the PRIDE archive [35] repository. It aids the reliability of identifications in heterogeneous MS experiments. It uses a modified version of the MS-Cluster, refined to increase the clustering quality. In contrast to MS Cluster, Pride uses Spectra ST to assess the quality of a spectrum. The clustering first joins clusters/points with the highest similarity rather than the first pair of clusters/points with a similarity above a set threshold. Additionally, it checks for non-fitting spectra in a cluster, i.e. if the similarity between a spectrum in a cluster and the consensus of the cluster is found to be less than a threshold, then the spectrum is removed from the cluster. The results from the PRIDE clustering are used to correct inaccurate annotations in the PRIDE database. Pride cluster also uses the cluster consensus to aid the construction of reliable spectral libraries.

Another application of clustering is shown in the paper "Sequencing-Grade De novo Analysis of MS/MS Triplets (CID/HCD/ETD) From Overlapping Peptides" [20]. This approach also uses a modified version of MS-Cluster to cluster spectra acquired by various fragmentation methods (Collision-induced dissociation (CID), Electron transfer dissociation (ETD), Higher-energy collisional dissociation (HCD)) to boost the interpretation of long and highly charged peptides.

The paper 'Comparison and Evaluation of Clustering Algorithms for Tandem Mass Spectra' [29] explores different clustering methods on Tandem Mass Spectra. Peptide annotations by Mascot peptide to spectrum matches for the non- clustered data are used to evaluate clustering algorithm as opposed to using annotations for consensus spectrum. The performance of each clustering algorithm is evaluated based on various evaluation metrics[2]. Adjusted Rand

[2]Adjusted Rand Index, purity, the proportion of Spectra remaining, retainment of identified spectra, the proportion of clustered spectra, the proportion of incorrectly clustered spectra.

Index (ARI) is used to compare different partitions of the same dataset. Purity is another evaluation metric used on the clusters, to control the quality. They discuss how the cluster analysis can be used in quality control of databases. Along with it, a new clustering algorithm called N-Cluster is proposed.

To summarize, we see that clustering finds various applications when applied on tandem mass spectrometry data. First, it reduces the time taken by a database search by replacing multiple spectra by a single representative. Second, it increases the number of identified spectra as it boosts the Signal-to-Noise Ratio (SNR) by combining many low-quality unidentifiable spectra to generate one high-quality identifiable spectra. Third, it increases the confidence with which de-novo interpretations are made and also helps to identify novel peptides. Fourth, it can be used as a quality control tool to detect wrongly annotated spectra in a large database. Fifth, the consensus spectrum aids in the construction of spectral libraries.

The output of any clustering method depends highly on the selection of good parameters. Each of the clustering algorithms mentioned requires a parameter value as an input to effectively carry out clustering. MS-Cluster needs a similarity threshold and a number of rounds. CAST requires set affinity. Unfortunately, the literature shows no way to select these parameters. In an attempt to overcome this we experiment with known internal cluster validation measures, to guide the parameter value selection.

1.2 Challenges:

All biological data has a certain amount of uncertainty associated with it. Tandem mass spectra are no exception. These uncertainties can arise from different sources such as limitation of mass spectrometers in terms of resolution accuracy, sensitivity, and mass range. Contamination of samples adds background noise in the data, making it noisy in nature. Though all uncertainties cannot be removed, continuous efforts are being taken to reduce the sources of uncertainties. Additional challenges are the lack of publicly available annotated datasets, and the absence of a standard method to validate the

database search output.

1.3 Contribution

In this dissertation, we apply various clustering techniques to data obtained from Tandem Mass Spectrometry, and study how it affects the number of acceptable peptide identifications. The contributions of this thesis are as follows.

Firstly, we use a wide variety of clustering methods to cluster tandem mass spectra to study its effect on the number of the peptide identified. We use the clustering algorithms from proteomics literature, along with clustering algorithms from data mining literature. To the best of our knowledge, there is no such comprehensive comparison of different clustering algorithms on tandem mass spectra in the context of peptide identification to be found in the literature yet.

Secondly, we propose a new method to extract clusters from HDBSCAN* hierarchies. The experimental results show that this approach outperforms the previous work MS-cluster and N-Cluster, and is comparable to other clustering techniques in the context of peptide identification.

Finally, we study two well-known cluster validation techniques namely Silhouette Width Criterion (SWC) and Density Based Cluster Validation (DBCV) to identify good parameter values for different clustering algorithms with an aim to increase peptide identification. We show that these approaches, unfortunately, do not work well in this context.

1.4 Outline

This thesis is structured as follows. In chapter 2 we give a basic background of Tandem mass spectra and the workflow of experiments required to understand the intuition behind the methodology, described in Chapter 3. Along with it, we discuss the clustering algorithms proposed in the proteomics literature, as well as the clustering algorithms proposed in data mining literature, which we apply on Tandem mass spectra. Chapter 3 explains a new methodology to extract clusters from HDBSCAN* hierarchies. It also gives details on pre-

processing spectra, similarity, and the post-processing techniques. Chapter 4 describes the details about the experimental setup. It also presents an extensive experimental evaluation of different clustering techniques on tandem mass spectra. Chapter 5 summarizes the dissertation and discusses future research possibilities.

Chapter 2

Background

2.1 Tandem Mass Spectrometry

Mass spectrometry (MS) is an analytical technique to detect, identify and quantify known and unknown molecules in a sample. This is done by analyzing the plot that measures the relative abundance of ions, which represents the structure and chemical properties of the sample molecules. A Mass Spectrometer is an instrument that generates this plot. A mass spectrometer has three components; namely 1) an ionization source 2) the mass analyzer 3) the detector. The figure 2.1 illustrates the general principle of mass spectrometry. A sample is ionized to produce gas phase ions. These ions are introduced into the ionization source of the instrument, where these molecular ions undergo fragmentation. The fragmented ions are extracted into the analyzer region of the mass spectrometer, where the ions are sorted and separated according to their mass (m) to charge (z) ratios (m/z). The separated ions are detected in the ion detector region of the mass spectrometer. The relative abundance of each of the resolved ionic species is recorded. This recorded signal is stored in a data system as a graph of a mass-to-charge vs. intensity called mass spectrum.

Tandem Mass spectrometry, also known as MS/MS, uses two mass spectrometers in tandem. As two spectrometers are used, ions are fragmented multiple times. Figure (2.2) depicts the formation of tandem mass spectra (MS/MS). A sample is injected into the first mass spectrometer (MS_1), where it is ionized, accelerated and analyzed, forming MS spectra. Then, ions of a

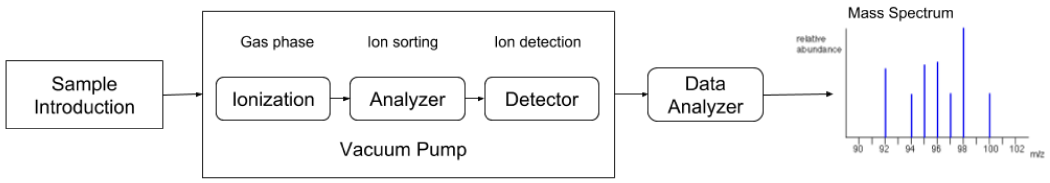


Figure 2.1: Mass Spectrometry: In the ionization source the sample components acquire their charges. In the mass analyzer, the components are separated according to their m/z values before they hit the detector. A computer connected to the instruments then construct the mass spectrum

specific range of mass to charge ratio are selected to be analyzed in a second Mass spectrometer MS_2. The ions produced in MS_1 are called parent ions or precursor ions. These parent ions break down into daughter ions, which are analyzed and then detected. These detected ions are processed with the help of data systems recording the MS/MS spectrum. While the diagram in Figure (2.2) indicates the use of two separate mass analyzers, some instruments utilize a single mass analyzer for both rounds of MS.

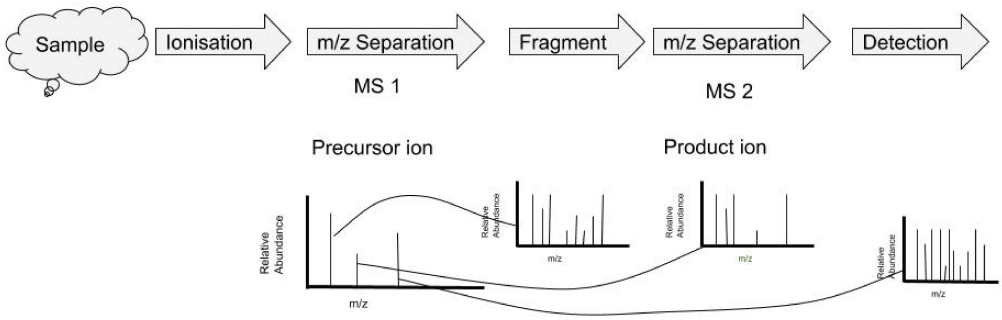


Figure 2.2: Tandem Mass Spectrometry

Figure 2.3 represents an entire workflow for protein identification. A typical bottom-up mass spectrometry experiment starts with a mixture of proteins. This mixture of proteins is enzymatically digested into peptides. Enzymes like trypsin, chymotrypsin, pepsin, papain, elastase are used in digestion, amongst which trypsin is the most common protease in such experiments. These peptides undergo fragmentation inside an ionization source (MS_1). A precursor ion range is then selected, and all the ions in this selected range are fragmented (MS_2). These fragmented ions are then picked up by the analyzer and detec-

tor of MS_2. The fragments are analyzed and recorded as MS/MS spectra. These acquired MS/MS spectra are uninterpreted and require identification to deduce the amino acid sequence of the peptides (MS/MS spectra). There are two ways to identify the uninterpreted spectra; 1) Database Search and 2) De-novo sequencing. In a database search, the acquired MS/MS data is searched against a theoretical database to yield a match. It returns all the possible peptide sequences for a spectrum. On the other hand, De-novo sequencing derives the peptide sequence without the aid of any theoretical databases. The list of identified peptides is used to infer proteins present in the original sample.

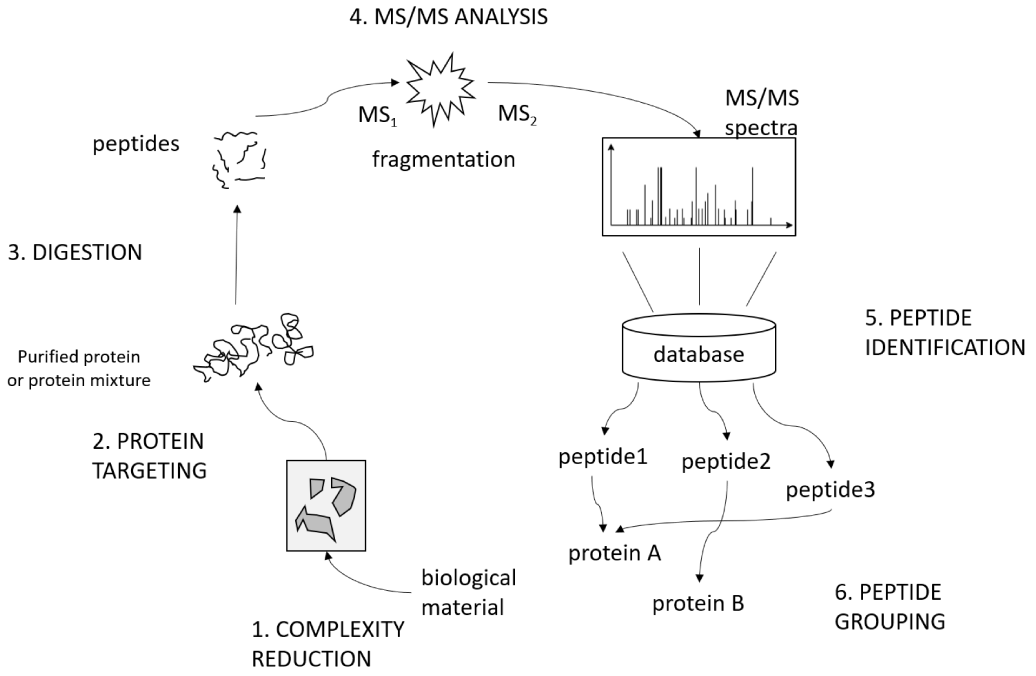


Figure 2.3: A typical experimental work flow for protein identification and characterization using MS/MS data.

2.1.1 Fragmentation of peptides

Proteins and peptides are naturally occurring, or artificially manufactured, chains of amino acids linked by peptide (amide) bonds. Proteins are long polymer chains of amino acids. On the other hand, peptides are short polymer

chains.[1] Figure 2.4 [1] depicts how two amino acids R_1 and R_2 on α-carboxyl group are linked in a chain by a bond called peptide bond. The peptide bond -OC-NH- is a covalent bond formed between the carboxyl group and the amino group by dehydration (removal of one molecule of water).

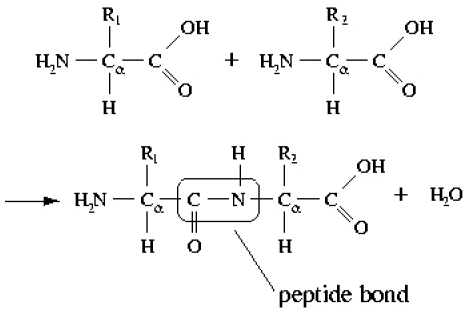


Figure 2.4: Peptide bond

The long repeating sequence of peptide bonds along with the α-carbon is called the peptide backbone. Figure 2.5 [16] shows an example of a polypeptide chain made up of a constant backbone (shown in black) and variable side chains (shown in green).

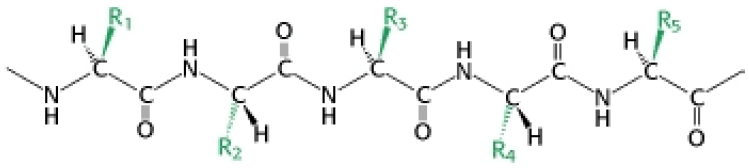


Figure 2.5: Polypeptide chain

A peptide inside an ionization source breaks into fragment ions by a process called fragmentation [2]. There are three different types of backbone bonds that can be broken to form peptide fragments: alkyl carbonyl *(CHR-CO)*, peptide amide bond *(CO-NH)*, and amino alkyl bond *(NH-CHR)*. When these backbone bond cleave, six main backbone fragment ions are formed, namely:

[1] An the amino acid is a molecule containing an amine group, a carboxylic acid and a varying side chain (R-varies)

[2] Fragmentation is the dissociation of energetically unstable molecular ions formed from passing the molecules in the ionization chamber of a mass spectrometer

a, b, c, x, y, z. Breakage of alkyl-carbonyl *(CHR-CO)* bond results in a and x ions. Breakage of peptide amide bond (CO-NH) results in b and y ions. The cleavage of amino alkyl bond results in c and z ions. If the charge is retained on the N-terminal fragment, the ion is classified as either a, b or c. If the charge is retained on the C-terminal, the ion type is either x, y or z. A subscript indicates the number of residues in the fragment.

Figure 2.6b depicts formation of ions a_2, b_2, c_2, x_2, y_2 *and* z_2 when the peptide in figure 2.6a cleaves. The two subscript shows that there are two residues in each fragment; either R_1, R_2 or R_3, R_4. Each peptide can break into two fragments at multiple sites. By the law of conservation of mass, the sum of both fragments is equal to the sum of all residual mass. Mass of the neutral peptide is equal to the sum of the mass of residues plus the nominal charge on 'N' terminal and the nominal charge on 'C' terminal.

From the figure 2.6;

$$b_n = [\text{residue masses} + 1]\text{these come from the N-terminus} \tag{2.1}$$

$$y_n = [\text{residue masses} + H_2O + 1] - \text{these come from the C-terminus} \tag{2.2}$$

$$\text{Mass of b ions} = \sum(\text{residue masses}) + 1(H^+) \tag{2.3}$$

$$\text{Mass of y ions} = \sum(\text{residue masses}) + 19(H2O + H^+) \tag{2.4}$$

$$\text{Mass of peptide} = \sum(\text{residue masses on fragment}) + 1(H^+) + 17(OH^-) \tag{2.5}$$

$$\text{Mass of b ion} + \text{Mass of y ion} = \text{Mass of peptide} + 2H^+ \tag{2.6}$$

$$\text{Mass of a ion} = \text{mass of b ion–28}(CO) \tag{2.7}$$

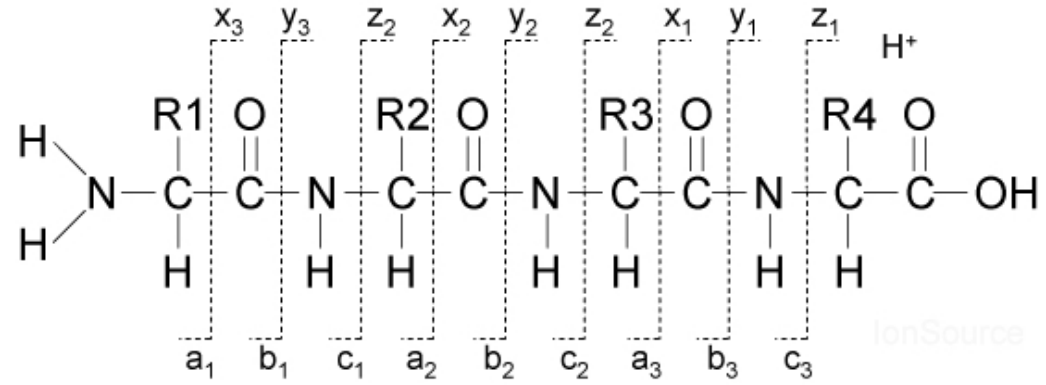
The most commonly occurring fragment types are *b-ions* and *y-ions* and partly *a-ions*. This is because the peptide amide bond *(CO-NH)* is the most vulnerable and the loss of *CO* from *b-ions* form *a-ions*. Many database search programs like MS-Tag [22] and SEQUEST [36] only consider these kinds of ions and in addition to some anions in their algorithms.

When a peptide is fragmented, not all the ions translates into the spectrum. Only the ions having a minimum of charge one are detected and translated onto the spectrum.

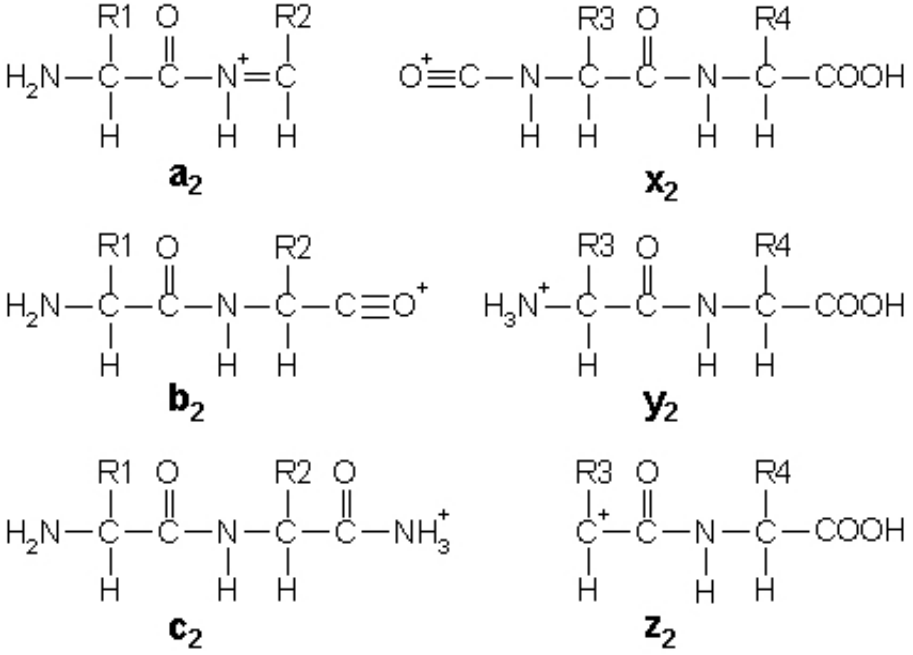
Figure (2.7) illustrates the fragmentation of a peptide having a sequence $NIDALSGMEGR$ and shows an example of how the fragments are translated in the spectrum. This peptide can get fragmented at ten different sites, but only a few of the fragments are observed in the spectrum. This is because the fragments did not carry any charge after fragmentation, so they went unobserved/undetected. When the fragmentation occurs between I and D, both the fragmented ions b and y *(b_2 and y_2)* are recorded but in the case of fragmentation that occurs between L and S only *y-ions* (y_5) are recorded. This is because *(y_5)* had more than charge one on it. However, *(b_2)* was not recorded, as it did not carry any charge. In the same figure, there are peaks which do not correspond to any fragments and which are considered as noise peaks.

2.1.2 Database search

In a database search, a peptide sequence is identified with the help of a sequence database. Each MS/MS spectrum acquired from the experiment is searched against a theoretical database. The database search returns possible matches with a score for each spectrum. The method/algorithm of comparison and the mathematical scoring between the acquired and the theoretical spectrum vary widely between different database search engines. The score generally represents the degree of match between the experimental spectrum and the theoretical spectrum. There are different database search engines which perform a database search. A few examples are Mascot [12], SEQUEST [36], InsPect [34] and X!Tandem [13]. Many of them are propriety and licensed



(a) All annotation sites of peptide



(b) Six backbone fragments.

Figure 2.6: Six main backbone fragments formed by fragmentation [14]

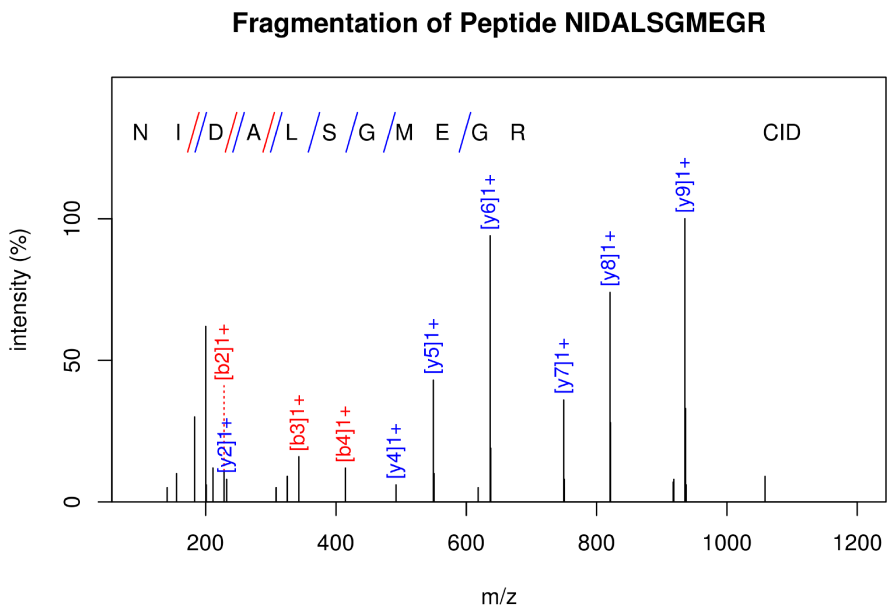


Figure 2.7: Fragmentation of peptide having sequence $NIDALSGMEGR$

software, and a few of them are free for academic use. Figure 2.8 [2] shows a schematic way of how proteins are digested into peptides and how a mass spectrum representing a peptide obtains a match from a sequence database.

One key requirement to obtain a successful match is that the database should have a homologous entry for a spectrum, that is being searched. If the database has no entry for a spectrum in advance, then it will fail to return a homologous match. Thus most database match algorithms fail to identify post-translational modifications and novel peptides.

In this thesis, we use the InsPect [34] database search tool to perform peptide identifications.

Validating Database Search

In order to identify a spectrum, it is searched against a protein sequence database whereby each experimental spectrum is computationally compared with the predicted spectra of candidate peptides in the database. The peptides are scored and ranked according to their degree of match to the input (experimental) spectrum, and the best-scoring peptide is chosen as the identification

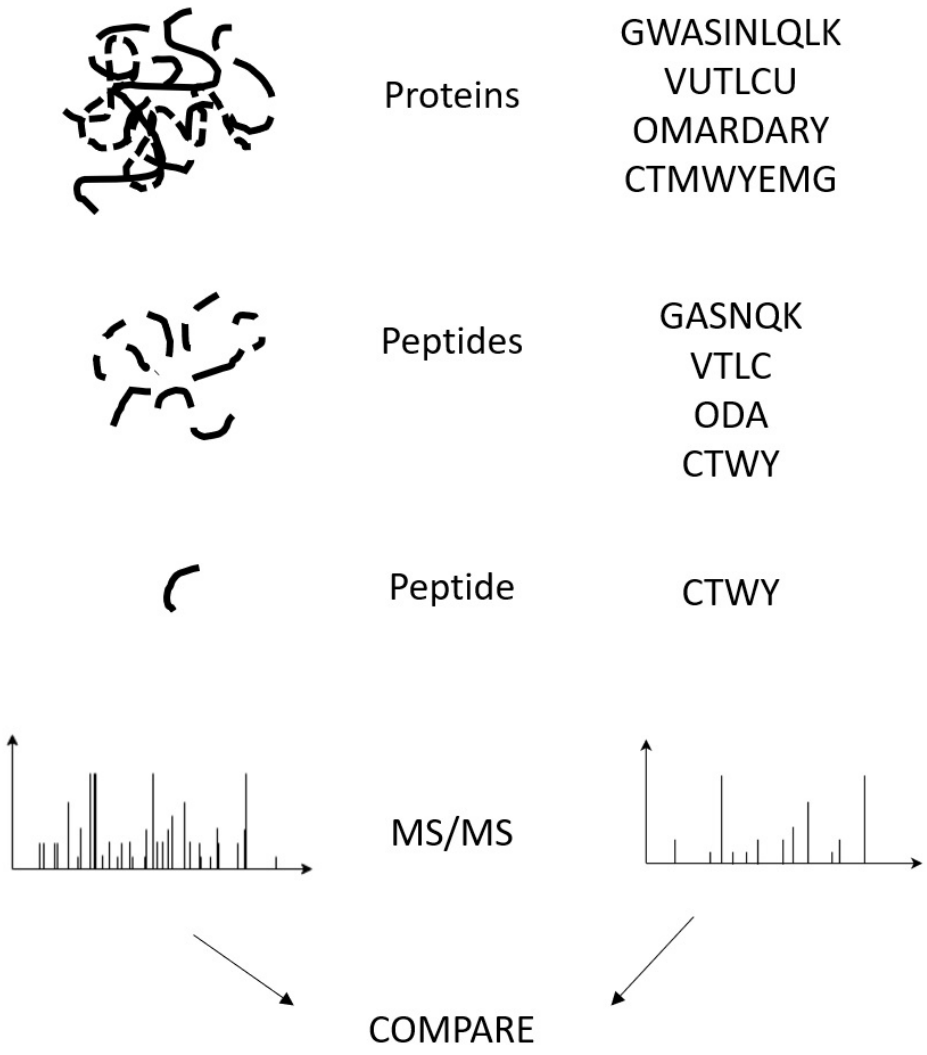


Figure 2.8: A schematic showing the concept of MS/MS database searching

of the spectrum. As a significant number of spectra are not correctly identified (due to various reasons), the set of matches obtained contain many false positive matches. Thus, one needs to filter the search results and asses the reliability of selected identifications. For a group of identifications rather than deciding which identifications are correct and which are incorrect, it is easier to estimate the proportion of incorrect identifications. This is the problem of False Discovery Rate (FDR). The target-decoy search strategy is a simple and useful tool to estimate FDR. The target-decoy search strategy allows the estimation of how many False Positives are associated with an entire dataset

and helps to select the above-mentioned threshold. In this strategy, the spectra in addition to being searched against the target database are also searched against an equal-size decoy database. Any hit in the decoy database is a false hit and helps to estimate the false positive rate. For an acceptable FDR value V,the cutoff for the matching score is selected such that the set of matches above the cutoff value has at most an FDR of V; all the matches below the cutoff are filtered out.

A decoy database is constructed by shuffling, randomizing or reversing the target database. There are two ways to search in a target-decoy database (TD). Either a spectrum can be searched separately in a target and a decoy database, or it can be searched against the concatenated database of target and decoy database, each with different assumptions on target decoy competition and false-positive estimation. Figure 2.9 taken from [4] illustrates the two database search strategies.

When a spectrum is searched separately in target and decoy databases, it yields one best match from target and decoy. Kall et al. [24] proposes a simple method to calculate FDR given by the equation 2.8

$$FDR_s = \frac{D}{T} \tag{2.8}$$

where, D is number of hits in decoy above threshold and T is number of hits in target database above threshold.

When a spectrum is searched against one unified target-decoy database, it gets a match either from target or decoy but not both. Elias et al. [15] provides a method to calculate FDR for target-decoy searches. This method has an underlying assumption that for any number of decoys (D) passing a given threshold, there are an equal number of false hits in target peptide spectrum match (PSMs) (T) above that threshold. Adding up the false hits in decoy and target, the number of false positives is, therefore, double of the decoy count above the threshold. The equation 2.9 shows how to calculate FDR in this case.

$$FDR_c = \frac{2XD}{T + D} \tag{2.9}$$

where, D is number of hits in decoy above threshold and T is number of hits in target database above threshold.

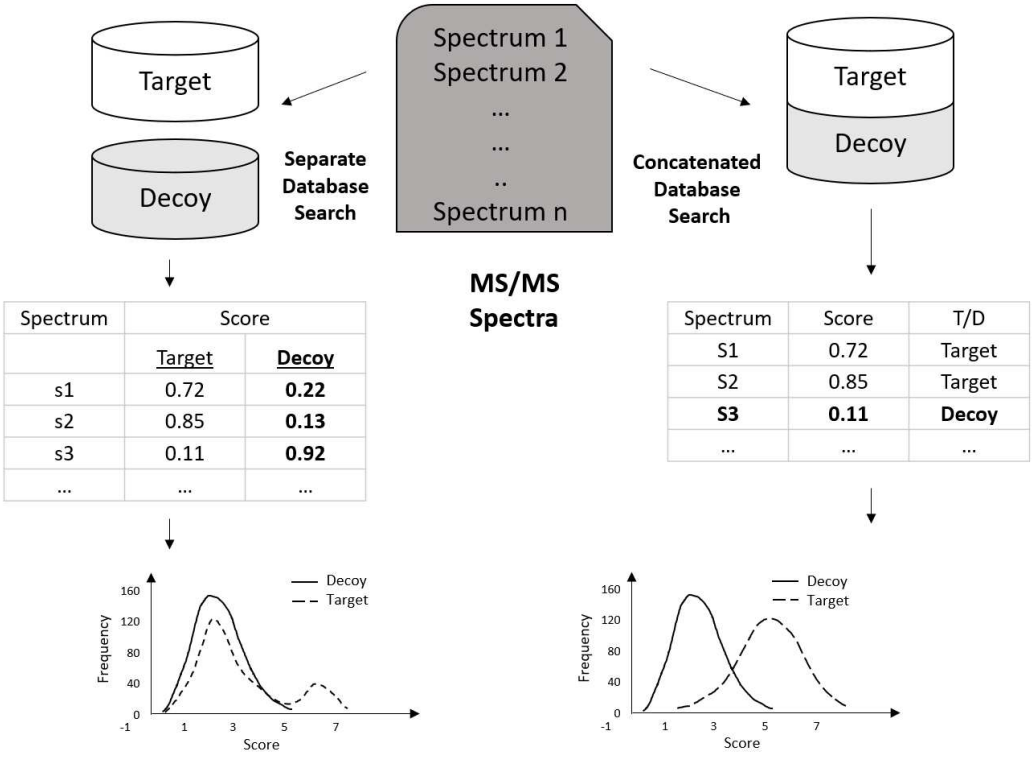


Figure 2.9: A diagram showing two database search strategies.

2.2 Clustering

Cluster analysis is an analytical and an exploratory tool, which helps us to study and analyze data. Cluster analysis divides data objects into groups called clusters. These groups contain data objects that are similar to each other and dissimilar to data objects in another group. Clusters are formed based on the information contained in the data objects and the relationship between them. Different clustering technique/algorithm divides the same set of data in different ways. Figure 2.10 shows different partitions of the same dataset. Thus, not all clustering technique would work well for all data.

Most common techniques such as K-Means [25] and spectral clustering

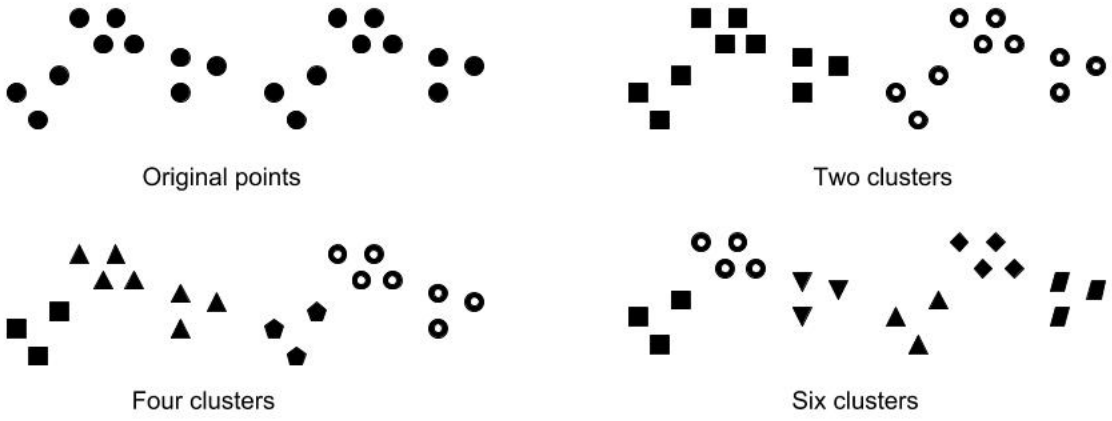


Figure 2.10: Different Ways of clustering same set of data points

[27] require the number of clusters being sought in advance. Other clustering techniques such as Brich [37] and Gaussian mixture models have parameters that are difficult to estimate. And few clustering techniques like mean shift algorithm [11] are not scalable. But, the nature of MS/MS dataset rules out the use of these algorithms, as it is impossible to guess the number of clusters in advance. Scalability of a clustering algorithm is important because of the sheer volume of data. Clustering techniques can be divided into partitioning, hierarchical and density-based methods. We explore our data on various hierarchical clustering techniques and density-based clustering techniques. In this section, we describe all the clustering techniques that we use from the literature.

For explanation of all the algorithms below, we will assume a dataset containing points D = $\{x_1, x_2, ..., x_n\}$ of n d-dimensional data objects. The pairwise distances can be represented by a $\left[n * n\right]$ matrix, X_{dist}, with any element of the matrix $X_{i,j}$ = dist(x_i, x_j), 1 $\leq$ i, j $\leq$ n, where dist(x_i, x_j) is some distance measure between data objects x_i and x_j.

2.2.1 DBSCAN

DBSCAN [17] is a clustering technique that views clusters as regions of high density separated by regions of low density. The algorithm defines a cluster as a maximal set of density-connected points. The algorithm DBSCAN is

described below in detail as presented by Ester et al. in [17]. It takes two global parameters ϵ and min_{pts}, which define the local density of a point X_p that belongs to dataset D.

- ϵ **:** maximum distance between two points to be considered in the same neighborhood. The definition of ϵ neighborhood of a point is given by definition 1.
- min_{pts} **:** Minimum number of points to be considered in the ϵ neighborhood.

Definition 1. ϵ **neighborhood of a point :** The ϵ neighborhood of a point X_p, denoted by $\mathrm{N}_\epsilon(\mathrm{X}_p)$, is defined by $\mathrm{N}_\epsilon(\mathrm{p})=\{q \in D | dist(X_p, X_q) \leq \epsilon\}$.

Every point X_p in the dataset D can be classified into 3 categories. The area where a point has more number of data points than min_{pts} in the ϵ neighborhood is called dense.

1. **Core point :** A point is a core point if it has more than a specified number of points, (min_{pts}) within ϵ.
2. **Border point :** A border point has less number of points than min_{pts} within ϵ, but is in the neighborhood of a core point.
3. **Noise :** A point which is neither a border point nor a core point. Definition 6 formally defines the noise which is also known as outliers in a dataset.

Figure 2.11 illustrates an example of core , border and noise points for min_{pts} 4 and ϵ as one unit distance.

Definition 2. Directly density reachable: A point X_p is *directly density reachable* from a point X_q w.r.t. ϵ, min_{pts} if

1. $X_p \in N_\epsilon(X_q)$ and
2. $|N_\epsilon(X_q)| \geq min_{pts}$(core point condition)

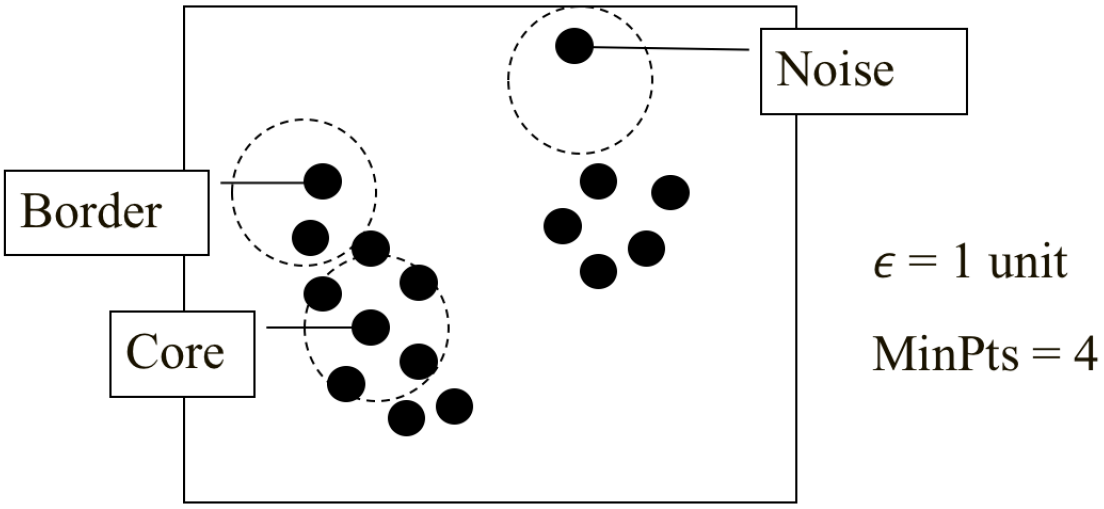


Figure 2.11: Core, Border and Noise point for $\min_{pts}$ four and ϵ as one unit distance.

Definition 3. Density reachable : A point X_p is density reachable from point X_q w.r.t. ϵ and min_{pts} if there is a chain of point $X_{p1}, X_{p2}, ...X_{pm}, X_{p1}$=$X_q$ such that X_{pi+1} is directly density reachable from X_{pi}

Figure 2.12 shows density reachability of two points X_p and X_q. Point X_p is density reachable form point X_q but the vice versa is not true. Thus density reachability is asymmetric.

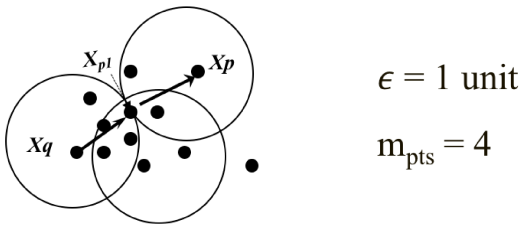


Figure 2.12: Point X_p is density reachable from point X_q through point X_p, but point X_q is not density reachable from point X_p.

Definition 4. Density-connected : A point X_p is density connected to a point X_q w.r.t. ϵ and min_{pts} if there is a point X_o such that both, X_p and X_q are density-reachable from X_o' w.r.t. ϵ and min_{pts}.

Figure 2.13 shows the density connectivity of points X_p and X_q via X_o.

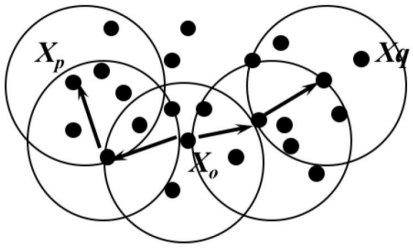


Figure 2.13: Point X_p and X_q are density-connected to each other by point X_o.

Formally the density based clusters are defined as:

Definition 5. Cluster: For a dataset of points D, a cluster C w.r.t. to min_{pts} and ϵ is defined as a non-empty subset if D satisfying the following condition.

- Maximality Condition: $\forall$ X_p, X_q: if $X_p \in$ C and X_q is density-reachable from X_p w.r.t. ϵ and min_{pts}, then $X_q \in$ C.
- Connectivity condition: $\forall$ X_p, $X_q \in$ C and X_q is density-connected to X_q w.r.t. ϵ and min_{pts}.

Definition 6. Noise: Let C_1,C_2,...,C_k be the clusters of the database D w.r.t. parameters ϵ_i and min_{ptsi}, i = 1,2,...,k. Then the noise is defined as a set of points in the database D that do not belong to any cluster C_i.

noise = $\{X_p \in D | \forall i : X_p \notin C_i\}$

Clustering starts with an arbitrary random point, which does not have a label. All the points in the ϵ neighborhood of this point are retrieved, if the number of points exceeds the min_{pts}, a cluster is started. Otherwise, the point is assigned a noise label. It is possible that a point which is initially labeled as noise point may later be found in a sufficiently sized ϵ- neighborhood of a

different point and is assigned a cluster label. If a data point X_p is found to be in a dense part of a cluster C, then, all the ϵ neighborhood points of this point X_p, by default becomes the part of the cluster C. This process continues until the complete density-connected cluster is found. Then, a new unlabeled data point is selected, and the entire process is repeated until all data points are assigned a label. Algorithm 1 describes the pseudo code for DBSCAN.

Algorithm 1 Pseudo algorithm for DBSCAN

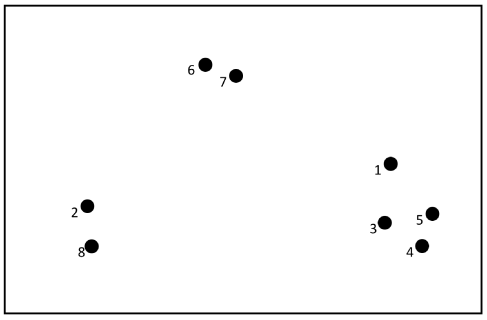
Input: DataSet D, ϵ, min_{pts}
Output: Cluster Labels

```
1:  procedure DBSCAN(cc)
2:      Given data set D, compute the distance matrix, X_dist
3:      for every point X_p in D, which is not assigned to cluster do
4:          compute number of the point in the ε neighborhood
5:          if number of points ≥ min_pts then          ▷ if core point or not
6:              Assign the point X_p and the             ▷ This will take
7:              points that are directly density         ▷ into account for
8:              reachable to the point                   ▷ all border points
9:              X_p a cluster label.
10:         end if
11:     end for
12:     for points in D, which as no cluster label do
13:         Assign them null label                       ▷ Noise Label
14:     end for
15: end procedure
```

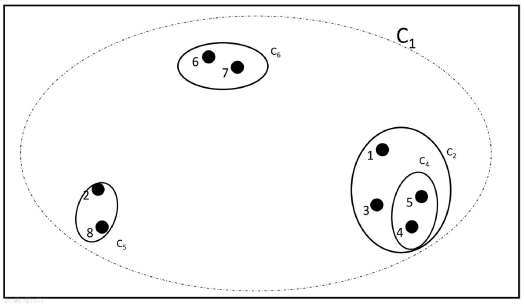
2.2.2 Hierarchical Clustering

Hierarchical clustering [23] algorithms build a hierarchy of clusters, a structure that is more informative than the unstructured set of clusters returned by flat clustering. Hierarchy is a set of nested clusters represented as a tree. This tree is formed when the data points iteratively change their cluster membership. This hierarchy of cluster and its sub-clusters is represented in the form of a tree and can be visualized with the help of a dendrogram. Figure 2.14 shows a visual example of data points, a clustered result, and the corresponding dendrogram [33].

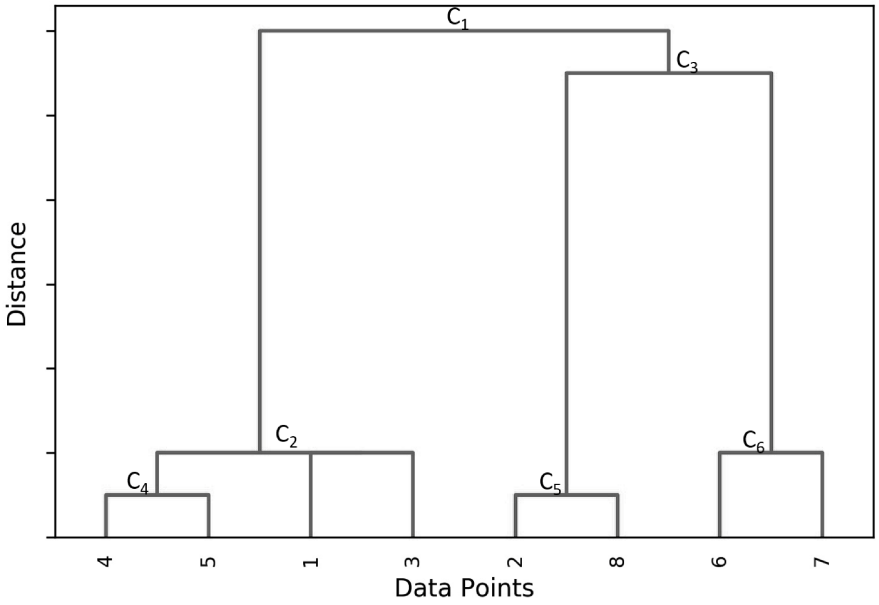
Hierarchical clustering can be categorized as agglomerative (bottom-up) and divisive (top-down) approaches. The divisive approach starts with all the



(a) Sample Data



(b) Venn Diagram of Clusters



(c) Dendrogram

Figure 2.14: Dendrogram and nested clusters on a Sample data

data objects initially belonging to one cluster. This cluster is successively split into sub-clusters based on some criteria until all the elements form singleton cluster. Agglomerative hierarchical clustering which follows a bottom-up approach starts with each data object initially representing a cluster of its own. These singleton clusters are progressively merged based on a distance function. They continue to merge until all the sub-clusters have merged into one large cluster. The figure 2.15 illustrates the difference between the agglomerative (bottom-up) and Divisive (Top-down) clustering approaches.

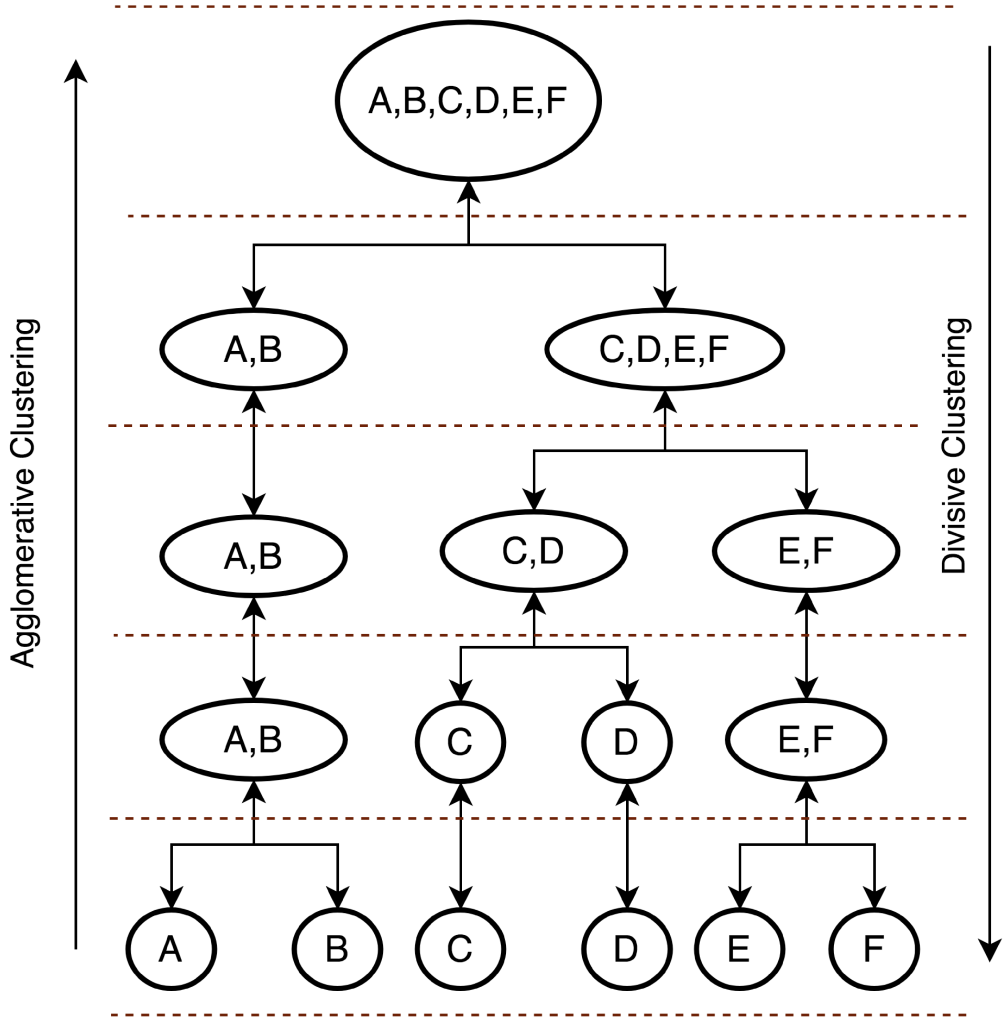


Figure 2.15: Difference between Agglomerative (bottom-up) and Divisive(top-Down) clustering approaches

Clusters are extracted by making a horizontal cut based on some distance

threshold.

Linkage criteria determine the distance between the sets of observations –i.e., the (dis)similarity between two clusters C_i and C_j. Three widely known linkage criteria that we use in our experiment are given by :

- Maximum or complete-linkage.

$$max\,\{dist(x_i, x_j) : x_i \in C_i,\ x_j \in x_j\} \tag{2.10}$$

- Minimum or single-linkage.

$$min\,\{d(x_i, x_j) : x_i \in C_i,\ x_j \in x_j\} \tag{2.11}$$

- Mean or average linkage.

$$\frac{1}{|C_i||C_j|} \sum_{x_i \in \mathcal{C}_i} \sum_{x_j \in C_i} dist(x_i, x_j) \tag{2.12}$$

  where |.| represents the cardinality of the set.

An illustration of the same is given in figure 2.16.

2.2.3 Hierarchical DBSCAN*- HDBSCAN*

The HDBSCAN* [10] algorithm is a hierarchical version of DBSCAN* which is a reformulation of DBSCAN [17]. DBSCAN* conceptually finds clusters as connected components of a graph in which the objects of dataset D are vertices, and every pair of vertices is adjacent only if the corresponding objects are ϵ-reachable w.r.t. the parameters ϵ and min_{pts}. It defines a density-based cluster based on core objects alone. We describe the algorithm HDBSCAN* in detail as presented in [10] .

Algorithm DBSCAN*

Definition 7. Core Object: An object X_p is called a core object w.r.t. ϵ and min_{pts} if its ϵ-neighborhood, $N_\epsilon(.)$, contains at least min_{pts} objects, i.e.,if $|N(X_p)| \geq min_{pts}$, where $\mathrm{N}(X_p) = \{X \in D | dist(X, X_p) \leq \epsilon\}$ and |.| denotes cardinality of the enclosed set. An object is called *noise* if the object is not a core object.

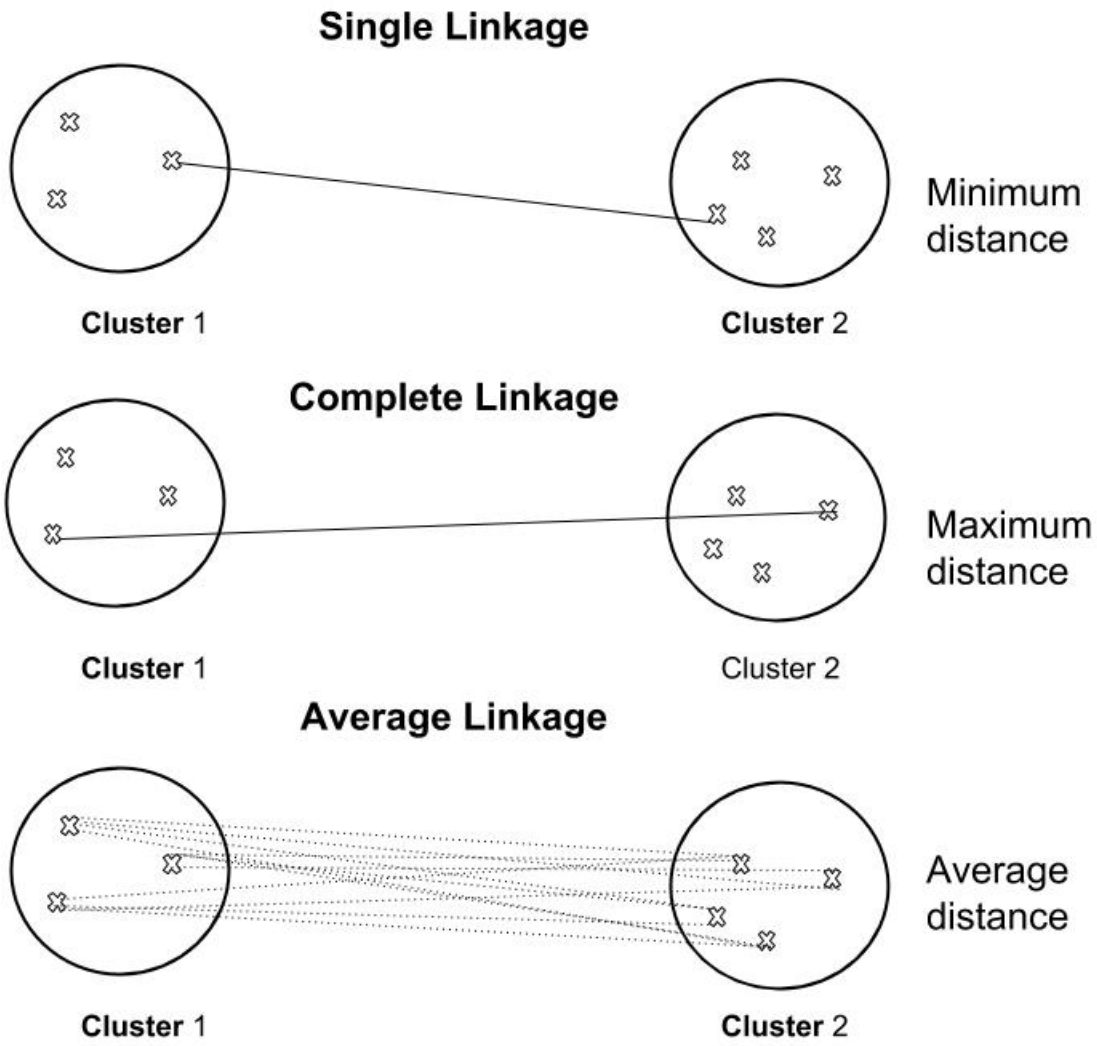


Figure 2.16: Linkage criteria for hierarchical clustering

Definition 8. ϵ-reachable: Two core objects X_p and X_q are -reachable w.r.t. and min_{pts} if $X_p \in N_\epsilon(X_q)$ and $X_q \in N_\epsilon(X_p)$.

Definition 9. Density-Connected: Two core objects X_p and X_q are density connected w.r.t. ϵ and min_{pts} if they are directly or transitively ϵ-reachable.

Definition 10. Cluster: A cluster C w.r.t. ϵ and min_{pts} is a non-empty maximal subset of D such that every pair of objects in C is density-connected.

HDBSCAN*

HDBSCAN* [9] is based on the concept that a hierarchy can be built from different levels of density. This density is based on different values of ϵ. HDBSCAN* can be explained based on the following definitions:

Definition 11. Core distance: The core distance, $d_{core}(x_p)$, of an object $x_p \in$ 'D' w.r.t. min_{pts} is the minimum distance between x_p to its min_{pts}-nearest neighbor including the point x_p.

Definition 12. ϵ core object: An object $x_p \in$ 'D' is called an ϵ-core object for every value of ϵ that is greater than or equal to the core distance of x_p w.r.t to min_{pts}.

Definition 13. Mutual reachability distance: The mutual reachability distance between two objects x_p and x_q in the dataset 'D' w.r.t. to min_{pts} is defined as $d_{mreach}(x_p; x_q) = \max\{ d_{core}(x_p), d_{core}(x_q), dist(x_p; x_q)\}$

Definition 14. Mutual Reachability graph: It is a complete graph $G_{m_{pts}}$, in which the objects of the data set D are vertices and the weight of each edge is the mutual reachability distance (w.r.t. m_{pts} between the respective pair of objects.

From definitions 10, 12 and 14, it can be deduced that the clusters created according to DBSCAN* w.r.t partitions for $\epsilon \in [0;1)$ can be produced in a nested and hierarchical way by removing edges in decreasing order of weight from the graph G_{mreach}. This is equivalent to applying Single Linkage on a transformed space of mutual reachability distances and then making a horizontal cut at ϵ. The set of connected components obtained are clusters while the singleton objects are noise objects. All possible levels in a hierarchy can be extracted by removing one edge at a time with decreasing values of ϵ starting with the highest value ϵ of from the graph G_{mreach}.[33]

A density-based cluster hierarchy represents the fact that an object o is noise below the level l that corresponds to o's core distance. To express this in a dendrogram, we can include an additional dendrogram node for o at level l

representing the cluster containing o at that level or higher. To construct this hierarchy, the Minimum Spanning Tree (MST) of the Mutual Reachability Graph G_{mreach} needs to be extended. Extension of the MST is created by adding self-edges" to each vertex with an edge weight equal to that of the core distance of o, $d_{core}(o)$. This extended MST can be used to construct the extended dendrogram by removing edges in decreasing order of weights. The hierarchy is computed by constructing Minimum Spanning Tree (MST) on the transformed space of mutual reachability distance. The HDBSCAN* hierarchy can be extracted from the MST by iteratively removing edges from the MST in decreasing order of their weights. Algorithm 2 describes a way to calculate HDBSCAN* hierarchy w.r.t min_{pts}.

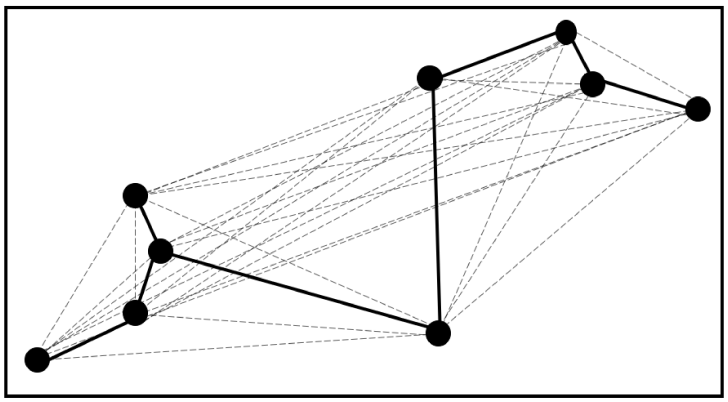


Figure 2.17: An example of a Minimum Spanning Tree (MST) generated from a complete graph where edges that form the MST are shown using solid lines and edges that are a part of the complete graph but not in the MST are shown using the dotted line.

Hierarchy Simplification

The hierarchy is simplified using the method proposed by Campello et al. [9]. The simplification of HDBSCAN* hierarchy is based on an observation about estimates of the level sets of continuous-valued probability density function (p.d.f.), which refers back to Hartigan's [21] concept of *rigid clusters*. For a given p.d.f., there are only three possibilities for the evolution of the connected components of a continuous density level sets when increasing the density level.

Algorithm 2 HDBSCAN* Main Steps

Input: Dataset D, Parameter m_{pts}
Output: HDBSCAN* hierarchy

1: **procedure** HDBSCAN(m_{pts})
2: Given data set D, compute the distance matrix, X_{dist}
3: From X_{dist} compute the core distances of all the data objects in D, w.r.t. m_{mpts}.
4: Compute a Minimum Spanning Tree of the Mutual Reachability Graph, G_{mreach}
5: Extend the MST to obtain MST_{ext}, by adding a self loop edge" for each vertex with weight equal to that of its core distance, dcore(x_p).
6: Extract the HDBSCAN* hierarchy as a dendrogram from MST_{ext}.

(a) All the objects are assigned to the same label, thus forming the root of the tree.

(b) Iteratively remove all edges from MST_{ext} in decreasing order of weights.

(i) Edges with the same weight are removed simultaneously.

(ii) After removal of an edge, labels are assigned to the connected components that contain one vertex of the removed edge. A new cluster label is assigned if the component has at least one edge in it, else the objects are assigned a null label, indicating it to be a noise object.

7: **Output:***Hierarchy*
8: **end procedure**

- The component shrinks but remains connected, up to a density threshold, at which either
- The component is divided into smaller ones, called a true split", (or)
- The component disappears.

Based on the above conditions, the HDBSCAN* hierarchy is simplified into a hierarchy consisting only those levels where there is a true split or the levels at which an existing cluster disappears. The other levels of the hierarchy, where the noise objects are removed from the cluster (where a particular component has shrunk), are not explicitly maintained in a simplified hierarchy.

For a connected component to be considered a cluster, many applications require that there be a minimum number of data objects in a group. HDBSCAN* hierarchy accommodates this by the use of a parameter min_{clSize}. The parameter min_{ClSize} specifies the smallest size of a connected component to be considered as a cluster.

The step 6.(b).(ii) of Algorithm 2, can be generalized to accommodate the parameter min_{clSize} and is described in Algorithm 3.

Algorithm 3 HDBSCAN* with optional parameter $\text{min}_{clsize} \geq 1$

1: 4.2.2. After removal of each edge, process the cluster that contained the removed edge (one at a time) as follows:

(a) Label spurious sub-components as noise by assigning them the null label. If all the sub-components of a cluster are spurious, then the cluster has disappeared. A sub-component is termed as spurious, if the number of vertices's in the sub-component are less than min_{clsize}.

(b) If there is a single sub-component of a cluster that is not spurious, then the original cluster label is maintained. This means that the cluster has shrunk.

(c) If there are two or more sub-components of a cluster that are not spurious, assign new labels to each of them. This means that the parent cluster has not split into two clusters.

Extraction of Prominent Clusters

To extract prominent cluster from the cluster tree by non-overlapping partitioning, that is either a parent cluster or its child clusters can be selected for the partition, but not both. Campello et al. [10] employ a Framework for Optimal Selection of Clusters (FOSC). It uses a bottom-up approach to select the clusters with the best total score from the cluster tree.

Cluster stability is the scoring method used to score each cluster in a cluster tree. The intuition behind stability is that more prominent clusters tend to *survive* for a longer duration after they appear. Stability of a cluster depends on the lifetime of a cluster as well as the individual density profiles of all data objects present in that cluster. This is because each data object belonging to a cluster can become noise at a density different from the density at which the cluster splits or disappears. The contribution of a data point X_o that belongs to cluster 'C_i' is equal to the difference between the density level at which X_o becomes a member of C_i and the density level at which X_o is no longer a member of C_i, and is defined as

$$\lambda_{max}(X_o, C_i) - \lambda_{min}(C_i) \tag{2.13}$$

where, $\lambda_{max}(x_p; C_i)$ is the maximum density at which the data object X_o belonged to the cluster, i.e., the density at which the object X_o or cluster C_i disappears or the density at which the cluster membership of the object is changed; and, $\lambda_{min}(C_i)$ is the threshold at which the cluster C_i first appeared.

Using the definition 2.13 , the stability $S(C_i)$ of a cluster C_i can be defined as

$$S(C_i) = \sum_{x_j \in C_i} \left(\lambda_{max}(x_p, C_i) - \lambda_{min}(C_i) \right) = \sum_{x_j \in C_i} \left(\frac{1}{\epsilon_{min}(x_j, C_i)} - \frac{1}{\epsilon_{max}(C_i)} \right) \tag{2.14}$$

Using this definition of cluster stability, we extract cluster from the cluster tree, with maximized overall aggregate stabilities.

Process every node except the root, starting from the leaf clusters (bottom-up), deciding at each node C_i whether C_i or the best-so-far selection of clusters in C_i's subtree should be selected. This is done by comparing the score of

cluster C_i with the combined total score of the best set of descendant clusters of C_i found so far (for leaf clusters, there will be no best set of descendants, and for the parents of leaf clusters, the best set of descendants will simply be its children). The group of clusters which has a higher total score is then passed up the tree to the clusters parent. This process continues until a final set of clusters with the highest possible total score is passed to the root cluster. This cluster set forms the flat partition, and labels are assigned correspondingly to its cluster members Figure 2.18 provides an example cluster tree.

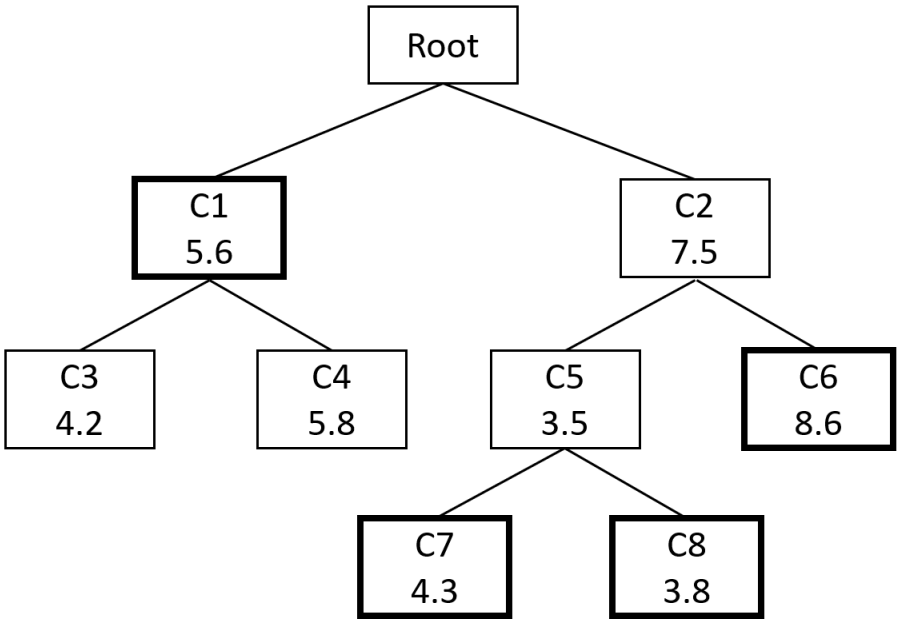


Figure 2.18: An example of cluster tree. Clusters with their stability values. Selected clusters (C1,C6,C7,C8) are in bold

2.2.4 Approximate Hierarchical Clustering (MS-Cluster)

Frank et al. [18] describe a new clustering method called Approximate Hierarchical Clustering (AHC). The MS-cluster approach uses this clustering algorithm to cluster mass spectra. This project uses this clustering technique as one of the methods. Algorithm (4) gives a simple code description for AHC.

The algorithm starts with the list of clusters consisting of all data objects in the dataset as singletons. It performs r rounds of clustering with decreasing similarity threshold τ. In each round, cluster C is compared sequentially to

Algorithm 4 A pseudo algorithm for approximate hierarchical clustering

Input: DataSet D, τ_{min},r
Output: Cluster Labels

```
1: procedure MS-CLUSTER(cc)
2:     Initialization
3:     δ ← (1−τ_min)/r
4:     Clusters ← {{X_1}, {X_2}, ..., {X_n}}
5:     r' ← 1
6:     while r'≤ r do
7:         for every cluster c in Clusters do
8:             for every cluster c' preceding c in Clusters do
9:                 if similarity (c,c') >τ then
10:                    append c to c'
11:                    remove c from Clusters
12:                end if
13:            end for
14:        end for
15:        r'=r'+1
16:    end while
17:    Output:Clusters
18: end procedure
```

clusters preceding in the cluster list. If the similarity between C and C' is above the set similarity threshold, then merge the two clusters, i.e., append data points in C to data points in C' and remove the cluster C from the cluster list. The similarity between two clusters is the similarity between their cluster representative.

2.2.5 Neighbor clustering (N-cluster)

The idea behind the N-Cluster algorithm, proposed by [29] Rieder et al. is that the center of a cluster should have many neighbors within some distance threshold c. The method works as follows. For every point, compute the number of neighbors within the distance of c. Select the point with the highest number of neighbors. Assign a new cluster label to all the points in the neighborhood of this point along with the point itself. Mark the points with only one neighbor (the point itself) as singletons. Remove all the points which are assigned a label from the dataset, and repeat the procedure on the remaining

dataset until all the points have a cluster label. The pseudo code for this algorithm is described in algorithm 5.

Algorithm 5 A pseudo algorithm for Neighbor clustering N-cluster

Input: Data Set D, c
Output: Cluster Labels

```
1: procedure N-Cluster(cc)
2:     for All points not labeled, calculate the number of neighbors in an c distance. do
3:         Elements with only one neighbor are singleton clusters; assign them individual cluster label (singleton cluster).
4:         Select the point with most neighbors and its neighbors and assign them a new cluster label.
5:         Remove the clustered points from the dataset D.
6:         Repeat steps 2 - 4 until all points are assigned a label.
7:     end for
8:     Output:Clusters
9: end procedure
```

2.2.6 Cluster validation

Internal validation is one type of clustering evaluation, which evaluates the goodness of clustering without any external information. The two internal cluster validation measures that we use to evaluate clustering methods are:

- Silhouette Width Criterion (SWC)
- Density based Cluster validation (DBCV)

We choose SWC because it is one of the commonly employed measures to validate clustering results. We also chose DBCV another cluster validation technique, as it is shown to perform better for density-based clusters [26].

Silhouette Width Criterion (SWC)

The silhouette width criterion [31] measures how similar an object is to other objects in the same cluster (cohesion) compared to other clusters (separation). The silhouette width of a point $X_i \in$ cluster C_k is defined as

$$s(i) = \frac{b(i) - a(i)}{\max((a(i), b(i)} \tag{2.15}$$

where,

$a(i)$ is defined as the mean distance of point the X_i to the other points in C_k.

$b(i)$ is defined as the smallest average distance of X_i to all points in any other cluster, of which X_i is not a member.

If $d(X_i, C_i)$ represents average similarity of X_i to all points of C_i, where $i \neq k$, then $b(i)$ is denoted as $\min_{C_i i \neq k}$ (d(i,C_k))

Silhouette width of the point is a quantity between -1 and 1: a value near 1 indicates that the point X_i is assigned to the right cluster whereas a value near -1 suggests that the point should be assigned to another cluster.

The mean of the silhouette widths for a given cluster C_k is called the *cluster mean silhouette* and is denoted as s_k:

$$s_k = \frac{1}{n_k} \sum_{i \in C_k} s_i \tag{2.16}$$

Finally, the *global silhouette index* is the mean of the mean silhouettes through all the clusters:

$$C = \frac{1}{K} \sum_{k=1}^{K} s_k \tag{2.17}$$

Adjusted Silhouette coefficient

Partitions generated with density-based algorithms like DBSCAN and HDBSCAN* may contain noise points. Silhouette width criterion when applied directly to the partitions generated by density-based clustering, would give a wrong measure, as it would consider all noise points as a cluster. Silhouette coefficient is not equipped to handle noise. To make a fair comparison with other clustering algorithms which do not produce noise, we need to find a corrective measure. One common remedy is to assign each noise point to a singleton cluster, but doing this will result in reduced overall separation, degrading the results of the measure. We follow the approach suggested by Moulavi et al. [26], where we remove all noise points, calculate the SWC and

penalize the score for lack of coverage. This approach helps to deal with noise appropriately. We first evaluate the silhouette measures only on cluster points and multiply it with the correcting coefficient given by equation 2.18

$$\text{correcting coefficient} = \frac{|O| - |N|}{|O|}, \tag{2.18}$$

where |O| is the number of objects in the dataset, |N| number of noise points.

Density based Cluster validation (DBCV)

Density-based cluster validation (DBCV) is developed by Moulvi et al. [26] explicitly for density-based clustering methods. It takes into account the properties of density and shape of clusters and also deals with noise. DBCV is defined in terms of the lowest density region in each cluster and the highest density region between pairs of clusters. We explain DBCV in detail as presented in [26]

Definition 15. Core Distance of an Object : The all-points-core-distance (inverse of the density) of an object o, belonging to cluster C_i w.r.t. all other n_i-1 objects in C_i is defined as:

$$a_{pts}coredist(o) = \left(\frac{\sum_{i=2}^{n_i} (\frac{1}{KNN(o,i)})^d}{n_i - 1} \right)^{-\frac{1}{d}} \tag{2.19}$$

Definition 16. Mutual Reachability Distance : The mutual reachability distance between two objects o_i and o_j in O is defined as

$$d_{mreach}(o_i; o_j) = \max(a_{pts}coredist(o_i); a_{pts}coredist(o_j); d(o_i; o_j)) \tag{2.20}$$

Definition 17. Mutual Reachability Distance Graph : The Mutual Reachability Distance Graph is a complete graph with objects in O as vertices and the mutual reachability distance between the respective pair of objects as the weight of each edge.

Definition 18. Mutual Reachability Distance MST : Let O be a set of objects and G be a mutual reachability distance graph. The minimum spanning tree (MST) of G is called MST_{MRD}

Definition 19. Density Sparseness of a Cluster : The Density Sparseness of a Cluster (DSC) C_i is defined as the maximum edge weight of the internal edges in MST_{MRD} of the cluster C_i, where MST_{MRD} is the minimum spanning tree constructed using a_{pts}coredist considering the objects in C_i.

Definition 20. Density Separation: The Density Separation of a Pair of Clusters (DSPC) C_i and C_j , $1 \leq i; j \leq n; i \neq j$, is defined as the minimum reachability distance between the internal nodes of the MST_{MRD^s} of clusters C_i and C_j .

Definition 21. Validity Index of a Cluster: We define the validity of a cluster C_i, $1 \leq i \leq n$, as:

$$V_c(C_i) = \frac{\min\limits_{1 \leq j, j \neq i} \Big(DSPC(C_i, C_j \Big) - DSC(C_i)}{\max \left(\min\limits_{1 \leq j, j \neq i} \Big(DSPC(C_i, C_j) \Big), DSC(C_i) \right)} \tag{2.21}$$

Definition 22. (Validity Index of a Clustering) The Validity Index of the Clustering Solution $C = C_i$, $1 \leq i \leq n$ is defined as the weighted average of the Validity Index of all clusters in C.

$$DBCV(C) = \sum_{i=1}^{i=n} \frac{|C_i|}{|O|} V_C(C_i) \tag{2.22}$$

DBCV index value lies in-between -1 and 1, and a higher value indicates better density based clusterings. A positive validity index of a cluster means that the cluster has better density compactness as compared to separation, on the other hand, a negative validity of index of a cluster means that the density inside a cluster is lower than the density that separates it from other clusters.

Chapter 3

Methodology

In this section, we define a new methodology to extract clusters from HDBSCAN* hierarchies. Furthermore, we describe all pre-processing and post-processing methods required to carry out experiments successfully.

3.1 Pre-Processing

The output of a mass spectrometry experiment is in the form of a raw file. The raw file is filtered and processed before analyzing. The raw files are converted to mzXML file format using msConvert [3]. We convert this mzXML file to individual text files with extension *.dta*. Each *dta* file contains a peak list representing a single mass spectrum (MS/MS). The first line of this file contains the mass of a singly protonated peptide (MH+) and the peptide charge state as a pair of space-separated values. Subsequent lines contain space separated pairs of fragment ion m/z and intensity values (I). Figure 3.1 depicts the peak list in the *.dta* file.

The number of peaks in each spectrum can vary from a few hundred to thousands. Most of these peaks are noise peaks. Thus, a method to remove unwanted noise peaks is profitable. From the literature, we follow two approaches to prune the noise peaks.

3.1.1 Top k peaks:

Frank et al.[18] choose the k strongest peaks, where k depends on the mass of the peptide also called as parent mass. They propose to select 15 peaks per

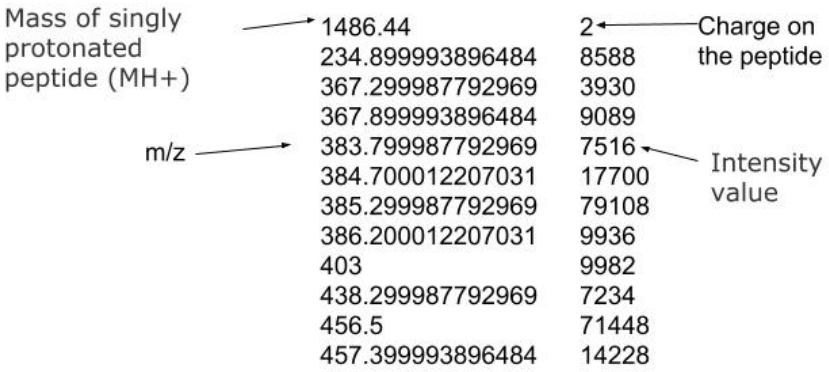


Figure 3.1: Peak list in a '.dta' file.

1000 Da of the parent mass. Equation 3.1 shows how to calculate the number of peaks k for each spectrum. Each spectrum is represented by the calculated top k strongest peaks.

$$k = \left\lfloor j \frac{Parent\ Mass\ (inDa)}{1000} * 15 \right\rfloor \tag{3.1}$$

where, $\lfloor X \rfloor$ denoted the integer part of X.

3.1.2 Top 20 peaks:

It is intuitive that two spectra that belong to the same peptide (have the same amino acid sequence) would a have strong correspondence between their N strongest peaks. N was found out to be 20 by Bandeira et al. [6] by analyzing the peak annotation histogram of spectra, which exhibited an extremely low percentage of b/y ion peaks outside top 20 intensity peaks.

3.1.3 Prefix Residue Mass Spectrum:

Given a spectrum S having peaks at masses $\{m_1, m_2, m_3..., m_n\}$ with intensity $\{I_1, I_1..., I_n\}$, and peptide mass equal to M_p, the inverse of S is defined as $\bar{S}$ having masses in $\{\bar{m_1}, \bar{m_2}, \bar{m_3}..., \bar{m_n}\}$ with intensities $\{I_1, I_1..., I_n\}$ where $\bar{m_i} = M_p - m, for 1 \leq i \leq n$.

The Prefix Residue Mass Spectrum (PRM) of a spectrum S is given by:

$$PRM(S) = S \cup \bar{S} \tag{3.2}$$

This method cannot be applied to the experimental spectra since an experimental spectrum contains m/z instead of mass for a peak. As we know, the peptides with a precursor charge +2, predominately produce peaks of charge +1, which makes m/z equals to m. Thus we calculate the PRM spectrum only for those spectra that have a peptide charge of +2. Each peak in a spectrum S with a peptide charge of +2, results in two peaks at, one at m and one at $m_P - m$ with the same intensity as in spectrum S, in the PRM spectrum. This PRM spectrum is symmetric around $m_p/2$.

3.2 Similarity as a Distance measure

A distance measure is crucial for any clustering algorithm. All clustering algorithms require some measure to determine (dis)similarity between the data points in order to cluster them. The performance of the clustering technique is heavily dependent on how good this measure is. We experiment with two different kinds of similarity measures which **we** transformed into distance measures:

- Cosine similarity.
- Spearmans correlation.

3.2.1 Converting a spectrum into its intensity vector

To calculate the similarity between two spectra S, S' (where each spectrum is a list of $[m/z, I]$ values), we reduce each spectrum to an intensity vector. To accomplish this, we first join the two sets of m/z values into a sequence M and filter out duplicate elements. Then we sort the elements in M according to their values to obtain the sequence $M' = [m_1, m_2, m_3, ...m_n]$. From the smallest value of M', m_1 to the largest value of M', m_1, we create bins of size 0.5, starting at the $\lfloor m_1 \rfloor$ to $\lceil m_n \rceil$.

Each bin represents one element (axis) of the vectors s and $s^{'}$ of the length $2 * (\lceil m_n \rceil - \lfloor m_1 \rfloor)$. For each bin, we check if S has peaks in the m/z range of the bin. If yes, then fill in the intensity value of the bin as the sum of all the

intensities of the peaks found in the range of the bin, else if there is no peak in the m/z range of the bin, we just set the value of the bin in vector s to 0. Similarly we calculate the vector s' for S'.

Note that, we consider two peaks to be comparable if their m/z values lie between 0.5 m/z units.

3.2.2 Cosine Similarity

The normalized dot product is used as a measure of similarity between two MS/MS spectra. This measure has been found to work with several groups approaching similar problems [7, 18, 28, 32]. The normalized dot product between two spectra S and S', represented by intensity vectors s_i and s'_i, respectively, is given by:

$$Similarity(S, S') = \frac{\sum_{i=1}^{t} s_i.s'_i}{\sqrt{\sum_{i=1}^{t}(s_i)^2 . \sum_{i=1}^{t}(s'_i)^2}} \tag{3.3}$$

This similarity between two spectra takes values that range from 0 (no common peak or different spectra) to 1 (total similarity). Equation 3.4 gives the dissimilarity between a pair of spectra and is used as a distance measure to carry out clustering.

$$Distance(S, S') = Dissimilarity(S, S') = 1 - Similarity(S, S') \tag{3.4}$$

3.2.3 Spearman Correlations

Pearson correlation between two spectra S and S' represented by intensity vectors s_i and s'_i, respectively, is the covariance of the two vectors divided by the product of the standard deviations and is defined by

$$r_{SS'} = Corr(s_i, s'_i) = \frac{Cov(s_i, s'_i)}{\sqrt{Var(s_i)}\sqrt{Var(s'_i)}} \tag{3.5}$$

Spearman correlation between S and S', given by $\rho(S, S')$ is a nonparametric version of the Pearson correlation and is defined as the Pearson correlation coefficient between the ranks of two sequences of intensities. As Spearman correlation does not assume that both datasets are normally distributed as

compared to Pearson correlation, we choose Spearman correlation as another method to calculate the similarity between two spectra. The value of the Spearman correlation varies between -1 and +1 with 0 implying no correlation. A correlation value of -1 or +1 implies an exact monotonic relationship.

Equation 3.6 gives the dissimilarity between a pair of spectra and is used as a distance measure to carry out clustering.

$$Distance(S, S') = Dissimilarity(S, S') = 1 - |\rho(S, S')| \tag{3.6}$$

where, $|X|$ denotes the absolute value of X.

3.3 Representative Consensus spectrum:

In this section, we describe two methods to generate a cluster representative called consensus spectrum. This consensus spectrum is searched against a spectral database to yield a match. A cluster representative (consensus spectrum) is generated by combining all the members of a cluster. We experiment with two known methods from the literature to generate a cluster representative, described below:

3.3.1 Average Consensus method:

Beer et al.[7] proposed a method of calculating cluster representative by the summing the intensity of peaks whose m/z is within the tolerance of 0.4 Da units. The m/z value of these peaks in the consensus spectrum is simply the average of m/z of the joined peaks.

3.3.2 Weighted Consensus method:

Another way to combine all the spectra, that belong to a cluster into a consensus spectrum is given by Frank et al. [18]. This method consolidates the peaks of all spectra in the cluster as a weighted average. Each consensus peak is assigned the m/z that equals the weighted average of the joined peaks' masses and intensity that equals the sum of the joined peaks' intensity (Two peaks are considered the same if the m/z is within 0.4 Da.).

Figure 3.2 shows how peaks are merged and a consensus spectrum is generated.

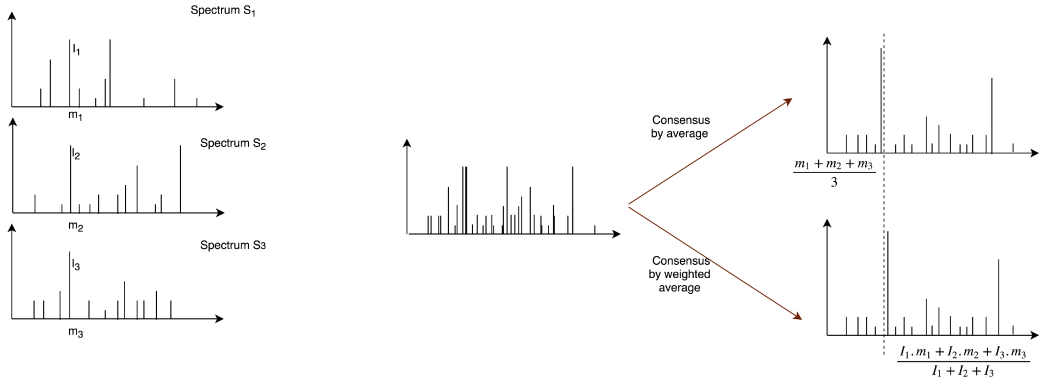


Figure 3.2: Methods to generate consensus spectra from spectra S_1, S_2, and S_3. First, all the peaks in these three spectra are merged into one spectrum. Due to mass error, we get bundles of peaks, instead of one single peak. These bundles of peaks are resolved into a single peak, either by using the average consensus method or by using the weighted consensus method.

3.4 Application of a clustering method to Tandem Mass Spectra

Figure 3.3 illustrates the pipeline followed to carry out experiments. An experiment starts with the conversion of raw data acquired from a mass spectrometry experiment to individual files representing one spectrum in the form of a peak list. Each spectrum in a dataset is pre-processed using the preprocessing techniques described above. A distance matrix is computed between these processed spectra, which is used by different clustering methods described above. Cluster representatives (consensus spectra) are generated from the clustering output. These consensus spectra are searched against their respective protein sequence database in target/decoy mode for a match. Matches were accepted with thresholding at ≤ 2 % FDR.

Along with using the already existing clustering methods, we propose a new method to extract clusters from HDBSCAN* hierarchies, which is described below.

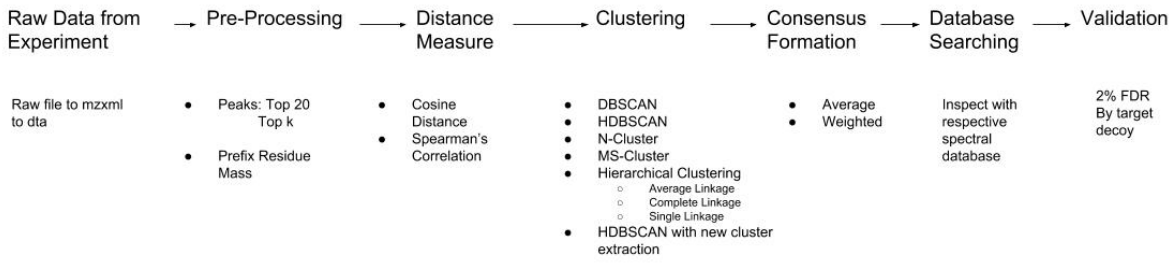


Figure 3.3: Pipeline of the entire experiment

3.5 Modification to HDBSCAN* cluster extraction

HDBSCAN* and DBSCAN clustering techniques discover arbitrarily shaped density-based clusters. By definition, all the points in a cluster discovered by HDBSCAN* and DBSCAN are density connected. In some cases, at a specific density, two points in the same cluster may not be similar enough to belong to the same group for this application. For example, if we look at the figure 3.4, HDBSCAN* and DBSCAN would place the points X_p and X_q into the same cluster. For this application, X_p and X_q represent two different spectra, which are very distant and have low similarity to each other. This results in a heterogeneous cluster of spectra (the cluster contains, spectra that belong to different peptide sequences), which is undesirable.

To cope with this issue, we propose to limit the diameter of each cluster to a selected $\max_{diameter}$. We do this by extracting clusters, whose diameter is not more than the set $\max_{diameter}$, from the HDBSCAN* hierarchies. The diameter of a cluster C_i is give by the definition 23.

Definition 23. Diameter of Cluster: Diameter of cluster $C_i = \{X_i\}$ i≥ 1 and i $\leq$ n is defined as the maximum distance between all pairs of points in the cluster C_i; and it is represented by the equation 3.7.

$$diameter(C_i) = \begin{cases} max(dist(X_i, X_j), \text{ where } X_i, X_j \in C_i, \text{ if } C_i \text{ is a non leaf node.} \\ 0, \text{ when } C_i \text{ is a leaf node.]} \end{cases} \tag{3.7}$$

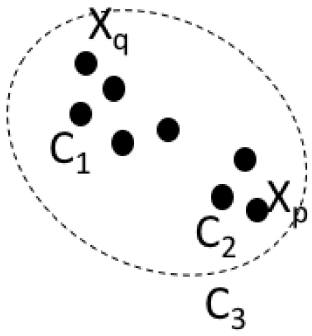


Figure 3.4: Point 'X_p' and 'X_q' are density-connected to each other, but they are not similar.

Our approach uses a bottom-up approach to extract clusters from the hierarchy. First, the hierarchy is simplified into a cluster tree with additional parameter $min_{Clsize} = 1$ using the algorithm described in 3. Nodes in the hierarchy are processed starting from leaf nodes, going up until the diameter of a clusters' parent is greater than $\max_{diameter}$ and the diameter of the cluster is less than $\max_{diameter}$. When this condition is satisfied, then the cluster is selected as part of the partition. For a leaf node, if the diameter of its parent is greater than $\max_{diameter}$, then all the children of this parent node are assigned a null label (noise cluster). Algorithm 6 describes a pseudo code to extract clusters from an HDBSCAN* hierarchy having a diameter not greater than $\max_{diameter}$.

Note: For min_{pts} equal to two, this method is equivalent to extracting clusters from a single linkage hierarchy.

3.5.1 How is this method of extraction different from extracting cluster by making an horizontal cut?

The HDBSCAN* hierarchy is a single linkage hierarchy on a transformed distance matrix (mutual reachability distance). For min_{pts} equal 2, it is equal to a single linkage hierarchy. For the sake of simplicity, we discuss the case where $min_{pts}$is equal to 2, where the mutual reachability distance and distance between the points is equal. When clusters are extracted by making a hori-

Algorithm 6 A pseudo algorithm for extracting clusters of specific diameter

Input: Cluster Tree, $\max_{diameter}$
Output: Cluster Labels

procedure EXTRACT CLUSTER(Cluster Tree)
    for all leaf nodes which have not been assigned a labels **do**
        Calculate diameter_{parent} = diameter(parent node)
        if $\text{diameter}_{parent} \geq \max_{diameter}$ **then**
            Assign all points in the parent node a noise label.
        else
            keep traversing up the cluster tree until the
condition diameter(clusters' parent)$\geq \max_{diameter}$ and
diameter(cluster) $\leq \max_{diameter}$ is TRUE:
            Select the child cluster and assign a cluster label to all the points in this subtree.
        end if
    end for
    Output:*Clusters*
end procedure

zontal cut in the hierarchy at some distance threshold d, the diameter of these clusters may or may not be $\leq d$. By making a horizontal cut at a distance of d all the edges with weight $\geq d$ are removed from the MST. Each of the connected components is considered as clusters.

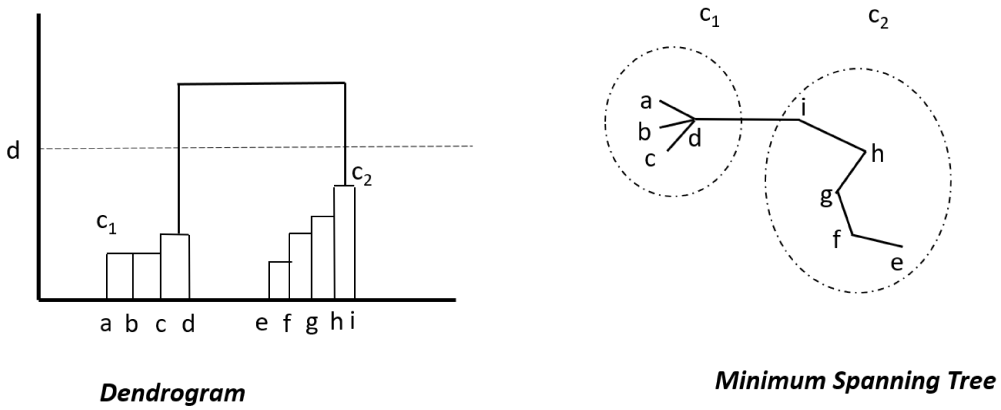


Figure 3.5: Dendrogram and the corresponding MST.

By construction, the weight of all the edges in the MST of connected components is $\leq d$, which means that the distance between two points is $\leq d$ only

if there exists an edge between them in the MST. The distance between other pairs of points in the cluster could be $\geq d$. Thus, it is very much possible that the diameter of a cluster can be greater than d. On the other hand, our method guarantees that the diameter of each cluster can never be greater than d.

Figure 3.5 shows a dendrogram and the MST for 9 points, namely a, b, c, e, f, g, h, i. in a 2D space. A cut is at a distance d, forms two connected C_1 and C_2. Amongst all pairwise distance in C_2, the distance between the points e and i is largest (because of the chain). Thus making the distance between the points e and i as the diameter of the cluster C_2.From the figure, it is evident that the distance between the points i and e is $\geq d$.

Chapter 4

Experiments, Results and Discussion

4.1 Experimental setup :

In this section, we describe all the details about the data and how experiments were carried out.

We use the published data [30] containing 27 MS/MS runs sampled from five sequenced organisms and four organisms without a sequenced genome. For our experiments, we use two MS/MS datasets from this published data belonging to two different organisms out of five, namely, (i) human (Homo sapiens, H, HeLa cell line), (ii) roundworm (Caenorhabditis elegans, C). Along with these, we use a smaller data set of 500 MS/MS spectra collected by Dr. Zukui Li at the University of Alberta. These spectra belong to 89 different peptides. We carry out pilot experiments on this smaller dataset. From the result of the pilot experiments, we select the best similarity and consensus generating method and parameters for clustering. The tests on the more massive datasets are carried out based on these selections.

The techniques described in Chapter 3, give us four different ways to pre-process the dataset. The first way is to transform each spectrum in a dataset to its strongest 20 peaks (we refer to this pre-processing as "top20"). The second way is to transform each spectrum in a dataset to its strongest k peaks where k depends on the mass of the precursor ion (we refer to this pre-processing as "top k"). Additionally, each spectrum pre-processed using the "top 20" and

"top k" can further be pre-processed to their PRM spectra. This generates two more pre-processed datasets, i.e., "top 20 PRM" and "top k PRM."

To speed up the calculation of the distance matrix, we use a heuristic where the similarity between the two spectra is only calculated if the precursor peptide mass of two spectra is within 2.5 Da (this will allow isotopes of a peptide to be in the same cluster); else the similarity is directly set to 0 for Spearman's correlation and 1 for cosine (most dissimilar).

The InsPect database search tool is used to perform peptide identification using default search parameters (precursor mass in the tolerances of 2.5Da and fragment ion tolerance of 0.5Da). The shuffle decoy database is created and added to the sequence database by the InsPect tool. The concatenated database (decoy and target) is used to perform peptide identification. The InsPecT F-score threshold value for accepting identifications was selected to ensure the true positive identification rate of 98%. That is 2% FDR.

Table 4.1 illustrates the sequence databases used for identification of peptides. These sequence databases (in the form of FASTA file format) are downloaded from UniProt, the universal protein knowledgebase [5]. UniProt has two sections, one that is manually reviewed and curator-evaluated and other that is computationally analyzed. Swiss-Prot is the manually annotated and reviewed section of the UniProt Knowledgebase. TrEMBL is the unreviewed section. As the origin of the smaller dataset is unknown, it is searched against all the entries in the swiss-port(Reviewed database). All the other data sets are searched against a combined database of Reviewed (Swiss-Prot) and Unreviewed (TrEMBL) of their respective sequence.[1]

Different data sets are evaluated to compare the clustering solutions produced by different clustering algorithms. By default, and unless explicitly stated otherwise, all experiments compare the quality of results regarding the number of unique peptides identified.

Experiments with different clustering methods are initially carried out on

[1]Reviewed (Swiss-Prot) - Manually annotated Records with information extracted from the literature and curator-evaluated computational analysis. Unreviewed (TrEMBL) - Computationally analyzed records that await full manual annotation.

Database	Number of Target Sequences.
Swiss-Prot (Reviewed)	555100
Human (Reviewed (Swiss-Prot) +Unreviewed (TrEMBL)	163115
Worm (Reviewed (Swiss-Prot) +Unreviewed (TrEMBL)	30847

Table 4.1: Protein databases used for database searches for different samples.

Clustering Method	Parameters	
MS-Cluster	Similarity Threshold : Number of Rounds:	0.4, 0.5, 0.6, 0.7, 0.8, 0.9 3
N-cluster	Distance Threshold :	0.05, 0.075, 0.095, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5
DBSCAN	$\min_{pts}$: ϵ :	2, 3, 4 0.1, 0.2, 0.3, 0.4, 0.5
HC- average	Threshold :	0.1, 0.2, 0.3, 0.4, 0.5
HC- complete	Threshold :	0.1, 0.2, 0.3, 0.4, 0.5
HC- single	Threshold :	0.1, 0.2, 0.3, 0.4, 0.5
HDBSCAN	$\min_{pts}$: $\min_{Clsize}$:	2, 3, 4 2, 3, 4
HDBSCAN-diameter	$\min_{pts}$: diameter :	2, 3, 4 0.075, 0.095, 0.1, 0.15, 0.2, 0.25, 0.3

Table 4.2: Parameter settings used for different clustering algorithms, when applied on smaller dataset of 500 spectra.

the smaller dataset of 500 spectra using the parameters described in table 4.2. These experiments are carried out on all possible combinations of four pre-processing methods, two similarity measures and two cluster consensus methods described in Chapter 3.

We use SWC to evaluate the partitions produced by different clustering techniques, on varied parameter settings. Additionally, we use DBCV to evaluate all density-based clustering algorithms.

We compare the results of different clustering algorithm on the more massive datasets based on the chosen similarity measure and cluster consensus method chosen for the smaller dataset. Table 4.3, shows the parameter values used by different clustering methods to cluster the data in the larger dataset.

For comparison on the smaller dataset, we use the clustering method AHC of MS-Cluster with our preprocessing and similarity method. For the larger datasets, MS-Cluster is implemented exactly as described in the paper.

Clustering Method	Parameters	
MS-Cluster	Similarity Threshold : Number of Rounds:	0.7, 0.75, 0.8, 0.85, 0.9, 0.95 3
N-cluster	Distance Threshold :	0.05, 0.1, 0.15, 0.2, 0.25, 0.3
DBSCAN	$\min_{pts}$: Epsilon :	2, 3 0.05, 0.1, 0.15, 0.2, 0.25, 0.3
HC- average	Threshold :	0.05, 0.1, 0.15, 0.2, 0.25, 0.3
HC- complete	Threshold :	0.05, 0.1, 0.15, 0.2, 0.25, 0.3
HC- single	Threshold :	0.05, 0.1, 0.15, 0.2, 0.25, 0.3
HDBSCAN	$\min_{pts}$: $\min_{Clsize}$:	2, 3 2, 3
HDBSCAN-diameter	$\min_{pts}$: diameter :	2, 3 0.05, 0.1, 0.15, 0.2, 0.25, 0.3

Table 4.3: Parameter settings used for different clustering algorithms, when applied on bigger dataset of tandem mass spectra.

4.2 Results and discussion

Amongst all the experiments that were carried out on the smaller dataset of 500 spectra, we discuss the results of clustering methods using cosine distance as similarity and the average consensus method to compute consensus spectra. The results of all other experiments on this smaller dataset are in the appendix. We show the results of each clustering method only for the parameter which identifies the highest number of peptides, to show the potential of the method to aid peptide identification.

Figure 4.1 shows the results of seven clustering methods, along with our method to extract cluster representative from HDBSCAN* hierarchies when applied on the smaller dataset of 500 spectra, with different pre-processing methods. (Here, the cluster consensus is calculated by average consensus method and the distance measure used is cosine distance.) For most of the clustering methods, We observe an increase in the number of unique peptides identified when the consensus spectra are searched against a database, as compared to non-clustered data. We observe that our clustering method reduces the number of spectra submitted for a database search, on an average **of** 50 % on the smaller dataset. It is also evident that our method of extracting clusters from the HDBSCAN* hierarchies performs better than the clustering methods

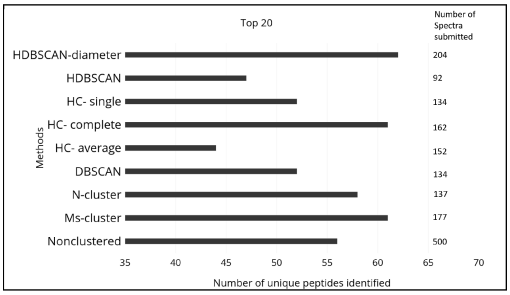
defined in the proteomic literature and is comparable to other clustering methods from the data mining literature. Additionally, preprocessing a spectrum into their PRM spectrum almost always benefits the peptide identification as compared to not processing them.

Figure 4.2 and 4.3, shows the comparison between the number of unique peptides identified when the partition with largest DBCV and SWC values are selected, and the highest number of unique peptides identified using the same clustering method (different partition).

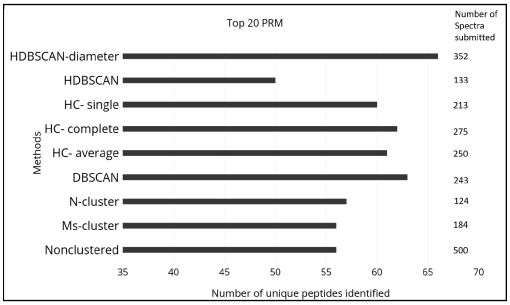
Figure 4.4 shows the results of eight clustering methods when applied to the dataset that has samples analyzed from humans. We see an increase in the number of unique peptides identified over non-clustered data for most cases. Our method of extracting clusters from HDBSCAN* hierarchies outperform all the clustering methods. Figure 4.5, shows the results of eight clustering methods when applied to the dataset that has samples analyzed from roundworms. We observe a slight decrease in the number of unique peptides identified when clustered data is searched against a database as compared to non-clustered data. This decrease in the number of identification could be because of many reasons, which are discussed below in section 4.2.1. On this dataset, we see that clustering fails to improve the unique number of peptides identified. Although the clustering methods we use fail to increase the number of identified peptides, our clustering approach leads to a higher number of peptide identifications than MS-Cluster and N-cluster.

On the larger data set that has samples analyzed from humans and roundworm, we observe between 6 % and 10 % decrease in the number of spectra that are searched for peptide identification.

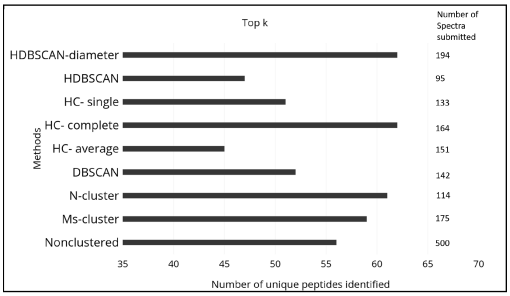
We also observe that HDBSCAN with stability as cluster extraction performs poorly as compared to all other clustering methods, almost all the time. It works better than hierarchical clustering with average linkage on the smaller dataset but works well for larger datasets. This irregularity could be due to the size of the dataset.



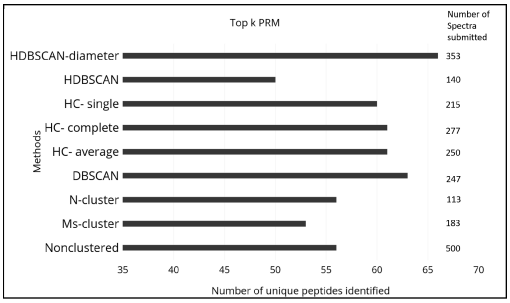
(a) Top 20 peaks.



(b) Top 20 PRM.



(c) Top k.

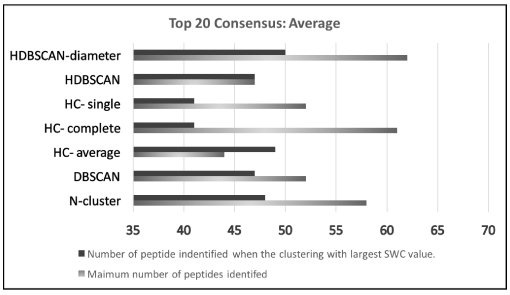


(d) Top k PRM.

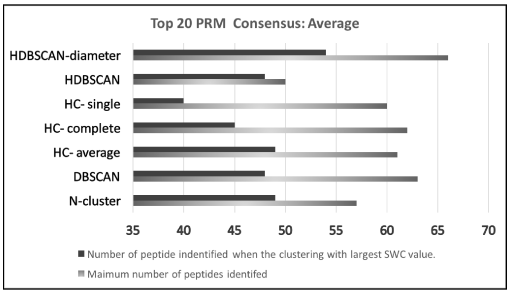
Preprocessing	Ms-cluster	N-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	Distance Threshold	(min_{pts}, ϵ)	Threshold	Threshold	Threshold	(min_{pts}, min_{clsize})	(min_{pts}, diameter)
Top 20	0.9	0.05	(2, 0.1)	0.1	0.1	0.1	(2, 2)	(3, 0.095)
Top 20 PRM	0.7	0.15	(3, 0.1)	0.1	0.1	0.1	(2, 2)	(3, 0.095)
Top k	0.9	0.075	(3, 0.1)	0.1	0.1	0.1	(2,3)	(2, 0.095)
Top k PRM	0.9	0.2	(3, 0.2)	0.1	0.2	0.1	(2, 2)	(3, 0.075)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

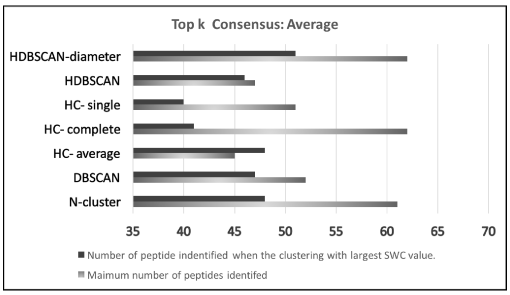
Figure 4.1: Results for smaller dataset of 500 spectra: Maximum number of peptides identified by 8 different clustering methods. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



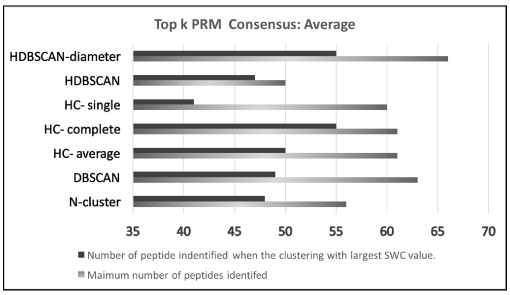
(a) Top 20.



(b) Top 20 PRM.

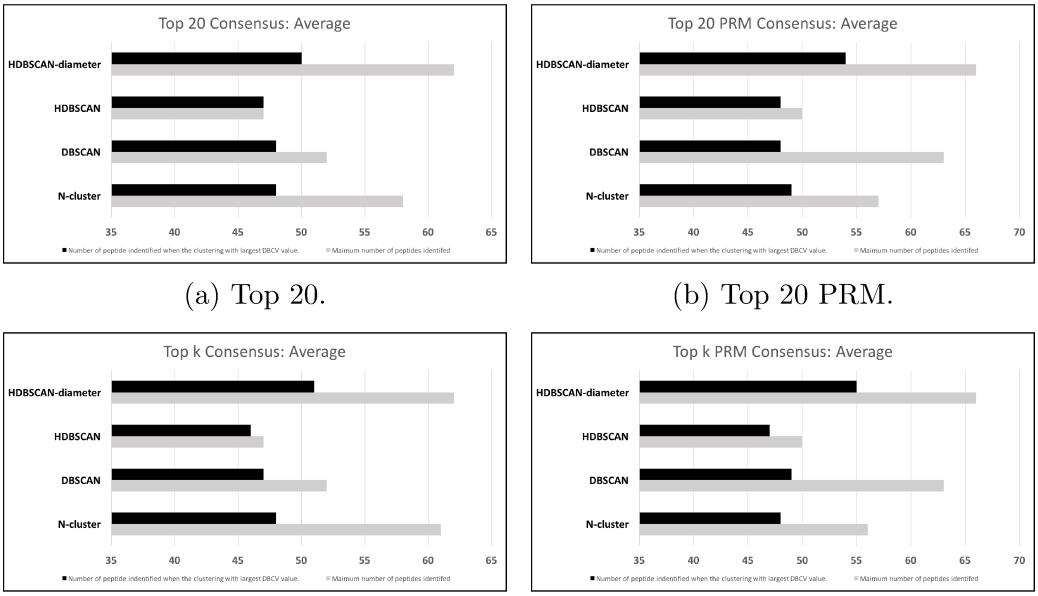


(c) Top k.



(d) Top k PRM.

Figure 4.2: Results for different cluster graph showing a comparison between the number of unique peptides identified when clustering with highest SWC is selected; as compared to the highest number of unique peptides identified by the same clustering method using other parameter for the smaller database. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



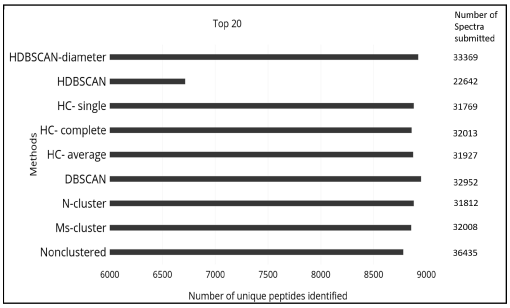
(a) Top 20.

(b) Top 20 PRM.

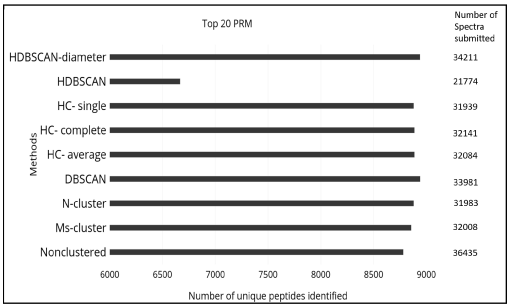
(c) Top k.

(d) Top k PRM.

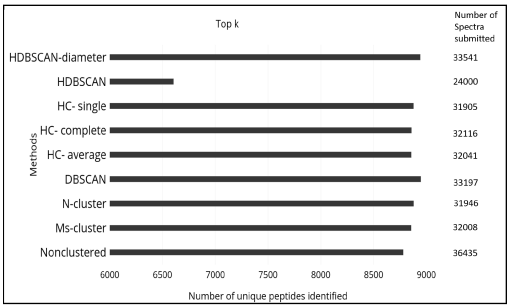
Figure 4.3: Results for different cluster graph showing a comparison between the number of unique peptides identified when clustering with highest DBCV is selected; as compared to the highest number of unique peptides identified by the same clustering method using other parameter for the smaller database. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



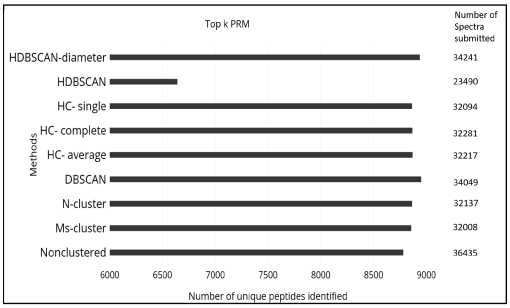
(a) Top 20.



(b) Top 20 PRM.



(c) Top k.

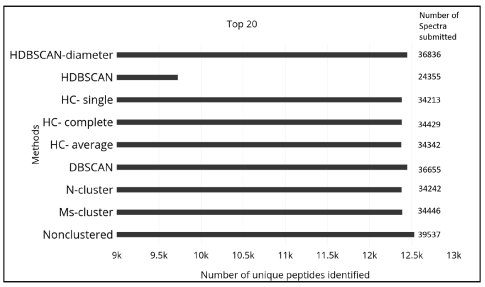


(d) Top k PRM.

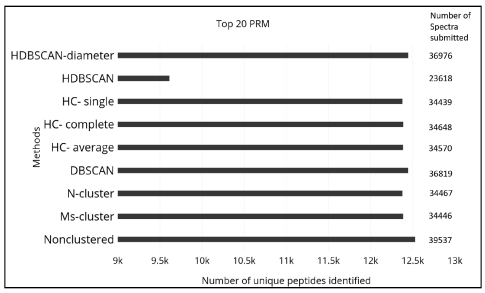
Preprocessing	Ms-cluster	N-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	radius	(min_{pts}, ϵ)	Threshold	Threshold	Threshold	(min_{pts}, min_{clsize})	(min_{pts}, diameter)
Top 20		0.05	(3, 0.1)	0.05	0.05	0.05	(3, 2)	(3, 0.1)
Top 20 PRM		0.05	(3, 0.05)	0.05	0.05	0.05	(3, 3)	(3, 0.05)
Top k	0.95	0.05	(3, 0.1)	0.05	0.05	0.05	(3, 2)	(3, 0.1)
Top k PRM		0.05	(3, 0.05)	0.05	0.05	0.05	(3, 2)	(3, 0.05)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

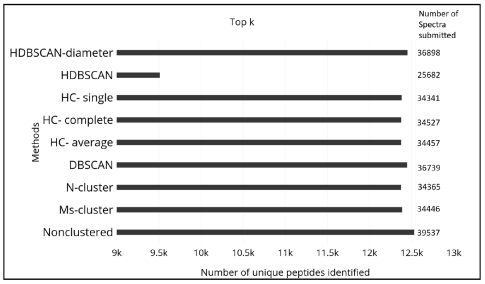
Figure 4.4: Results for human dataset: Maximum number of peptides identified by 8 different clustering methods. (Distance: Cosine distance; Cluster representative: Average Consensus Method.



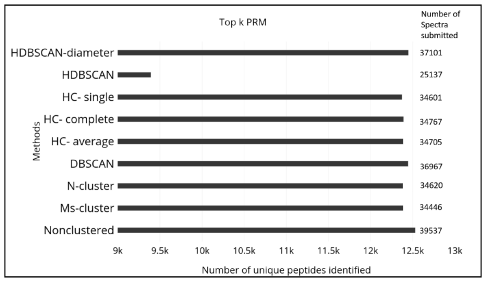
(a) Top 20.



(b) Top 20 PRM.



(c) Top k.



(d) Top k PRM.

Preprocessing	Ms-cluster	N-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	radius	(min_{pts}, ϵ)	Threshold	Threshold	Threshold	(min_{pts}, min_{clsize})	(min_{pts}, diameter)
Top 20		0.05	(3, 0.05)	0.05	0.05	0.05	(3, 2)	(3, 0.05)
Top 20 PRM		0.05	(3, 0.05)	0.05	0.05	0.05	(3, 2)	(3, 0.05)
Top k	0.95	0.05	(3, 0.05)	0.05	0.05	0.05	(3, 2)	(3, 0.05)
Top k PRM		0.05	(3, 0.05)	0.05	0.05	0.05	(3, 2)	(3, 0.05)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

Figure 4.5: Results for roundworm dataset: Maximum number of peptides identified by 8 different clustering methods. (Distance: Cosine distance; Cluster representative: Average Consensus Method.

4.2.1 How does clustering improve performance?

Often the spectra (peptides) from a mass spectrometry experiment show more similarity to other experimental spectra (peptides) obtained in the same or different experiments than to the theoretical spectrum (peptides) in a database. Spectra that are not identified when searched individually against a database can, however, get a match through cluster membership when the consensus spectrum get a match in the database. This identification through cluster membership accounts for the increase in the unique number of peptides identified, when clustered data is searched.

We observe cases where many spectra, when searched individually, never yield a match from the database, but when combined in the form of cluster consensus spectra, yield a match from the database through cluster membership.

We also observe a few unusual cases where some spectra were identified when submitted individually but, were not identified when using a clustering approach. This happens when they are part of clusters whose consensus spectra does not receive a match from the database.

There are three explanations for this; one reason could be the noise signals in the unidentifiable spectra in a cluster are so high that it cancels the true signals of the identifiable spectra in the consensus spectra. A second reason could be that the spectra which remained unidentified when searched individually are Protein post-translational modifications(PTMs). When the unmodified spectra are combined with these PTM spectra, the consensus spectrum can no longer be identified in a traditional database search, as it contains modified signals. A Third reason might be that the clustering results were not optimal due to the wrong choice of parameter. Estimating the optimal parameter values is a work in progress.

4.2.2 Why does internal cluster validation not work?

Parameter selection plays a crucial role in the output of any clustering algorithm. In order to find good parameter values for optimal clustering (evaluated

by the number of unique peptides identified), we experimented with two internal cluster validation measures namely SWC and DBCV, which are described in Chapter 3.

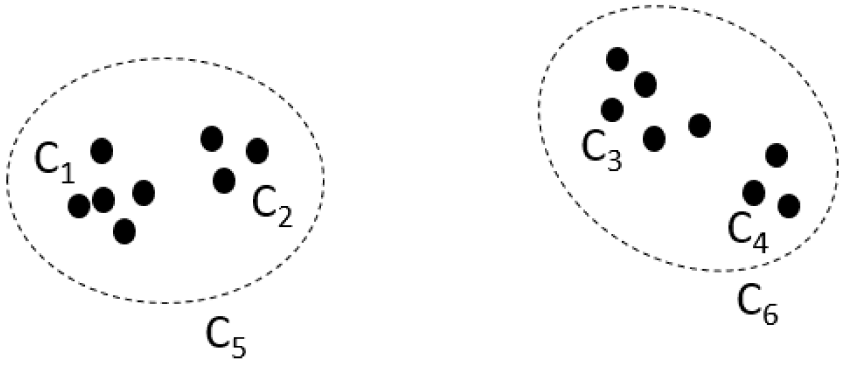


Figure 4.6: Two clusters C_5 and C_6 have high SWC and DBCV as compare to partition C_1, C_2, C_3 and C_4.

From the figures 4.2 and 4.3, we observe that the number of unique peptides identified, when we select the partition with largest SWC or DBCV value, is significantly lower than the best number of unique peptides identified that can be obtained by the same clustering method using a different parameter value.

For each partition, SWC measures compactness (how close the points are within a cluster) and Separation (distance between two clusters). DBCV measures, the density between two clusters (the maximum edge between two clusters in MST_{MRD}) and density within the clusters (maximum edge inside a cluster).

A group of spectra from the same peptide is typically very compact, but the group can still be relatively close to another group. Thus, making the two groups of peptides not very well separated while there may still be a large gap to the next closest group of spectra. Hence, well-separated clusters selected by SWC and DBCV on tandem mass spectra generally contains spectra from more than one peptide. The selected clusters tend to be too large and heterogeneous for this application, and contain multiple smaller significant clusters. Thus,

clusters selected by DBCV and SWC, are not indicative of a correct partition.

An illustration of the above observation can be seen from the figure 4.6. Notice that SWC and DBCV would have the highest value for the partition C_5 and C_6; but for the purpose of our problem, the ideal clusters would be C_1, C_2, C_3 and C_4.

4.2.3 Why does HDBSCAN with stability as cluster extraction yields poor results?

From the experimental results, we see that HDBSCAN with stability as cluster extraction performs poorly as compared to all other clustering methods, almost all the time. Cluster's stability is a measure of how long a cluster 'survives' within the hierarchy, as well as how many objects are part of the cluster.

For this application, clusters of peptides having the same amino acid sequence are very small and compact. They are generally at the lower levels of a cluster hierarchy.

Figure 4.7, illustrates nine mass spectra belonging to two different peptide sequences ,*GYSFVTTAER* and *DFPLANGER*, in a dendrogram. Cluster stability selects C_3 over C_1, and C_2, as it is more dense and stable for a longer amount of time. For our application, C_1 and C_2 are considered to be good clusters as C_1 represents the group of peptides having sequence *GYSFVTTAER* and C_2 represents *DFPLANGER*. From the figure, C_1 and C_2 joins into cluster C_3 at an epsilon value larger 0.3, where epsilon is the smallest distance between C_1 and C_2. From the literature, we know that the probability of peptides belonging to the same sequence is low when the distance between them is greater than 0.3. Thus, making the points belonging to C_3 too dissimilar to belong to the same cluster. Hence, cluster stability ends up selecting wrong clusters and making the cluster C_3 a heterogeneous cluster. Cluster consensus of a heterogeneous cluster is expected to have more noise than signal, thus decreasing the number of peptide identification.

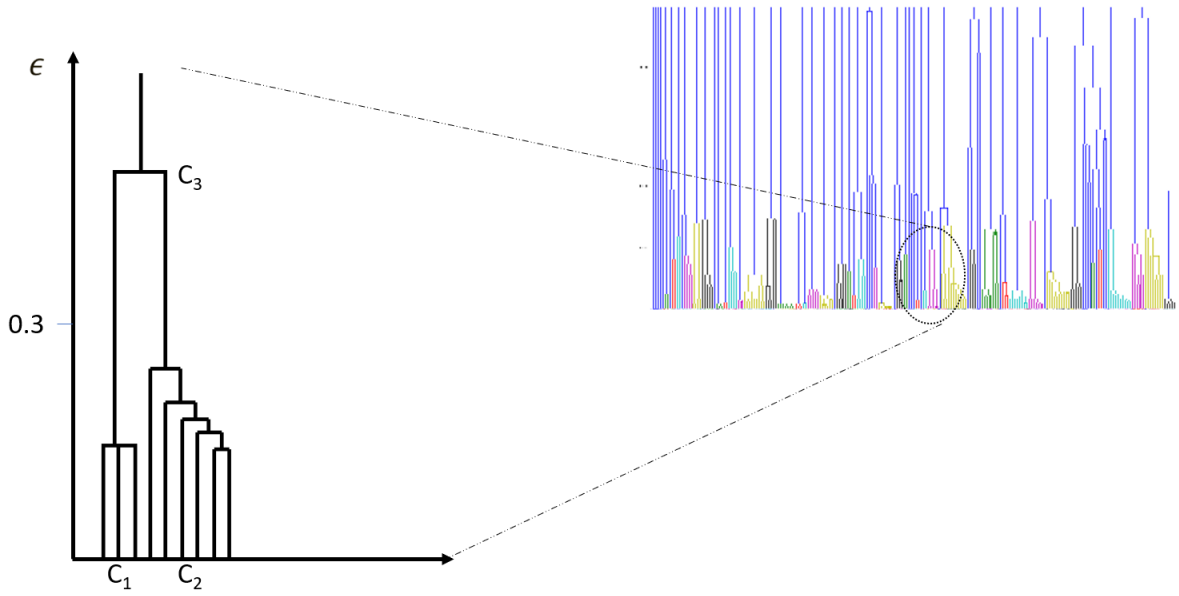


Figure 4.7: Dendrogram showing how two groups of peptides merge in HDBSCAN* hierarchy.

4.2.4 How does cluster consensus boost signal-to-noise ratio (SNR) and improve identification?

The intuition behind generating consensus spectra is the underlying fact that the probability of noise peaks consistently occurring at the same position across spectra of a cluster is low. While generating a consensus spectrum, we sum the intensities over multiple spectra. This summation of intensities leads to an increase in the intensity of correct peaks as compared to noise peaks, which results in boosting of the signal. Thus, consensus spectra retain high-intensity true peaks, which aids better peptide identification.

4.2.5 Why our method yields better results than existing methods?

Our method of extracting clusters from HDBSCAN* hierarchies outperforms the two existing clustering methods in the proteomics literature which are MS-Cluster and N-cluster.

MS-Cluster performs poorly because its approximate hierarchical clustering

(AHC) algorithm is sensitive to the ordering of data points, which leads to mislabeling/ misclassification of data points.

Consider a case where there are 3 points A, B, and C to be clustered by the AHC algorithm in the same order. Assume that first the two points A and B are compared, and merged as the $similarity(A, B) \geq$ similarity threshold. Next, assume that point C is more similar to point A than point B. Point C is however compared with the consensus of A and B. The similarity between point C and the consensus of the point A and the point B may not be high enough to join the point C into the cluster. Thus, assigning points A and B two different cluster label, which is not desirable.

This is not the case with our method since it extracts cluster from HDBSCAN* hierarchies. The points in the HDBSCAN* hierarchy are merged in increasing order of their similarity, and this avoids the above misclassification.

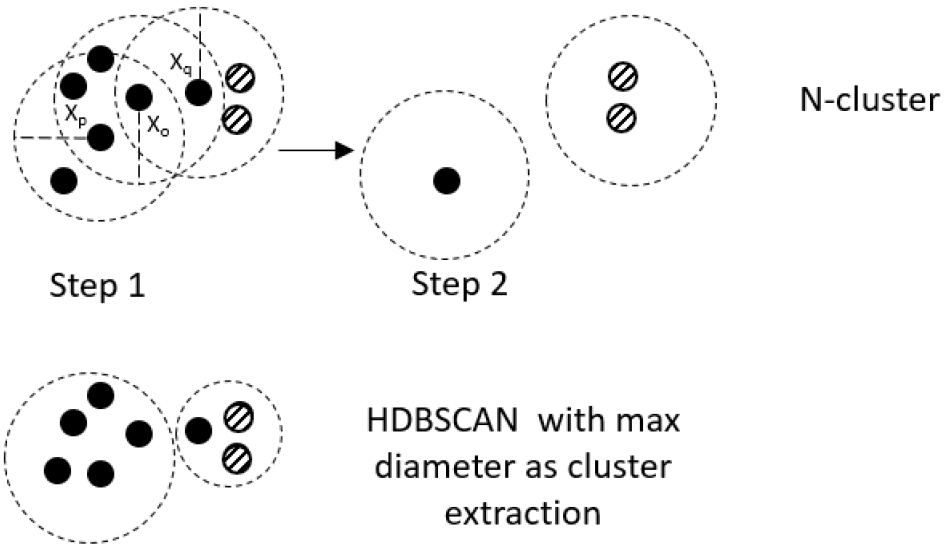


Figure 4.8: Figure showing how N cluster and HDBSCAN-diameter extracts clusters on same set of points.

N–cluster forms clusters sequentially by selecting the point with the highest number of neighbors within a distance threshold 'r' first. Consider the points in the figure 4.8, N-cluster would first calculate the number of points in the 'r' neighborhood of each point. Then it selects the point with the highest number

of neighbors within a distance threshold 'r' which is X_o in this case. It can be seen that x_q is thus assigned to x_o's cluster, although x_q is closer to the set of striped points.

On the other hand, cluster extraction from a hierarchy with $\max_{diameter}$ as a parameter would assign two different labels to X_q and X_o, which is more desirable.

All three algorithms require a thresholding parameter, either as a similarity threshold, distance threshold, or maximum diameter. Besides, our method also requires an additional parameter called $\min_{pts}$ to create the hierarchy. We also observed that $\min_{pts}$ 3 work well in almost all cases. Thus, reducing the parameter only to selecting maximum diameter which makes our approach comparable to the other methods regarding the number of parameters.

Chapter 5

Conclusion

Data obtained from an experiment of tandem mass spectrometry is voluminous and redundant in nature. Clustering takes advantage of these redundancies and replaces duplicate spectra by a single representative. This reduction in the number of spectra accelerates database searches as performed by various search engines like InsPect [34], Mascot [12], Sequest [36]. This single representative spectrum has a boosted signal, which aids the confidence with which a peptide is identified. It thus, increases the number of acceptable peptide identifications.

In this dissertation, we study the effects of applying various clustering techniques on the number of acceptable peptide identifications from data obtained from Tandem Mass Spectrometry. We experiment with a wide variety of clustering methods to cluster tandem mass spectra. This includes the clustering algorithms from proteomics research namely, MS-Cluster and N-cluster. Along with the algorithms from data mining research namely, DBSCAN, HDBSCAN and hierarchical clustering (single, average, and complete linkage).

We propose a new method to extract clusters from HDBSCAN hierarchies and compare them with seven clustering algorithms mentioned based on the number of unique peptides identified. We experiment with two similarity measures, four pre-processing methods and two methods to form consensus spectra.

Key points that we observe from experiments are 1) Clustering generally increases the number of unique peptides identified and decreases the number of points (spectra) that undergo a database search. 2) Cosine similarity is

a better measure of similarity as compared to Spearman's' correlation. 3) Constructing consensus spectra by the consensus average method works better than those done by the weighted consensus method. 4) Calculating PRM for spectra almost always aids the identification process.

We also observed that the traditional data mining algorithms were comparable to the MS- cluster and N cluster and often outperform them. Our method of extracting clusters from an HDBSCAN* hierarchy, by limiting the diameter, outperformed the previously existing state of the art clustering algorithms MS-Cluster and N-cluster in terms of the number of unique peptides identified.

We also explored internal cluster validation methods to estimate good parameter values for different clustering techniques. We observed that it did not work well in this application, and the reason was discussed in section 4.2.2. A method to select parameter values for good clustering in the context of peptide identification remains an open problem.

5.0.1 Future Research :

In the future, we plan to run more experiments on other available datasets of single MS/MS runs as well as from multiple MS/MS runs. One area worth exploring in the future is, how to select good parameter values for clustering in the context of this application. One could also aim to develop a clustering algorithm, which is parameter free for tandem mass spectra. Mass spectrometry is a vast field, which is evolving continuously and there is a scope of improvement for every step in the pipeline.

References

[1] URL: http://virtuallaboratory.colorado.edu/CLUE-Chemistry/chapters/chapter9txt-2.html. 11

[2] *A schematic showing the concept of MS/MS database searching*. 1999. URL: http://what-when-how.com/proteomics/tutorial-on-tandem-mass-spectrometry-database-searching-proteomics. 15

[3] Ravali Adusumilli and Parag Mallick. "Data conversion with ProteoWizard msConvert." In: *Proteomics*. Springer, 2017, pp. 339–368. 39

[4] Suruchi Aggarwal and Amit Kumar Yadav. "False discovery rate estimation in proteomics." In: *Statistical Analysis in Proteomics*. Springer, 2016, pp. 119–128. 17

[5] Rolf Apweiler et al. "UniProt: the universal protein knowledgebase." In: *Nucleic acids research* 32.suppl_1 (2004), pp. D115–D119. 50

[6] Nuno Bandeira et al. "Shotgun protein sequencing by tandem mass spectra assembly." In: *Analytical chemistry* 76.24 (2004), pp. 7221–7233. 2, 40

[7] Ilan Beer et al. "Improving large-scale proteomics by clustering of mass spectrometry data." In: *Proteomics* 4.4 (2004), pp. 950–960. 3, 42, 43

[8] Amir Ben-Dor and Zohar Yakhini. "Clustering gene expression patterns." In: *Proceedings of the third annual international conference on Computational molecular biology*. ACM. 1999, pp. 33–42. 3

[9] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. "Density-based clustering based on hierarchical density estimates." In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2013, pp. 160–172. 28, 29

[10] Ricardo JGB Campello et al. "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies." In: *Data Mining and Knowledge Discovery* 27.3 (2013), pp. 344–371. 26, 32

[11] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis." In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002), pp. 603–619. 19

[12] John S Cottrell and U London. "Probability-based protein identification by searching sequence databases using mass spectrometry data." In: *electrophoresis* 20.18 (1999), pp. 3551–3567. 13, 65

[13] Robertson Craig and Ronald C Beavis. "TANDEM: matching proteins with tandem mass spectra." In: *Bioinformatics* 20.9 (2004), pp. 1466–1467. 13

[14] Ingvar Eidhammer et al. *Computational methods for mass spectrometry proteomics*. John Wiley & Sons, 2008. 14

[15] Joshua E Elias and Steven P Gygi. "Target-decoy search strategy for mass spectrometry-based proteomics." In: *Proteome bioinformatics*. Springer, 2010, pp. 55–71. 17

[16] *erg JM, Tymoczko JL, Stryer L. Biochemistry. 5th edition. New York: W H Freeman; 2002. Section 3.2, Primary Structure: Amino Acids Are Linked by Peptide Bonds to Form Polypeptide Chains.* URL: `https://www.ncbi.nlm.nih.gov/books/NBK22364/`. 11

[17] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: 96.34 (1996), pp. 226–231. 19, 20, 26

[18] Ari M Frank et al. "Clustering millions of tandem mass spectra." In: *Journal Of Proteome Research* 7.1 (2008), pp. 113–122. ISSN: 1535-3893. 3, 33, 39, 42, 43

[19] Johannes Griss et al. "PRIDE Cluster: building a consensus of proteomics data." In: *Nature methods* 10.2 (2013), p. 95. 4

[20] Adrian Guthals et al. "Sequencing-grade de novo analysis of MS/MS triplets (CID/HCD/ETD) from overlapping peptides." In: *Journal of proteome research* 12.6 (2013), pp. 2846–2857. 4

[21] John A Hartigan. "Clustering algorithms." In: (1975). 29

[22] CR Jiménez et al. "Searching sequence databases over the internet: protein identification using MS-Fit." In: *Current protocols in protein science* 14.1 (1998), pp. 16–5. 13

[23] Stephen C Johnson. "Hierarchical clustering schemes." In: *Psychometrika* 32.3 (1967), pp. 241–254. 23

[24] Lukas Käll et al. "Assigning Significance to Peptides Identified by Tandem Mass Spectrometry Using Decoy Databases." In: *Journal of Proteome Research* 7.1 (2008). PMID: 18067246, pp. 29–34. DOI: `10.1021/pr700600n`. eprint: `https://doi.org/10.1021/pr700600n`. URL: `https://doi.org/10.1021/pr700600n`. 17

[25] James MacQueen et al. "Some methods for classification and analysis of multivariate observations." In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297. 18

[26] Davoud Moulavi et al. "Density-based clustering validation." In: *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM. 2014, pp. 839–847. 35–37

[27] Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." In: *Advances in neural information processing systems*. 2002, pp. 849–856. 19

[28] Smriti R Ramakrishnan et al. "A fast coarse filtering method for peptide identification by mass spectrometry." In: *Bioinformatics* 22.12 (2006), pp. 1524–1531. 42

[29] Vera Rieder et al. "Comparison and evaluation of clustering algorithms for tandem mass spectra." In: *Journal of proteome research* 16.11 (2017), pp. 4035–4044. 4, 34

[30] Vera Rieder et al. "DISMS2: A flexible algorithm for direct proteome-wide distance calculation of LC-MS/MS runs." In: *BMC bioinformatics* 18.1 (2017), p. 148. 49

[31] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65. 35

[32] Stephen E Stein and Donald R Scott. "Optimization and testing of mass spectral library search algorithms for compound identification." In: *Journal of the American Society for Mass Spectrometry* 5.9 (1994), pp. 859–866. 42

[33] Talat Iqbal Syed. "Parallelization of Hierarchial Density-Based Clustering Using Mapreduce." Msc dissertation. University of Alberta, 2015. 23, 28

[34] Stephen Tanner et al. "InsPecT: identification of posttranslationally modified peptides from tandem mass spectra." In: *Analytical chemistry* 77.14 (2005), pp. 4626–4639. 13, 15, 65

[35] Juan Antonio Vizcaíno et al. "The proteomics identifications database: 2010 update." In: *Nucleic acids research* 38.suppl_1 (2009), pp. D736–D742. 4

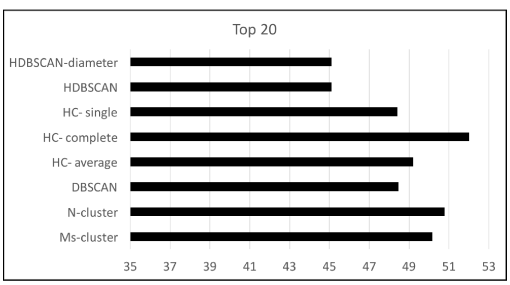
[36] John R Yates et al. "Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database." In: *Analytical chemistry* 67.8 (1995), pp. 1426–1436. 13, 65

[37] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: an efficient data clustering method for very large databases." In: *ACM Sigmod Record*. Vol. 25. 2. ACM. 1996, pp. 103–114. 19

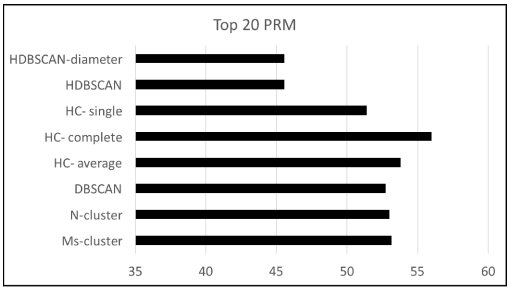
Appendix A

Other experimental results

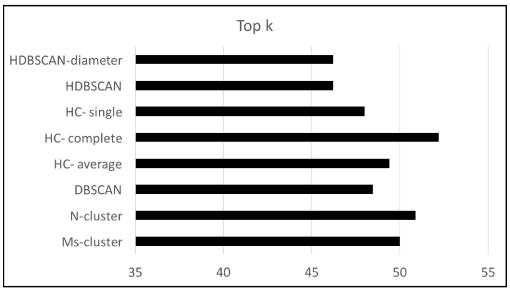
In this section we presents results of other experiments carried out on the smaller dataset. along with it, there are graphs showing the average number of unique peptides identified across all parameter by various clustering methods on each dataset.



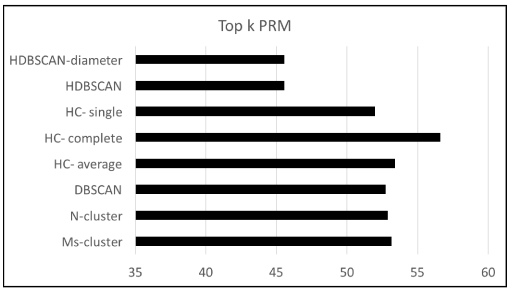
(a) Top 20.



(b) Top 20 PRM.

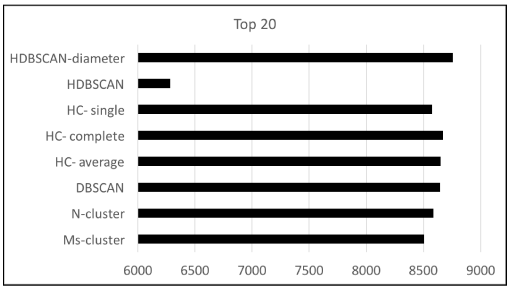


(c) Top k.

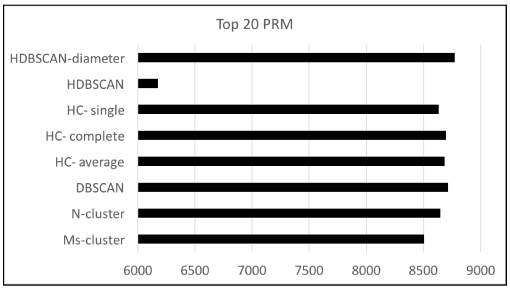


(d) Top k PRM.

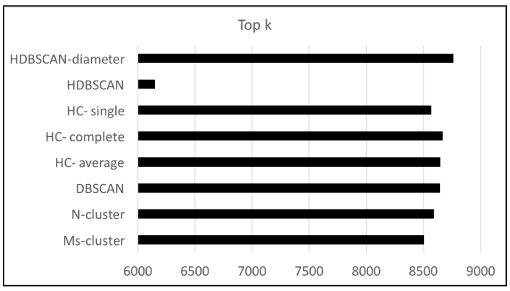
Figure A.1: Average number of unique peptides identified by different clustering algorithms over all the parameters on the smaller dataset. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



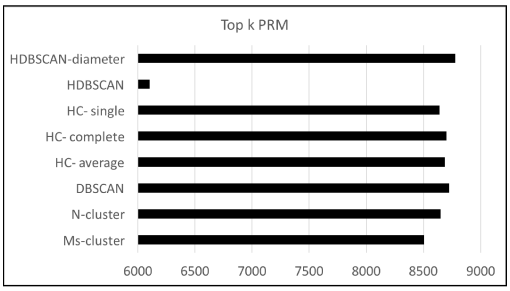
(a) Top 20.



(b) Top 20 PRM.

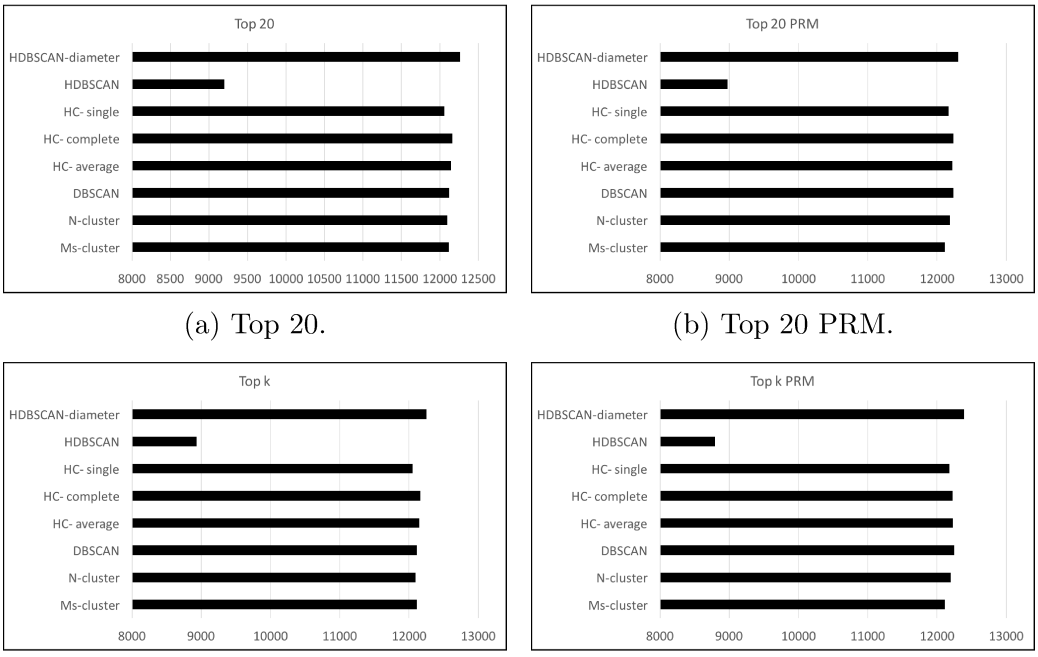


(c) Top k.



(d) Top k PRM.

Figure A.2: Average number of unique peptides identified by different clustering algorithms over all the parameters on the human dataset. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



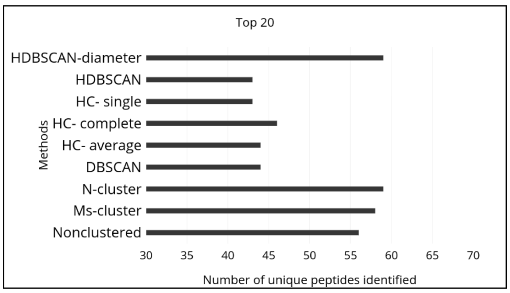
(a) Top 20.

(b) Top 20 PRM.

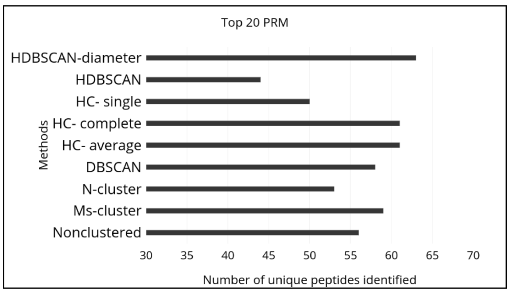
(c) Top k.

(d) Top k PRM.

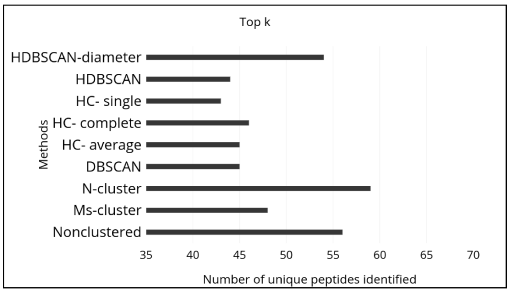
Figure A.3: Average number of unique peptides identified by different clustering algorithms over all the parameters on the roundworm dataset. (Distance: Cosine distance; Cluster representative: Average Consensus Method.)



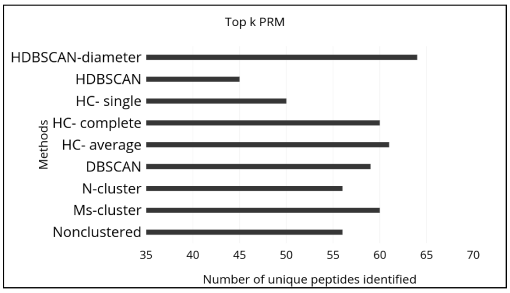
(a) Top 20 peaks.



(b) Top 20 PRM.



(c) Top k.

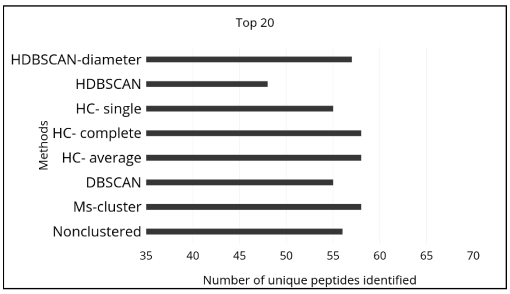


(d) Top k PRM.

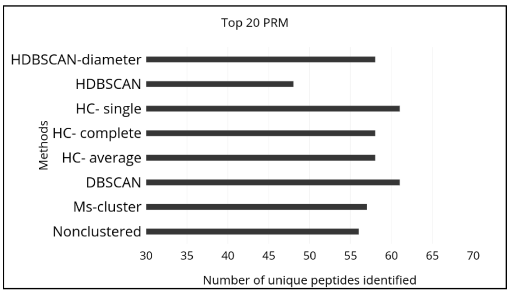
Preprocessing	Ms-cluster	N-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	radius	(min_{pts}, ϵ)	Threshold	Threshold	Threshold	(min_{pts}, min_{clsize})	(min_{pts}, diameter)
Top 20	0.9	0.5	(4, 0.1)	0.1	0.1	0.1	(2, 2)	(2, 0.075)
Top 20 PRM	0.9	0.05	(3, 0.1)	0.1	0.1	0.1	(2, 3)	(3, 0.075)
Top k	0.9	0.5	(4, 0.1)	0.1	0.1	0.1	(2, 3)	(4, 0.075)
Top k PRM	0.9	0.05	(3, 0.1)	0.1	0.1	0.1	(2, 2)	(3, 0.095)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

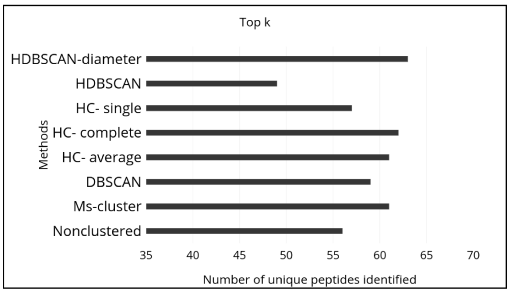
Figure A.4: Results for smaller dataset of 500 spectra: Maximum number of peptides identified by 8 different clustering methods. (Distance: Cosine distance; Cluster representative: Weighted Consensus Method.)



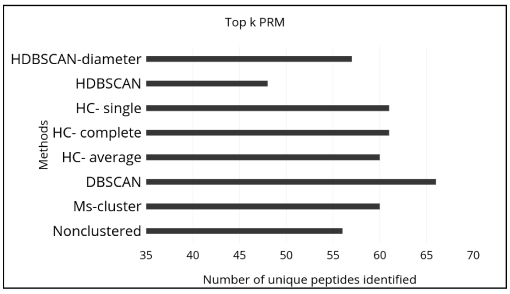
(a) Top 20 peaks.



(b) Top 20 PRM.



(c) Top k.

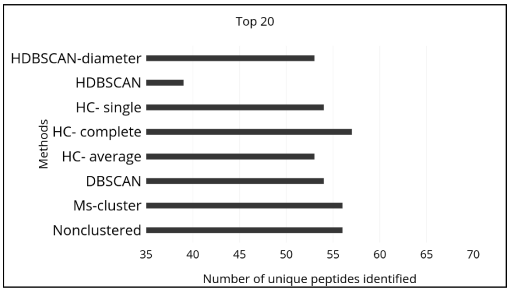


(d) Top k PRM.

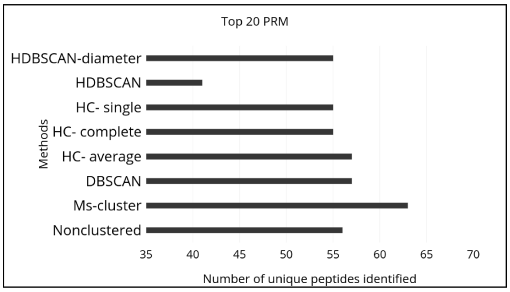
Preprocessing	Ms-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	(min_{pts},ϵ)	Threshold	Threshold	Threshold	(min_{pts},min_{clsize})	(min_{pts},diameter)
Top 20	0.4	(3, 0.1)	0.4	0.4	0.3	(2, 2)	(3, 0.25)
Top 20 PRM	0.5	(2, 0.5)	0.2	0.2	0.5	(2, 3)	(2, 0.2)
Top k	0.4	(3, 0.2)	0.4	0.4	0.2	(2, 2)	(2, 0.15)
Top k PRM	0.6	(3, 0.3)	0.4	0.4	0.5	(2, 3)	(2, 0.25)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

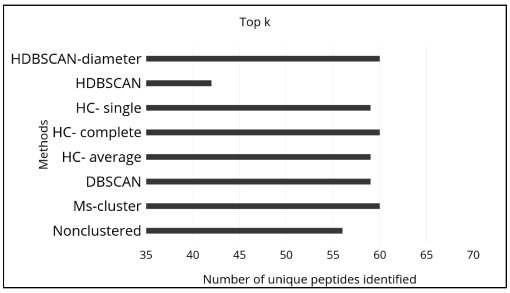
Figure A.5: Results for smaller dataset of 500 spectra: Maximum number of peptides identified by 7 different clustering methods. (Distance: Spearman correlation; Cluster representative: Average Consensus Method.)



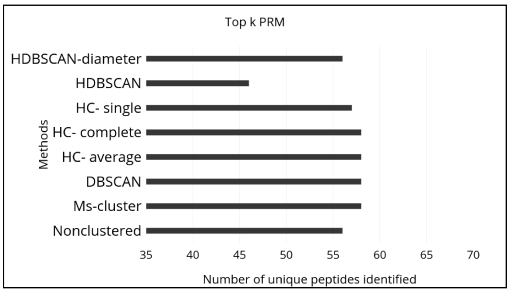
(a) Top 20 peaks.



(b) Top 20 PRM.



(c) Top k.



(d) Top k PRM.

Preprocessing	Ms-cluster	DBSCAN	HC- average	HC- complete	HC- single	HDBSCAN	HDBSCAN-diameter
Parameters	Similarity Threshold	(min_{pts}, ϵ)	Threshold	Threshold	Threshold	(min_{pts}, min_{clsize})	(min_{pts}, diameter)
Top 20	0.8	(2, 0.1)	0.1	0.4	0.1	(2, 2)	(3, 0.25)
Top 20 PRM	0.4	(3, 0.4)	0.5	0.2	0.2	(2, 2)	(2, 0.2)
Top k	0.7	(3, 0.2)	0.3	0.4	0.2	(2, 2)	(3, 0.2)
Top k PRM	0.4	(3, 0.3)	0.4	0.5	0.3	(2, 2)	(2, 0.15)

(e) Parameters for different clustering method which yields the highest number of peptide identification, shown in the graph above.

Figure A.6: Results for smaller dataset of 500 spectra: Maximum number of peptides identified by 7 different clustering methods. (Distance: Spearman correlation; Cluster representative: Weighted Consensus Method.)