seit 1558

Identification of Small Molecules using Mass Spectrometry

A fully automated pipeline proposing similar molecules and compound classes

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik

der Friedrich-Schiller-Universität Jena

von Dipl.-Bioinf. Florian Rasche

geboren am 04. Mai 1983 in Bielefeld

Gutachter:

1. Prof. Dr. Sebastian Böcker, Friedrich-Schiller-Universität Jena
2. Dr. Aleš Svatoš, Max-Planck-Institut für chemische Ökologie, Jena
3. Prof. Dr. Oliver Kohlbacher, Eberhard-Karls-Universität Tübingen

Abstract

Metabolites, small molecules that are produced by an organism, possess a broad range of functions from energy provision to the transfer of complex messages. A large number of metabolites still remain unknown. As metabolites often directly influence the phenotype, the biology of an organism can not be fully understood without uncovering most of its metabolites. Additionally, a newly found metabolite may serve as lead for the discovery of new drugs, especially antibiotics.

Two major techniques exist for the discovery of metabolites: nuclear magnetic resonance provides full structural information, but lacks sensitivity. In contrast, mass spectrometry (MS) provides much less information, but requires little amount of sample and allows for high-throughput analysis. Here, we present algorithms and workflows for the fully automated analysis of tandem MS spectra from small molecules.

In the first step, we annotate spectra with a fragmentation tree. Such a tree assigns molecular formulas to the peaks, and proposes fragmentation reactions between them. Graph theoretical formulation of this task leads to the NP-hard Maximum Colorful Subtree problem. We present algorithms for the *de-novo* calculation of fragmentation trees, based on the spectra alone. Using an ILP formulation the tree calculation is usually faster than the spectra can be acquired. Mass spectrometry experts confirm that the trees agree well with their knowledge of fragmentation chemistry. Additionally, we can use fragmentation trees to improve molecular formula determination by isotopic pattern.

The next step in the pipeline compares the fragmentation tree of an unknown with reference fragmentation trees. We propose to use tree alignment for this, since alignments define similarity in a biologically meaningful way. To perform tree alignments, we adapt a dynamic programming algorithm by Jiang *et al.* (1995). Unfortunately, tree alignment is again NP-hard on unordered trees such as fragmentation trees. But the runtime is only exponential in the maximum out-degree of the tree. Fragmentation trees usually have small out-degrees, rendering the approach feasible.

There are several possibilities how to process the fragmentation tree alignment similarities: For evaluation, we compare these scores with structural similarities of the corresponding compounds, and find these two measures highly correlated, even if the spectra have been measured on different instrument types. Using tree similarities as input for hierarchical clustering results in groups that agree well with chemical compound classes. We also developed FT-BLAST, a tool to search a database of reference trees for an unknown tree. In addition to finding highly similar compounds from the database, it can filter out spurious hits by applying a decoy database strategy. Evaluations show that most of the remaining hits are meaningful.

We apply the full workflow starting with molecular formula determination to a biological sample from Icelandic poppy (*P. nudicaule*). Clustering the unknowns together with reference compounds, enabled the prediction of compound classes for some unknowns. The FT-BLAST analysis gave hints to structural features of the unknowns. An independent manual analysis of the unknowns confirmed our findings.

In addition, reference fragmentation trees can be annotated with structural features using an *in-silico* fragmentation approach. Although theoretical formulation of this problem turns out to be one NP-hard problem nested in another, we managed to develop a branch-and-bound heuristic for this task. In future, it may help to further interpret tree alignments.

This workflow may help researchers in the dereplication of drug leads by telling them if a promising compound is similar to an already tested lead early in the process. Another application may be the reliable reconstruction of metabolic networks from mass spectrometric data, similar to Watrous *et al.* (2012).

Zusammenfassung

Metaboliten, kleine Moleküle die von einem Lebewesen produziert werden, besitzen vielfältige Funktionen zum Beispiel als Energiespeicher oder als komplexer Botenstoff. Eine große Zahl von Metaboliten ist noch unbekannt. Weil Metaboliten oft direkt den Phänotyp beeinflussen, können die Prozesse eines Organismus nicht vollständig verstanden werden, ohne den Großteil seiner Metaboliten zu kennen. Außerdem können neue Metaboliten als Leitstruktur für die Medikamentenentwicklung dienen.

Zur Entdeckung von Metaboliten werden hauptsächlich die folgenden zwei Techniken eingesetzt: Kernspinresonanzspektroskopie kann die vollständige Struktur der Substanz aufdecken, ist aber nicht sehr sensitiv. Im Gegensatz dazu liefert die Massenspektrometrie (MS) weniger Informationen, kommt aber auch mit einer viel kleineren Probenmenge aus und ermöglicht daher eine Hochdurchsatzanalyse. In dieser Arbeit stellen wir Algorithmen für die vollautomatische Analyse von Tandemmassenspektren kleiner Moleküle vor.

Zunächst werden die Spektren mit einem Fragmentierungsbaum annotiert. Ein Fragmentierungsbaum ordnet den Peaks Summenformeln zu und postuliert Fragmentierungsreaktionen zwischen diesen. Die Formulierung dieser Aufgabe mithilfe der Graphentheorie führt zum NP-harten Maximum Colorful Subtree Problem. Wir stellen Algorithmen für die Berechnung von Fragmentierungsbäumen ohne Zuhilfenahme einer Datenbank vor. Bei Verwendung eines ganzzahligen linearen Programms können die Bäume meist schneller berechnet, als die Spektren gemessen werden. Massenspektrometrie-Experten bestätigen, dass die Bäume sich gut mit ihrem Wissen über Fragmentierungsreaktionen erklären lassen. Fragmentierungsbäume können auch verwendet werden, um die Identifizierung der Summenformel durch Isotopenmuster zu verbessern.

Im nächsten Schritt wird der Baum eines unbekannten Stoffes dann mit Referenzbäumen verglichen. Wir verwenden Baumalignments zu diesem Zweck, da Alignments Ähnlichkeit biologisch sinnvoll definieren. Um Baumalignments zu berechnen, haben wir einen Algorithmus von Jiang *et al.* angepasst. Dieser basiert auf dynamischer Programmierung. Leider ist das Alignieren von ungeordneten Bäumen, wie Fragmentierungsbäumen wieder NP-schwer. Allerdings wächst die Laufzeit nur exponentiell mit dem höchsten Ausgangsgrad. Fragmentierungsbäume haben meist kleine Ausgangsgrade, was den Ansatz praktikabel macht.

Es gibt verschiedene Möglichkeiten mit den Ähnlichkeiten der Fragmentierungsbäume weiterzuarbeiten: Zu Testzwecken vergleichen wir diese Ähnlichkeit mit der strukturellen Ähnlichkeit der entsprechenden Substanzen. Diese Ähnlichkeiten sind stark korreliert, sogar wenn die Spektren mit verschiedenen Spektrometer-Typen gemessen wurden. Wenn wir die Stoffe basierend auf ihren Fragmentierungs-

baumähnlichkeiten clustern, bilden sich Gruppen, die in etwa chemischen Stoffklassen entsprechen. Zusätzlich haben wir FT-BLAST entwickelt, ein Programm zur Suche von Fragmentierungsbäumen in einer Referenzdatenbank. Neben der Tatsache dass es sehr ähnliche Substanzen in der Datenbank finden kann, kann es die Signifikanz der Treffer mithilfe einer Köderdatenbank berechnen. Tests zeigen, dass der Großteil der Treffer über einer Signifikanzschwelle sinnvoll sind.

Wir verwenden den gesamten Ablauf zur Identifizierung einer biologischen Probe des Islandmohns (*P. nudicaule*). Durch das Clustern der unbekannten Substanzen zusammen mit Referenzmessungen konnten wir die Stoffklassen einiger Unbekannter vorhersagen. Außerdem gab die FT-BLAST-Analyse Aufschluss über charakteristische Teilstrukturen. Eine unabhängige manuelle Analyse bestätigte unsere Ergebnisse.

Referenzfragmentierungsbäume können außerdem mit Strukturformeln annotiert werden, indem man einen *in-silico* Fragmentierungsansatz benutzt. Obwohl es sich bei der theoretischen Formulierung dieses Problems um zwei ineinander geschachtelte NP-schwere Probleme handelt, konnten wir eine Branch-and-bound-Heuristik für das Problem entwickeln. Diese kann in Zukunft helfen, Baumalignments noch besser zu interpretieren.

Die hier vorgestellt Interpretation von Spektren könnte bei der Dereplizierung von Leitstrukturen für Medikamente helfen, da in einer frühen Phase bereits festgestellt werden kann, ob die neue Leitstruktur einer schon getesteten Struktur ähnelt. Eine weitere Anwendung könnte die verlässliche Rekonstruktion von metabolischen Netzen aus Massenspektrometrie-Daten sein, ähnlich dem Ansatz von Watrous *et al.* (2012).

Acknowledgements

As good science is always teamwork, a lot of people supported me on this project. Here I would like to express my gratitude to them.

I thank my supervisor, Sebastian Böcker, for numerous fruitful discussions, great ideas, and interesting insights into many areas of bioinformatics. His door was always open for questions of any kind and I am grateful for his support and understanding concerning personal problems that arouse during the time of this work.

The same is true for my co-workers, especially Franziska Hufsky and Kerstin Scheubert, who continue the fragmentation tree project. Its been a pleasure working with you! I also thank the rest of the mass spec team, Martin Engler, Marvin Meusel and Imran Rauf, for the great work together. Malte Brinkmeyer, Bao Bui, Thasso Griebel, Frank Mäurer, Anton Pervukhin and Anke Truss warmly welcomed me in the group, and I hope I could extent this welcome to Raffael Fassler, François Nicolas, Florian Sikora, and Sascha Winter. We had a great time together. Kathrin Schowtka always helped with bureaucracy and various other problems. Thanks a lot!

I would also like to thank our collaborators in the tandem mass spectrometry project, Aleš Svatoš, Marco Kai and Ravi Maddula of the Max-Planck-Institute for Chemical Ecology, Jena, Christoph Böttcher and Steffen Neumann of the Leibniz-Institute for Plant Biochemistry, Halle, and Rolf Müller and Daniel Krug from Saarland University, Saarbrücken. Thanks for your explanations, your patience and the trust in our work. I thank Masanori Arita (University of Tokyo, Japan) and Takaaki Nishioka (Keio University, Japan) for making the MassBank data available and Miroslav Strnad (Palacký University, Olomouc, Czech Republic) and Evangelos Tatsis (MPI-CE Jena) for providing samples.

Several students worked on their diploma theses in the field of tandem MS analysis and their ideas and results helped a great deal in the development of this work. Thus, I'm grateful to Kai Dührkop, Birte Kehr, Tamara Steijger and Thomas Zichner for their excellent work. The student assistants Martin Bens, Antje Biering, Claudia Dahl, Markus Fleischauer, Johannes Helmuth and Michael Probst implemented the methods described here into a graphical user interface software. Thank you for digging deep into the realms of software development.

I thank Sebastian Wernicke, my supervisor while I was a student research assistant, for introducing me to scientific work. My projects would not have been as successful without his foundations.

Last but not least, I would like to thank my wife Sarah for her support, for catalyzing ideas and bearing my moods and, of course, my daughter Maya for many hours of joy.

Preface

This thesis presents large parts of my research in the automated analysis of tandem mass spectra from small molecules. During this work, I was member of the Bioinformatics Group led by Professor Sebastian Böcker at the Friedrich-Schiller-Universität Jena. My studies were supported by the university's basic funding.

Most of the results presented here have been published in [12, 76–78] and were achieved in cooperation with my supervisor Sebastian Böcker, our collaborators Aleš Svatoš, Macro Kai, Ravi Kumar Maddula and Christoph Böttcher, my colleagues Franziska Hufsky, François Nicolas, Kerstin Scheubert and Imran Rauf and, last but not least, my diploma students Tamara Steijger and Thomas Zichner.

Together with other co-workers I also participated in the analysis of glycan mass spectra [7, 8], the calculation of fragmentation trees from MS^n data [83, 84] and GC-MS spectra [45] as well as a faster method to calculate fragmentation tree alignments [44]. Before working with the Bioinformatics group, I implemented two methods for biological network analysis under supervision of Sebastian Wernicke and Prof. Rolf Niedermeier [103–105].

The main results of this thesis are presented in Chapters 3–6, whose results are also presented in the following publications:

Chapter 3 describes the calculation of fragmentation trees using results from both [77] and [78]. Sebastian Böcker and I developed the concept and the dynamic programming approach, which I also implemented. I designed the scoring function and developed and implemented the insertion heuristic. I was also involved in the development of the other heuristics and the ILP.

Chapter 4 presents evaluation of the trees against expert knowledge. Here, I only performed the comparison with the Mass Frontier results, since the manual evaluation had to be performed by skilled biochemists.

Chapter 5 deals with the alignment concept to compare fragmentation trees. It is published in [76]. In this project, I participated in the development of the method, the scoring and the significance estimation. I implemented the scoring, decoy database generation and q-value calculation. Furthermore, I performed large parts of the analyses.

In Chapter 6, an approach for the annotation of fragmentation trees with molecular structures is described. These results have been published at the *9th Workshop on Algorithms in Bioinformatics* (WABI 2009) [12]. Here, I worked on the problem formulation and developed the algorithms together with all co-authors.

For the remainder of this thesis, I will use "we" as the first person pronoun, as it is common in scientific literature. This may be interpreted as "the reader and I" or as "my collaborators and I", whichever suits best in the situation.

Contents

1 Introduction

Analysis of biological sequences has long been the focus of bioinformatics research. The genome as well as proteins can be represented by sequences of nucleotides or amino acids. But as modeling approaches proceed, it becomes evident that neither the study of the genome and transcriptome, nor the proteome is sufficient to fully understand cell processes.

The genome gives a static image of what an organism is able to do, but no hints to what is currently happening in the organism or cell. (For simplicity, we ignore epigenetic effects like methylation here.) The proteome, in contrast, reveals information about the cell's current state. But it is only half of the picture: Some proteins such as actins, tubulins and keratins directly influence the appearance of a cell. But a large class of proteins – enzymes – is used to influence and modify small molecules, which in turn determine the appearance of the cell as cell walls, membranes, energy sources and in several other functions. As a prominent example, Mendel observed the color of pea flowers. Their color originates from an anthocyanin produced by the plant [93] and thus, from a metabolite. Mendel was lucky that the connection between this metabolite and some gene was straightforward, despite several transformations in between. Otherwise his studies might have been inconclusive.

Both metabolites and proteins can also serve as signaling molecules. Small molecules are advantageous to bridge large distances, e.g. between organisms. Metabolites are also often used as attack and defense molecules between species, as they can enter another cell more easily and can be produced in sufficient amounts faster and with less energy investment. Thus antibiotics, toxins and stress signals are often metabolites. These properties render metabolites highly interesting in pharmacy [22]. According to Li and Vederas, by 1990, 80 % of all drugs were derived from or inspired by metabolites [61].

Therefore, the field of metabolomics, the study of all metabolites in a cell or an organism, has emerged over the last years [59]. The results in this area helped to gain a more comprehensive understanding of the cell as a whole [35].

Metabolites are difficult to identify since their structure cannot be deduced from the genomic sequence. Only for molecules produced by polyketide synthases and non-ribosomal peptide synthases a certain predictability has been established [13]. Thus, a vast number of metabolites remains unknown [24]. Particularly, plants, fungi and bacteria synthesize up to 20 000 different metabolites per species [35], presenting a huge pool of potentially bioactive substances.

Typically, nuclear magnetic resonance (NMR) is used to identify the structure of an unknown metabolite [27, 32]. It provides detailed information on the structure, but requires large amounts of purified metabolite, in the range of milligrams. Often the metabolite of interest is only present in very low concentrations and hard to purify, impeding the acquisition of such amounts. If the model organism, a plant, for example, cannot be cultivated easily in high numbers, this further complicates the task. NMR is thus not applicable for an high throughput setup.

In contrast, mass spectrometry (MS) is far more sensitive than NMR, and does not require a purified sample. In contrast, a tissue extract or biofluids can be analyzed with a minimal effort of sample preparation, by using a chromatography method before conducting the spectrometric measurement. According to Patti *et al.* [74] liquid chromatography coupled mass spectrometry (LC-MS) detects more metabolites than any other technique and is thus most suitable for the screening of unknown metabolites.

But LC-MS does not provide as much information about a metabolite as NMR. In the simplest form, only masses and, if the data has high quality, molecular formulas can be determined. In a more complex variant, called tandem mass spectrometry, the compound is fragmented and masses of the fragments are recorded. This reveals some information about the structure of a compound. The standard approach is now to compare a fragmentation spectrum with other such spectra in a database. But this is only successful in 25% of the cases [2]. A manual inspection of the unidentified potential metabolites is time-intensive and, given their large number, impractical [74].

1.1 Contribution of this Work

In this thesis, we will present a workflow to greatly simplify identification of unknown metabolites. This workflow is able to automatically retrieve a list of similar (not necessarily identical) compounds from a database, and assign significance values to the hits. Additionally, it depicts relationships between known and unknown compounds by clustering. We use this clustering to even postulate compound classes for unknown metabolites, where this is unambiguously possible.

In our pipeline, we first annotate the spectra with molecular formulas without the use of any database. Then, we perform a database search using the annotation instead of the spectra directly. Through this concept we were able to overcome several limitations of spectral comparisons for database search: We can identify compounds even if the reference spectra have been measured on a different instrument type, such as QToF versus Orbitrap instruments. Using a decoy database strategy, we can assign significances to database hits. This measures the reliability of a database hit. Lastly, we are able to retrieve hits not only from identical but also from similar substances. This has previously been possible as a side product [25]. In contrast, our comparison model has been deliberately designed to account for spectrum changes caused by similar compounds. The significance estimation allows for the assessment of such similar hits for the first time. Thus, we were able to improve on previous results for the search of similar substances in spectral libraries.

Based on the results of such a search, several applications are possible: During the search for novel antibiotics, time is often wasted in the structural elucidation of a potential drug, which then turns out to be highly similar to a known antibiotic.

Although automated structural elucidation by mass spectrometry will most likely remain impossible, the similarity search could sort out such unpromising candidates early in the pipeline. Additionally, further analysis of promising candidates is simplified by already knowing the compound class and eventually some structural features. Similarities calculated between a large number of analytes could also be used to postulate metabolic networks and biosynthesis pathways, thus improving our knowledge about the less common parts of the metabolism.

To present the analysis pipeline, this thesis is structured as follows: We start with introducing the main concepts of metabolomics and mass spectrometry in Chapter 2. In Chapter 3, the method to automatically annotate the spectra is presented. For the annotation, we use fragmentation trees, which assign molecular formulas to peaks and connect these formulas with fragmentation reactions. We have developed several algorithms for the calculation of fragmentations trees. These are presented in the chapter and their performance is assessed on several sets of spectra. Chapter 4 contains a manual evaluation of the automatically generated annotations. Additionally some fragmentation reactions have been verified by multi-stage mass spectrometry. The results of this validation can also be found in the chapter.

Chapter 5 presents the algorithm used to compare the fragmentation trees, that is, the annotations of the spectra. The alignment concept is used for the comparison, allowing for insertions and deletions as well as mismatches of neutral losses. We evaluate the tree comparison by comparing the chemical similarity of reference compounds with the fragmentation tree similarity of their mass spectra. We also show that a clustering based on fragmentation tree similarity agrees well with known compound classes. Lastly, we demonstrate the capabilities of the database search approach using a *leave-one-out* evaluation. In addition, we use biological measurements from poppy to show that the approach is applicable to high-throughput metabolomics data.

In Chapter 6, we investigate an alternative use for fragmentation trees. Given a structure hypothesis, we try to annnotate the fragmentation tree with structural formulas. This can be useful in two ways: On the one hand, reference fragmentation trees may be checked for agreement with the structure of the reference compound. On the other hand, it can assessed how well a structure hypothesis fits to the given fragmentation tree, and thus, to the spectrum. Structure hypotheses may be taken from compound databases, which are much larger than spectrum databases. The chapter present two algorithms for the problem and evaluates their performance.

Finally, we conclude in Chapter 7 with a summary of the results as well as an outlook to future improvements and possible applications of the approach.

1.2 Graph Theoretical Notation

In this thesis we will use graphs to formalize tasks and problems. Thus, we give a short introduction on graph theory here.

Definition 1.1 (Graph)**.** An *undirected graph* $G = (V, E)$ is a pair of a vertex set V and an edge set $E \subseteq \binom{V}{2}$, where $\binom{V}{2}$ denotes the collection of all two-element subsets of V. An undirected edge $e = \{u.v\}$ connects vertices u and v. In a *directed graph*, edges are ordered pairs $e = (u, v)$ and hence $E \subseteq V \times V$. We often use $e = uv$, instead of $e = (u, v)$ for directed edges to improve readability.

As our graphs represent real-world processes, we link vertices and edges of the graph to arbitrary objects called *labels*. In this work, labels are molecular formulas and thus multi-sets of elements. Note that with this definition, a label need not be unique. This is sometimes defined differently, depending on the application at hand. Real-valued weights can be used to define the importance of an edge or vertex.

Definition 1.2 (Weighted Graph)**.** If there is a function $w : E \mapsto \mathbb{R}$ defined on the edge set, a graph $G = (V, E)$ is *edge-weighted.* We call $w(e)$ the *weight* of the edge e. Analogously, if a function $w_V : V \mapsto \mathbb{R}$ exists, we call G *vertex-weighted.*

A weighted graph is usually assumed to be edge-weighted. Another property vertices may have is a color.

Definition 1.3 (Colored Graph)**.** Given a set of colors C, if there is a function $c : V \mapsto C$ a graph $G = (V, E)$ is *vertex-colored.* We call $c(v)$ the *color* of the vertex v.

Often, color is used to denote the type of a vertex, or, in our case, vertices that share the same origin. Colors may also be defined on edges, but edge colors are not used in this work.

Definition 1.4 (Colorful)**.** A graph is *colorful* if every color occurs at most once in the graph, that is every vertex possesses a unique color.

Selection of parts of a graph leads to subgraphs. This can be helpful, e.g., to distinguish between meaningful and less important information for the application at hand.

Definition 1.5 (Subgraph)**.** A graph $G' = (V', E')$ is a subgraph of the graph $G = (V, E)$ iff $V' \subseteq V$ and $E' \subseteq E$.

A directed graph which contains no directed cycles is called a *directed acyclic graph* (DAG). It represents a hierarchy of objects. We call a DAG *transitive*, if its edge relation $E \subseteq V \times V$ is transitive.

An *arborescence* is a DAG, that does not contain cycles even if its edges were considered undirected and whose edges all point away from a particular vertex called the *root.* For simplicity, we call arborescences *trees* in the rest this work, although trees are commonly defined on undirected graphs. We will call the vertices of such a tree *nodes*, to easily distinguish them from the vertices of general graphs.

In such trees, each node can have only one incoming edge, coming from a node called its *parent.* Outgoing edges from a node lead to its *children.* All nodes on the path from the root to a node are called this node's *predecessors.* This includes the root node. Lastly, the *out-degree* of a node (and also a vertex) is the number of its outgoing edges.

2 Biochemical Background and Mass Spectrometry Concepts

This chapter introduces the biochemical concepts and mass spectrometric techniques required for the understanding of this work. We start by introducing the chemical objects analyzed throughout this work, namely molecules. Small molecules active in a biological cell are called metabolites, thus a short overview on the study of metabolites follows. Afterwards mass spectrometry is introduced as the analytical technique producing the data this thesis aims to analyze. Finally, we present previous work on the computational analysis of small molecule mass spectra.

This chapter can only present a brief overview on the topics mentioned, the following books give more details: Mortimer on general chemistry [66], Weckwerth on metabolomics [102], Gross on mass spectrometry [38], and Eidhammer *et al.* on computational approaches in mass spectrometry [31].

2.1 Molecules

Living cells contain a multitude of different molecules, fulfilling diverse tasks like energy supply, information storage and compartment separation. Molecules (or in some cases compounds, see below) are the chemical entities detectable using mass spectrometry. Thus, we will introduce their basic concepts here, starting with the atoms.

Atoms Atoms are the building blocks of all matter. Atoms consist of a nucleus surrounded by electrons. The nucleus in turn contains protons and neutrons. The number of protons in the nucleus, called *atomic number*, mainly determines the chemical properties of the atom. All atoms with the same proton number belong to the same chemical element. Carbon, for example, has six protons in the nucleus. As protons are positively charged, neutrons prevent the protons from repelling each other and thus keep the nucleus stable. For this work neutrons are mostly relevant because they change the mass of an atom. For a certain element several numbers of neutrons can occur in the nucleus, resulting in atoms with the same chemical properties, but a different mass. These variants are called the *isotopes* of an element. An electrically neutral atom contains as many negatively charged electrons as there are protons in the nucleus.

Covalent bonds Electrons are organized in shells around the nucleus. The first shell may contain two electrons, the second eight, and further shells even more electrons.

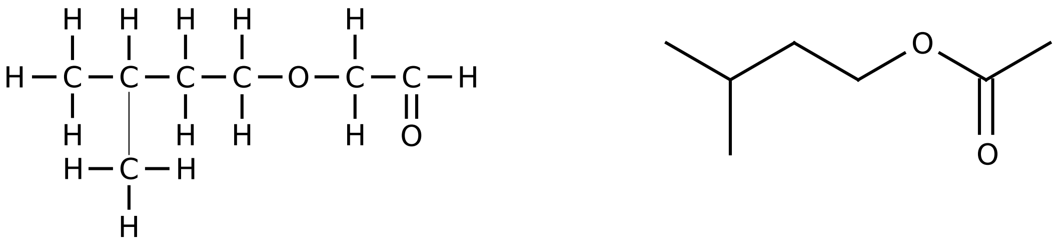


Figure 2.1: Lewis (left) and skeletal formula (right) of isoamyl acetate, a banana flavor.

Possessing a complete shell full of electrons is energetically optimal. Atoms may share electrons to reach this state, e.g., two hydrogen atoms share their two electrons to complete their first shell and form the molecule H_2. The electrons now circle both nuclei, which connects the atoms. Chemists refer to this connection as a covalent bond. The group of atoms connected this way is a *molecule*. In contrast to a molecule, a *compound* can also be connected by other types of chemical bonds, such as *hydrogen bonds* or *van-der-Waals-forces*. If there are enough weak bonds between molecules, they may stay connected during mass spectrometric measurements. Thus, we will use the term compound throughout this thesis, although in most cases molecule would be correct, too.

Molecular formulas By counting the different types of atoms in a molecule or compound, one obtains its *molecular formula*. Ethanol, for example, has the molecular formula C_2H_6O. This elemental composition determines the mass of the compound. Thus, the molecular formula is the only information that can be obtained by mass spectrometry without fragmenting the molecule.

Structural formulas The widespread Lewis structure visualizes the structure of an molecule by displaying the atoms by their elemental symbol and drawing lines between them to represent bonds, see figure 2.1 for an example. A complete Lewis structure also contains lines next to the atoms to represent electron pairs not involved in bonds. These are often omitted for simplicity. Also, the carbon symbol, and hydrogens attached to carbon atoms can be omitted, resulting in skeletal formulas. Such a structure can be represented as a graph in computer science. It must be understood that it contains no information about the three-dimensional layout of the molecule. As it is already hard to determine information about the 2D structure using mass spectrometry, we ignore that molecules are in fact three dimensional in this work.

Compound classes The term *compound class* is not exactly defined. Molecules may fall into the same group, because they share a common reactive group (e.g. amino acids), a substructure (e.g. flavonoids), have certain chemical properties (e.g. alkaloids) or similar biological functions (e.g. nucleotides). Which property is used to define compound classes depends on the application at hand. For the evaluation of our newly developed methods, we will use a mixture of all these class types, as we retained the annotation of the data performed by biochemistry experts. The IUPAC retains an official list of structure-based compound classes [67]. But the most comprehensive database of compound classes are the medical subject headings (MeSH) with nearly 20 000 chemistry related entries [68]. MeSH terms are organized in a tree structure from general to specific terms. A compound may have several MeSH terms.

Molecular mass The weight of atoms, molecules and compounds is measured in *Dalton (Da)*. By definition, an atom of the carbon isotope 12 C in its ground state has a mass of 12 Da. Thus, a proton weighs *approximately* 1 Da. One Dalton equals $1.660538921 \times 10^{-27}$ kg. Another name for the Dalton is *unified atomic mass unit*, abbreviated as u. The mass of a molecule or compound can be calculated by summing the masses of its atoms. Due to bond energies this is not absolutely accurate, but definitively accurate enough for the analyses presented here.

2.2 Metabolites

Metabolites are the substrates and products of chemical reactions taking place in living cells. This would include all compounds in the cell, but the term is usually restricted to small molecules, with the threshold being somewhere around 1000 Da. Usually, the products of polymerisation are no longer considered metabolites. This implies that proteins and DNA are not metabolites, but amino acids and nucleotides are. A notable exception are polyketides, which are metabolites.

Traditionally, metabolites are divided into primary and secondary metabolites. The metabolites necessary for growth, development and reproduction are classified as primary metabolites, all others are called secondary. Signalling, defense against pathogens, protection against abiotic stress, and attraction of mating partners through smell or colour are some of the various functions fulfilled by secondary metabolites.

Most primary metabolites are well investigated, as they occur in large amounts and are usually present in several, if not all, organisms. In contrast, secondary metabolites are only produced in small quantities and are often specific to a species. Their structures are highly diverse, only restricted by the limitations of organic chemistry, see for example the cubic-shaped structure of tetrodotoxin, the famous poison of puffer fish (Figure 2.2).

There exist several metabolite databases, all with a slightly different focus: The Kyoto Encyclopedia of Genes and Genomes (KEGG) [49], the Human Metabolome database (HMDB) [107], the METLIN database [91], and the Madison Metabolomics Consortium Database (MMDB) [23]. The KNApSAck database aggregates species-metabolite relationships [88] and the MetaCyc database focuses on metabolic pathways [20]. The PubChem database is not focused on metabolites, but is the largest freely

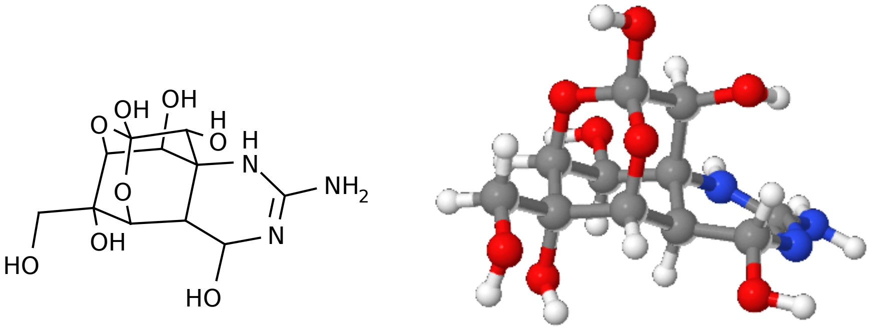


Figure 2.2: Structural formula (left) and ball-and-stick model of tetrodotoxin. It has a roughly cubic three dimensional structure. By convention, carbon atoms are coloured grey, oxygen red, nitrogen blue and hydrogen white.

accessible collection of chemical compounds in general [100]. It currently contains 30 million structures. Additionally, there exists several databases for specific metabolite classes, such as lipids (LMSD [97]) or natural products (DNP [18]).

Despite the huge number of structures stored in such databases, many secondary metabolites can still not be found in any of these databases and thus remain unknown. On the other hand secondary metabolites are of interest in many areas of biotechnology. They often serve as leads in drug development: Li and Vederas [61] estimate that by 1990 80% of drugs where either metabolites or analogs inspired by them. Fields of application include antibiotics, antimalarials, immunosuppressants and anticancer drugs.

Therefore, a sensitive high-throughput method for the identification of new metabolites is highly sought.

2.3 Mass Spectrometry

Mass spectrometry allows for high-throughput analysis of chemical compounds. As the name implies, it is able to record the masses of the analyzed compounds with an accuracy of a few parts-per-million. In its simplest form it can thus be seen as a very exact scale. A typical mass spectrometer consists of three parts: An ion source, where the molecules get charged, a mass analyzer, that separates the molecules by their mass-to-charge ratio, and a detector, which approximately measures the number of incoming ions.

2.3.1 Ion Sources

Most processes in a spectrometer are based on electric fields and currents. Unfortunately, neutral molecules are only little influenced by electricity. Thus, the molecules have to be charged to respond to the measurement. This so called ionization is performed in the ion source. Several methods to ionize a sample exist. They can either keep the analyte molecule intact (soft ionization) or fragment it (hard ionization).

The soft *electrospray ionization* (ESI) is most relevant in this thesis [106]. Here, the liquid sample is pressed through a small metal capillary, whose tip is one pole of an electric field. At this tip, sample and solvent molecules get charged, and through electro magnetic repulsion small drops of about 10 μm diameter emerge. From these drops more and more solvent evaporates. This increases surface charge, which at some point forces the drops to divide. The end result of this process are charged sample ions in the gas phase. There are several theories, how exactly these arise, for details, see [38]. Typically, the resulting ions either carry an additional proton ([M+H]+, M denotes the sample molecule), or have lost one in negative mode ([M-H]-), but sample ions may also form through the addition of other ions, such as sodium ([M+Na]+) or ammonium ([M+NH4]+). Larger sample molecules often receive multiple charges.

Other soft ionization techniques include the *matrix assisted laser desorption/ionization* (MALDI) [50]. Here the sample is embedded into a cristalline matrix. In the ion source, the matrix is then evaporated by laser pulses, releasing charged molecules into the gas phase. This is typically applied for the analysis of proteins. *Atmospheric pressure chemical ionization* (APCI) is similar to ESI [16]. Here sample and solvent are evaporated through heat first, and then the solvent is ionized in the gas phase by applying a high voltage. The solvent then transfers the charge to the sample molecules, thus the name chemical ionization. For some compound classes this works better than ESI, but has the disadvantage that more molecules are fragmented during ionization.

Hard ionization techniques, that fragment the molecule, do not play a role in this thesis, as the adaption of the presented methods to these techniques requires major modifications [45]. Nevertheless, *electron ionization* (EI) will be described here shortly, as it is the most widely used ionization technique in metabolomics [36]. When EI is applied, the sample is usually already in the gas phase, e.g., since a gas chromatography has been performed beforehand. For ionization, a high energy electron beam is shot at the sample. This results in the removal of electrons from the sample molecules and thus the formation of radical ions. Additionally the molecules fragment due to the high energy applied by the beam.

2.3.2 Mass Analyzers

A mass analyzer separates ions by their mass-to-charge ratio m/z. The unit *Thompson* (Th) is sometimes used for this ratio in mass spectrometry, although it is officially dimensionless. Various methods exist to achieve separation by mass-to-charge. For this work, four of them are relevant, namely time-of-flight (ToF), quadrupole, linear ion trap and Orbitrap mass analyzers.

Time-of-flight mass spectrometry is based on the fact that objects of different mass gain different velocities if accelerated by the same force ($\vec{a} = \frac{\vec{F}}{m}$). Acceleration is performed by a uniform electric field. Thus, the force acting on a molecule is

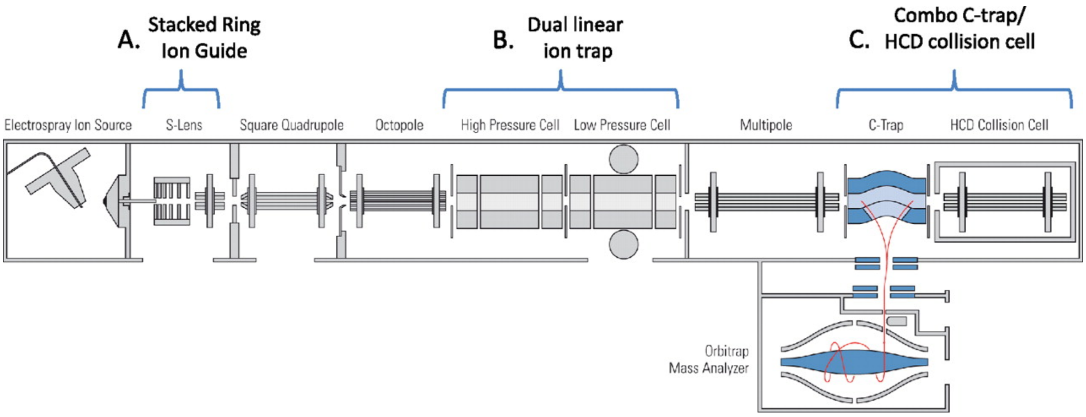


Figure 2.3: Schematic drawing of an LTQ Orbitrap spectrometer. The sample is injected on the left, the Orbitrap analyzer is located bottom right. To the very right the HCD collision cell is shown. Figure taken from [73].

proportional to its charge ($\vec{F} = z \cdot \vec{E}$). Resolving the two equations we get $\frac{m}{z} = \frac{\vec{E}}{\vec{a}}$. Here, both mass and charge are unknown, thus only their ratio can be determined. This holds for all mass analyzers, since electric fields are always used to influence the ions. The velocity of an ion is determined by measuring the time an ion takes to fly through a field free flight tube. The longer the flight tube the more accurate the measurement. Therefore current high performance instruments possess long flight tubes. The tube of the Bruker MaXis, for example, has a length of 2.5 m.

Quadrupole analyzers follow a completely different principle. Here, four parallel rods are connected to an AC power source. The resulting electro magnetic fields cause ions to spiral through the center of the rods. For a fixed AC frequency, only ions with a certain mass to charge ratio can safely fly through the rods, without colliding with them. Thus, the quadrupole allows for filtering of certain ions. Quadrupole analyzers are not very accurate compared to ToF or Orbitrap analyzers, since it is difficult to achieve a narrow isolation window due to the underlying physics.

A *linear ion trap* like a quadrupole consists of four parallel rod-shaped electrodes. But in the linear trap the ends of the rod are insulated from the centers allowing to create electric fields that are different from the center at both ends of the unit. By holding all three fields at the correct AC frequencies, there are different modes of operation: The trap can store all ions, store only ions of a certain mass and let all others pass, or let only ions of a certain mass pass through the trap. As with quadrupole analyzers the disadvantage is the low mass accuracy. State-of-the-art instruments can reach 150 ppm, but 500 ppm are not uncommon. Their huge advantage is that fragmentation spectra can be measured with only one analyzer: By filling the trap with gas, the ions currently stored can be fragmented, allowing for multi-stage fragmentation spectra.

Orbitrap analyzers trap ions on a trajectory around a spindle shaped electrode (Figure 2.3). Ions of the same mass-to-charge ratios have the same trajectory in moving back and forth along the spindle. These oscillations of the ions induce an alternating current in two metal plates located near the electrode. This current can be detected and its frequency allows for a very accurate calculation of the mass-to-charge ratio. No further detector is required. To ensure the ions enter their stable trajectories, they are bundled by a C-shaped ion trap and focused through several electrodes before entering the trap. With their high mass accuracy, their wide dynamic range and low maintenance requirements, Orbitraps have become a common instrument in biological sample analysis.

2.3.3 Detectors

The *Faraday cup* is the most simple type of detector. When ions hit a metal cup, it is slightly charged. In regular intervals, the cup is discharged by grounding it. The current flowing during the grounding is proportional to the number of ions detected during the last interval. But since the current induced by a low number of ions is extremely small, this method is not very sensitive.

Thus, it is necessary to amplify the ion signal before measuring it electronically. This is achieved by *electron multipliers*. Here, the ions hit a special material that emits electrons when hit by charged particles of high energy. Through this process, called secondary emission, the material releases more electrons than charged particles previously hit the plate. A strong amplification is achieved by a cascade of such plates. The charge increase on the last plate is then high enough to be detected even if the original ion input was not.

Photo multipliers are based on the same principles, but amplify a photon beam instead of electrons. The ions to be measured therefore need to be converted to photons by a phosphorescent screen. This additional conversion renders photo multipliers a little less sensitive than electron multipliers. But in a photo multiplier less contaminants assemble, as it can be completely sealed. This results in less maintenance and higher lifetime.

Micro channel plates (MCPs) consist of millions of small multipliers assembled together. Each multiplier is about 10 μm in diameter. Due to their fast response time and their large detection area, they are used in most up-to-date mass spectrometers.

2.4 Tandem Mass Spectrometry

Using the three components above, either intact molecules can be measured (using a soft ionization technique) or fragments are recorded, but the information about the original molecule is lost (using hard ionization). In the latter case, even fragments of several sample molecules may occur in the same spectrum. To gain both, information about the intact molecule and the fragment spectrum, two mass spectrometric measurements are required, one before and one after fragmentation. Two concepts to achieve this exist.

For the first concept coupling of mass analyzers is required. The first analyzer filters for ions of a specific mass. These fly through a collision cell where fragmentation occurs. Afterwards, the fragment ions are separated in the second analyzer and finally detected, resulting in a fragment spectrum. The collision cell is usually a quadrupole, hexapole or octopole, operated in such a way that all ions can pass the cell. Since first and second measurements take place in different analyzers, this is called *tandem-in-space*.

The second concept is *tandem-in-time*. Here, the two measurements take place in the same analyzer, which has to be able to trap ions. A first measurement is performed, then all ions that shall not be fragmented are released. The ions remaining in the trap are then fragmented (usually by releasing a gas into it), and the second measurement is performed. Multiple fragmentation (MS^n) is possible by repeating the procedure. Since a trap is required, this approach only produces low accuracy data not suitable for the methods presented in this thesis.

In both cases, the ion selected for fragmentation is called the *precursor ion* of the resulting *fragmentation spectrum*. For fragmentation between the analysis steps several techniques are available, the most relevant will be described here.

2.4.1 Collision Induced Dissociation

Collision induced dissociation (CID) is the technique most commonly used with small molecules. Most data presented here have been measured using this technique. Here, the collision cell is filled with a neutral gas (hydrogen, nitrogen or argon). The fast ions collide with these molecules resulting in fragmentation determined by complex rules of the gas phase chemistry. In case of singly charged ions, one part, the fragment ion, retains the charge, where as the other part becomes a so called *neutral loss* or *radical loss*, that can no longer be detected in the spectrometer. Through an acceleration field in front of the collision cell, the intensity of the collisions and thus the degree of fragmentation can be adjusted. This acceleration energy is measured in electron volt (eV) and typically ranges from 5 to 100 eV.

2.4.2 Higher-Energy Collisional Dissociation

In an Orbitrap instrument fragmentation using *higher-energy collisional dissociation* (HCD) is possible. Normal CID fragmentation in the Orbitrap happens in the first mass analyzer, a linear ion trap. Even at high energies, this can lead to poor fragmentation [72]. An octopole collision cell can be attached to the C-trap of the instrument and filled with gas to enable collisions. This usually results in a larger diversity of fragment ions. Some spectra analyzed in this work were measured using this technique.

A plethora of other fragmentation methods exist, but most of them are not regularly applied to analyze small biological molecules. Electron capture dissociation (ECD) and its variants electron transfer dissociation (ETD) and electron detachment dissociation (EDD) are becoming widespread in proteomics. They produce fragments that do not from when using CID. Unfortunately, the efficiency of these techniques grows quadratically with the charge of the sample ion. Thus, it is not too useful for the analysis of mostly singly charged small molecules. For a recent overview of all mass-spectrometric techniques applied to analyze small molecules, see [54].

2.4.3 Common Tandem MS Instruments

Certain combinations of mass analyzers have proven useful and are commercially available:

Triple Quadrupole In triple quadrupoles (QqQ) the first quadrupole is used to filter the precursor ion and the second serves as collision cell. The third quadrupole scans through the whole mass range, and thus allows for recording a fragment spectrum. Since quadrupoles only allow for a resolution of 0.1 Th, spectra measured on triple quadrupoles do not possess a high mass accuracy. The advantage of a triple quadrupole is that it can also detect which intact ions produce fragments of a fixed mass, by exchanging the operation modes of quadrupoles one and three.

Quadrupole Time-of-Flight Quadrupole time-of-flight (QToF) instruments work similar to triple quadrupoles. The first analyzer is a quadrupole, followed by a quadrupole or hexapole collision cell. As the name implies, the difference lies in the second analyzer. It is a time-of-flight analyzer which allows for much higher mass accuracy. Current instruments such as the Bruker Maxis can reach a mass accuracy of 1 ppm. It is thus a good choice for the analysis of unknown small molecules and several data sets presented in this thesis have been measured using QToF instruments with accuracies between 20 and 50 ppm.

LTQ Orbitrap Orbitrap analyzers are commonly coupled with a linear trap, resulting in an LTQ Orbitrap mass spectrometer. The linear trap serves as a mass filter by letting ions of a specific mass out of the linear trap into either the collision cell, or the Orbitrap analyzer directly. This results in fragment ion spectra with high mass accuracy. A major advantage of this setup is that the ions can be transferred back and forth between the linear trap and the Orbitrap allowing for multiple fragmentation. This is the second instrument setup suitable for analyzing unknown metabolites, and data from this type of instrument is used in this thesis.

Several other combinations of analyzers are possible, but less relevant in the analysis of small molecules. See [38] for a detailed list.

2.5 Chromatography Methods

For complex microbiological samples, e.g. cell extracts, a separation is necessary to obtain fragmentation spectra of only a single compound. In metabolomics, chromatography is commonly used to achieve this. It separates components of a mixture between a stationary and a mobile phase. It makes use of the fact that some compounds of the mixture will interact stronger with the mobile phase, others stronger with the stationary phase, thus requiring different times to pass along the stationary phase. *Column chromatography*, where the stationary phase is placed in a tube called the *column*, can easily be coupled with a mass spectrometer. Thus, this is the type of chromatography relevant here.

The chromatographic method most widely applied in metabolomics is *gas chromatography* (GC). Here, the mobile phase is an inert gas such as helium or nitrogen. Gas chromatography is cheaper than other chromatography methods and offers a high resolution. Unfortunately, it has two major disadvantages: Firstly, it is usually coupled with electron ionization, leading to a fragmentation of the sample molecules before mass spectrometric measurement. Thus, the mass of the unfragmented molecule remains unknown. Secondly, non-gasous samples, such as cell extracts, have to be evaporated. For this, the analytes must be volatile and temperature stable, which is the case only for a fraction of the metabolites.

Liquid chromatography (LC) avoids these disadvantages. Here the mobile phase is liquid and the stationary phase consists of porous beads packed into a metal tube. See [31] for a short introduction. The measurements presented in this thesis use silica beads with C_{18} hydrocarbon chains attached. These interact strongly with non-polar substances of the mixture through van-der-Waals forces. The mobile phase is gradually changed from water to non-polar acetonitrile, which first washes out the polar analytes and later the unpolar ones. This leads to a separation that is orthogonal to the separation by mass in the spectrometer. In modern chromatography systems pressures up to 100 MPa are used to allow for a dense stationary phase with good separation and still retain short flow times between 30 and 60 minutes.

The time a substance requires to pass through the column is called *retention time*. Due to the complexity of the interactions in the column, it is nearly impossible to relate this time to a measure of polarity, e.g., dielectric constants. Thus, the retention time is used together with its mass to refer to a certain compound, although this is only valid for this specific experiment.

LC requires only small amounts of sample (typically 25 mg of tissue or about 1 million cells) [74]. Since no evaporation is necessary, a much wider range of metabolites can be analyzed. Additionally, a soft non-fragmenting ionization is usually applied after LC, allowing the mass of the unfragmented analyte to be recorded. This is the reason why only data from LC-MS and those where the sample was directly injected into the spectrometer are interpreted in this thesis.

2.6 Signal Processing in Mass Spectrometry

In the following chapters, we will assume that the result of a mass spectrometric measurement is a list of peaks, that is mass-over-charge and intensity pairs. In reality, this list is the result of several signal processing steps which we will shortly describe here. For more details refer to [31].

The instrument measures the ion current as an analog signal, the results are then converted by an analog digital converter, and transmitted to a computer. The plot

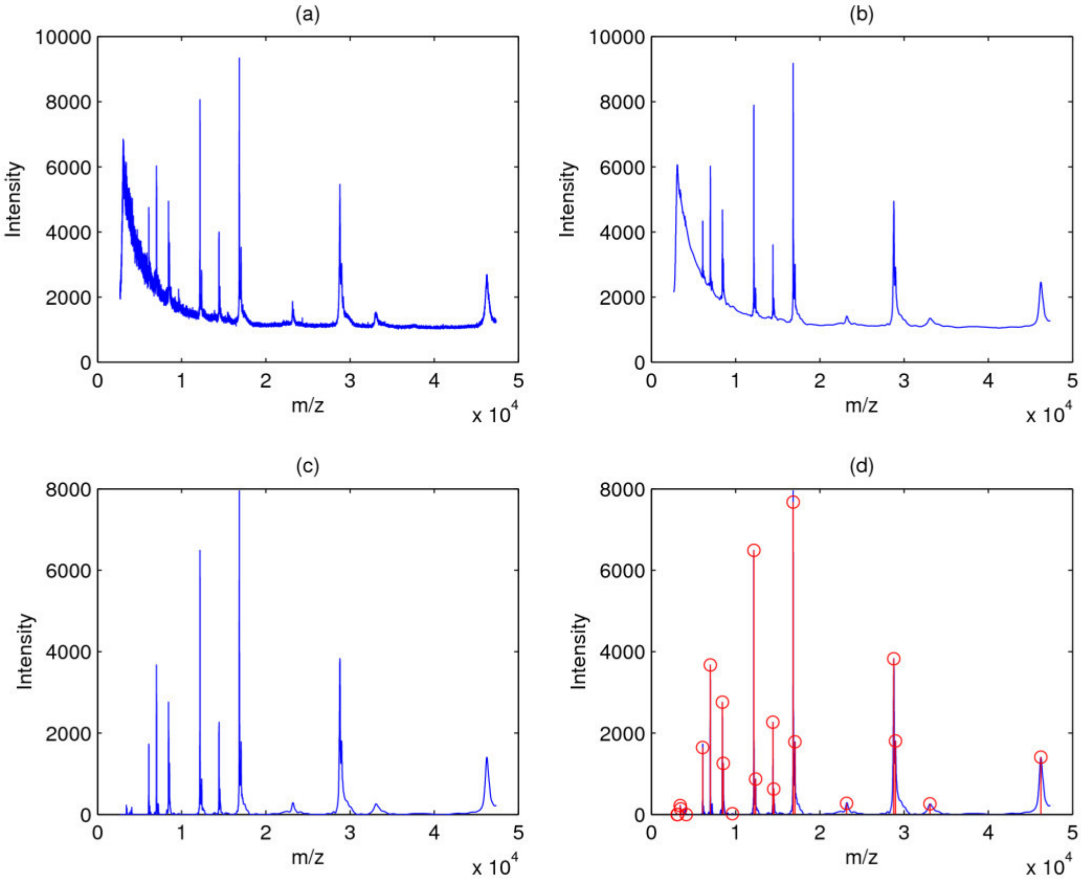


Figure 2.4: Data preprocessing steps: (a) Unprocessed raw spectrum (b) Spectrum after smoothing (c) Smoothed and baseline corrected spectrum (d) Peaks identified in the spectrum. Figure taken from [110].

obtained this way is called the *profile spectrum*. The following steps are required to obtain a high quality peaklist from these raw data:

- *Smoothing* To filter electronic noise, a gaussian filter with a small width below 1 Da is often applied [31].

- *Baseline correction* The baseline stems from ubiquitous ions, see Figure 2.4 (a) for an example. It has to be subtracted to obtain meaningful peak intensities. Unfortunately, the baseline level varies over m/z values. Commonly, a sliding window approach is used to determine the baseline in a certain m/z region, see for example [4].

- *Peak picking* In this step the (pseudo-)continuous profile spectrum is converted into the histogram-like peak list spectrum. The simplest approach would be to calculate the minima and maxima of the profile spectrum, where the maxima denote peak centers and the minima denote transition between peaks. The maximal peak height, peak area or its full-width at half mean can then be used as intensity, whereas the weighted average determines its m/z value. More sophisticated methods use wavelet transformations to detect peaks [58]. This has the additional advantage of rendering baseline correction unnecessary.

- *Calibration* In order to achieve high accuracy a calibration of the measurement is required. For tandem MS experiments, this is performed by measuring a standard solution from time to time. This is called external calibration as opposed to internal calibration, where known standards are added to the sample, resulting in reference peaks within the spectrum. The latter method is more accurate, but not applicable as soon as ion are filtered during the experiments.

Only after these steps have been performed, manual or automated interpretation of the spectra is possible. Methods for the automated analysis of such peaklists are the main part of this thesis. Before we present these in the next chapter, we will give a short overview of other methods for this task.

2.7 Spectral Libraries

The most straightforward approach to automatically analyze a fragmentation mass spectrum is to compare it to reference spectra from a database. If a very similar spectrum is found, the compound generating the sample spectrum has been identified. Several algorithms exist for this task [95]. Unfortunately, such an identification requires that a reference measurement of the compound exists and that the two spectra are sufficiently similar. The latter may not be the case for tandem MS measurements, where many different experimental setups exist. Thus, reference spectra need to be available from the same instrument type [5], ideally even on exactly the same instrument. As this is usually impossible, Oberacher *et al.* developed a more sophisticated comparison, detecting small but significant similarities [70, 71].

The fact that many molecules detected cannot be found in any database or metabolite repository can make library search unsuccessful, even with the best search methods [74]. Demuth *et al.* developed a search algorithm optimized to retrieve similar compounds in case a compound is not in the database [25], but they cannot assess the quality of the hits.

Several databases storing tandem MS spectra exist. The *MassBank* database contains about 30 000 spectra [43]. Its most notable feature is that most of its data is publicly available, which is unusual among spectral libraries. It is developed and maintained by a consortium of mainly Japanese metabolomics research institutes.

The *METLIN* database is provided by the Scripps Center for Metabolomics and Mass Spectrometry [91]. It contains 55 000 spectra of 10 000 metabolites. It can be searched free of charge via a web-interface, but its data is not available for download. The *Human Metabolome Database* (HMDB) stores mostly NMR spectra, but for about 900 compounds, mass spectra are also available [107]. Similar to the METLIN database it can be freely searched, but the spectra cannot be downloaded for automated processing.

With nearly 100 000 tandem MS spectra of 13 000 compounds the *NIST* library is one of the most comprehensive spectral libraries [98]. It is not restricted to metabolites, but covers all sorts of compounds. Unfortunately, access to it is only commercially available, but even then the spectra can only be searched using the software included. Full access to the data for different analyses is prohibited.

Databases of EI spectra The *NIST* library contains an even larger number of EI spectra (nearly 250 000), but the restrictions above apply. Fortunately, smaller, but publicly available databases exist: Probably, the most renowned are the Golm Metabolome Database (GMD) with 1 400 spectra [56] and the *FiehnLib* with 2 200 spectra.

2.8 Related Work

This section describes several methods that can be used if a reference spectrum is not available. They either form the basis for the work of this thesis or complement it.

2.8.1 Molecular Formula Determination Using Isotope Pattern

When analyzing single MS spectra the first step is to find all molecular formulas that have the measured mass. Formally, this leads to the MONEY CHANGING PROBLEM, which asks for all combinations of coins that can be used to pay a certain amount. Böcker *et al.* proposed a fast solution for this problem [10] and adapted it to real valued masses [9]. But even at high mass accuracy, there are usually many molecular formulas for a given mass.

As additional information we can use the fact, that nearly every element has several stable isotopes. These isotopes are integrated in molecules in exactly the proportion they occur in the environment. Thus, a molecular ion creates a cluster of peaks with distances of about one proton mass between neighboring peaks, an *isotopic pattern*. For a given molecular formula, the relative intensities of the cluster peaks as well as their exact masses can be calculated by folding [9, 21].

The candidate formulas are now ranked by the similarity between their theoretical isotopic pattern and the measured one. Böcker *et al.* derived a similarity score using Baysian statistics [9], which performs well on spectra from a ToF instrument. Recently, Pluskal *et al.* showed that a straightforward spectrum comparison performs better on a dataset from an Orbitrap instrument [75].

Such an analysis of the isotopic pattern is the foundation to determine the molecular formula of an unknown metabolite. Its results can be improved using fragmentation information, see Chapter 3. Fragmentation information alone seems to be insufficient to determine the molecular formula.

2.8.2 Substructure Identification Using Spectral Trees

Sheldon *et al.* introduced spectral trees as a means to visualize the relationships between several MS^n spectra of the same compound [87]. In 2011, Rojas-Chertó *et al.* presented a method to identify the molecular formula of an unknown based on spectral trees [81]. As relationships between the spectra are known, and they do not aim to produce a representation of the fragmentation events, they can assume that the spectral tree also represents relationships between the peaks. To identify the formula they use a bottom up approach. Small peaks will likely have few explanations, and these restrict the explanations on the next level. Their approach works well for compounds below 500 Da, if there are no noise peaks in the spectra. They ensure this by repeating the measurements as often as the chromatography allows and using only peaks that occur in at least 40% of these repetitions [82]. This limits its applicability in a high-throughput setup, where the compounds of interest often appear at low concentrations and thus quickly elute from the chromatography column.

This method also results in spectral trees annotated with molecular formulas (from the bottom-up formula calculation). In [82], Rojas-Chertó *et al.* develop a method to compare these trees based on calculating binary fingerprints. This comparison is then used for database searching and common substructures are extracted from the hit lists. These substructures are chemically meaningful and allow for conclusions about the sample compound. The method displays a so-called "neighborhood behavior", meaning that a high spectral tree similarity implies a high chemical similarity. Unfortunately, the reverse is not true, thus the method currently is not very sensitive.

2.8.3 In-Silico Fragmentation

Chemical compound databases contain many more compound structures than reference spectra. The free PubChem database, for example, contains 30 million structures [100]. Thus, being able to calculate a theoretical spectrum from a structure would dramatically increase the number of reference spectra available. The calculation of such a spectrum is called *in-silico fragmentation.* This approach is followed by the commercial software "Mass Frontier"[1]. It uses a set of rules, derived by experts, to predict the spectrum. This rule-based approach makes it less applicable to compound classes whose fragmentation is not fully understood. Additionally, they show low accuracy for compounds above 300 Da, as to many fragmentation pathways can explain any fragment mass. The same drawbacks apply for its competitor, "ACD/MS Fragmenter"[2]. Hill *et al.* demonstrated that spectra predicted by "Mass Frontier" can at least help identify the molecular formula of a compound for which no reference spectrum exists [41].

A different approach of *in-silico fragmentation* tries to explain as many peaks in the spectrum as possible using a given structure. This can then be performed for all candidate structures with the correct weight, or the correct formula. Heinonen *et al.*

[1] Mass Frontier 7.0 Spectral Interpretation Software, Thermo Fisher Scientific, Waltham (MA), USA
[2] ACD/MS Fragmenter, ACD/Labs, Toronto, Canada

presented the first algorithm for this task, a mixed iteger linear program [40]. Unfortunately, even the subproblem of deciding whether a fragment of fixed weight can be cut from the molecule using at most a given amount of energy is NP-hard. Calculations on molecules above 500 Da are thus quite slow. A common query with five to ten candidates often takes days to calculate. A problem similar to this one is presented in Chapter 6.

Wolf *et al.* use a heuristic to solve this problem in their MetFrag software [108]. To my knowledge this is the only usable and free approach for *in-silico fragmentation*. It can account for fragmentation spectra annotated with molecular formulas by passing it the exact masses of the annotations and setting a low mass error. Thus, it can include fragmentation tree information. The major disadvantage of the approach is that it cannot handle structural rearrangements which occur regularly during fragmentation.

2.8.4 Compound classification by machine learning

Varmuza *et al.* present a machine learning approach to classify spectra of unknown compounds [99]. For GC-EI-MS spectra, they define several features based on general knowledge about EI spectra. Then they choose certain substructures and compound classes for which they train classifiers. The classifiers are based on linear discriminant analysis and neural networks using radial basis functions, which were state of the art back in 1996. Unfortunately, to reach a precision of at least 90% their classifiers had to reject 40–70% of the spectra, leading to only a little reliable information about the unknown compound.

A similar method has already been proposed by Buchanan *et al.* as part of the DENDRAL project [17]. Except from a few examples, no evaluation of that tool is available, but given the machine learning methods of the time I speculate that it was not suitable for practical use.

2.8.5 Assessing Statistical Significance of Database Hits

When searching a database it is important to differentiate between true and spurious hits. One of the database entries will always have the highest similarity to the query but that does not make it a meaningful hit. This is where BLAST pioneered in the field of sequence similarity search. For the analysis of protein spectra, methods have been developed to assess the significance of a spectral library hit [69]. They can be subdivided into two types: *Target-decoy searching* [48] and *empirical Bayes approaches* [51].

Here, I will only cover the target-decoy strategy, as it will be applied in Chapter 5. To employ the approach, a decoy database with random entries is generated. It has to fulfill three criteria: It should have no entries in common with the real database, it should not contain real peptides, and a hit in the decoy database should be as likely as a wrong hit in the real database. Exceptions from this criteria are possible, as long as they are unlikely. In proteomics, a decoy database can simply be generated by reversing the peptide sequences of the real database, except for the last letter. (Due to experimental setup, true peptides always end in K or R.) Of course, more involved approaches for this task exist.

The query is then searched in both databases and results are combined. Now, it can be assumed that in a given result list, there are as many spurious hits from the real database as there are hits from the decoy database. Thus a *false discovery rate* (FDR) can be calculated by dividing the number of decoy hits by the number of hits from the real database. Usually, an FDR threshold is defined, and the longest result list with an FDR below the threshold is returned. To assess the quality not only of a result list, but of an individual hit in that list, the *q-value* is used. The q-value of a single hit is the smallest FDR for which this hit still occurs in the output list.

Kim *et al.* developed an approach to really calculate the number of peptide spectra that will receive a score large or equal to the score of a given hit [52]. This enables the calculation of FDRs without the need for a decoy database.

3 Calculation of Fragmentation Trees

When using tandem mass spectrometry, fragments usually result from a subsequent series of fragmentation events. We model these fragmentation cascades using fragmentation trees (Fig. 3.1): The nodes of this tree are labeled with the molecular formulas of the molecule and its fragments, whereas the (directed) edges correspond to fragmentation reactions and, equivalently, neutral or radical losses. The root of the fragmentation tree is labeled with the unfragmented ion. Fragmentation trees

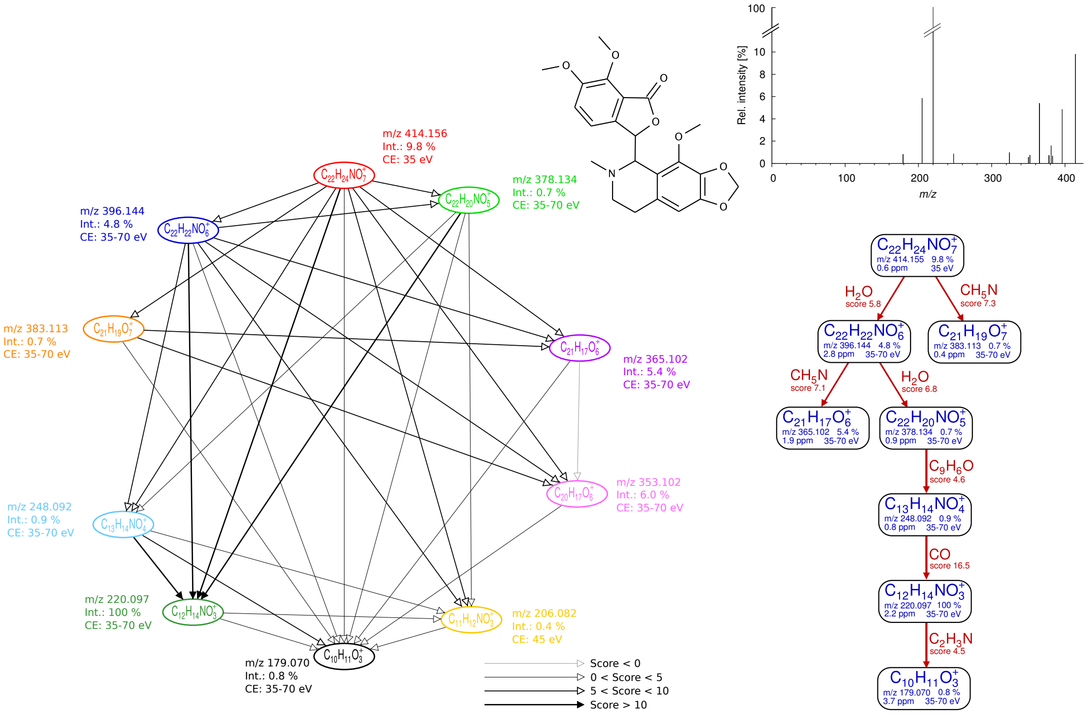


Figure 3.1: Left: Fragmentation graph for (S,R)-noscapine ($C_{22}H_{23}NO_7$) using Orbitrap data. Nodes of the same color correspond to annotations of one measured peak (m/z, intensity, and collision energies). Edges correspond to potential neutral losses. The weight of edgess is encoded by different line types. Right: The corresponding hypothetical fragmentation tree of noscapine computed by our method. Nodes (blue) correspond to peaks in the tandem mass spectra and their annotated molecular formula (CE is range of collision energies), edges (red) correspond to hypothetical neutral losses.

can be easily represented in a computer. Additionally, they render even complicated fragmentation processes easily susceptible, see Figure 4.4 on page 43.

Here, we present a method to calculate fragmentation trees solely from the spectral data, without the use of any database. Thus, this approach can be used to interpret the spectra of unknowns, that have never been characterized before. In Chapter 5, we will demonstrate the application of the method on such unknowns.

In this chapter, we describe how to transform the problem of finding the most likely fragmentation tree into a graph theoretical problem, namely, the Maximum Colorful Subtree problem. For this, we need the calculate meaningful weights for both fragments and fragmnetation reaction annotations, which are also presented here. We will develop exact algorithms as well as heuristics for solving the Maximum Colorful Subtree problem and evaluate the performance of these algorithms on real spectra as well as randomly generated data. We conclude by describing not only how we can use the fragmentation tree calculation for molecular formula prediction of an unknown compound.

3.1 Generation of the Fragmentation Graph

For ease of presentation, we will assume in this chapter, that the molecular formula of the compound is known. Thus, we can assume that there is only one molecular formula for the precursor ion peak. Section 3.8 shows how to overcome this assumption and use fragmentation trees for the prediction of molecular formulas.

To obtain a high number of fragments it is beneficial to measure spectra of the same precursor peak at several collision energies. Before the calculation, we merge these spectra back into a single peaklist. For that, we consider peaks with less than 50 mDa distance to represent the same fragment ion and combined to a single one by calculating their signal intensity weighted mean. This relatively large mass window was found to improve the mass accuracy of the data. We do not scale intensities, since this would compromise comparisons between the peaks between spectra taken at different collision energies.

To calculate a fragmentation tree, we first transform the spectrum into a fragmentation graph. Fragment masses are replaced by a set of molecular formulas within mass accuracy around the fragment mass and every possible reaction between fragments is drawn. This graph contains all possible fragmentation trees as subgraphs. In detail, for every peak of the fragmentation spectrum, we compute all molecular formulas that are within the mass accuracy of the instrument, and that are subformulas of the compound molecular formula. Additionally, we discarded formulas that did not obey Senior's third theorem [85]. It states that the sum of valences has to be greater than or equal to twice the number of atoms minus one. We considered this reasonable, since in the KEGG COMPOUND database, only 0.16% of substances violate this rule [9].

We use these molecular formulas as the vertices of a fragmentation graph, see Figure 3.1 for an example. Vertices are colored so that two molecular formulas corresponding to the same peak, also receive the same color. Two vertices are connected by a directed edge if the second molecular formula is a sub-formula of the first. We assign a weight to the vertices representing the likelihood that the molecular formula of this vertex is the correct fragment formula that resulted in the corresponding peak. Edges receive a weight relative to the likelihood that the corresponding fragmentation reaction is real. The detailed calculation of these weights is described in the next sections. A fragmentation graph is a directed acyclic graph, since fragments can only loose, never gain, weight. More mathematically speaking, since the sub-formula relation defines a partial order on the molecular formulas and thus on the vertices. This graph has only one vertex with no incoming edges (commonly called source), namely the one annotating the molecular ion peak. This will become the root of the fragmentation tree.

3.2 Scoring Fragments – Weighting Vertices

The scoring or weighting of the fragmentation graph is based on the probability that a certain vertex or edge is "true": Trees will be assessed by our algorithm based on the sum of these scores. To this end, it is reasonable to assign scores based on log likelihoods or log odds, which enable a statistical interpretation of the outcome (i.e., maximum likelihood): Summing log likelihood equals the log product of these likelihoods, and the location of the maximum is identical with both likelihood and log likelihood. This concept is used for both vertices, that is fragments and edges, that is fragmentation reactions or neutral losses.

Scoring Mass Accuracy. For vertices, we use log odds to differentiate between the model (the peak is truly a fragment with the proposed molecular formula) and the background (the peak is noise). We can use the mass difference between the measured peak and the molecular formula to assess the likelihood of the peak being true (model): Mass differences are usually assumed to be normally distributed [46, 111], and we calculate this likelihood as the two-sided area under the Gaussian curve with standard deviation 1/3 of the relative mass error [9].

Scoring Peak Intensity. For the background model, we cannot use the mass of the peak since, in general, noise peaks may appear at any mass. But we can use the peak intensity for this purpose: Evaluations have shown that noise peak intensities are roughly exponentially distributed; see for instance Fig. 4 in [37]. Let $\lambda \exp \lambda x$ be the exponential distribution with parameter λ, where x is the peak intensity. The likelihood of observing a noise peak with intensity y or higher is

$$\mathbb{P}(\text{intensity} \geq y) = \int_y^\infty \lambda \exp \lambda x \, \mathrm{d}x = \exp -\lambda y \tag{3.1}$$

Taking the natural logarithm, we reach $-\lambda y$ for intensity y. Since this likelihood appears in the denominator of the log odds term, we simply add the peak intensity, multiplied by a constant representing the noise in the spectrum, to the score.

Hetero to carbon ratio. All biochemical compounds possess a backbone of carbon atoms. Thus it is reasonable to punish molecular formulas that possess an extraordinary high number of non-hydrogen, non-carbon atoms (called hetero atoms) in relation to the carbon atoms. The hetero to carbon ratio of the KEGG database [49] is normal distributed with mean 0.59 and standard deviation 0.56. We use the log value of this distribution at the hetero to carbon ratio of the candidate molecule. But the ratio of a fragment is strongly influenced by the ratio of its predecessor. Hence, we only apply this score to the formula candidates of the precursor ion. The hetero to carbon ratio of the fragments is taken into account differently, see next section.

Prior probabilities Finally, we can use prior probabilities, computing the odds ratio that any peak is not noise: We add a constant b, being the logarithm of this odds ratio, to each vertex score. This has proven unnecessary when using raw intensities of the instrument, but is mentioned here for completeness.

The resulting score for a vertex is pulled to each of its incoming edges, so that the resulting graph is solely edge-weighted. This simplifies further calculations.

3.3 Scoring Fragmentation Reactions – Weighting Edges

When weighing edges and thus fragmentation reactions, we consider common neutral losses, implausible losses, radical losses, unlikely neutral losses containing only one atom type, the mass of the loss, collision energies, and the ratio between carbon and hetero atoms.

Scoring neutral losses There are certain neutral losses that appear often when analyzing organic and biological compounds. We have created a short list of these common neutral losses; see Table 3.1. We reward the occurrence of a combination of up to three losses from the list by adding $\log(\gamma/n); \gamma > 1$ to the score, where γ is a parameter that has to be chosen individually for each dataset, and n is the number of combined common losses. Combinations may represent groups detaching together or the loss of an intermediate peak, but these cases are not as strongly rewarded.

Analysis by MS experts revealed that certain neutral losses are usually not occurring during fragmentation, but are chosen from time to time by our algorithm. Thus, we have created a list of "implausible" losses, see Table 3.2. If a neutral loss equals an entry of this list, its score is significantly decreased.

Additionally, we penalize losses consisting purely of carbon or purely of nitrogen with $\log(\epsilon)$, $\epsilon \ll 1$, as these are unlikely neutral losses.

Radical losses. The formation of radical fragments is possible by CID fragmentation, though not very common. If a radical fragment is formed, one of the radical losses of Table 3.3 is usually involved. Thus a radical formation with one of these losses is not

loss name	loss formula
Hydrogen	H_2
Water	H_2O
Methane	CH_4
Ethene	C_2H_4
Ethine	C_2H_2
Butene	C_4H_8
Pentene	C_5H_8
Benzene	C_6H_6
Formaldehyde	CH_2O
Carbon monoxide	CO
Formic acid	CH_2O_2
Carbon dioxide	CO_2
Acetic acid	$C_2H_4O_2$
Ketene	C_2H_2O
Propionic acid	$C_3H_6O_2$
Malonic acid	$C_3H_4O_4$
Malonic anhydride	$C_3H_2O_3$
Pentose equivalent	$C_5H_8O_4$
Deoxyhexose equivalent	$C_6H_{10}O_4$
Hexose equivalent	$C_6H_{10}O_5$
Hexuronic equivalent acid	$C_6H_8O_6$
Ammonia	NH_3
Methylamine	CH_5N
Methylimine	CH_3N
Trimethylamine	C_3H_9N
Cyanic Acid	$CHNO$
Urea	CH_4N_2O
Phosphonic acid	H_3PO_3
Phosphoric acid	H_3PO_4
Metaphosphoric acid	HPO_3
Dihydrogen vinyl phosphate	$C_2H_5O_4P$
Hydrogen sulfide	H_2S
Sulfur	S
Sulfur dioxide	SO_2
Sulfur trioxide	SO_3
Sulfuric acid	H_2SO_4

Table 3.1: The common neutral losses used for fragmentation tree calculations. If an entry of this table occurs in a fragmentation step, the score of the step is significantly increased.

“loss name”	loss formula
“Dicarbon monoxide”	C_2O
“Tetracarbon monoxide”	C_4O
“Unsaturated cyclopropane”	C_3H_2
“Unsaturated cyclopentane”	C_5H_2
“Unsaturated cycloheptane”	C_7H_2

Table 3.2: The *implausible losses* used for fragmentation tree calculation in Chapter 5. If an entry from this table occurs in a hypothetical fragmentation step, the score of this step is significantly decreased.

loss name	loss formula
Atomar hydrogen	$H^{\cdot}$
Oxygen radical	$O^{\cdot}$
Hydroxy radical	$^{\cdot}OH$
Methyl radical	$^{\cdot}CH_3$
Methoxy radical	$CH_3O^{\cdot}$
Propyl radical	$^{\cdot}C_3H_7$
tert-Butyl radical	$^{\cdot}C_4H_9$
Phenoxy radical	$C_6H_5O^{\cdot}$

Table 3.3: The *radical losses* used for fragmentation tree calculation in Chapter 5. If an entry from this table occurs in a hypothetical fragmentation step, this is not penalized. Other radical losses are not forbidden, but the score of the corresponding step is significantly decreased.

penalized. All other radical formations are punished by subtracting a certain amount from their score.

Mass of the loss. We penalize large losses by $\log(1 - \frac{\text{mass neutral loss}}{\text{parent mass}})$. This is not justifiable chemically, as large losses are equally likely to occur as small ones. But this score favors fragmentation cascades over fragmentations directly from the precursor ion. This is desirable as star-like trees do not give much information on the fragmentation process. By this score, we ensure that fragments may only be inserted too deep, never or rarely too high in the tree. This results in the phenomenon of “pull-ups” described in Section 3.4

Collision energies Measuring tandem spectra at several distinct collision energies, allows to deduce that some peak cannot be a direct fragment of another peak, if it appears at a lower energy than its presumed predecessor. This conclusion also applies, if there is a spectrum with an intermediate energy, where neither the peak nor its predecessor appear. These cases are strongly punished by adding $\log \alpha$, $\alpha \ll 1$ to the score. If there is no spectrum where both peaks appear, but neither a spectrum where none of the peaks occurs, the peaks may or may not be directly connected. Thus, this situation is slightly punished with $\log \beta$, $\alpha < \beta < 1$ [11].

Hetero to carbon ratio As described in the previous section, the hetero to carbon ratio is a good measure for the biochemical plausibility of a formula. But, since the ratio of a fragment is strongly influenced by the ratio of its predecessor, we do not want to punish an unusual ratio multiple times. Thus a penalty is only given if the ratio of this fragment is worse than its predecessor, i.e. further away from the mean. This is achieved by subtracting the hetero to carbon ratio score of the predecessor from this fragments score. Only if this results in a negative value, this value is added to the edge score [11].

3.4 Assumptions for Fragmentation Tree Calculation

To be able to calculate Here, we describe the assumption made, when describing the complex fragmentation process by a fragmentation tree. Another assumption is necessary to be able to formulate the calculation as an optimization problem.

Different fragmentation pathways may lead to fragments with identical molecular formulas or even identical structure. This is quite easy to see but, unfortunately, makes it practically impossible to formulate our task as an optimization problem: a small fragment may be generated from almost all other fragments, but we only want to record the most likely explanation. Hence, we slightly oversimplify the problem: We demand that each fragment in the fragmentation spectrum is generated by a single fragmentation pathway. That means that any fragment may have at most one "parent fragment" from which it is generated. Thus, we search for a tree inside the fragmentation graph. This allows us to simplify our problem: For every vertex in the fragmentation tree except for the root, which corresponds to the unfragmented compound, we select exactly one incoming edge. Hence, we can move the weight of each vertex into the incoming edges and assume that the fragmentation graph is edge-weighted.

There is one exception to the above reasoning: Assume that some fragment f_3 is cleaved from fragment f_2, and that f_2 is in turn cleaved from f_1. Solely from the tandem MS data and without additional structural information, we cannot rule out that f_3 is directly cleaved from f_1. But this information is implicitly encoded in a fragmentation tree: the fragmentation may occur from the fragment's direct parent in the tree, or from any of its parents. If it is later decided by manual inspection (Chapter 4) or automated annotation using structural information (Chapter 6) that the fragment should rather originate from a fragment higher in the cascade, the fragment is "pulled up" in the tree. Thus we refer to such an event as *"pull-up"*.

Similar to fragmentation pathways resulting in the same fragment, several fragments may result in a single peak in the fragmentation spectrum. We argue that this is extremely rare. On the other side, we have to make sure, that each peak intensity and mass accuracy contributes to the score only once. Thus, we consider it reasonable, to demand that every peak is annotated and thus scored at most once. In our formalism the fragmentation tree has to be colorful: Each vertex color and, hence, each peak in the fragmentation spectrum may occur at most once. Think of it as forcing the algorithm to make a decision. If this restriction were not applied, the algorithm always would choose all explanations.

3.5 Formal Problem Definition

Calculating fragmentation trees under the above mentioned restrictions leads to the Maximum Colorful Subtree problem [11].

Maximum Colorful Subtree problem.
Given a vertex-colored DAG $G = (V, E)$ with colors $\mathcal{C}$ and weights $w : E \to \mathbb{R}$. Find the induced colorful subtree $T = (V_T, E_T)$ of G of maximum weight $w(T) := \sum_{e \in E_T} w(e)$.

This is a special case of the edge-weighted Graph Motif problem, see [89] for an overview. Scheubert *et al.* [84] present the related Colorful Subtree Closure problem for analyzing multiple mass spectrometry data. Ljubíc *et al.* [62] presented an Integer Linear Program for the related Prize-Collecting Steiner Tree problem. The Maximum Colorful Subtree problem is NP-hard [34] as well as APX-hard [28] even on binary trees. Furthermore, on general trees it has no constant factor approximation [28, 90].

3.6 Algorithms for the Maximum Colorful Subtree Problem

In this section, we briefly review exact algorithms as well as heuristics for the Maximum Colorful Subtree problem. We conclude the section with two new heuristics for the problem.

For vertices u and v, let $c(v)$ be the color assigned to v and $w(u, v) \in \mathbb{R}$ the weight of the edge uv. Throughout the rest of the paper we denote the number of colors in $G = (V, E)$ by k. Note that $k \leq p$ can be as large as the number of peaks in the spectrum; but we can also choose a smaller k to decrease running times, limiting our attention to, say, the k most intense peaks.

3.6.1 Exact Methods

Dynamic programming The problem can be solved exactly using dynamic programming over vertices and color subsets [29]. Let $W(v, S)$ be the maximal score of a colorful tree with root v and color set $S \subseteq C$. Now, table W can be computed by the following recurrence [11]:

$$W(v,S) = \max \begin{cases} \max\limits_{u: c(u) \in S \setminus \{c(v)\}, vu \in E} W(u, S \setminus \{c(v)\}) + w(v,u) \\ \max\limits_{(S_1,S_2): S_1 \cap S_2 = \{c(v)\}, S_1 \cup S_2 = S} W(v,S_1) + W(v,S_2) \end{cases}$$

where, obviously, we have to exclude the cases $S_1 = \{c(v)\}$ and $S_2 = \{c(v)\}$ from the computation of the second maximum. Using the above recurrence with the initial condition $W(v, \{c(v)\}) = 0$, we can compute a maximum colorful tree in $O(3^k k\,|E|)$ time and $O(2^k\,|V|)$ space. The exponential running time and space make the algorithm useful only for small size instances. The running time can be somewhat improved to $O(2^k \cdot \mathit{poly}(|V|, k))$ by using the Möbius transform and the inversion technique of Björklund *et al.* [6]. However, the technique only works for suitably small integer weights.

Guillemot and Sikora [39] suggest a different approach using multilinear detection [57] for input graphs with unit weights. Their algorithm requires $O(2^k \cdot \mathit{poly}(|E|, k))$ time and only polynomial space. The algorithm can be adopted to integer weight graphs in a straight forward manner but the resulting algorithm would be pseudo-polynomial, i.e., its running time would depend polynomially on the integer weights thus making it impractical for our purposes. To the best of our knowledge, neither the above algorithm nor the dynamic programming with Möbius transform of the previous paragraph have been used in implementations.

Brute force For small instances a brute-force approach is suggested in [11]. The idea is to find a maximum subtree for each possible combination of vertices forming a colorful set. We then search a maximum subtree in a colorful DAG. Clearly, when all edge weights are positive, then the maximum subtree is a spanning tree. This can be found by a simple greedy algorithm, choosing the maximum weight incoming edge for each vertex but the root. With arbitrary edge weights, the problem becomes NP-hard. We solve the problem naively by iterating over all combinations of vertices whose best incoming edge has a negative weight. The brute-force approach is obviously not practical when either the number of combinations is large or when there are many vertices whose maximum incoming edge has negative weight.

Integer Linear Programming. Integer Linear Programs (ILPs) have proven useful in providing quick exact solutions to NP-hard problems. To construct an ILP for the MAXIMUM COLOURFUL SUBTREE problem, let us define a binary variable x_{uv} for each edge uv of the input graph. For each color $c \in C$ let $V(c)$ be the set of all vertices in $G = (V, E)$ which are colored with c. Then, the following objective and constraints represent the problem:

$$\max \sum_{uv \in E} w(u,v) \cdot x_{uv} \tag{3.2}$$

$$\begin{aligned}
\text{s.t.} \quad \sum_{u \text{ with } uv \in E} x_{uv} &\leq 1 && \text{for all } v \in V \setminus \{r\}, && (3.3)\\
x_{vw} &\leq \sum_{u \text{ with } uv \in E} x_{uv} && \text{for all } vw \in E \text{ with } v \neq r, && (3.4)\\
\sum_{uv \in E \text{ with } v \in V(c)} x_{uv} &\leq 1 && \text{for all } c \in C, && (3.5)\\
x_{uv} &\in \{0,1\} && \text{for all } uv \in E. && (3.6)
\end{aligned}$$

In the above integer program, the constraints set (3.3) ensures that the feasible solution is a forest, whereas the constraints set (3.5) make sure that there is at most one vertex of each color present in the solution. Finally, (3.4) requires the solution to be connected. Note that in general graphs, we would have to ensure for every cut of the graph to be connected to some parent vertex. That would require an exponential number of constraints [62]. But since our graph is directed and acyclic, a linear number of constraints suffice.

3.6.2 Heuristics

Greedy Heuristic A simple greedy heuristic has been proposed in [11]. It works by considering the edges according to their weights in descending order. The edge being considered is added to the result, if it does not conflict with the previously picked edges. The algorithm continues until all positive edges are considered and the resulting graph is connected. An edge conflicts with another if they either are incoming edges to the same vertex or are incident edges to different vertices of the same color. Finally, we prune the leaves which are attached by negative weight edges in the resulting spanning tree. We refer to the above heuristic as *greedy* in the rest of the paper.

Insertion Heuristic Another greedy strategy is to consider colors in some ordering and for the current color add an vertex of that color that promises the maximum increase of the score and attaches it to the already calculated tree. The resulting heuristic, called *insertion heuristic* in the rest of the paper, begins with only the root as the current partial solution. The heuristic greedily attaches vertices labeled with unused colors. For every vertex u with unused color, and every vertex v already part of the solution, we calculate how much we gain by attaching u to v. To calculate the gain of attaching u to v, we take into account the score of the edge vu, as well as the possibility of rerouting other outgoing edges of v through u. The vertex with maximum gain is then attached to the solution, and edges are rerouted as required.

Tree Completion Heuristic As noted before, the dynamic programming approach of Section 3.6.1 works only for small inputs. We now present a heuristic that combines DP with the greedy approaches. For a small enough constant b, the heuristic works by first computing the maximum colorful subtree consisting of at most b vertices, which we call *backbone* of a candidate solution. Next, we complete the backbone by using one of the greedy heuristics discussed above.

When using the insertion heuristic, vertices of the remaining colours are now added to the computed backbone according to the rules of this heuristic. Similarly, the *greedy* heuristic can be used to complete the tree by starting with the backbone and applying the greedy heuristic on the remaining edges. In our experiments, we use the *insertion* heuristic for tree completion since it achieved consistently better scores. This heuristic is referred to as DP_b, where b is the size of the backbone computed exactly.

3.7 Algorithm Evaluation

In our study, we analyze spectra from three real and two randomly generated datasets. The *Orbitrap* dataset consists of mass spectra of 38 compounds with a mass accuracy of 10 ppm [77]. The *Micromass* dataset [41] contains spectra of 100 compounds with an accuracy of 50 ppm, while the *QSTAR* dataset [11] consists of 36 mass spectra with 20 ppm accuracy. Except for one additional compound in the *Orbitrap* dataset, these datasets are identical to those used for tree evaluation, see Section 4.1 and the respective publications for details.

For each compound in the above three datasets, we assume that we know the correct molecular formula and construct a directed acyclic graph as described in Section 3.1. We use the same scoring as the tree evaluation in Chapter 4 to weight the edges.

Our first randomly generated dataset, called *Random*, consists of 17 DAGs with the number of colors ranging from 20 to 100 in steps of 5. We generate 3 vertices for every color except the root, which is a unique vertex with color 0. For k colors, our DAG consists of $3(k-1)+1$ vertices and $\frac{9}{2}k(k-1)-6(k-1)$ edges, where each vertex with color i is connected to all vertices colored $i+1,\ldots,k$ and has weight drawn from normal distribution with mean -8 and standard deviation 10. And finally the last dataset we consider is called *Hard* which is generated with Model RB of Xu and Li [109] that produces instances of *constraint satisfaction problem (CSP)* with exponential resolution proof complexity (with high probability). For each $10 < c \leq 30$, we generate a CSP instance with constraint sizes c and convert these CSP instances to the instances of MAXIMUM SUBTREE problem using a standard reduction to the *maximum independent set* problem and the reduction presented in [78].

We implemented the exact algorithms based on the dynamic program, the integer linear program and the brute-force approach of Section 3.6.1. We also implemented the *greedy* heuristic and the tree completion heuristic with backbone size 10 and 15 (DP_{10}, DP_{15}) that use *insertion* to complete the backbone. To evaluate an heuristic on an instance of the problem, we consider its *performance ratio*, i. e., the ratio of the weight of generated solutions versus the optimal.

The algorithms are implemented in Java 1.6 by using an adjacency list representation for graphs. In the *DP* algorithm, we use the Java `long` data type to represent sets of colors as bitsets. This limits the maximum possible size of the color set to 64. Memory usage becomes prohibitive long before this number is reached. The experiments were run on a Lenovo T400 laptop powered with dual core Intel P8600 at 2.40 GHz with 2 GB of RAM and running Ubuntu Lucid Lynx as operating system.

Our implementation is single threaded and does not exploit the availability of multiple cores in the system. The integer linear programming solvers, however, use multiple cores. We run the experiments with default heap size on a Sun Java server virtual machine.

For our applications, an algorithm is sufficiently fast if it runs in less than ten seconds, since this is usually faster than the data can be acquired. Among the exact algorithms, only the ILP (Section 3.6.1) managed to solve all instances of our datasets. With the *Gurobi Optimizer*[1] the running time stayed under 5.6 minutes per instance while for about 95% of the instances, it terminated in at most 5 seconds. With CPLEX[2], the running time was usually comparable and we were able to solve all instances of the real world datasets except one in under 7.5 minutes. The CPLEX solver was noticeably slow on the *Hard* dataset where it did not finish in 2 hours for larger instances in the dataset.

The brute force algorithm mentioned in Section 3.6.1 runs fast on most instances of the *Orbitrap* and *QSTAR* dataset. Due to the high mass accuracy and the small compound sizes in these datasets there are only few explanations per peak and thus few vertices with the same color in the input graph. Edges with negative weights are rare. Therefore only a few calls to the spanning tree algorithm are necessary, resulting in short running times (under a second for all but three instances in *Orbitrap* and *QSTAR*) and exact solutions. But for two compounds from the *Orbitrap* dataset and 37 compounds in the *Micromass* dataset, the algorithm does not terminate in 12 hours and a week, respectively.

The DP algorithm (Section 3.6.1) was able to solve the *QSTAR* instances exactly, since the number of colors and vertices is small for this dataset. The algorithm failed on the other two datasets, since either the exponential memory usage became infeasible or running times exceeded several days.

In Figure 3.2 and 3.3, we present performance ratios achieved by several heuristics. The tree completion heuristics (DP_{10} and DP_{15}) work very well on *Micromass*, *Orbitrap* and *Random* datasets with an output tree of weight at least 80 percent of the optimal for the DP_{15} variant. On the other hand, the *greedy* heuristic performs inferior to both DP_{10} and DP_{15}.

The *insertion* heuristic performs better than the *greedy* heuristic in our experiments on real datasets, while on the *Random* dataset, the *greedy* heuristic generate trees with better scores. All heuristics completely fail on the *Hard* dataset, where most of the time they return the empty tree as output. We also observe improved performance in general for the tree completion heuristic as we increase the parameter, i. e., the size of the backbone computed exactly. But the performance increase is only marginal for real-world datasets, as can be seen in Figures 3.2 and 3.3. The tree completion heuristic becomes infeasible when the size of the backbone is 25. In this case, more than half of the instances from Orbitrap and Micromass datasets fail to terminate in less than a week.

[1] *Gurobi Optimizer 4.5.* Houston, Texas: Gurobi Optimization Inc., April 2011.

[2] *IBM ILOG CPLEX Optimization Studio 12.3.* Armonk, New York: IBM Corporation, June 2011.

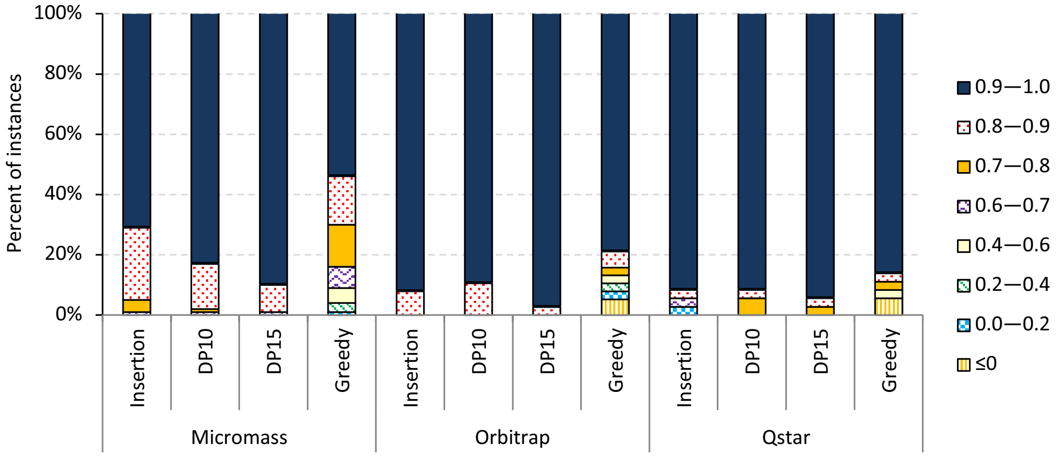


Figure 3.2: Performance ratios achieved by different heuristics on *Micromass*, *Orbitrap*, and *QSTAR* datasets.

The *insertion*, *greedy* and DP_{10} heuristics are fast with running times well under a second, whereas the DP_{15} heuristic terminates in less than 8 seconds for all instances. The algorithm based on integer programming also finished in at most 16 seconds for any instance, while it was actually faster on most of the instances. Both integer programming solvers showed comparable performance except on the *Hard* instances, where CPLEX was noticeably slower than the Gurobi solver. Figure 3.4 and 3.5 present the breakdown of datasets depending on how much time it took to solve them using different algorithms. Note that the running times mentioned do not include the time needed to construct the graph representations from MS data.

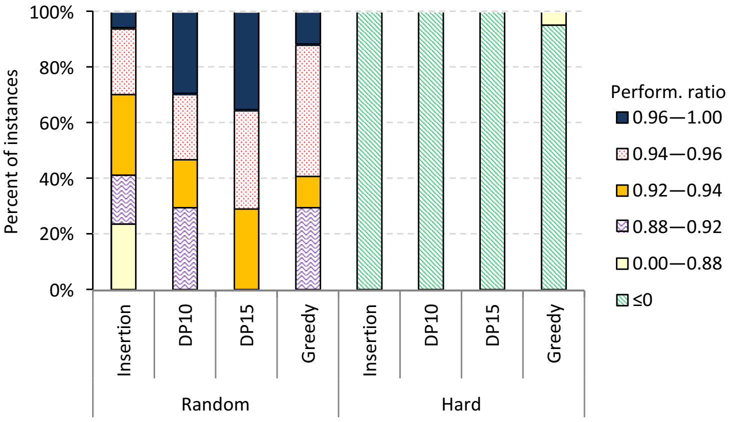


Figure 3.3: Performance ratios achieved by different heuristics on *Random* and *Hard* datasets.

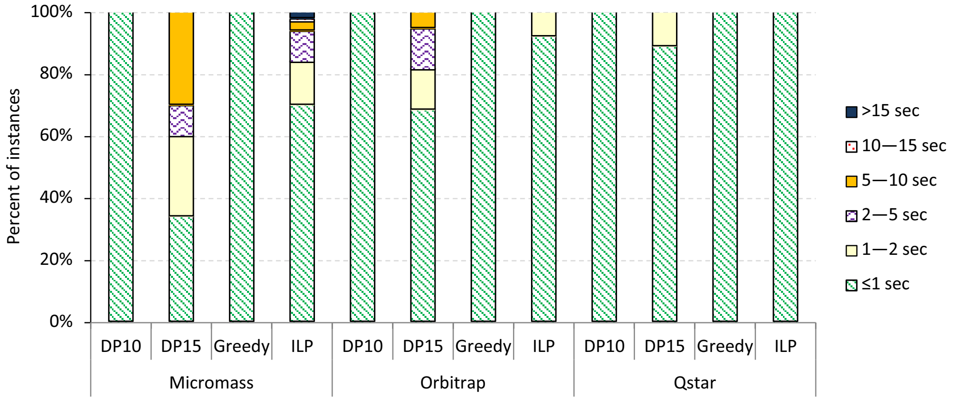


Figure 3.4: Running times taken by different heuristics on *Micromass*, *Orbitrap* and *QSTAR* datasets, where ILP denotes the algorithm based on integer programming with Gurobi solver.

Since the structure of the tree is highly relevant for tree alignment (Chapter 5) and strucutral annotation of trees (Chapter 6), it is most probably beneficial to find exact solutions. Our tests show that the integer linear program performs best on this task.

3.8 Using Fragmentation Trees to Determine Molecular Formulas

Numerous methods to determine the molecular formula of small molecules without any user interaction have been published recently [9, 11, 41, 53, 80].

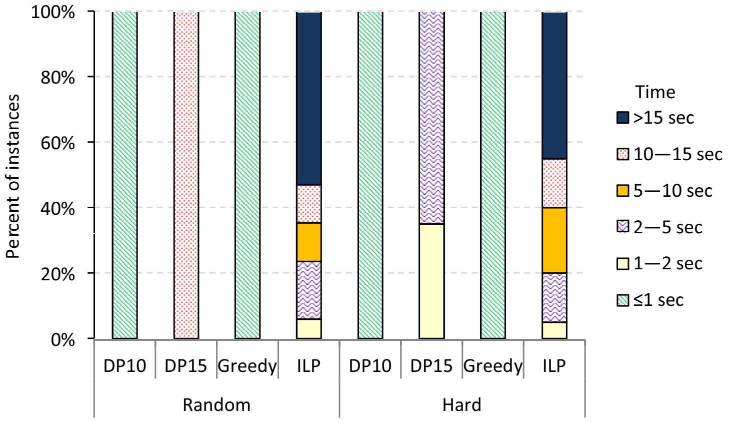


Figure 3.5: Running times taken by different algorithms on *Random* and *Hard* datasets, where ILP denotes the algorithm based on integer programming with Gurobi solver.

To use fragmentation trees for molecular formula prediction, we use each molecular formula within mass accuracy of the precursor ion mass as candidate formula. For each of these formulas, we generate a fragmentation graph and calculate the best scoring tree as described. The score of this tree is then used as score for its candidate formula as proposed in [11]. When determining the molecular formula, hundreds of instances (one per candidate formula) have to be solved for a single compound, but only the scores are relevant. In this case, the tree completion heuristic with parameter 10 provides a good performance ratio of 95% on average, and very fast running times.

Unfortunately, this approach alone is insufficient to determine the formulas of large, complex metabolites. To this end, we use isotope pattern either from additional single MS measurements or the precursor ion of the MS^2 spectrum, if the mass filter was wide enough to let the isotopes pass. We calculate scores based on the isotopic pattern using the method by Böcker *et al.* [9].

Finally, we combine results of the two identification methods: This combined score is $5 \log p_{iso} + s_{frag}$ where p_{iso} is the likelihood from the Bayesian analysis of isotope patterns, and s_{frag} is the score of the fragmentation pattern analysis. The constant is chosen to make the scores comparable.

4 Evaluation of Fragmentation Tree Quality

To ensure that fragmentation trees calculated in the previous chapter are a viable explanation of the spectrum, we calculated trees from three different datasets using the DP_{15} heuristic as described in the previous chapter and experts evaluated their quality in [77]. At the time of the evaluation, the DP_{15} algorithm was the most exact algorithm that was computationally feasible. As manual evaluation is extremely time consuming, we refrained from repeating this step after development of feasible exact methods, but random samples indicate that tree quality further increases with an exact method.to

In this chapter, we present the comparison of our trees against expert knowledge and multi-stage mass spectrometry. We also evaluate our annotations against a software tool with an entirely different annotation approach.

4.1 Datasets and Parameter Choice

We will evaluate fragmentation trees from three data sets in the remainder of this chapter (see Table 4.1). These were measured with two different instrument types from three different manufacturers and demonstrate that our method is applicable to a wide variety of data. The first data set consists of 37 compounds, mostly representing plant secondary metabolites, measured on an Orbitrap mass spectrometer [77]. The second contained 42 compounds measured on an API QSTAR [11]. The third data set with 102 compounds was measured on a Micromass Q-TOF instrument [41]. Two compounds from this data set were excluded, since precursor peaks had mass accuracy worse than 50 ppm. Tables A.1, A.2 and A.3 in the appendix list the molecules of the data sets.

Instrument	ppm[a]	CID (eV)	#[b]	mass range	average
Orbitrap [77]	5	35,45,55,70	37	152.0–822.4 Da	345.2 Da
API QSTAR [11]	20	15,25,45,55,90[c]	42	89.0–441.2 Da	207.5 Da
Micromass QTOF [41]	20	10,20,30,40,50	100	137.1–609.3 Da	372.5 Da

Table 4.1: Datasets used in this study. [a]Mass accuracy of the measurement, [b]number of compounds used for evaluation, [c]three to five distinct collision energies were measured of each compound.

For the calculation of the fragmentation trees, we did not use the list of implausible losses, as it is a result of this study. Additionally, we were not aware of the fact that radical formation may occur during fragmentation. Thus, radical ions were completely forbidden for this calculation. To merge the spectra we applied a relatively large mass window of 50 mDa. This improved mass accuracy in the QSTAR dataset.

For our analysis, the following parameter values were used: For all datasets set $\lambda = 0.1$ and $\epsilon = 10^{-4}$, as well as $\alpha = 0.1$ and $\beta = 0.8$, the defaults in [11]. Parameters γ, b were chosen to capture instrument-specific properties: For example, the QSTAR instrument produces relatively few fragment peaks, but these often reflect typical losses. For the Orbitrap data, we use $\gamma = 10$ and $b = 5$; for the Micromass QTOF data $\gamma = 10$ and $b = 0$; and for the API QSTAR data $\gamma = 1000$ and $b = 0$.

These parameters are either naturally occurring, such as the mass accuracy, published as default values [9,11] or chosen ad-hoc. We did not optimize the parameters used, since in this study the amount of training data is small compared to the number of parameters in the optimization. In this case, an optimization may lead to so called “overfitting”, resulting in poor generalization results.

4.2 Evaluation against Expert Knowledge

Mass spectrometry experts experienced in the structural elucidation of natural products manually evaluated the fragmentation trees of all 79 compounds in the Orbitrap and API QSTAR dataset.

For these two datasets, accurate isotopic patterns from single MS measurements were available. We combined the isotopic pattern comparison from [9] with the calculation of a fragmentation tree for each candidate molecular formula, as described in Section 3.8. With this approach we assigned the correct molecular formula to each of the compounds in these two datasets, see Tables A.1 and A.2 for detailed results.

From now on, we only consider the trees of the correct molecular formulas. As manual evaluation, fragmentation trees were compared with expected fragmentation patterns that the experts deduced from the provided chemical structures and merged CID spectra. All known fragmentation reaction mechanisms were taken into account, see [14, 42, 64, 86] for the details. First, the theoretical fragmentation pathway was formulated based on the fragmentation rules for protonated even-numbered electron ions. Individual edges in the pathway were compared to those in the fragmentation tree, and matching losses were assigned as “correct”. Numerous agreements between neutral losses listed in Table 3.1 and manually assigned fragmentation steps were found. The experts found “pull-ups” as defined in Section 3.4 in some trees. We evaluate those “pull-up” edges as “correct”, since without a given structural formula and solely MS^2 data the “correct” case cannot be distinguished from our method’s suggestion. Some losses assigned by the automated method cannot be ruled out, but experts were unable to rationalize them in a fragmentation pathway; these losses are annotated as “unsure”. Edges which result in molecular fragments with questionable stability under experimental conditions, and those that cannot be explained via a

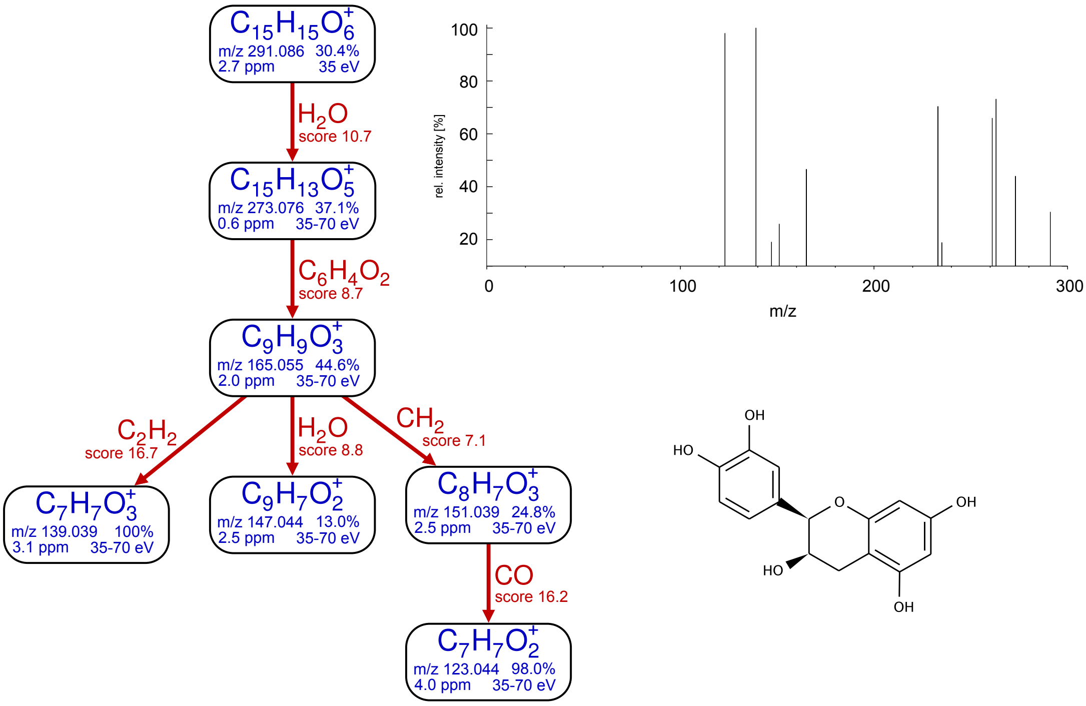


Figure 4.1: Hypothetical fragmentation tree of (-)-epicatechine ($C_{15}H_{14}O_6$) computed by our method using Orbitrap data. Nodes (blue) correspond to peaks in the tandem mass spectra and their annotate molecular formula (CE is the range of collision energies), edges (red) correspond to hypothetical neutral losses.

"pull-up", were assigned as "wrong'. Whenever possible, literature sources were used to support the assignment; however, not all references provided useful data due to a lack of well-evaluated CID fragment spectra of metabolites in the literature.

For protonated (-)-epicatechine, we now describe in detail how we evaluated the fragmentation tree (Figure 4.1): The fragmentation pathway was based on the CID fragmentation of structurally related kaempferol [63] as no reliable literature on (-)-epicatechin exists. Obvious water and $C_6H_4O_2$ o-chinone neutral losses followed by another water loss (nodes 291, 273, 165, 151) were found in the calculated tree and annotated as "correct". A loss of CO from node 151 is possible, but the abundant m/z 123.045 is more likely formed by an retro Diehls-Alder reaction from protonated (-)-epicatechin. Acetylene loss (edges between nodes 165-139) cannot be excluded; however, carbene (CH_2) loss is rather unlikely and considered "wrong". For example, edges between nodes 273-165-139 can be combined to the expected neutral loss of 3,4-dihydroxyphenyl acetylene, so the loss of acetylene was reconsidered as "correct". Pull-up of edges between nodes 291-273-165-151 results in a total loss of $C_7H_8O_3$, and the carbene loss can be considered as "correct" by pull-up. However, as this loss is not very common, it was annotated as "unsure". In all evaluated trees, similar reasoning processes were used to evaluate the hypothetical fragmentation trees. We find that the losses of O, C, N, $: CH2$, C_2O, C_4O, C_3H_2, C_5H_2, C_7H_2 were the edges most frequently annotated as "wrong".

For the Orbitrap dataset, 352 of 458 neutral losses (76.9%) were assigned as "correct", 57 (12.4%) as "unsure", and 49 (10.7%) as "wrong". In cases of methoxylated aromatic compounds, well-pronounced radical losses, namely $^{\cdot}CH_3$,$HO^{\cdot}$ and $CH_3O^{\cdot}$, were not presented in the calculated trees of compounds such as berberine or emetine although the corresponding peaks were found in the spectra. It should be understood that this is solely a problem of the objective function used, not of the general approach: We will include radical losses in a subsequent program version. For the QSTAR dataset, 286 of 350 losses (81.7%) were assigned as "correct", 51 (14.5%) as "unsure", and only 13 (3.7%) as "wrong". For 15 fragmentation trees in the Orbitrap dataset and 22 trees in the QSTAR dataset, all losses in the tree were annotated as "correct". See Tables A.4 and A.5 in the appendix for details. In general, the calculated trees are very close to the experts' assignment, which is remarkable if we consider the comparatively simple optimization objective the automated assignment is based on, compared to years of experience on the human side. Note that the experts knew the correct molecular structure during evaluation, whereas it is unknown to our method during fragmentation tree calculation.

4.3 Evaluation Using Multi-stage MS

Since the Orbitrap mass spectrometer is capable of recording multi-stage spectra, the experts evaluation could be refined by MS^3 and MS^4 spectra. Those were measured for the ions of high abundance in the CID spectra, and of those forming branching nodes of calculated trees. The resulting multi-stage MS spectra were manually annotated.

For (S,R)-noscapine, MS^3 of m/z 414→396 and 414→220 transitions were recorded using a linear trap for a precursor ion preparation/selection and an orbitrap analyzer for ion detection. Additionally, MS^4 of m/z 414→396→378 was obtained, see Fig. 4.2. Transition 414→396 supported the direct formation of m/z 378 and 365 fragment ions from m/z 396. A strong peak at m/z 381, not included in the calculated tree and corresponding to radical loss ($CH_3^{\cdot}$), was noticed. Transition 414→396→378 did not support the direct edge between m/z 378 and fragment ions 248 and 220. Those are likely formed directly from protonated noscapine. The lineage of ions 179 from m/z 220 was confirmed by transition 414→220; however, the most intense is a methyl radical loss providing m/z 205. When comparing the pathway this data suggests with the tree calculated from MS2 data, five edges are "correct" and two are pull-ups. No wrong assignment was made, see Fig. 4.3.

For the more complex tree of chelidonine in Fig. 4.4, MS3 data also strongly supported the calculated fragmentation tree; see Fig. 4.5 and 4.6. The main backbone pathway (354-323-295-293-275-247) was fully supported with one exception. The edge connecting nodes 295-293 is incorrect (due to the loss of molecular hydrogen), as m/z 293 is formed from m/z 323 by the loss of formaldehyde, and nodes 323 and 293 are

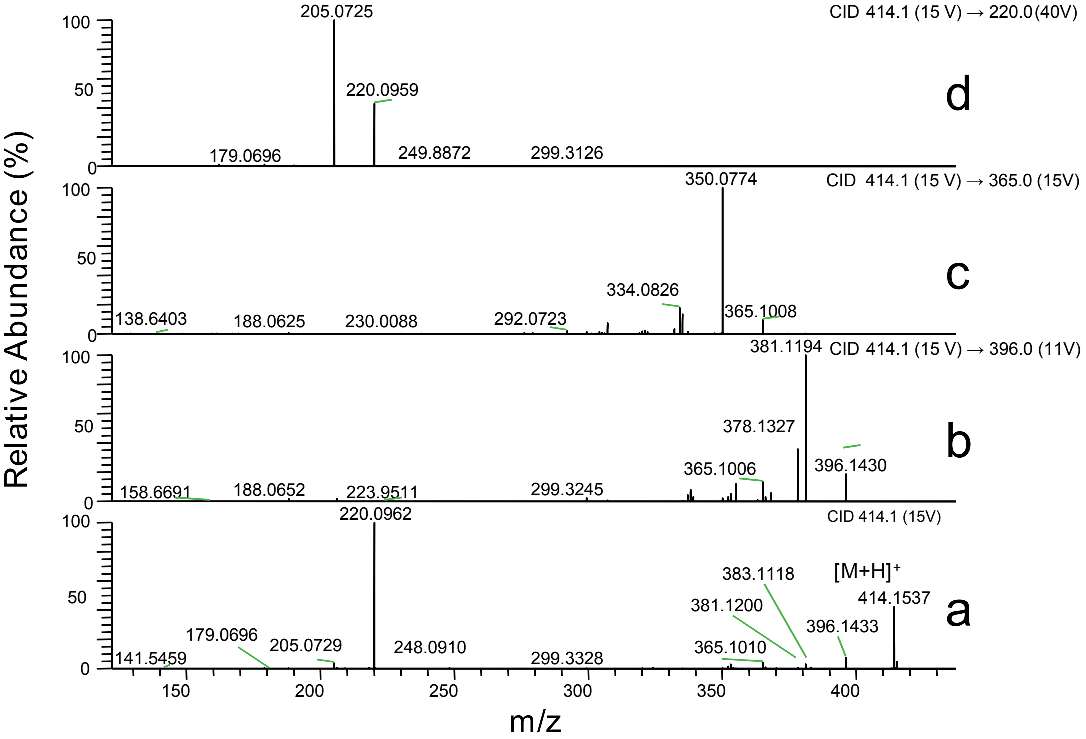


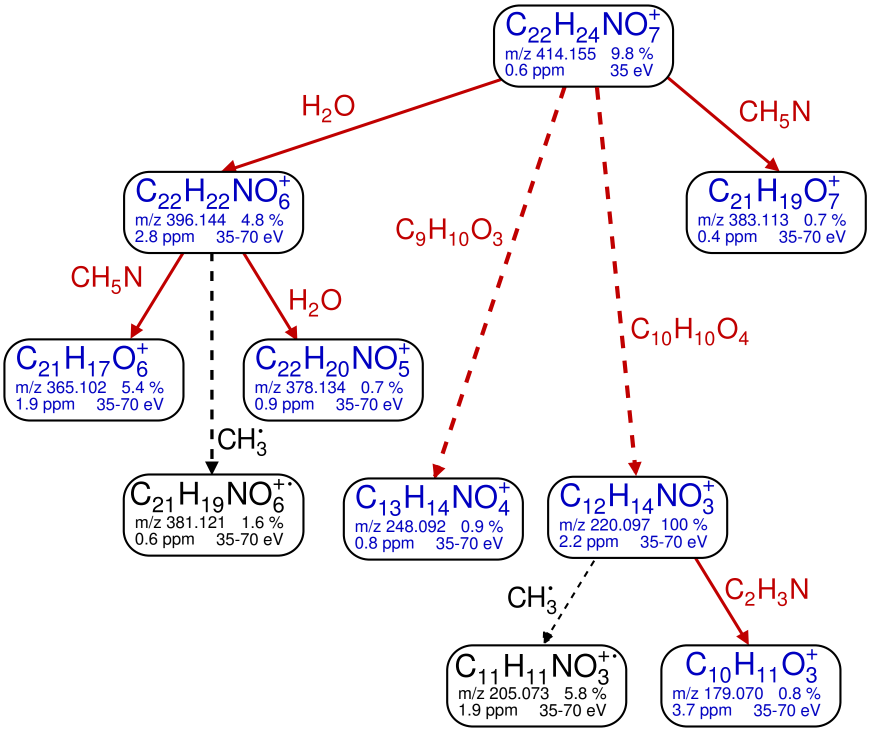
Figure 4.2: Multi-stage MS experiments performed with protonated (S,R)-noscapine generated by electrospray and analyzed with an Orbitrap XL instrument. Fragmentation was realized in linear trap using He as collision gas. (a) CID MS^2 spectrum generated from molecular adduct ion [M+H]+ using 15 V in linear trap (other used CID voltages given in brackets). (b–d) MS^3 spectra; transitions are given in inserts in bold, used collision energies are indicated in brackets.

directly connected. Node 295 remains in the tree but forms a new branch (323-295, loss of CO). The third generation 305 node can be formed both from nodes 323 and 326. This connection is not visible in the calculated tree as this would violate the tree property.

Altogether, the multi-stage MS experiments demonstrated the close similarity of calculated trees and MS^n-spectra-derived fragmentation pathways and supports the annotation of “correct”, “unclear”, and “wrong” neutral losses in experts’ evaluation; see Tables A.4 and A.5 in the appendix.

4.4 Evaluation against Mass Frontier

For further evaluation of our method, we compare the molecular formulas our method assigns to the peaks, with the predictions of the Mass Frontier software. For this,



$C_{22}H_{24}NO_7^+$ m/z 414.155 9.8 % 0.6 ppm 35 eV
H_2O
CH_5N
$C_9H_{10}O_3$
$C_{10}H_{10}O_4$
$C_{22}H_{22}NO_6^+$ m/z 396.144 4.8 % 2.8 ppm 35-70 eV
$C_{21}H_{19}O_7^+$ m/z 383.113 0.7 % 0.4 ppm 35-70 eV
CH_5N
H_2O
$C_{21}H_{17}O_6^+$ m/z 365.102 5.4 % 1.9 ppm 35-70 eV
$C_{22}H_{20}NO_5^+$ m/z 378.134 0.7 % 0.9 ppm 35-70 eV
$CH_3^{\bullet}$
$C_{21}H_{19}NO_6^{+\bullet}$ m/z 381.121 1.6 % 0.6 ppm 35-70 eV
$C_{13}H_{14}NO_4^+$ m/z 248.092 0.9 % 0.8 ppm 35-70 eV
$C_{12}H_{14}NO_3^+$ m/z 220.097 100 % 2.2 ppm 35-70 eV
$CH_3^{\bullet}$
C_2H_3N
$C_{11}H_{11}NO_3^{+\bullet}$ m/z 205.073 5.8 % 1.9 ppm 35-70 eV
$C_{10}H_{11}O_3^+$ m/z 179.070 0.8 % 3.7 ppm 35-70 eV

Figure 4.3: (S,R)-Noscapine experimental fragmentation pathway from MS^3 and MS^4 experiments; numbers below the formulas represent m/z ratios. Edges in red and nodes in blue are present in the calculated fragmentation tree (Fig. 3.1 on page 21); the dashed edges represent pull-ups. Black nodes and dashed black edges represent intense ions which are missing in the tree. Five correct edges, two pull-ups, and no wrong neutral loss annotations were found by experimental validation and expert evaluation.

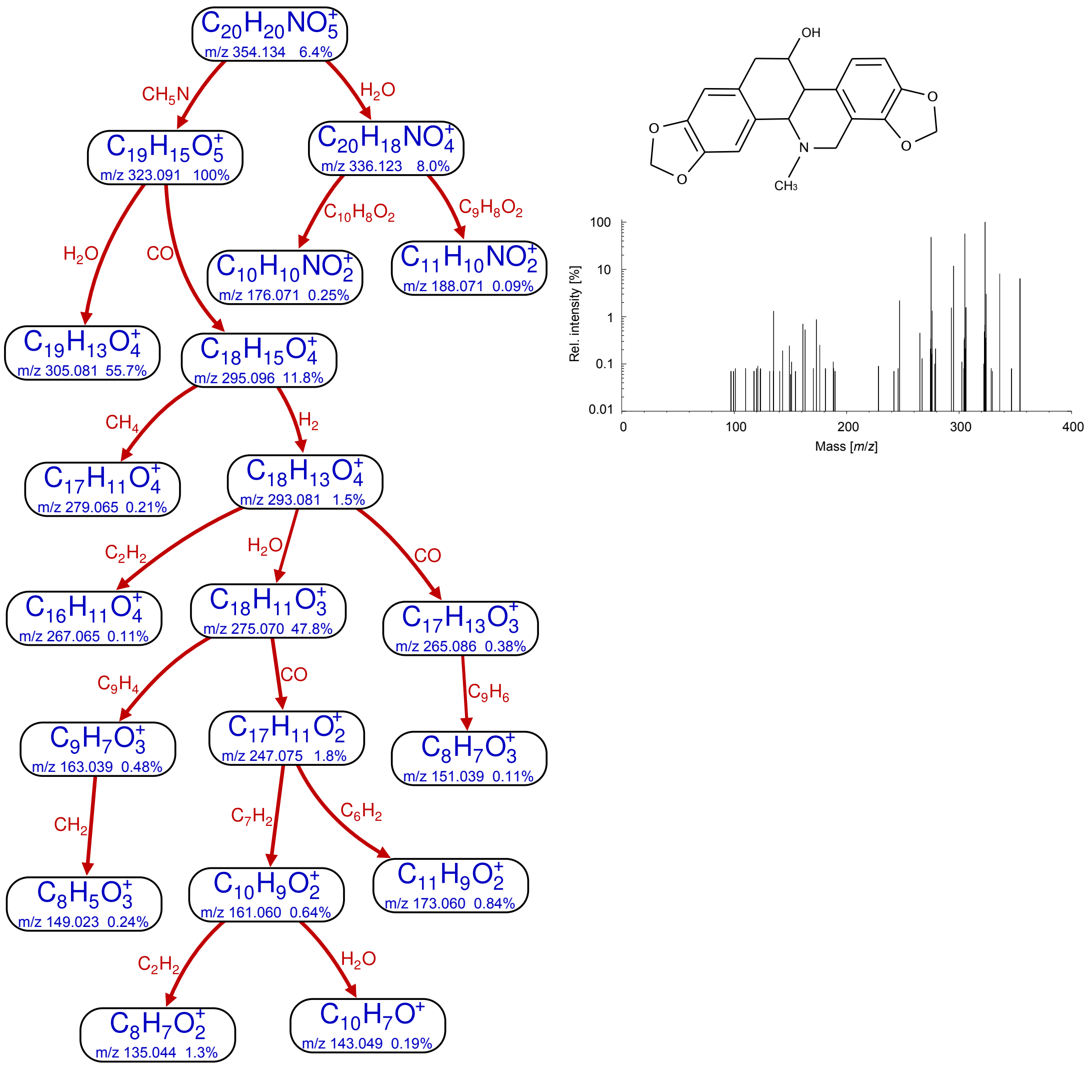

C20H20NO5+ m/z 354.134 6.4%
CH5N
H2O
C19H15O5+ m/z 323.091 100%
C20H18NO4+ m/z 336.123 8.0%
C10H8O2
C9H8O2
H2O
CO
C10H10NO2+ m/z 176.071 0.25%
C11H10NO2+ m/z 188.071 0.09%
C19H13O4+ m/z 305.081 55.7%
C18H15O4+ m/z 295.096 11.8%
CH4
H2
C17H11O4+ m/z 279.065 0.21%
C18H13O4+ m/z 293.081 1.5%
C2H2
H2O
CO
C16H11O4+ m/z 267.065 0.11%
C18H11O3+ m/z 275.070 47.8%
C17H13O3+ m/z 265.086 0.38%
C9H4
CO
C9H6
C9H7O3+ m/z 163.039 0.48%
C17H11O2+ m/z 247.075 1.8%
C8H7O3+ m/z 151.039 0.11%
CH2
C7H2
C6H2
C8H5O3+ m/z 149.023 0.24%
C10H9O2+ m/z 161.060 0.64%
C11H9O2+ m/z 173.060 0.84%
C2H2
H2O
C8H7O2+ m/z 135.044 1.3%
C10H7O+ m/z 143.049 0.19%
OH
O
O
N
O
O
CH3
Rel. intensity [%]
100
10
1
0.1
0.01
0
100
200
300
400
Mass [m/z]

Figure 4.4: Hypothetical fragmentation tree of chelidonine (C20H19NO5) computed by our method using Orbitrap data. Nodes (blue) correspond to peaks in the tandem mass spectra and their annotated molecular formula, edges (red) correspond to hypothetical neutral losses.

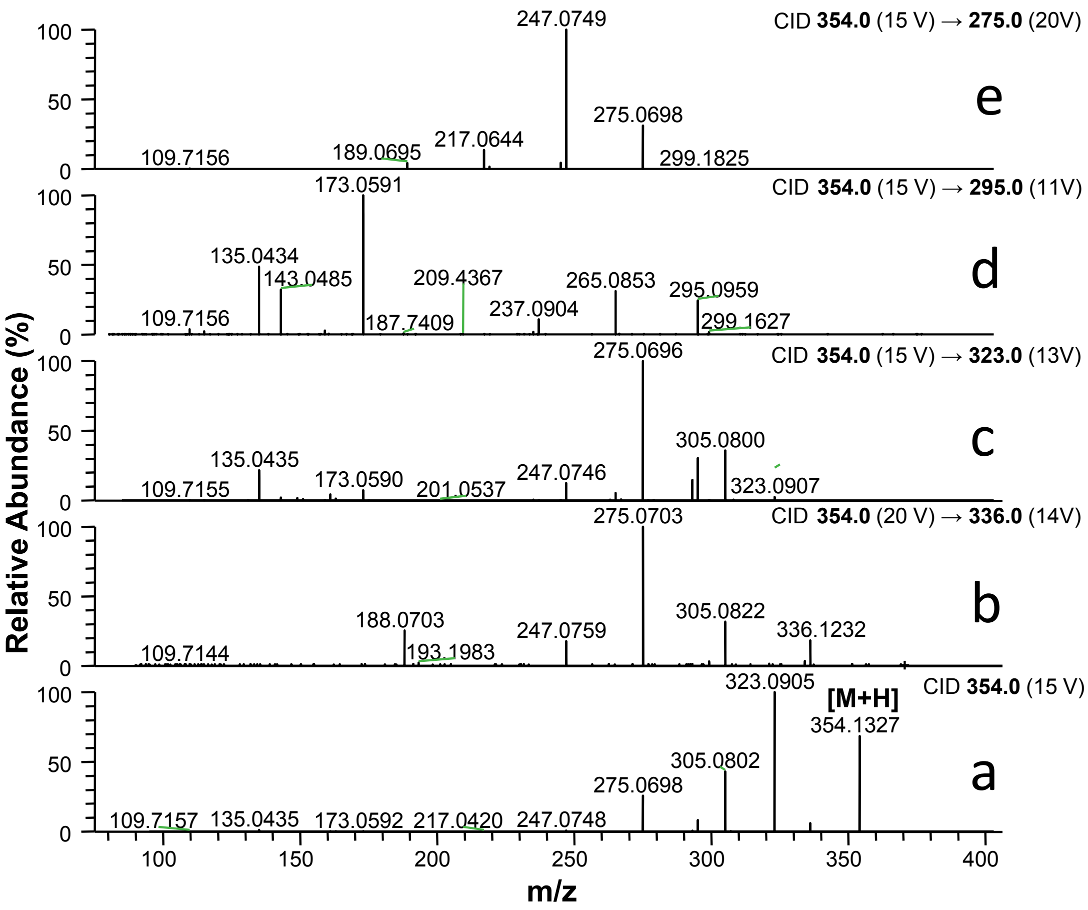


Figure 4.5: Multi-stage MS experiments performed with protonated chelidonine generated by electrospray and analyzed with an Orbitrap XL instrument. Fragmentation was realized in linear trap using He as collision gas. (a) CID MS^2 spectrum generated from molecular adduct ion [M+H]+ using 15 V in linear trap (other used CID voltages given in brackets). (b–e) MS^3 tandem mass spectra; transitions are given in inserts in bold, used collision energies are indicated in brackets.

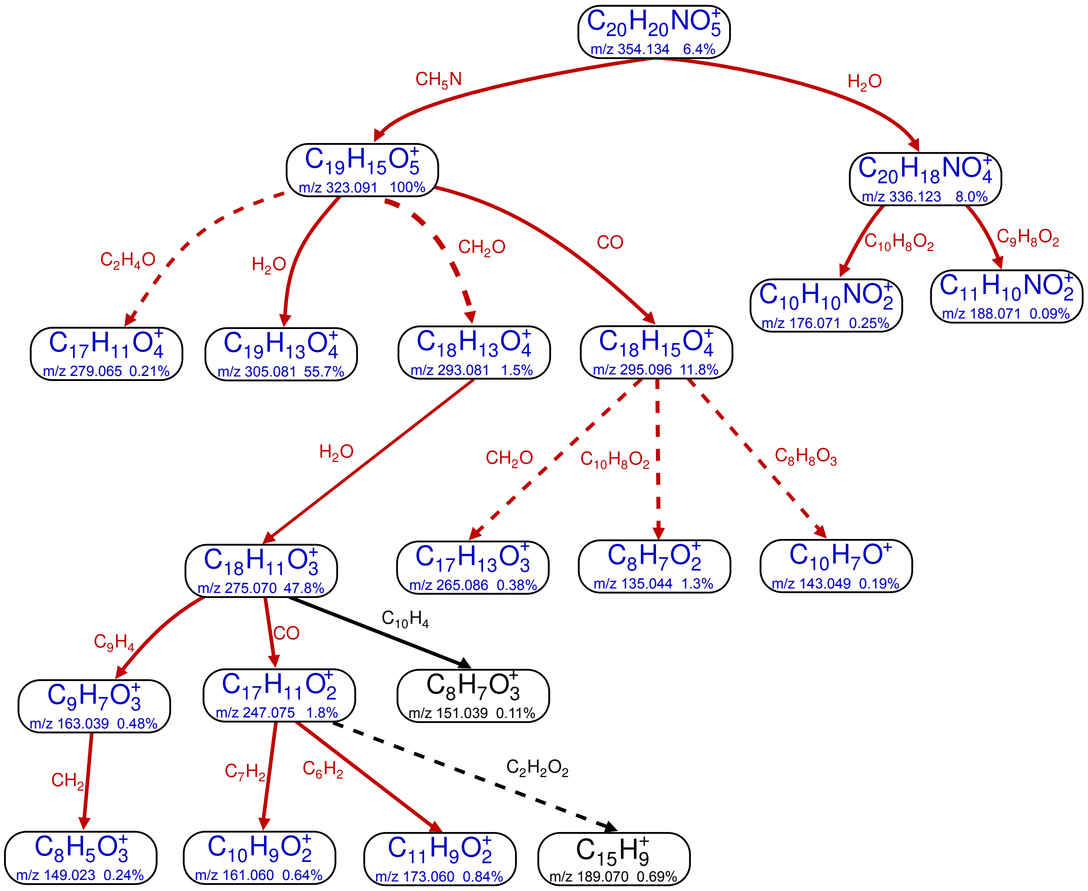


Figure 4.6: Fragmentation pathway from chelidonine MS^3 and MS^4 experiments. Nodes in blue and edges in red correspond to the calculated tree (Fig. 4.4). Dashed red edges represent "pull-ups". Solid black edges differ from the calculated tree. Dashed black edges were not present in the tree. We omitted peaks not occuring in the MS^2 spectra. The annotation of some peaks below one percent intensity could not be verified by multi-stage MS, these peaks are also not shown.

we use the Micromass QTOF dataset. Hill *et al.* performed these predictions using Mass Frontier Version 4 with a different goal in mind [41]. They used Mass Frontier in protonated ion mode with "rules" fragmentation mechanism and a reaction number of 5. Given the molecular structure of the compound, Mass Frontier predicts tandem mass spectra, which we match to the observed data.

For the 1072 peaks that both tools annotate, the same molecular formula is assigned in 97.3% of the cases (1043 peaks). This is an excellent agreement, taking into account the completely different paradigms of the two tools: Mass Frontier knows the molecular structure but not the experimental MS data, whereas our tool knows the experimental MS data but not the molecular structure. The probability that such an agreement can happen by chance (significance) is below 10^{-167}. This is the probability that, by uniformly drawing a molecular formula at 50 ppm for each peak, we reach the observed number of 1043 matching molecular formulas or an even higher number. See Table A.6 in the appendix for detailed results.

Because Mass Frontier tends to annotate peaks of small mass, the number of candidate molecular formulas for a peak annotated by Mass Frontier is small. To further demonstrate the good agreement between the tools, we discarded all matched peaks with only one possible annotation, keeping 444 peaks with 3.9 explanations on average. For these peaks, we reach a match with Mass Frontier in 93.7% of the cases (significance as above). To assess this agreement, we compared Mass Frontier predictions against two other predictors: A random peak annotator that selects an arbitrary molecular formula within the mass accuracy, reaches only 35.6% agreement with Mass Frontier (significance 0.51). The naive approach, which always uses the molecular formula with the smallest mass difference to each peak, would reach 71.8% agreement (significance 10^{-61}). Clearly, agreement between Mass Frontier and our approach is much higher.

5 Alignment of Fragmentation Trees

This chapter presents a method the automated comparison of fragmentation trees (FTs) and demonstrates various applications of this method using four different datasets. One of these has been measured in a high-throughput setup for an untargeted metabolomics screen, underlining the applicability of the method in such an experimental setup. The comparison is based on tree alignment. Alignments have proven useful in many areas of bioinformatics due to their ability to accurately estimate the similarity between structured objects, the most prominent example being sequence alignments.

After presentation of the alignment algorithm and the introduction of the test datasets, we will introduce three applications of fragmentation tree alignment, see Figure 5.1 for an overview. The first approach is only applicable to reference datasets of known compounds and used to evaluate the method and its parameters. It correlates the similarity score of two fragmentation trees with the Tanimoto structural similarity score of the corresponding compounds. This results in a single correlation coefficient per dataset, which we use to easily assess the methods quality.

In the second application we use the similarity scores to cluster a set of known and/or unknown compounds. This can be helpful in two ways: On the one hand by grouping unknowns together with reference compounds, the compound class of an may be predicted, if it falls into a group of reference compounds of the same class. On the other hand, even if no reference data is available, a clustering of the unknown measurements gives a first overview over the dataset. An experienced experimenter may even spot the compounds of interest for his study from such a clustering.

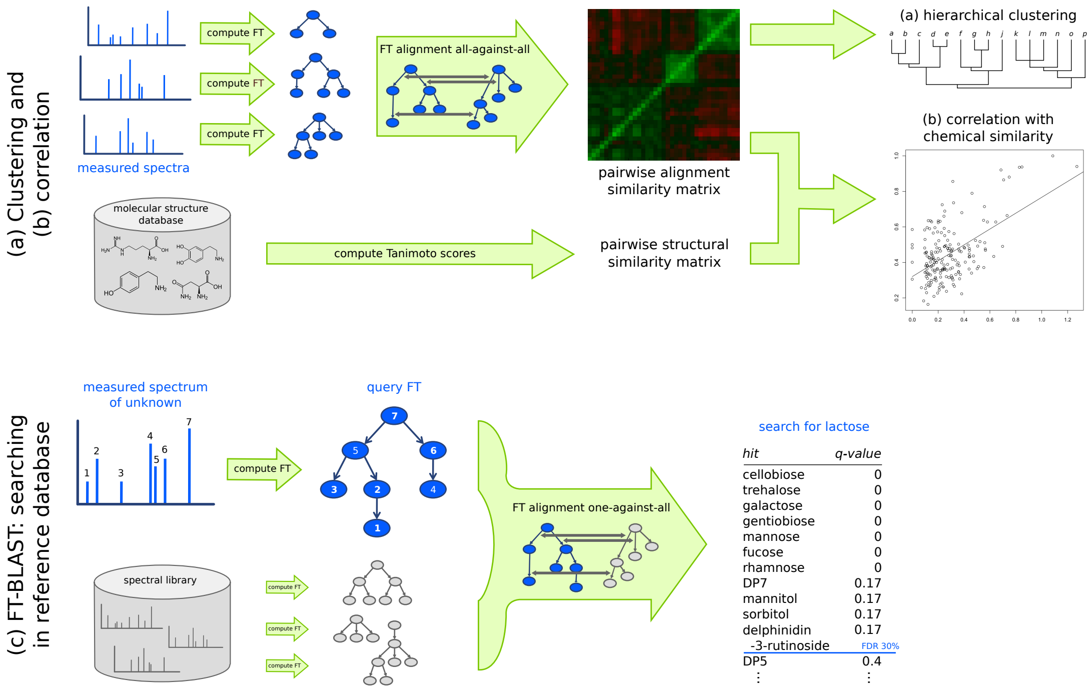
The third application is database searching. Here, we search an unknown compound against a database of reference trees. As this results in an output similar to BLAST for sequence alignment, we name this workflow fragmentation tree local alignment search tool, FT-BLAST for short. A major advantage of FT-BLAST over common spectral library searches is, that FT-BLAST allows for a significance estimation of its hits, using a decoy database strategy similar to Section 2.8.5.

5.1 Alignment Algorithm

For the automated comparison of fragmentation trees we use pairwise *local alignments*. To apply this concept it is necessary to define a similarity measure on the edges (losses) and nodes (fragments) of two fragmentation trees (Table 5.1). The similarity of two trees is then defined as the sum of the scores from all aligned edge pairs. Insertion and deletion are possible through the introduction of “gap” nodes and edges. Additionally, we allow two nodes to be joined and aligned against a single node of the other tree.

This allows for a fragmentation to be matched, although an intermediate peak has not been detected in one of the spectra. We search for subtrees of the original trees that maximize the similarity score, because the molecular structures of the compounds are not *identical* but subtree similarity indicates structural resemblance. Figure 5.2 shows such an local fragmentation tree alignment.

Tree alignments have been proposed in the context of RNA structure comparison and efficient algorithms have been developed to compute them [47]. In contrast to RNA trees, fragmentation trees are unordered, as there is no meaningful ordering of the losses of some fragments. Aligning unordered trees is computationally hard,



hit	*q-value*
cellobiose	0
trehalose	0
galactose	0
gentiobiose	0
mannose	0
fucose	0
rhamnose	0
DP7	0.17
mannitol	0.17
sorbitol	0.17
delphinidin -3-rutinoside	0.17
DP5	0.4
⋮	⋮

Figure 5.1: Workflows elaborated for the analysis of tandem MS data. Apart from choosing analysis parameters such as mass accuracy, no user interaction is required. Workflows (a) and (c) are targeted at compounds that are *not* in any database. (a) Clustering of known and unknown compounds using an all-against-all pairwise fragmentation tree alignment, followed by hierarchical clustering. (b) Correlating fragmentation tree alignment similarities and chemical similarities for a set of reference compounds. (c) Searching for an unknown compound in databases of reference compounds (either tandem mass spectra or fragmentation trees) using FT-BLAST. This method will return hits (similar compounds) even if the true compound is not in the database. Molecular structures are required only to compute chemical similarities (correlation analysis) or to annotate FT-BLAST hits.

namely MAX SNP-hard [47]. The following exact dynamic programming algorithm computes the alignment of unordered trees:

The goal is to compute the maximal score $S(T_1, T_2)$ of a local alignment between two trees T_1, T_2. Let $N(v)$ denote the children of any node v in T_1 or T_2. In the following, let u be a node of T_1, and v a node of T_2. Let $D[u, v]$ be the maximal score of a local alignment of two subtrees of T_1, T_2, where the subtree of T_1 is rooted in u, and the subtree of T_2 is rooted in v. For $A \subseteq N(u)$ and $B \subseteq N(v)$ we define $D_{u,v}[A, B]$ to be the score of an optimal local alignment with subtree rooted in u and v, respectively, such that *at most* the children A of u and B of v are used in the alignment. Note that all children A of u and B of v can be used, but also, any subset

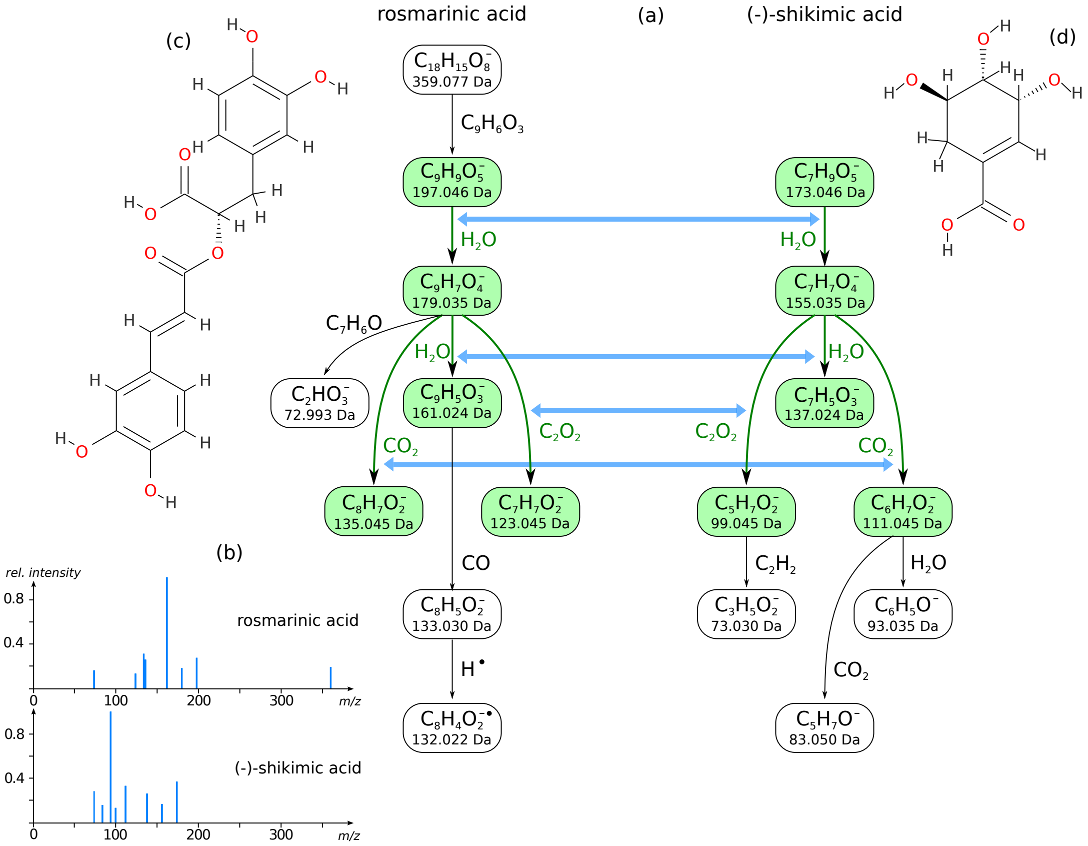


Figure 5.2: Optimal fragmentation tree alignment for rosmarinic acid (8 losses) and (-)-shikimic acid (7 losses) from the MassBank dataset (a). The FT fingerprint similarity (from -1 to $+1$) of the mass spectra is $+0.24$. (b) Fragmentation mass spectra of rosmarinic acid and (-)-shikimic acid used for computing fragmentation trees. The mass spectra do not share common peaks. Molecular structures of rosmarinic acid (c) and (-)-shikimic acid (d). PubChem Tanimoto score of the compounds is 0.50.

is allowed, including the empty set. Clearly, we have $D_{u,v}[A,\emptyset] = D_{u,v}[\emptyset, B] = 0$ for all A, B. Now, $D[u,v] = D_{u,v}[N(u), N(v)]$ holds.

We initialize $D_{u,v}[A,B] = 0$ for $A = \emptyset$ or $B = \emptyset$. In the recurrence, we distinguish three cases, namely *match* (including mismatches), *deletion*, or *insertion*, where the latter two are symmetric to each other. For non-empty sets $A \subseteq N(u)$ and $B \subseteq N(v)$ we get

$$\begin{aligned}
D_{u,v}[A,B] &= \max\Big\{0, \mathit{match}_{u,v}[A,B], \mathit{delete}_{u,v}[A,B], \mathit{insert}_{u,v}[A,B]\Big\}\\
\mathit{match}_{u,v}[A,B] &:= \max_{a\in A, b\in B}\Big\{D[a,b] + D_{u,v}\big[A-\{a\}, B-\{b\}\big] + \delta(ua,vb)\Big\}\\
\mathit{delete}_{u,v}[A,B] &:= \max_{a\in A, B'\subseteq B}\Big\{D_{a,v}\big[N(a),B'\big] + D_{u,v}[A-\{a\}, B-B'] + \delta(ua,\lambda)\Big\}\\
\mathit{insert}_{u,v}[A,B] &:= \max_{A'\subseteq A, b\in B}\Big\{D_{u,b}\big[A',N(b)\big] + D_{u,v}[A-A', B-\{b\}] + \delta(\lambda,vb)\Big\}
\end{aligned}$$

where $\delta(ua,vb)$ denotes the score of the losses attached to edges ua and vb, and $\delta(ua,\lambda), \delta(\lambda,vb)$ accordingly. Finally, we compute the maximal score of a local alignment of T_1, T_2 as

$$S(T_1,T_2) = \max_{u\in T_1, v\in T_2} D[u,v].$$

Merging two losses in T_1 or T_2 requires two additional symmetric cases, namely *join* and *disjoin* for merging in tree T_1 or T_2, respectively. Here, we define

$$\begin{aligned}
D_{u,v}[A,B] &= \max\Big\{0, \ldots, \mathit{insert}_{u,v}[A,B], \mathit{join}_{u,v}[A,B], \mathit{disjoin}_{u,v}[A,B]\Big\}\\
\mathit{join}_{u,v}[A,B] &:= \max_{\substack{a\in A\\ b\in B}} \max_{\tilde{a}\in N(a)}\Big\{D[\tilde{a},b] + D_{u,v}\big[A-\{a\}, B-\{b\}\big] + \delta(u\tilde{a},vb)\Big\} + \delta_{\text{merge}}\\
\mathit{disjoin}_{u,v}[A,B] &:= \max_{\substack{a\in A\\ b\in B}} \max_{\tilde{b}\in N(b)}\Big\{D[a,\tilde{b}] + D_{u,v}\big[A-\{a\}, B-\{b\}\big] + \delta(ua,v\tilde{b})\Big\} + \delta_{\text{merge}}
\end{aligned}$$

where $\delta(u\tilde{a},vb)$ denotes the score for the combined losses on the path from u to $\tilde{a}$ with the loss of edges vb, and $\delta(ua,v\tilde{b})$ analogously.

This allows for only one child node to be joined with its parent, all other children are discarded. As this is clearly not desirable, a new approach allowing a several child nodes to be merged with their parent has been developed [44], but results presented here are based on the above algorithm. First evaluations show that results improve only marginally using the new algorithm.

Note that we modify the recurrence by Jiang *et al.* [47] for solving the problem in three ways: First, we also consider edge similarities. Second, we computed local alignments for maximum subtree similarity by adding a “zero-case” to the recurrence, corresponding to the leaves of the subtree. Third, we score *join nodes* to account for the non-appearance of intermediate fragmentation steps, as described above.

Computational complexity is normally not an issue as the algorithm is efficient if the trees do not contain nodes with many outgoing edges. This is usually not the case in fragmentation trees, where fragments rarely have more than five daughter fragments.

	Event	Score
losses	Basic match score	+5
	Modification for each non-hydrogen atom	+1
	Basic mismatch score	-2
	Modification for each non-hydrogen atom	-0.5
fragments	Basic match score	+5
	Modification for each non-hydrogen atom	+1
	Basic mismatch score	-3
	Modification for each non-hydrogen atom	±0
	Insertion/deletion score	±0
	Merging losses modification	±0

Table 5.1: Scoring neutral losses and fragments.

To be able to evaluate the tree alignment concept, we implemented this algorithm in Java 1.6.

5.2 Scoring Fragmentation Tree Alignments

Since we base our fragmentation tree alignment on losses and fragments, we need a scoring function to evaluate pairs of losses, as well as pairs of fragments. In our scoring we distinguish three main cases for two losses nl_1 and nl_2. Those cases are a match $nl_1 = nl_2$, a mismatch $nl_1 \neq nl_2$, or an insertion/deletion (indel) where either $nl_1 = \lambda$ or $nl_2 = \lambda$ is a gap symbol. A summary of scores can be found in Supplementary Table 5.1. In detail, we define:

- For a *match*, we assign a positive score. This score depends on the size of the losses, since agreement between larger losses is more significant than between smaller ones. We set $\delta(nl, nl) := 5 + \#atoms$ where $\#atoms$ is the number of non-hydrogen atoms in the loss nl (that is, all carbon and hetero atoms).
- For a *mismatch* we assign a negative score, that increases when the losses get more dissimilar. We set $\delta(nl_1, nl_2) := -5 - \#diff$ where $\#diff$ is the number of non-hydrogen atoms in the symmetric difference between the two losses. As an example, $nl_1 = C_2H_3O_2$ and $nl_2 = C_4H_4O_1N_1$ differ in two carbon, one oxygen, and one nitrogen atoms, a total of four non-hydrogen atoms, so $\delta(C_2H_3O_2, C_4H_4O_1N_1) = -5 - 4 = -9$.
- For an *insertion/deletion* we set $\delta(nl_1, \lambda) = \delta(\lambda, nl_2) = 0$, as deleting nodes from the alignment implicitly reduces the score that can be reached.
- Finally, we will allow two subsequent losses to be *merged* in one of the tree. Here, we set $\delta_{\text{merge}} := \pm 0$. We do not penalize merged losses, as merging losses implicitly reduces the score that can be reached by the alignment.

Scoring of fragment pairs is somewhat similar. For two fragments f_1 and f_2 we again distinguish between match $f_1 = f_2$ and mismatch $f_1 \neq f_2$. To correctly compare trees measured in negative and positive mode, we "neutralize" the fragment ion formulas by adding or subtracting a hydrogen atom.

- For a *match*, we assign a positive score depending on the size of the fragment. We set $\delta(f, f) := 5 + \#atoms$ where $\#atoms$ is the number of non-hydrogen atoms in the fragment f (that is, all carbon and hetero atoms).
- For a *mismatch* we assign a negative score not depending on the symmetric difference between the two fragments. We set $\delta(f_1, f_2) := -3$ for $f_1 \neq f_2$. In this way, we allow for matching losses even when the corresponding fragments show no similarity.

If compounds are isotopically labeled, we treat the labeled element as identical to the unlabeled. As an example, losses H_2O and HDO would receive a score of +6.

5.3 Test Data and Pre-computations

To evaluate the tree alignment we analyzed spectra from three reference datasets (5.2). The *Orbitrap* dataset contains 97 compounds, measured on a Thermo Scientific Orbitrap XL instrument. The *MassBank* dataset [43] consists of 370 compounds measured on a Waters Q-Tof Premier spectrometer. The *QSTAR* dataset with its 44 compounds is the same as in the previous chapter [77]. The masses of all compounds ranged from 75 Da to 1258 Da.

For the Orbitrap dataset, 26 spectra analyzed in the previous chapter have been reused. Additionally, several zeatins, amino acids, glucosinolates, and sugars have been newly measured to yield 97 compounds in total. Table B.1 in the appendix lists all compounds of the dataset. For experimental details on the new data, see [76]. 41 compounds (zeatins, sugars, lipids, bicuculline) were measured at a single collision energy. Some of these compounds show rather few fragments.

The MassBank dataset was downloaded from the MassBank database [43] at `http://www.massbank.jp/`, accession numbers PR100001 to PR101056. These spectra were measured on a Waters Q-Tof Premier instrument at the RIKEN Plant Science Center (Yokohama, Japan) by F. Matsuda, M. Suzuki, and Y. Sawada. We discarded 47 compounds where the measurement of the *unfragmented* molecule mass deviated more than 10 ppm from the theoretical mass, leaving us with 370 compounds. By visual inspection of mass spectra and fragmentation trees, we decided to use an accuracy of 50 ppm. Table B.2 in the appendix lists all compounds of this dataset.

5.3.1 Biological Data from Icelandic Poppy

The fourth dataset stems from a biological sample that has been extracted from different organs of the plant *Papaver nudicaule*. Measurements were performed in an untargeted mode, that is, the instrument selected intense peaks of the survey scan

Name	Orbitrap	MassBank	QSTAR
Mass accuracy (ppm)	< 5	≈ 50	20
collision energy (eV)	between 5 and 150[a]	ramp 5–60, 30[b]	15,25,45,55,90[a]
Number of compounds	97	370	44
Mass range (Dalton)	75.0 – 1257.4	90.0 – 822.4	89.0 – 450.2
Median / average mass	342.1 / 346.2	230.0 / 298.0	174.6 / 212.1
FTs with 1+ losses[c]	93	343	44
FTs with 3+ losses	77	242	43
FTs with 5+ losses	65	157	32
FTs with 7+ losses	51	103	28
Major compound classes	zeatins (24), amino acids (19), glucosinolates (14), sugars (12), benzopyrans (11)	flavonoids (85), carboxylic acids (76), amino acids (73), nucleotides (65), sugars (22)	amino acids (21), cholines (18), amines (4)
Compound details	Table B.1	Table B.2	Table B.3

Table 5.2: Datasets used in this study. The MassBank dataset consists of ramp spectra; the other datasets were measured at discrete collision energies. 26 compounds of the Orbitrap dataset were fragmented using higher-energy collisional dissociation (HCD). For these compounds we used fragmentation energies between 5 and 95 arbitrary units. [a]Between 1 and 20 different collision energies. [b]Some compounds were also measured at 30 eV discrete collision energy. [c]Number of fragmentation trees (FT) with x or more losses

for fragmentation. For details on the experimental setup, see again [76]. The data contained 489 non-empty fragmentation spectra of 89 potential compounds.

Unfortunately, the isotopic pattern recorded during the measurement were not of the required quality. Thus, we apply the following procedure to ensure that the correct molecular formula is chosen: We applied the method to determine the molecular formula as described in Section 3.8. Then, we checked that the highest scoring molecular formula of the combined analysis was among the TOP 5 formulas of the isotope analysis, as well as among the TOP 5 of the fragmentation pattern analysis alone. This was the case for 29 compounds, which formed our poppy dataset of unknowns.

It must be understood that even in cases were we cannot unambiguously determine the molecular formula from the data, it is possible to use the fragmentation tree alignment setup described in this paper: In case of doubt about the molecular formula of an unknown, we can use the trees of several molecular formula hypotheses as queries or clustering input.

5.3.2 Calculation of Fragmentation Trees

From these data, we calculated fragmentation trees as described in Chapter 3. The ILP was used to generate optimal trees. Different from Table 3.1 hydrogen (H_2) was no longer considered common. Instead, methanol (CH_4O) was added to the common losses. The punishment of implausible losses, as well as the calculation of radical losses as described in Section 3.3 is enabled.

Some compounds did not fragment significantly, resulting in hypothetical fragmentation trees with an insufficient number of losses. Especially amino acids and carboxylic acids have mostly less than three losses. This is due to current instruments limited mass range at 50 Thomson, too high for small amino acids like glycine and alanine.

5.4 Normalization of Scores and Fingerprinting

After an alignment score of two trees has been calculated using the algorithm and scoring described above, alignment scores have to be normalized, since the scores are highly dependent on the size of the trees: Large trees may receive higher scores simply because they possess more edges that can be matched. Therefore, we normalize by the score that a *perfect match* would obtain. Since we do local alignments, a perfect match means that the one tree is a subtree of the other one. The same score is obtained by aligning this subtree with itself, $S(T_i, T_i)$. So, we normalize the score by

$$S_0(T_1, T_2) = \frac{S(T_1, T_2)}{\big(\min\big\{S(T_1, T_1), S(T_2, T_2)\big\}\big)^c} \tag{5.1}$$

where $c \in [0, 1]$ is the normalization parameter. Here, $c = 1$ corresponds to a full normalization by the perfect match score, whereas $c = \frac{1}{2}$ corresponds to the square root of this value. We do not to choose the full score for normalization, since it is much more likely for a very small tree to be a subtree of another tree, than it is that a medium-size or large tree is a subtree of another tree. To this end, $c = 1$ favors small trees and discriminates against large trees, whereas no normalization ($c = 0$) favors large trees. In our study, we choose $c = \frac{1}{2}$.

Instead of directly using normalized scores, we found that an additional re-evaluation of similarities is useful: When two compounds are structurally similar, they should show comparable fragmentation tree similarities to *any* other compound. To this end, we use the scores of one compound against all others as its *fingerprint* or *feature vector*. We compare two compounds by comparing their fingerprints using the Pearson product-moment correlation coefficient r (*Pearson correlation coefficient* for short) that measures the linear dependence of two variables $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_n)$:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{5.2}$$

with $-1 \leq r \leq +1$. Here, $\bar{X}$ denotes the mean of $X_1, \ldots, X_n$. We refer to the resulting score as *FT fingerprint similarity*

5.5 Clustering

To cluster compounds based on their fragmentation trees, we compute pairwise alignments of fragmentation trees for all compound pairs, as explained in Section 5.1. We normalize the alignment scores and compute fingerprints. This results in a matrix of pairwise similarities. To this matrix, we apply hierarchical clustering or, more precisely, UPGMA (Unweighted Pair Group Method with Arithmetic Mean) agglomerative clustering [92].

There might be better methods to cluster compounds based on fragmentation tree similarity. We have chosen hierarchical clustering as it is well-known, particularly in the context of analyzing gene expression data [26], and its outcome is easily susceptible.

It is understood that for fragmentation trees with few losses, clustering results will become somewhat arbitrary: In the extreme case of a single neutral loss, similarity or dissimilarity to any other fragmentation tree can easily be spurious. Thus, we limit clustering to fragmentation trees with three ore more losses. See Table 5.2 for the number of compounds included in the clustering. This is probably not a shortcoming of the method, but rather the problem that certain compounds do not “fragment sufficiently” under tandem MS, resulting in mostly uninformative fragmentation spectra. This problem may possibly be overcome by using multiple MS.

We first analyze the Orbitrap dataset. We discarded 20 compounds as the resulting fragmentation trees showed less than three losses. The resulting clustering is depicted in Figure 5.3. We observe that clusters are very homogeneous: There is a perfect glucosinolate cluster containing all 14 glucosinolates, a perfect zeatin cluster containing all 21 zeatins, and an almost perfect sugar cluster containing all nine sugars, plus one anthocyanin and one carboxylic acid. Furthermore, there is an almost perfect amino acid clusters containing seven of the nine amino acids plus one alkaloid. Similarly, there is a perfect benzopyran cluster containing six of the eleven benzopyrans.

For the MassBank dataset, we had to discard 128 compounds with less than three losses. Here, we find a large group of flavonoids (81 with 3+ losses), nucleotides (54), amino acids (33), carboxylic acids (26), and sugars (17). Clustering with collapsed mostly-homogeneous clusters is depicted in Supplementary Figure 5.4. We observe an almost perfect cluster of 64 flavonoids containing only two non-flavonoid compounds. For amino acids we find five perfect clusters containing 22 of the 33 amino acids in total. Similarly, we find four carboxylic acid clusters containing ten carboxylic acids plus one other compound. For nucleotides there are seven small perfect clusters, containing 32 nucleotides in total, and a large cluster containing 16 nucleotides but also four sugars and two sugar alcohols.

The QSTAR dataset contains biogenic amino acids and complex choline derivatives [11]. We observe a well partitioning of the compounds into amino acids, amines and cholines, see Figure 5.5 for the hierarchical clustering.

To show the applicability of our method between measurements from different instruments, we performed a combined dataset clustering: We cluster all compounds from the Orbitrap, MassBank and QSTAR datasets for fragmentation trees with 5+ losses, leaving us with 157 compounds from the MassBank dataset, 65 compounds

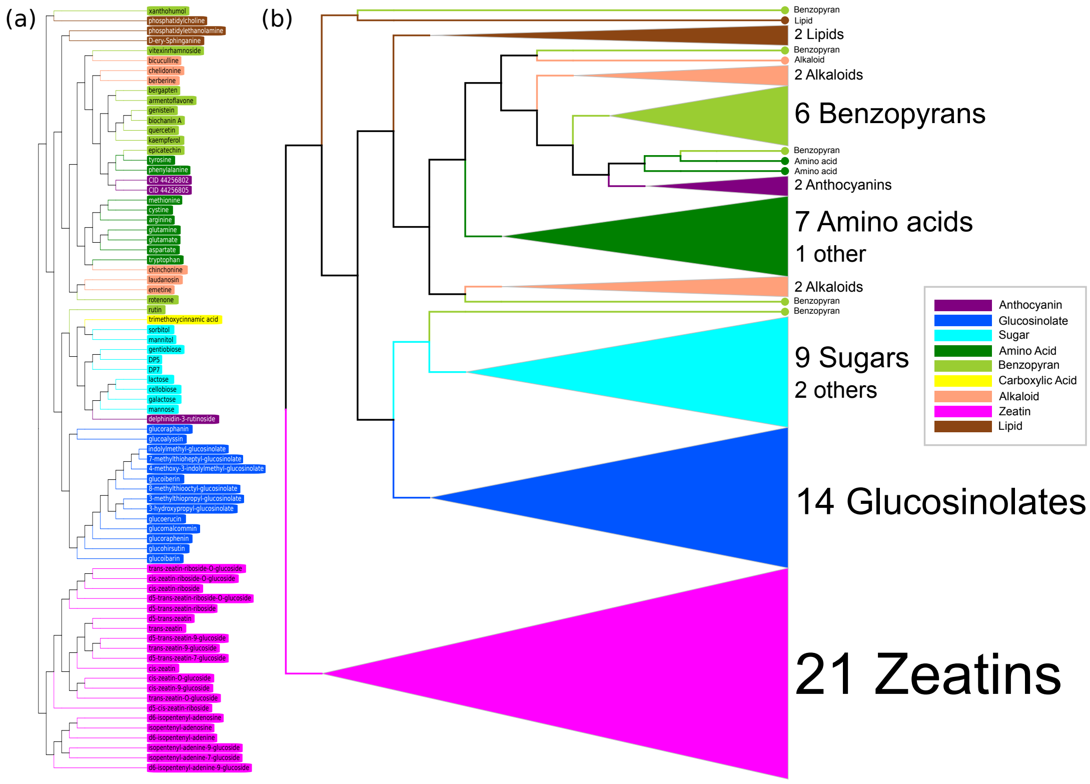


Figure 5.3: (a) Hierarchical clustering of the Orbitrap dataset (compounds with 3+ losses) (b) The same clustering, where (mostly) homogeneous cluster have been collapsed. homogeneous clusters.

from the Orbitrap dataset, and 32 compounds from the QSTAR dataset. We report results in Figure 5.6. We observe a large amino acid cluster containing three amino acids from the MassBank, three amino acids from the Orbitrap and 17 amino acids from the QSTAR dataset. Furthermore, eight sugars from MassBank and eight sugars from Orbitrap form a large cluster with six sugar alcohols and five carboxylic acids from MassBank. The only remaining glucosinolate from MassBank forms a perfect cluster with the 13 remaining glucosinolates from Orbitrap. Finally, an almost perfect cluster of 27 nucleotides from MassBank forms a subcluster of the almost perfect zeatin cluster, containing 15 zeatins from Orbitrap and four nucleotides from MassBank. This demonstrates that the structures of the fragmentation trees are highly similar although the fundamental differences between Q-Tof and Orbitrap mass analyzers.

5.5.1 Clustering of the Poppy Dataset

To determine compound classes of the unknown in the poppy dataset, we performed an all-against-all alignment using the poppy fragmentation trees plus the Orbitrap treess.

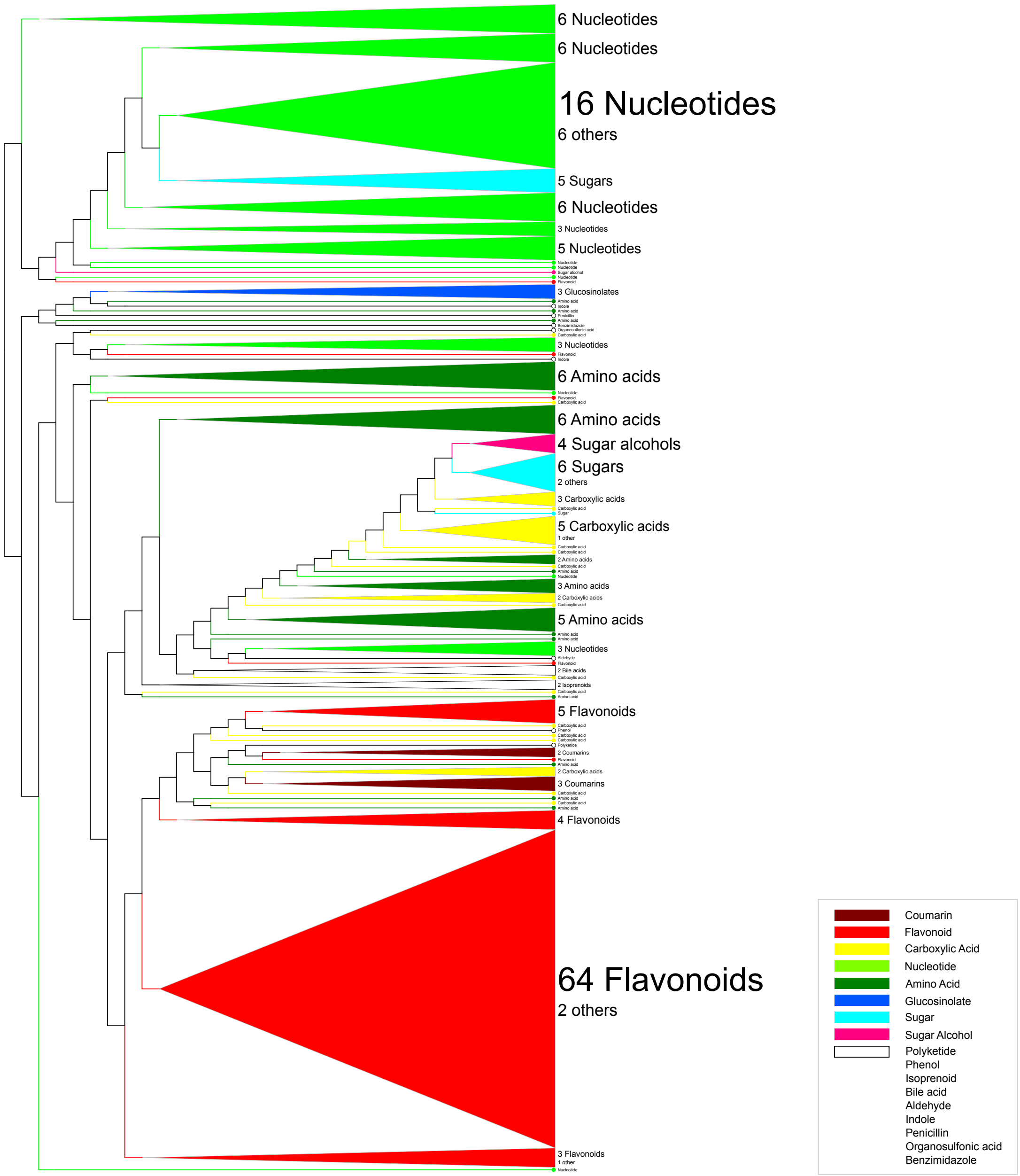


Figure 5.4: Hierarchical clustering of the MassBank dataset (compounds with 3+ losses) where (mostly) homogeneous clusters have been collapsed.

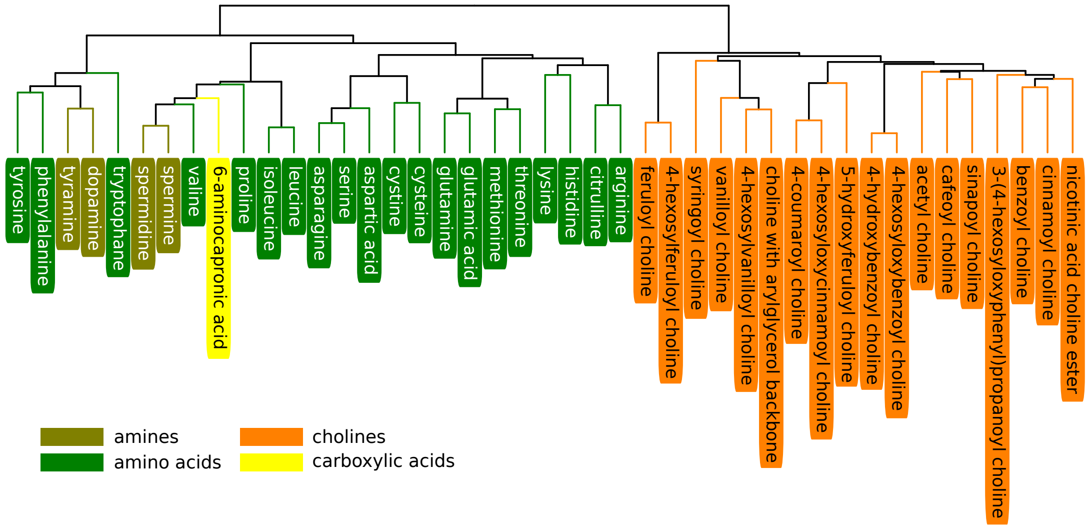


Figure 5.5: Hierarchical clustering of the QStar dataset (compounds with 3+ losses).

Scores were normalized and fingerprint similarities were calculated as described in Section 5.4.

We cluster the unknowns together with the reference measurements from Orbitrap. We used all fragmentation trees with at least one loss to include as many reference compounds as possible. Figure 5.7 shows the clustering of the unknown compounds from poppy together with the Orbitrap reference dataset. Mass spectrometry experts identified eight compounds in the sample by manual analysis of the spectra. All manually identified unknowns are grouped into their respective cluster. On top of the figure one can see the alkaloid cluster with four reference alkaloids and the four manually identified “unknowns”. The 400 Da compound probably is also an alkaloid. Since it is located at the border of the cluster, more reference alkaloids are required for a reliable classification. Since the unknown at 229 Da falls into the amino acid cluster, we consider it at least strongly related with amino acids. The 277 Da molecule is probably a sugar, or contains a sugar moiety. With the limited reference data, it is not possible to assign a group to the 438 and 537 Da compounds, but we may assume that they are neither related to zeatins nor to glucosinolates, as no unknown falls into these well-separated clusters. Manual interpretation also failed to identify the compounds, NMR analysis is currently being performed. Additionally, our analysis correctly shows that a contamination with mass 338 Da, measured during a blank column run, is similar to the lipids. Database search and manual validation identified it as erucamide (PubChem CID 5365371), an additive originating from the plastic ware used for sample collection.

Results from the FT-BLAST and clustering analysis should be seen as strong hints towards a compound class. This can point towards unknowns of interest and simplify a downstream analysis, e.g. using NMR.

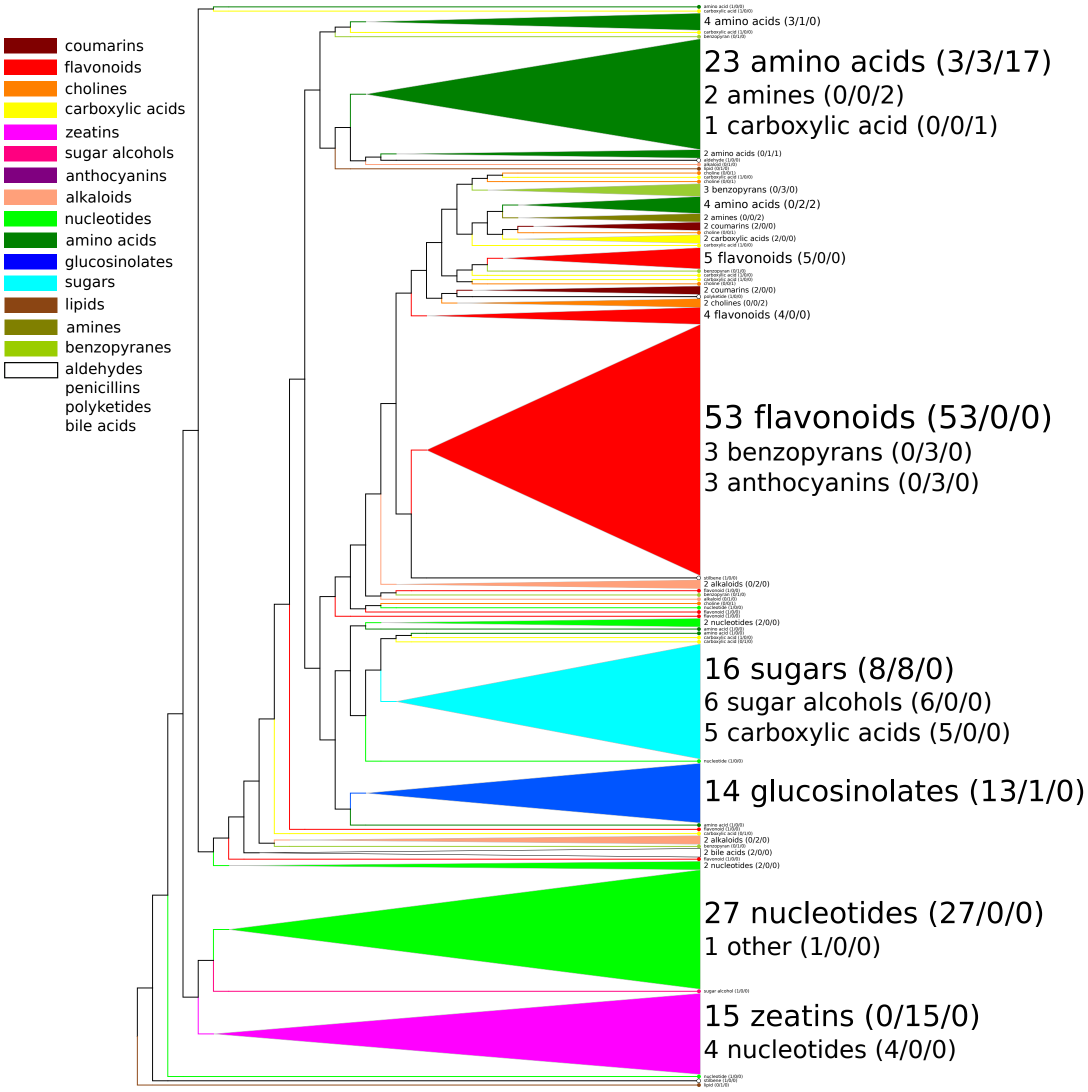


Figure 5.6: Combined dataset clustering, fragmentation trees with 5+ losses, $N = 254$. For better visualization, we have collapsed mostly homogeneous clusters. Number of compounds from different datasets are given as “(MassBank/Orbitrap/QSTAR)”. Compounds of the same or similar classes but from different datasets, such as amino acids or sugars, cluster together. A nucleotide cluster (from MassBank) forms a subcluster of the zeatin cluster (from Orbitrap).

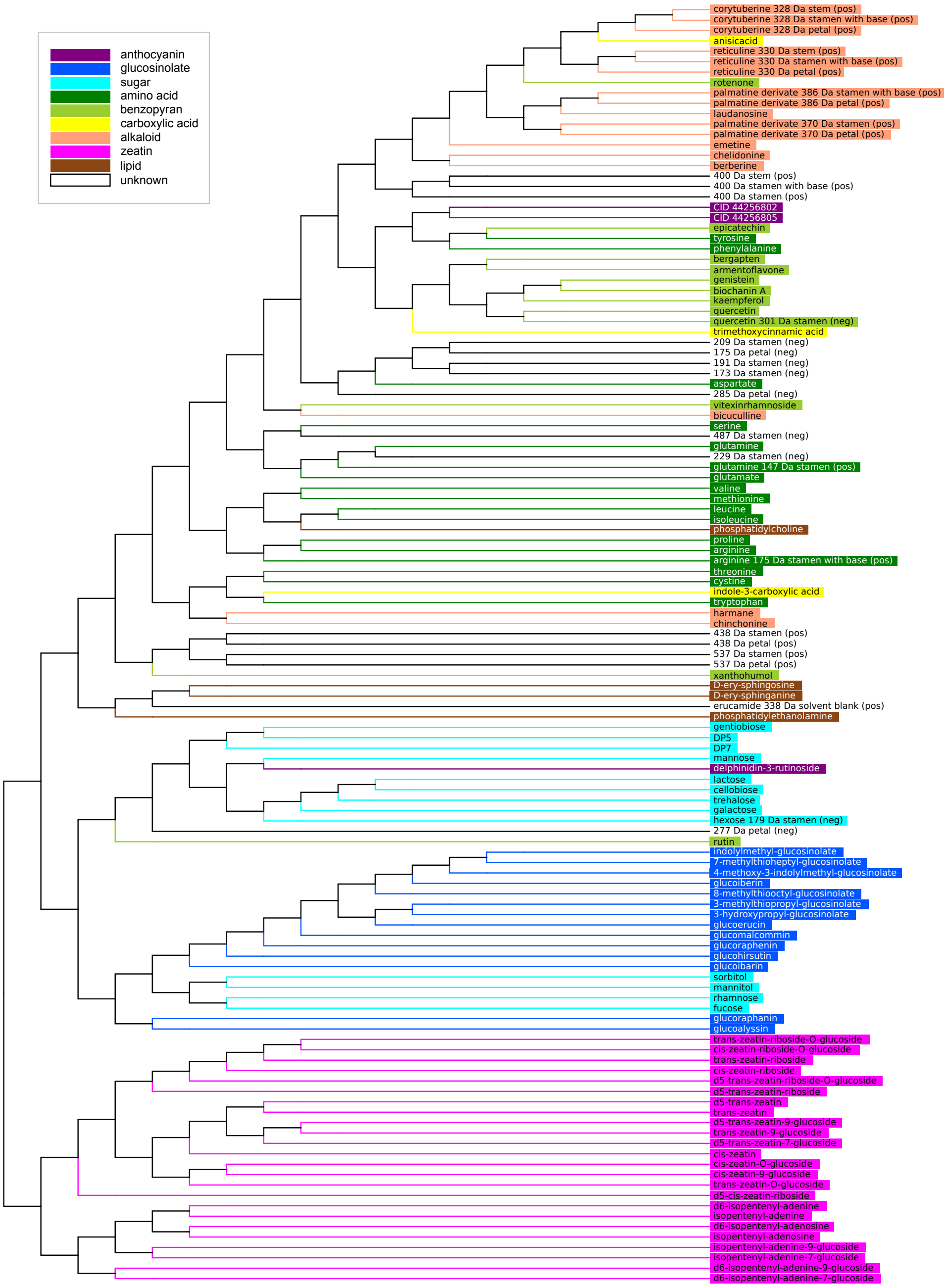


Figure 5.7: Clustering of the poppy and the Orbitrap datasets, fragmentation trees with 1+ losses. Colored compounds are known references. Many unknown compounds form a cluster together with several alkaloids (top of the figure). Other unknowns end up in amino acid or sugar clusters. The poppy sample most likely contained no glucosinolates and zeatins, as no unknowns can be found among these clusters.

5.6 Correlation with Chemical Similarity

As all of the compounds in our datasets are references with known molecular structure, we can estimate their structural similarity, termed *chemical similarity* in the following. This allows us to compare chemical similarities with our fragmentation tree alignment-based similarities. This is meant as a proof-of-concept: In applications, we obviously do *not* know the molecular structure of the unknown query compound. But our results clearly show the correlation between these similarity values.

For measuring correlation, we Pearson correlation coefficient r, that measures the linear dependence of two variables, see Equation (5.2) in Section 5.4. We also compute the *Spearman correlation coefficient* ρ that is the Pearson correlation coefficient of the ranked variables. The values X_i, Y_i are each converted to ranks $x_i, y_i \in \{1, \dots, n\}$, and

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n}\left(x_i - \frac{n+1}{2}\right)\left(y_i - \frac{n+1}{2}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \frac{n+1}{2}\right)^2 \sum_{i=1}^{n}\left(y_i - \frac{n+1}{2}\right)^2}} \qquad (5.3)$$

where again, $-1 \leq \rho \leq +1$. Ties can be broken by assigning fractional ranks. Computations of correlation coefficients were carried out using the program language R.

To judge the level of correlation between the two similarities, we stress that these are not two measurements where, say, by the laws of physics, one expects a linear dependence. This being said, we argue that any Pearson correlation coefficients $r > 0.5$ ($r^2 > 0.25$) can be regarded as strong correlation. This is even more so since two different chemical similarity scores based on comparing molecular (sub-)structures, namely PubChem/Tanimoto and another Tanimoto score that uses Molecular ACCess System (MACCS) fingerprints [30], show a Pearson correlation of less than $r = +0.82$. This may be seen as the upper bound that any correlation between spectra and molecular structures can reach. Similarly, a Spearman correlation coefficient of $\rho > 0.5$ ($\rho^2 > 0.25$) indicates a strong but possibly non-linear correlation.

Again, we normalize fragmentation tree alignment scores by perfect match score using $c = \frac{1}{2}$ in (5.1), and compute fingerprints of the compounds as described in Section 5.4. To show the effect of the fragmentation tree size on the correlation with chemical similarity, we differentiate between those compounds with fragmentation trees that have at least 1+, 3+, 5+, and 7+ losses, respectively. See Table 5.2 for the number of compounds remaining in the different datasets. For a dataset with n compounds, this results in $\binom{n}{2} = \frac{n(n-1)}{2}$ compound pairs where we can correlate the two similarity values. We stress that we do not measure the similarity of a compound against itself: Any method for comparing fragmentation patterns should be able to pick up the similarity of two *identical* patterns. Including such self-comparisons would result in even higher correlation coefficients.

Many different similarity scores have been developed in chemoinformatics to compare molecular structures [60]. We concentrate on one of the most commonly used frameworks [3], namely binary fingerprint representations with Tanimoto similarity

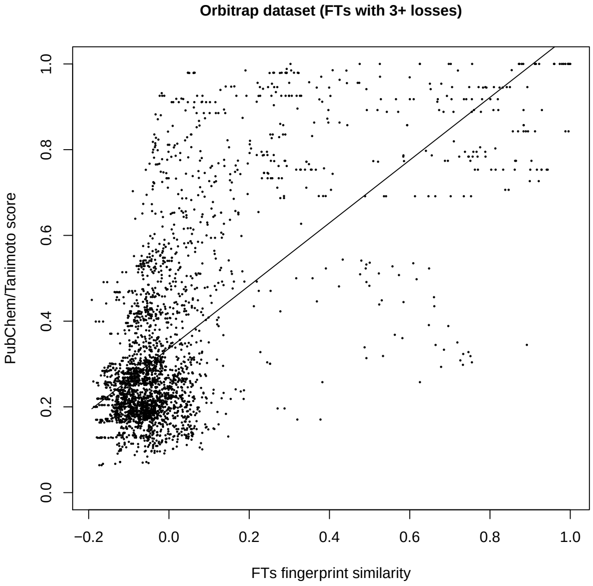
		only compounds with			
Dataset	correlation method	1+ losses	3+ losses	5+ losses	7+ losses
Orbitrap	Pearson r	0.65	0.67	0.64	0.58
	Pearson r^2	0.42	0.45	0.41	0.34
	Spearman ρ	0.45	0.47	0.48	0.51
	Spearman ρ^2	0.20	0.22	0.23	0.26
	no. alignments N	4278	2926	2080	1275
MassBank	Pearson r	0.50	0.60	0.67	0.68
	Pearson r^2	0.25	0.36	0.45	0.46
	Spearman ρ	0.43	0.52	0.64	0.71
	Spearman ρ^2	0.18	0.27	0.41	0.50
	no. alignments N	58653	29161	12246	5253
QSTAR	Pearson r	0.63	0.62	0.55	0.51
	Pearson r^2	0.40	0.38	0.30	0.26
	Spearman ρ	0.64	0.64	0.61	0.55
	Spearman ρ^2	0.41	0.41	0.37	0.30
	no. alignments N	946	903	496	378
Between-dataset	Pearson r	0.49	0.52	0.55	0.58
	Pearson r^2	0.24	0.27	0.30	0.34
	Spearman ρ	0.37	0.40	0.38	0.43
	Spearman ρ^2	0.14	0.16	0.14	0.18
	no. alignments N	51083	32351	17309	9565

Table 5.3: Correlation of chemical similarity (PubChem/Tanimoto) with fragmentation tree similarity, for all datasets and different restrictions on the number of losses. For the between-dataset correlation, only compound pairs from different datasets are considered. We also report the number of alignments (compound pairs) N for every set.

scores (Jaccard indices) [79]. We use the fingerprints of the PubChem database [100] as implemented in the Chemistry Development Toolkit version 1.3.37 [96].

5.6.1 Results of the Correlation Analysis

The results of the correlation analysis are listed in Table 5.3. All three datasets show a good correlation ($r \geq 0.50$). We reach the best correlation ($r = +0.65$) for the Orbitrap dataset that contains many compound classes (Figure 5.8, trees with 3+ losses). For the QSTAR dataset comprised of only two major compound classes we still reach a very strong Pearson correlation of $r = +0.63$. But even for the MassBank dataset with mass accuracy much worse than 10 ppm there is a good correlation, which increases to very strong Spearman correlation $\rho = +0.71$ for fragmentation trees with 7+ neutral losses.



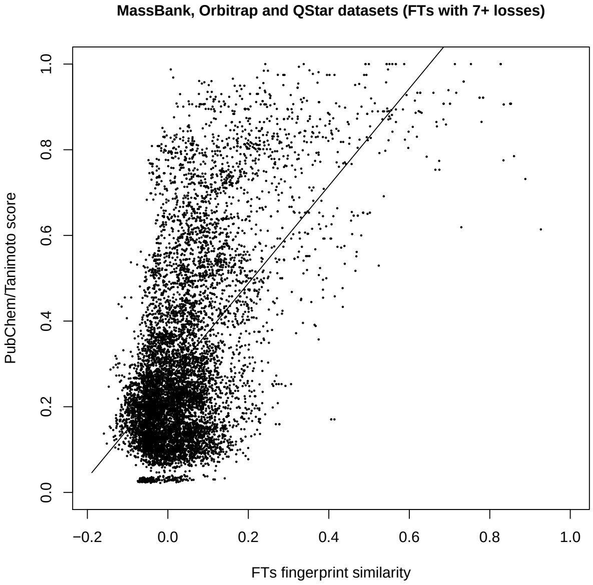


Figure 5.8: Correlation and regression line: FT fingerprint similarity (x-axis) plotted against chemical similarity measured by PubChem/Tanimoto score (y-axis). Top: Orbitrap dataset, fragmentation trees with 3+ losses, $N = 2926$. Pearson correlation is $r = +0.67$ ($r^2 = 0.45$) and Spearman correlation is $\rho = +0.47$ ($\rho^2 = 0.22$). Bottom: between-datasets analysis, each compound from one dataset is compared to all compounds from the other two datasets. Only fragmentation trees with 7+ losses are considered, $N = 9565$. Pearson correlation is $r = +0.58$ ($r^2 = 0.34$) and Spearman correlation is $\rho = +0.43$ ($\rho^2 = 0.18$).

The correlation coefficients of the MassBank dataset increase by limiting the correlation analysis to fragmentation trees with more neutral losses. This may appear evident, since correlation with chemical similarity requires that information is present in the fragmentation trees. Nevertheless, the correlation coefficients of the QSTAR and the Orbitrap datasets decrease when limiting the analysis to bigger trees. Interestingly, also the correlation between the MACCS/Tanimoto scores and the PubChem/Tanimoto scores of these two datasets decreases from $r = +0.79$ to $r = +0.74$ for the QSTAR dataset, respectively from $r = +0.81$ to $r = +0.74$ for the Orbitrap dataset. We believe that the weaker correlation of fragmentation trees with more losses is an artifact of our data. Some compound classes fragment better than others, and limiting the compounds to bigger fragmentation trees implies limiting the compound subsets to less compound classes. For example, in the QSTAR dataset 13 of the 16 fragmentation trees with less than seven losses are cholines. Thus, the reduced subset consists of 64% amino acids. Possibly, a strong correlation within only one or few compound classes is more difficult, since fragmentation trees of one compound class are very similar and not sensitive enough to predict small differences between the structures.

5.6.2 Correlation between Datasets

To demonstrate that the strong correlation coefficients are not artifacts (measuring all compounds with one instrument and by one person), we performed a between-datasets analysis: Each compound from each dataset (Orbitrap, MassBank, QSTAR) is compared to each compound from the other two datasets. This is done to separate the intra-dataset correlation from the inter-dataset correlation. We reach Pearson correlation $r = +0.49$ ($r^2 = +0.24$) for the between-datasets analysis, and $r = +0.58$ ($r^2 = +0.34$) for fragmentation trees with 7+ losses (Figure 5.8). The results in Table 5.3 indicate that the method is robust against differences in sample preparation, instruments, and raw data processing methods. This may allow us to search for compounds in "mixed" databases where we do not limit the search to reference compounds measured under similar conditions as the query compound, see the next section. In this way, we may considerably enlarge the set of reference compounds for identifying the unknown.

5.6.3 Peak Counting Score

In this section, we would like to check whether the correlation is really attributed to the fragmentation tree alignment, or whether it could also be reached by spectral comparison.

For this, we tested different variants of the shared peak counting score. First, beside counting only similar peaks, also similar *parent losses* (mass differences to the parent peak) were counted. We tried various combinations of scoring peak masses and/or loss masses. Second, also considered the mass differences between two peaks, where two peaks with a lower mass difference receive a higher score. We tested a log likelihood-based scoring, based on the observation that mass differences in a well-calibrated mass spectrum are normally-distributed [9,111]. Third, we include the intensities and masses of the matching peaks by scoring two matching peaks with $peakmass^3\sqrt{peakintensity}$ as suggested by Stein and Scott [55, 95]. The second and the third attempt did not improve the correlation with the chemical similarity score. It turned out that the unweighted peak counting score gave best correlations, so we report its results here. In the end, we normalized the shared peak counting score similar to the normalization of the fragmentation tree alignment score by perfect match score using $c = 1.0$, and compute the fingerprints of the compounds as described in Section 5.4. Again, this normalization produced best results.

We find that the correlation of the peak counting scores with chemical similarity (Tanimoto/PubChem) is — in all cases but two — weaker than for the tree alignment scores. Noteworthy is the large increase in Pearson correlation when analyzing the between-dataset: Whereas the peak counting score reaches a Pearson correlation coefficient of only $r = +0.38$ ($r^2 = 0.14$), Pearson correlation for the tree alignment fingerprint score is $r = +0.49$ ($r^2 = 0.24$).

5.7 Fragmentation Tree Basic Local Alignment Search Tool

The classic way of analyzing tandem MS data is database searching. fragmentation tree alignments can also be used for this task. Given the tandem MS spectra of an unknown compound, we computed its fragmentation tree, then aligned it to all fragmentation trees in our target database, and ranked hits according to fingerprint similarity. Target fragmentation trees are constructed from tandem MS data, possibly on the fly. Searching for a “known” compound in a target database is a task that has already been thoroughly studied. We concentrated on the much more intriguing case of where we could not find the query compound in the target database.

An important point is to differentiate between true and spurious hits. Obviously, one of the fragmentation trees has maximal similarity among all trees in the database, but this does not mean that this best hit is a good hit. We employ a *decoy database strategy*, see Section 2.8.5.

5.7.1 Calculation of Decoy Fragmentation Trees

To assess the significance of hits, we generated a decoy database: For each tree in the target database, a tree in the decoy database is constructed. For a target tree with m edges, we randomly generate a decoy tree with m edges. Unfortunately, we have no statistical model of the structure of fragmentation trees; at the same time, we believe that the topology of fragmentation trees is extremely important for the alignment. To this end, we chose to generate *decoy fragmentation trees* from an independent dataset. We computed trees for the fragmentation data from 102 compounds measured on a Micromass QTOF, published by Hill *et al.* [41]. Using compounds from an independent dataset has two advantages: On the one hand, these are true fragmentation trees, so decoy trees are structurally "similar" to the true trees. On the other hand, this is an independent dataset, so any similarity to true fragmentation trees must be fully at random. Using the Hill *et al.* dataset [41] has the additional advantage that resulting trees are large, allowing us to compute subtrees more easily: To generate a random tree with m losses, we first discard all decoy trees with less than m edges. From the remaining, we randomly select one tree, where larger trees are chosen with higher probability: A tree with m' edges is chosen with weight $m' - m + 1$. Starting with a random edge, we build a subtree from this tree by randomly adding incident edges to the subtree, until the subtree has size m edges. The root of the decoy tree is assigned the same molecular formula as the root of the target tree. We then label the edges and remaining nodes of the decoy tree: We randomly choose a loss from the target database, respecting multiplicities. So, whereas the structure of the tree and the succession of losses is random, the losses of a decoy fragmentation tree have the same "occurrence pattern" as those in the target database. The label for the target node of this edge is defined by subtracting the chosen loss from the label of its source node. In case the resulting molecular formula is invalid (the loss is not a sub-formula of the source node molecular formula), a new loss is selected. If no loss that would result in a valid formula exists, the whole tree is discarded, and the tree generation is restarted from scratch.

5.7.2 False Discovery Rate Calculation

From this construction, we may assume that spurious hits in the target database and hits in the decoy database are equally likely: The decoy trees are similar to true fragmentation trees with respect to size, tree topology, losses, and molecular formula of the parent compound. We also assume that hits in the decoy database are never "true" hits: It is extremely unlikely to construct a tree which, by chance, is also part of the target database.

We align our sample tree to every tree in the combined database, containing both target and decoy trees, and sort the results with respect to score (fingerprint similarity). We report hits from the true database only. Assume we are given a *False Discovery Rate* (FDR) threshold t, such as $t = 30\,\%$. If the TOP $n_\mathrm{T} + n_\mathrm{D}$ in the combined search contains n_T hits from the target database and n_D hits from the decoy database, then we calculate a FDR of $n_\mathrm{D}/n_\mathrm{T}$ for this list. We search for the largest set of top hits with FDR $n_\mathrm{D}/n_\mathrm{T} \leq t$. For each hit, we compute the *q-value* as the smallest FDR under which this hit is reported in the output.

5.7.3 Leave-one-out Evaluation

We want to evaluate our method for those cases where the compound is *not* found in the database. To this end, we pursue a *leave-one-out* evaluation: For each compound, we deliberately delete the corresponding fragmentation tree from the database before searching for it. We then compute an alignment score against all remaining compounds (both targets and decoys) in our dataset. As usual, these values are normalized by perfect match score with exponent 0.5 and used as fingerprints. Pearson correlation between the fingerprints is calculated and used as final fingerprint similarity score. We sort compounds with respect to fingerprint similarity, and estimate the FDR as described above.

In Table 5.4 we report search results for the Orbitrap dataset (compounds with 1+ losses, $N = 93$) with FDR threshold $t = 30\%$. In the table, we report the following compound classes as being similar: Since anthocyanins are made up of sugars and benzopyrans, they are regarded as being similar to both classes; as glucosinolates contain a sugar moiety, these classes are also regarded as being similar. One can see that the search results of glucosinolates, sugars and zeatins contain almost exclusively compounds of the respective group. Some benzopyrans receive several hits from their own and similar groups, whereas for other benzopyrans, no hits are found. Possibly, the corresponding spectra are of lower quality, or the chemical similarity to other benzopyrans is weak. Only few hits were found for the alkaloids. We attribute this to the fact that we have relatively few reference compounds available for the diverse class of alkaloids. We find almost no hits for amino acids, carboxylic acids, and lipids. Here, fragmentation trees were often too small to identify any hits.

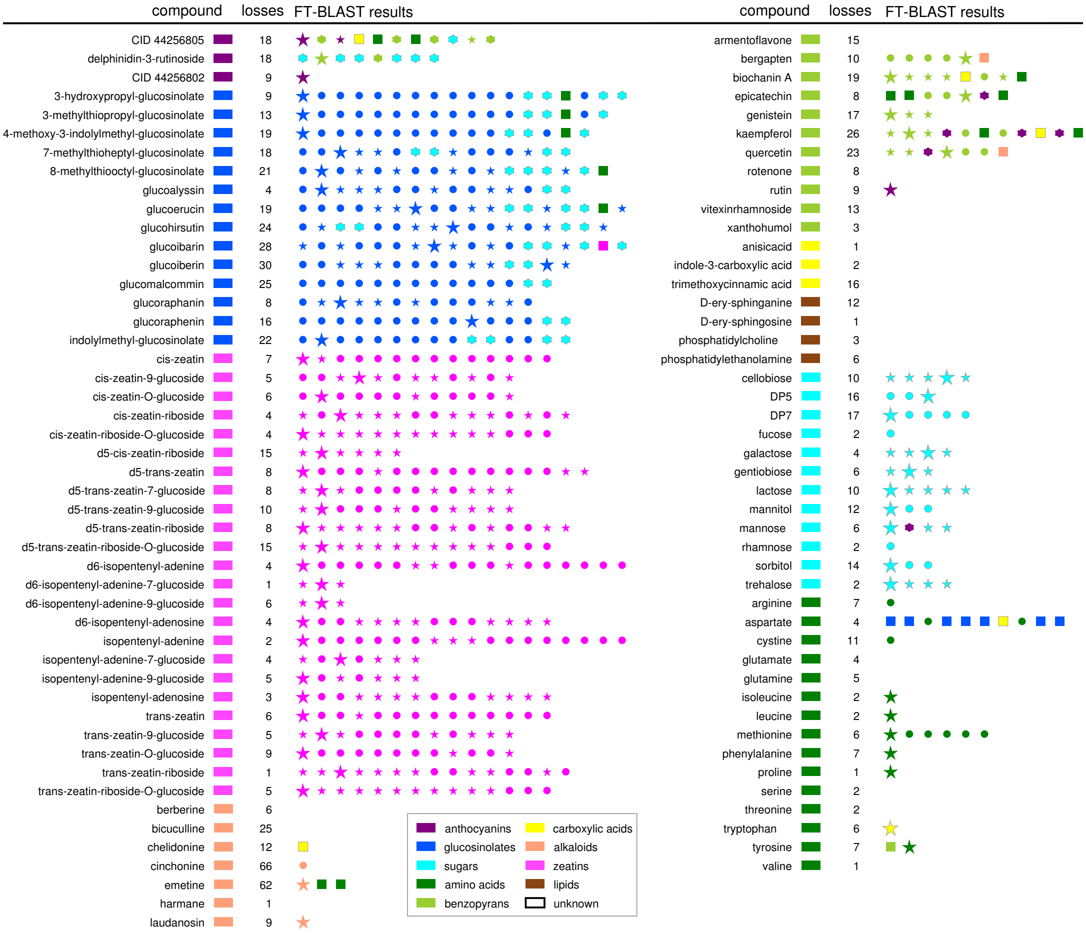
Overall, we return 557 compounds from the same group, 63 compounds from a similar group, 270 compounds with best or high PubChem/Tanimoto score, and only 31 compounds which do not fall into any of the above categories. In 33 cases (35%) we return the compound with highest chemical similarity at the top position; in 56 cases (60%) this compound is in the TOP 3.

In case we search for a fragmentation tree in the database where we did not exclude the query tree, our method recovered the correct tree in all cases. More precisely, the similarity of a fragmentation tree to itself, is highest among all trees in the dataset. Finding a "known compound" in a database is not a complicated task, and could be also done using methods based on spectral comparison. But we report this result here to show that our method will also "find the knowns", not only the unknowns.

5.7.4 Average Tanimoto Structural Similarity of FT-BLAST Hits

To assess the quality of the FT-BLAST hitlists, we report the average Tanimoto structural similarity score of the hits returned by FT-BLAST. We calculated the Tanimoto score of the query compound and each hitlist entry. We then averaged either over all hits with an FDR below the threshold of 30% for the FT-BLAST approach, the five best scoring hits disregarding the FDR for the TOP 5 approach, or only those hits both within the FDR threshold and the TOP 5 for the combined approach. Now we average over all 93 queries (Orbitrap trees with 1+ losses) to reach the final values of 0.76 for FT-BLAST, 0.67 for TOP 5, 0.78 for the combined approach. The TOP 5 approach is identical to Demuth *et al.* [25], the others are only adapted to the fact that an FDR estimation is available. Of course, this analysis is performed on the *leave-one-out* results.

Identical to Demuth *et al.* [25] we analyzed the Tanimoto scores $T(h)$ of the first h hits with h ranging from one to the number of compounds. Again, we did not use the FDR estimation but considered all scores obtained by a *leave-one-out* analysis. We then averaged over all compounds (Fig. 5.9). As Demuth *et al.* we compared these results with pseudo hitlists containing randomly ordered compounds (minimum value) and compounds arranged in descending order in accordance with the Tanimoto scores (upper limit). The average Tanimoto scores of our hitlists decrease from 0.78 ($h = 1$) to 0.34 ($h = 92$). The upper limit is between 0.90 ($h = 1$) and 0.34 ($h = 92$), and the


FT-BLAST results
anthocyanins
carboxylic acids
glucosinolates
alkaloids
sugars
zeatins
amino acids
lipids
benzopyrans
unknown

compound	losses
CID 44256805	18
delphinidin-3-rutinoside	18
CID 44256802	9
3-hydroxypropyl-glucosinolate	9
3-methylthiopropyl-glucosinolate	13
4-methoxy-3-indolylmethyl-glucosinolate	19
7-methylthioheptyl-glucosinolate	18
8-methylthiooctyl-glucosinolate	21
glucoalyssin	4
glucoerucin	19
glucohirsutin	24
glucoibarin	28
glucoiberin	30
glucomalcommin	25
glucoraphanin	8
glucoraphenin	16
indolylmethyl-glucosinolate	22
cis-zeatin	7
cis-zeatin-9-glucoside	5
cis-zeatin-O-glucoside	6
cis-zeatin-riboside	4
cis-zeatin-riboside-O-glucoside	4
d5-cis-zeatin-riboside	15
d5-trans-zeatin	8
d5-trans-zeatin-7-glucoside	8
d5-trans-zeatin-9-glucoside	10
d5-trans-zeatin-riboside	8
d5-trans-zeatin-riboside-O-glucoside	15
d6-isopentenyl-adenine	4
d6-isopentenyl-adenine-7-glucoside	1
d6-isopentenyl-adenine-9-glucoside	6
d6-isopentenyl-adenosine	4
isopentenyl-adenine	2
isopentenyl-adenine-7-glucoside	4
isopentenyl-adenine-9-glucoside	5
isopentenyl-adenosine	3
trans-zeatin	6
trans-zeatin-9-glucoside	5
trans-zeatin-O-glucoside	9
trans-zeatin-riboside	1
trans-zeatin-riboside-O-glucoside	5
berberine	6
bicuculline	25
chelidonine	12
cinchonine	66
emetine	62
harmane	1
laudanosin	9
armentoflavone	15
bergapten	10
biochanin A	19
epicatechin	8
genistein	17
kaempferol	26
quercetin	23
rotenone	8
rutin	9
vitexinrhamnoside	13
xanthohumol	3
anisicacid	1
indole-3-carboxylic acid	2
trimethoxycinnamic acid	16
D-ery-sphinganine	12
D-ery-sphingosine	1
phosphatidylcholine	3
phosphatidylethanolamine	6
cellobiose	10
DP5	16
DP7	17
fucose	2
galactose	4
gentiobiose	6
lactose	10
mannitol	12
mannose	6
rhamnose	2
sorbitol	14
trehalose	2
arginine	7
aspartate	4
cystine	11
glutamate	4
glutamine	5
isoleucine	2
leucine	2
methionine	6
phenylalanine	7
proline	1
serine	2
threonine	2
tryptophan	6
tyrosine	7
valine	1

Table 5.4: Results of the leave-one-out FT-BLAST analysis for the Orbitrap dataset, see text for details. Results are ordered according to fingerprint similarity score. Circles correspond to hits in the same compound class as the query compound, hexagons to hits from a "similar" compound class, see text for details. Boxes correspond to hits from all other classes. A large asterisk indicates the compound with the highest chemical similarity (PubChem/Tanimoto), and small asterisks indicate other hits with chemical similarity above 0.85. Symbols are colored by the class of the compound.

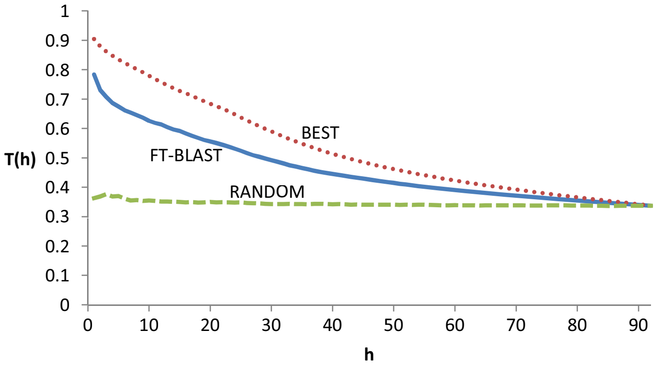


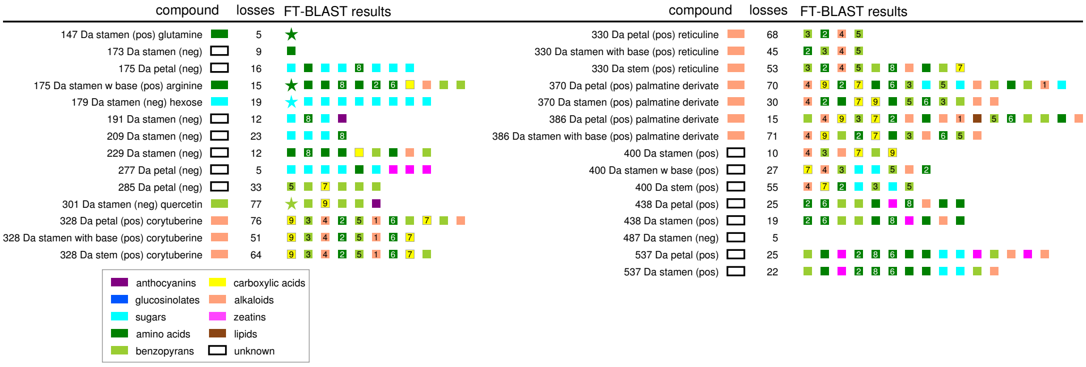
Figure 5.9: Average Tanimoto scores $T(h)$ between query structures and the first h structures from hitlists obtained by FT-BLAST without using FDR estimation (FT-BLAST), pseudo hitlists containing the database structures with maximum Tanimoto score to query structure (BEST) and randomly selected pseudo hitlists (RANDOM). All three analyses were performed on the Orbitrap dataset.

minimum value is about 0.34 for all h. All three values converge to 0.34 as this is the average Tanimoto score of all pairwise different compounds. Compared to Figure 1 in [25], the correlation values of FT-BLAST are considerably higher.

5.7.5 FT-BLAST Analysis of Poppy Data

To search for the unknown compounds in the icelandic poppy extracts, we calculated the all-against-all alignment of the fragmentation trees from poppy, those from the Orbitrap dataset, and the decoy trees generated from Orbitrap data. Again, normalization and fingerprint calculation was performed based on this similarity matrix. We then searched for the unknown compounds in the database of knowns (Orbitrap). The FDR was again 30%. Results of this analysis are shown in Table 5.5.

As mentioned above, eight compounds of the dataset were manually identified by experts. FT-BLAST identified glutamine, arginine and quercetin by returning the respective references from the Orbitrap dataset as first hit. For the hexose (179 Da) galactose and mannose are the first hits. The unknown is most likely glucose, which was not in our reference, so FT-BLAST suggests other hexoses. Four other compounds were manually identified as alkaloids. The 328 Da feature is corytuberine, the 330 Da compound is reticuline. We consider the 370 Da feature as hydrogenated and hydroxylated palmatine. The 386 Da unknown is again hydrogenated and hydroxylated palmatine, but additionally with an methyl-group and a broken double bond. Unfortunately, our reference dataset only contained few alkaloids. Our list of search results always contains the alkaloid laudanosine, which is most similar to the manual identifications. In case of corytuberine, chelidonine is always among the TOP3. These two alkaloids are extremely similar. The non-alkaloid hits are also reasonable:



compound		losses	FT-BLAST results
147 Da stamen (pos) glutamine	🟩	5	★
173 Da stamen (neg)	□	9	🟩
175 Da petal (neg)	□	16	🟦 🟩 🟦 🟦 🟩8 🟦 🟦 🟦
175 Da stamen w base (pos) arginine	🟩	15	★ 🟩 🟩 🟩8 🟩 🟩2 🟩6 🟨 🟧 🟢 🟢
179 Da stamen (neg) hexose	🟦	19	★ 🟦 🟦 🟦 🟦 🟦 🟦 🟦 🟦
191 Da stamen (neg)	□	12	🟦 🟩8 🟦 🟪
209 Da stamen (neg)	□	23	🟦 🟦 🟦 🟩8
229 Da stamen (neg)	□	12	🟩 🟩8 🟩 🟩 🟨 🟢 🟩 🟧 🟢
277 Da petal (neg)	□	5	🟦 🟦 🟦 🟦 🟩 🟦 🟥 🟥 🟥
285 Da petal (neg)	□	33	🟢5 🟢 🟨7 🟢 🟢 🟢
301 Da stamen (neg) quercetin	🟢	77	★ 🟢 🟨9 🟢 🟢 🟪
328 Da petal (pos) corytuberine	🟧	76	🟨9 🟢3 🟧4 🟩2 🟢5 🟧1 🟩6 🟢 🟨7 🟢 🟧
328 Da stamen with base (pos) corytuberine	🟧	51	🟨9 🟢3 🟧4 🟩2 🟢5 🟧1 🟩6 🟨7
328 Da stem (pos) corytuberine	🟧	64	🟨9 🟢3 🟧4 🟩2 🟢5 🟧1 🟩6 🟨7 🟢

🟪 anthocyanins	🟨 carboxylic acids
🔵 glucosinolates	🟧 alkaloids
🟦 sugars	🟥 zeatins
🟩 amino acids	🟫 lipids
🟢 benzopyrans	□ unknown

compound		losses	FT-BLAST results
330 Da petal (pos) reticuline	🟧	68	🟢3 🟩2 🟧4 🟢5
330 Da stamen with base (pos) reticuline	🟧	45	🟩2 🟢3 🟧4 🟢5
330 Da stem (pos) reticuline	🟧	53	🟢3 🟩2 🟧4 🟢5 🟢 🟩8 🟧 🟩 🟢 🟨7
370 Da petal (pos) palmatine derivate	🟧	70	🟧4 🟨9 🟩2 🟨7 🟩 🟩6 🟢3 🟦 🟢5 🟦 🟧 🟢 🟩 🟢 🟧1 🟦
370 Da stamen (pos) palmatine derivate	🟧	30	🟧4 🟩2 🟩 🟨7 🟨9 🟩 🟢5 🟩6 🟢3 🟢 🟧 🟧
386 Da petal (pos) palmatine derivate	🟧	15	🟢 🟧4 🟨9 🟢3 🟨7 🟩2 🟧 🟩 🟧 🟧1 🟫 🟢5 🟩6 🟢 🟢 🟩 🟧
386 Da stamen with base (pos) palmatine derivate	🟧	71	🟧4 🟨9 🟢 🟩2 🟨7 🟩 🟢3 🟧 🟩6 🟢5 🟧
400 Da stamen (pos)	□	10	🟧4 🟢3 🟧 🟨7 🟢 🟨9
400 Da stamen w base (pos)	□	27	🟨7 🟧4 🟢3 🟦 🟦 🟢5 🟧 🟩2
400 Da stem (pos)	□	55	🟧4 🟨7 🟩2 🟦 🟢3 🟦 🟢5
438 Da petal (pos)	□	25	🟩2 🟩6 🟢 🟢 🟩 🟥 🟩8 🟧 🟩 🟩
438 Da stamen (pos)	□	19	🟩2 🟩6 🟢 🟢 🟩 🟩8 🟥 🟩 🟧 🟩
487 Da stamen (neg)	□	5	
537 Da petal (pos)	□	25	🟢 🟩 🟥 🟩2 🟩8 🟩6 🟩 🟩 🟦 🟦 🟥 🟢 🟧 🟥 🟧
537 Da stamen (pos)	□	22	🟢 🟩 🟥 🟩2 🟩8 🟩6 🟩 🟩 🟦 🟦 🟢 🟧

Table 5.5: Searching poppy data in the Orbitrap dataset. A large asterisk indicates the correct identification. Search results mentioned in text and frequent search results are indicated by a boxed number, namely chelidonine (1), phenylalanine (2), laudanosine (3), rotenone (4), bergapten (5), tyrosine (6), trimethoxycinnamic acid (7), glutamate (8), and anisic acid (9). Symbols are colored by compound class.

Phenylalanine is the biosynthetic precursor of these alkaloids. Benzopyrans and hydroxylated alkaloids only differ by the fact that the oxygen is not in the ring system but attached to it as hydroxy group, and anisic acid (the carboxylic acid occurring in all hit lists) is again very similar to phenylalanine.

6 Structural Annotation of Fragmentation Trees

In this chapter, we develop an automated method to annotate the fragment peaks of a *known* compound with molecular structures. This process is called *in-silico* fragmentation, and several approaches for it exist, see Section 2.8.3 for an overview on these approaches.

The new feature of our approach is that we use previously calculated fragmentation trees to guide the prediction of fragment structural formulas. We combine this idea with the multi-step fragmentation model by Heinonen *et al.* [40]. In this optimization-based concept, the overall bond energy of the bonds broken during fragmentation is minimized.

There exist various applications for this approach. The typical use of *in-silico* fragmentation is the assessment of structure database hits. This has been proposed by Hill *et al.* [41] and fully automated by Wolf *et al.* [108]. It works as follows: If you have a list of database hits for your unknown compound, based on exact mass or molecular formula alone, you perform in-silico fragmentation for each of the hits based on the measured tandem MS spectrum. The database structure that can explain the most peaks with the lowest fragmentation energy is then ranked as the most likely compound for the measurement.

Another application is the verification of reference fragmentation trees. For the workflows of the previous chapter, we require reference trees. Results of these workflows are likely to become better if the quality of reference trees is increased. Here we will show how reference trees can be improved using *in-silico* fragmentation.

The third application is closely related to the second: If we are able to annotate the reference fragmentation trees with structures and find an unknown whose tree is similar to, say, two reference trees, then we can check which fragmentation cascades are shared by the trees and try to identify the structural feature responsible for this cascade based on the structural annotations of the reference trees.

6.1 One-step Fragmentation Problem

To be able to annotate complete trees, we need to be able to annotate a node, given a structural annotation of its parent. As this annotates one fragmentation step, we name this problem *One-step fragmentation.* Note, that single-step fragmentation [40] is a special case of one-step fragmentation. With single-step fragmentation, the fragments are always cleaved from the parent molecule, whereas one-step fragmentation allows an arbitrary starting fragment. Formal definition of one-step fragmentation based on molecular graphs leads to the Edge-Weighted Graph Motif problem.

Let Σ be the alphabet of elements in our molecules, such as $\Sigma = \{\mathrm{C}, \mathrm{H}, \mathrm{N}, \mathrm{O}, \mathrm{P}, \mathrm{S}\}$. A *molecular structure* M consists of a simple, undirected, connected graph where all vertices are labeled with elements from Σ, and edges are weighted by positive weights $w(e) > 0$. The elements of Σ will be called *colors* in this context. The *molecular formula* indicates how many vertices of each color are present in a molecular structure, e.g., $C_{20}H_{30}NO_8$. For *one-step fragmentation*, we are given a molecular structure M and a molecular formula f over Σ, and we try to find a connected subgraph of M that can be cleaved out with minimum costs, that is, minimum sum of energies for all cleaved bonds, and that has colors corresponding to f.

Edge-Weighted Graph Motif problem. Given a vertex-colored edge-weighted graph $G = (V, E)$ and a multiset of colors C of size k, find a connected subgraph $H = (U, F)$ of G such that the multiset of colors of U equals C, and H has minimum weight $w(H) := \sum_{\{u,v\} \in E, u \in U, v \in V \setminus U} w(\{u, v\})$.

This problem is NP-hard [34] as well as APX-hard [28] even on binary trees. Sikora*et al.* give an overview of the problem [89]. The releated Graph Motif problem, where no edge weights exist, and one asks whether any such subgraph exists, is NP hard even for bipartite graphs of bounded degree and two colors [33].

In the following, we will present a randomized and an exact branch-and-bound algorithm to solve the Edge-Weighted Graph Motif problem. Note that both algorithms can also calculate sub-optimal solutions. This will be required to annotate complete fragmentation trees. A third algorithm is given in [12], but its results were inferior to those presented here.

6.1.1 Random Separation

Cai *et al.* [19] proposed a randomized technique called *random separation* based on color-coding [1]. The key idea of random separation is to partition the vertices by coloring them randomly with two different colors. Then, connected components are identified and appropriate components are tested for optimality. Random separation has proven useful especially when the input graph has bounded degree. This is the case for molecular structures, where vertex degrees are bounded by the valences of elements.

We now apply random separation to the Edge-Weighted Graph Motif problem. Let k be the cardinality of the color multiset C. We search for a substructure $H = (U, F)$ that minimizes $w(H)$, where $|U| = k$. Let $N(U)$ denote the neighborhood of U in G. Given a graph $G = (V, E)$ and a random separation of G that partitions V into V_1 and V_2, there is a $2^{-(k+|N(U)|)+1}$ chance that U is entirely in V_1 and its neighborhood $N(U)$ is entirely in V_2 or vice versa. We use depth-first search to identify the connected components in V_1 and V_2. Simultaneously, colors are counted and costs for the partition are calculated. If the colors of a connected component correspond to the colors of the given multiset C and the costs are smaller than the costs of the best solution so far, the connected component is stored. In order to find the optimal solution with error probability ϵ, the procedure has to be repeated $\lceil |\log \epsilon| \,/\, |\log(1 - 2^{-(k+kd)+1})| \rceil$ times, where d is the maximum vertex degree in G.

The worst-case running time of this approach is as follows: Coloring takes $O(|V|)$ time. Depth first search has a running time of $O(|V| + |E|)$ but since molecular structures have bounded degree, the overall running time of one trial is $O(|V|)$. Accordingly, the overall running time of the random separation algorithm is $O(|\log \epsilon|\, 2^{(k+kd)} \cdot |V|)$. Recall that d is bounded in molecular structures. Also note that the term kd is due to the neighborhood of U in G. In our experiments, we observe that one-step fragmentation usually requires only few bonds to break. In this case, we can substitute the worst case estimation kd with maximal number b of bonds breaking in a fragmentation step. In our implementation, b is an input parameter that can be used to reduce the number of trials and, hence, to decrease running time. Obviously, b has to be chosen large enough to guarantee that the optimal solution is found with high probability.

6.1.2 Branch-and-Bound

The second algorithm is a classical branch-and-bound algorithm. It branches over edge sets that might break during a fragmentation step. Given an edge set, its deletion might separate G into a set of connected components. Similar to the random separation approach, depth first search is used to identify components that might be selected as a solution. If a solution has been found, its costs are used as an upper bound for pruning. The user can specify the maximum number of bonds b that may break during one single fragmentation step. We then try to cut out a solution with exactly $b' = 1, \ldots, b$ edges.

Since the costs of a solution correspond to the sum of weights of deleted edges, it is not necessary to iterate over all possible edge sets. To efficiently traverse the search tree, we use an edge set iterator that avoids edge sets with too high costs. Edges are sorted in increasing order with respect to their weight. Now, we can easily iterate over all edge sets of a fixed cardinality such that the edge sets have increasing costs. Thus, as soon as a solution with b' edges has been found, or the costs exceed that of the best solution found so far, all following edge sets of the same cardinality will have higher costs and can be omitted.

Sorting edges costs $O(|E| \log |E|)$ time. Running time of the depth first search is $O(|V|)$, as explained for random separation. The branch-and-bound algorithm iterates over $O(|V|^b)$ edge sets. This results in an overall running time of $O(|E| \log |E| + |V|^b)$. Unfortunately, running time is exponentially in b. But if the number of bonds that break in one single fragmentation step is small and bounded, b can be assumed as a constant and hence, the algorithm can be executed in polynomial time.

6.1.3 Enabling Structural Rearrangements

In general, this model assumes that bonds break during fragmentation, but no new bonds are formed. Unfortunately, this is not the case. The process of atoms changing

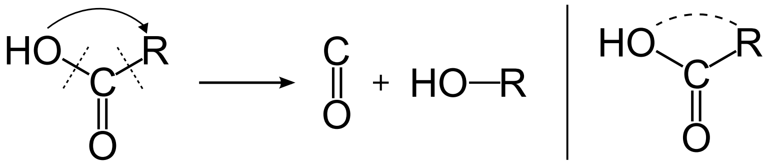


Figure 6.1: Left: The rearrangement of the carboxyl group. Without rearrangement, cleavage of carbon monoxide from this structure would also release hydroxide. Right: Pseudo edge (dashed) added to the molecular graph to account for this rearrangement. Figure taken from [94]

place in the molecule due to new bond formation is called rearrangement. In the following, we describe how we cater for some types of rearrangements.

Since hydrogen atoms are often subject to rearrangements, we do not include them in our calculations. We can support minor structural rearrangements such as carboxyl group rearrangements. These occur frequently as a result of cyclizations, see Figure 6.1. We model this using pseudo-edges as shown in the figure. As a result, we have to adapt the edge weights of all involved edges. Our model is not biochemically correct but enables us to reconstruct fragmentation trees with minor rearrangements.

6.2 Multistep Fragmentation Model

Unfortunately, solving the one-step fragmentation problem is not sufficient, since fragmentation pathways consist of consecutive fragmentation steps [40], where fragments can be cleaved from other fragments. Here, we represent such pathways by *fragmentation trees.*

For the *multistep fragmentation* model, we are given a molecular structure M and a fragmentation tree T. We want to assign sub-structures to the nodes of the fragmentation tree that match their molecular formulas, such that the total cost of cutting out the substructures, over all edges of the fragmentation tree, is minimized. Clearly, it does not suffice to search for the optimal graph motif in every fragmentation step independently, since following fragmentation steps may be cleaved from a suboptimal substructure with lower total costs.

With this approach not use prior information about metabolite fragmentation such as fragmentation rules except edge weights that represent bond energies. Thus, our approach is applicable for *any* compound, even if fragmentation of its class has not been thoroughly studied.

6.2.1 A Beam Search Algorithm for Multistep Fragmentation

In order to find a fragmentation process consistent with the given fragmentation tree, we use a search tree. Since it does not suffice to take the fragment with minimum costs in every fragmentation step, our heuristic allows the user to specify a number p so that in every step, the best p fragments are considered. For each such fragment,

		multistep heuristic		
Mass (Da)	#comp.	RS	BB-3	BB-5
< 100	1	< 1 s	< 1 s	< 1 s
100–200	17	23.2 s	0.1 s	0.2 s
200–300	16	50.7 min	0.7 s	6.0 s
300–400	3	4.8 h	6.7 s	55.7 s
400–500	5	> 1 day	0.5 s	5.6 s
> 500	1	> 1 week	10.2 min	4.6 h

Table 6.1: The average running times of the algorithms: neighborhood for random separation has been estimated with $b = 5$, branch-and-bound allowed $b = 3$ (BB-3) and $b = 5$ (BB-5) bonds to break. Multistep fragmentation considered the 5 best fragments in every step.

we build up the search tree recursively, and accept the fragment that results in lowest total costs. Additionally, we check whether moving up a node in a fragmentation tree by one level will decrease the total cost of fragmentation. To do so, we compare the total costs of cleaving fragment f and all subsequent fragments from its parent, with the total costs of cleaving them from its grandfather. This way, we allow and can identify pull-ups in the tree (See Section 3.4). In our calculations, we found that $p = 5$ results in good annotations, while keeping running times feasible.

6.3 Experimental Results

We implemented our algorithms in Java 1.5. Running times were measured on an Intel Core 2 Duo processor, 2.5 GHz with 3 GB memory. For the random separation algorithm, we use an error probability $\epsilon = 0.1\%$, so that the optimal solution will be found with a probability of 99.9%. In the multistep fragmentation evaluation, we set $p = 5$, thus, keeping the five best substructures in each fragmentation step.

As test data we used 35 fragmentation trees of the QStar dataset from Section 4.1 on page 37 and 8 trees of spectra from an LTQ Orbitrap XL instrument (Thermo Fisher Scientific) using the same experimental setup as described in Section 4.1 except for using PQD fragmentation, calculated using the parameters of the Orbitrap dataset in Section 4.1 and the DP_{10} heuristic.

Detailed information about running times of the multistep heuristic using the different approaches can be found in Table 6.1. One can see that the branch-and-bound algorithms outperforms the more sophisticated algorithm. The random separation algorithm performs fast for small instances, but requires several days for molecules > 400 Da.

Fig. 6.2 shows how the running times of the algorithms depend on the size of the molecular structure M and on the size of the fragments. It illustrates that the running time of the branch-and-bound algorithm mainly depends on the size of M, particularly

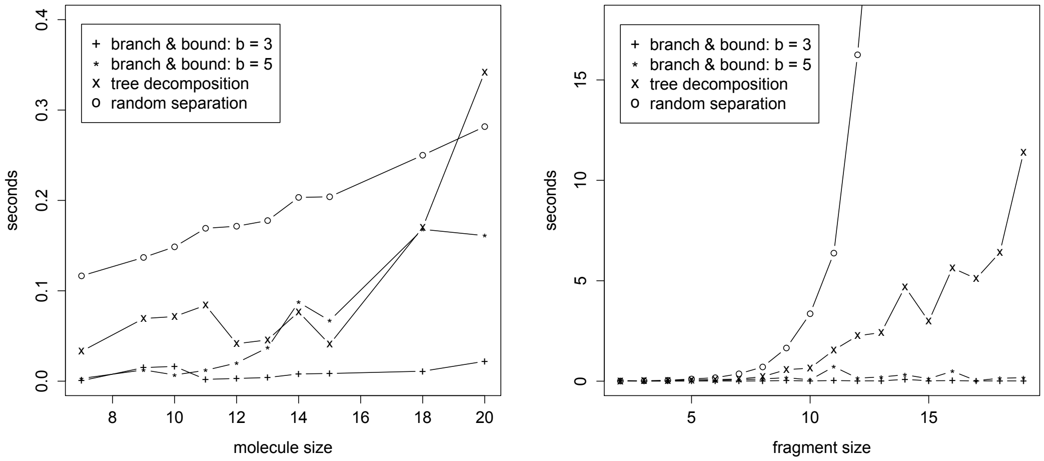


Figure 6.2: Running time comparison of the three algorithms: The left diagram shows the average running time depending on the molecule size $|M|$ given a fixed fragment size of six. In the right diagram the average running time for several molecule sizes in dependence on the fragment's size is displayed.

for larger b. Finally, running time of the random separation algorithm depends mainly on fragment size.

For all instances that finished computation using the random separation algorithm, we reach identical total costs as for the branch-and-bound algorithm. Annotations differ only marginally in the sequence of cleavages. The annotations found by the branch-and-bound algorithm with $b = 3$ and $b = 5$ also have identical costs. This supports our assumption that instances based on molecular graphs do not resemble the full complexity of the Edge-Weighted Graph Motif problem.

Our annotations turned out to be valuable to validate the fragmentation trees proposed by the methods of Chapter 3. Our analysis of the annotated fragmentation trees identified peaks in several fragmentation trees that were annotated with a molecular formula but probably are noise peaks. These peaks have a low intensity and are also scored very low. In our analysis, we were unable to assign a fragment to the molecular formula. For example, the 250 Da fragment of hexosyloxycinnamoyl choline was identified as noise peak. The score of the corresponding fragmentation step is very low compared to the others, and a fragment with formula $C_{10}H_{20}NO_6$ cannot be cleaved from the corresponding structure without major rearrangements. We also identified an intense peak that could not be annotated with any fragment. Consultation with an expert resulted in the conclusion that the spectrum was contaminated.

Furthermore, we identified three nodes in two fragmentation trees that had been inserted too low into the fragmentation tree, and pulling them up one level resulted in a fragmentation pattern with significantly decreased total costs. In two other fragmentation trees, we identified nodes where pulling-up results in slightly reduced costs. A closer look at the fragmentation patterns revealed that in these cases, two competitive paths might co-occur.

7 Conclusion

In this thesis, we have presented several algorithms and concepts for the analysis of small compounds using high-accuracy tandem mass spectrometry. Whereas previous approaches are based heavily on databases, this work focuses mainly on the identification of novel compounds, which are neither contained in spectral nor in compound databases.

We developed fragmentation trees as a means to annotate fragmentation spectra and formulated their calculation as a graph theoretical problem. Unfortunately, the underlying problem is NP-hard. A first heuristic already produced good results on real mass spectra from three different instruments. Manual inspection of the fragmentation trees showed that 78.9% predicted fragmentation reactions were correct according to mass spectrometry experts. Some trees have been validated by multi-stage mass spectrometry, which can to a certain extent reveal the occurring reactions. Again, a good agreement between the experimental data and the predictions was found. For one dataset, computationally predicted spectra were available and most of their annotations matched our annotations of the measured spectra. Thus, good quality of the fragmentation trees was established.

We have developed an integer linear program (ILP) for the calculation of fragmentation trees. This enables the evaluation of our heuristic results against exact solutions. To determine an exact solution for an instance the ILP takes at most 15 seconds. Although this is slower than several heuristics also proposed in this work, it is still clearly faster than the spectra can be acquired. Evaluations show that the exact solution is preferable in case the tree structure is relevant for further processing, whereas a good heuristic is sufficient, when only the score of the tree is relevant, to determine the molecular formula of a fragmented compound, for example.

Fragmentation trees are already helpful as they save the investigator the tedious work of manually annotating the spectrum with molecular formulas, but still do not provide an automated identification of the measured compound. Full structural elucidation will most likely be impossible using only mass spectrometry. But by using the fragmentation tree alignment concept presented in this work, information about the compound class, similar compounds and structural elements can be revealed. This is achieved by comparing the fragmentation tree of the unknown against reference fragmentation trees. To achieve a high quality similarity search, we were able to transfer the local alignment concept from sequence comparisons as well as a significance calculation technique from proteomics to fragmentation trees.

Evaluations on three different real datasets show that fragmentation tree fingerprint similarities correlate well with structural similarities of the compounds. Correlation with the Tanimoto score used by PubChem ranged as high as $r = +0.68$ ($r^2 = 0.46$) for large enough trees from one dataset. Intriguingly, we also reached a good correlation between the datasets, indicating that fragmentation tree alignment is independent of the instrument type used. Hierarchical clustering based on the similarity matrix grouped compounds of the same class together. Thus, we were able to predict the compound class of some unknown molecules from Icelandic poppy as they fell into well-separated clusters of reference compounds. Finally, we presented FT-BLAST, a database search tool, which can not only retrieve similar compounds from a reference database, but also assess their significance using a decoy database strategy. By using an FDR threshold of 30 % the hits had an average structural similarity of 0.76 to the query, indicating that only highly similar hits are returned. Thus, our method predicts compound classes as well as similar compounds of an unknown molecule in a fully automated pipeline.

An alternative path is to use fragmentation trees as basis for an *in silico fragmentation* approach. Given the structure and the fragmentation tree of a compound, we search for an assignment of fragment structures that is energetically optimal. For this, we developed a branch-and-bound heuristic that repeatedly needs to solve the NP-hard Edge-Weighted Graph Motif problem. For this problem, we present two algorithms, one based on a randomized technique and an exact branch-and-bound algorithm. In evaluations on real data, we found that the simple branch-and-bound approach performs best. By applying this approach to fragmentation trees from reference data, we were able to detect some inconsistencies in the trees. Improving the quality of reference trees will likely also improve the results of an alignment against those trees. Additionally, the method could be used to assess structure hypotheses for the compound, e.g. retrieved from a compound database.

7.1 Future Work

As the concept of fragmentation trees is still relatively young, it yet has to uncover its full potential. Current scores for fragmentation tree calculation have been chosen ad-hoc based on the suggestions of a few experts and minor improvements have been made to address frequently occurring errors. With sufficient data available a statistically sound scoring could be learned from the loss frequencies in the data. This, of course, has to be done under the cautious eye of a mass spectrometry expert, to prevent amplification of erroneous assignments.

In a similar fashion, scores for the tree alignment may be improved as well. Due to an easily susceptible outcome (e.g. the correlation with structural similarity or the average structural similarity of all FT-BLAST hits) a straightforward optimization approach is possible, if overfitting can be avoided. Currently, the method also has difficulties to compare small trees with rather large ones, due to the fact that several losses in the large tree are combined to a single loss in the small tree. The join operation has been introduced to cover this case, but it currently combines only two losses to keep calculations feasible. Improved alignment algorithms may allow for joins over three or more nodes. An improved generation of decoy trees would increase the accuracy of q-values, but this requires a thorough understanding of fragmentation tree characteristics.

A tool for database querying, like FT-BLAST, is only as good as the database it searches. Currently, FT-BLAST is based on flat files and reference measurements the user supplies. To make it a success a fragmentation tree database has to be developed. Depositing data in such a repository should be as easy as possible to encourage its use. Fragmentation trees in the database could either be calculated automatically from the spectra, or curated trees could be uploaded manually.

The idea to use fragmentation trees as an aid to predict fragmentation in silico could be combined with the faster heuristic by Wolf *et al.* [108]. This might improve the search in compound databases, although structural rearrangements will still present a major difficulty.

Approaches to calculate fragmentation trees from different types of fragmentation data are already under development by my colleagues. Scheubert *et al.* proposed an algorithm for the calculation of fragmentation trees from multi-stage mass spectrometry data [84]. The resulting COMBINED COLORFUL SUBTREE PROBLEM turns out to be computationally hard to approximate. Results based on an heuristic look promising, but a sufficiently fast exact algorithm is still to be found.

When trying to calculate a fragmentation tree from GC-MS data with the hard EI fragmentation, another problem arises: The mass of the unfragmented compound is unknown. Thus, Hufsky *et al.* focus on detecting the molecular ion peak and predicting a molecular formula for the compound [45]. This can be done by calculating all trees rooted in one of the peaks in the higher mass range and selecting the best scoring one.

Various applications of the fragmentation tree alignment concept are possible: The identification of the compound class may be helpful for the dereplication of novel drug leads, for example, potential antibiotics. In this field, a large amount of time and money is often wasted by identifying a compound, of which a derivate is already known. Compound classification could be improved and adjusted to this application by using more involved classifiers based on machine learning.

Recently, Watrous *et al.* published an approach to generate metabolic networks from tandem MS data [101]. This concept was pioneered by Breitling *et al.* [15] using single MS data. The idea is to use the similarity between the spectra and a set of known bio-reactions to infer potential metabolic reactions between compounds, which do not even have to be identified. But current approaches lack specificity. Here, FT-BLAST could help by assigning q-values rather than similarities to the reaction candidate. This might help to filter spurious candidates. Still, problems arise as such an inferred network tends to be “somewhat” transitive: If A is made from B and B from C, then A and C will be similar. Here, a graph algorithm will be necessary to thin-out the network in a sensible way. Less complex, but also highly relevant is the analysis of drug degradation. Here, all compounds a body produces by converting a certain drug have to be identified. This may be achieved by similar methods as above. Full identification of the degradation process is a requirement for the approval of a drug by health authorities.

Finally, the results of an FT-BLAST analysis might help in elucidating the structure of an unknown compound. By identifying the compound class and perhaps detecting a common substructure among the related compounds found by FT-BLAST, hints can be given towards molecular structure. It might then be possible to generate all structures fulfilling these criteria, e.g., using MOLGEN [65]. To select among such a reduced number of candidates, a simple 1D NMR or an NMR measurement with a small amount of sample might be sufficient. This would greatly reduce the effort necessary for structure elucidation.

Although the molecules playing the central part in this thesis are small, even from a molecular part of view, they may have a huge impact on the macroscopic level. Hopefully, these approaches may aid to gain insight into new metabolic processes, ultimately leading to a better understanding of life by bridging the gap between genotype and phenotype. Eventually, these methods might lead to the discovery of new drugs or at least give hints, where not to search. But even if all of that is not the case, it has been fun to find out how much information you can get by smashing something into pieces and then putting these pieces onto a scale.

Bibliography

[1] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J ACM*, 42(4):844–856, 1995. 72

[2] M. Baker. Metabolomics: From small molecules to big ideas. *Nat Methods*, 8:117–121, 2011. 2

[3] P. Baldi and R. W. Benz. BLASTing small molecules–statistics and extreme statistics of chemical similarity scores. *Bioinformatics*, 24(13):i357–i365, 2008. 61

[4] M. Bellew, M. Coram, M. Fitzgibbon, M. Igra, T. Randolph, P. Wang, D. May, J. Eng, R. Fang, C. Lin, J. Chen, D. Goodlett, J. Whiteaker, A. Paulovich, and M. McIntosh. A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution lc-ms. *Bioinformatics*, 22(15):1902–1909, Aug 2006. 15

[5] R. J. Bino, R. D. Hall, O. Fiehn, J. Kopka, K. Saito, J. Draper, B. J. Nikolau, P. Mendes, U. Roessner-Tunali, M. H. Beale, R. N. Trethewey, B. M. Lange, E. S. Wurtele, and L. W. Sumner. Potential of metabolomics as a functional genomics tool. *Trends Plant Sci*, 9(9):418–425, 2004. 16

[6] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. of ACM Symposium on Theory of Computing (STOC 2007)*, pages 67–74. ACM press, New York, 2007. 29

[7] S. Böcker, B. Kehr, and F. Rasche. Determination of glycan structure from tandem mass spectra. In *Proc. of Computing and Combinatorics Conference (COCOON 2009)*, volume 5609 of *Lect Notes Comput Sci*, pages 258–267. Springer, Berlin, 2009. ix

[8] S. Böcker, B. Kehr, and F. Rasche. Determination of glycan structure from tandem mass spectra. *IEEE/ACM Trans Comput Biology Bioinform*, 8(4):976–986, 2011. ix

[9] S. Böcker, M. Letzel, Zs. Lipták, and A. Pervukhin. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25(2):218–224, 2009. 17, 22, 23, 34, 35, 38, 65

[10] S. Böcker and Zs. Lipták. A fast and simple algorithm for the Money Changing Problem. *Algorithmica*, 48(4):413–432, 2007. 17

[11] S. Böcker and F. Rasche. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinformatics*, 24:I49–I55, 2008. Proc. of *European Conference on Computational Biology* (ECCB 2008). 27, 28, 29, 30, 31, 34, 35, 37, 38, 55

[12] S. Böcker, F. Rasche, and T. Steijger. Annotating fragmentation patterns. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI 2009)*, volume 5724 of *Lect Notes Comput Sci*, pages 13–24. Springer, Berlin, 2009. ix, 72

[13] H. B. Bode and R. Müller. The impact of bacterial genomics on natural product research. *Angew Chem Int Ed Engl*, 44:6828–6846, 2005. 1

[14] S. Bourcier and Y. Hoppilliard. Use of diagnostic neutral losses for structural information on unknown aromatic metabolites: An experimental and theoretical study. *Rapid Commun Mass Spectrom*, 23:93–103, 2009. 38

[15] R. Breitling, D. Vitkup, and M. P. Barrett. New surveyor tools for charting microbial metabolic maps. *Nat Rev Microbiol*, 6(2):156–161, 2008. 81

[16] A. P. Bruins. Mass spectrometry with ion sources operating at atmospheric pressure. *Mass Spectrom Rev*, 10:53–77, 1991. 9

[17] B. Buchanan, G. Sutherland, and E. A. Feigenbaum. *Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry*, volume 4 of *Machine Intelligence*, page 209. Edinburgh University Press, 1969. 19

[18] J. Buckingham, editor. *Dictionary of Natural Products.* Chapman & Hall/CRC press, London, 2005. 8

[19] L. Cai, S. M. Chan, and S. O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proc. of International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, pages 239–250. Springer, Berlin, 2006. 72

[20] R. Caspi, T. Altman, J. M. Dale, K. Dreher, C. A. Fulcher, F. Gilham, P. Kaipa, A. S. Karthikeyan, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, S. Paley, L. Popescu, A. Pujar, A. G. Shearer, P. Zhang, and P. D. Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res*, 38:D473–D479, 2010. 7

[21] J. Claesen, P. Dittwald, T. Burzykowski, and D. Valkenborg. An efficient method to calculate the aggregated isotopic distribution and exact center-masses. *J Am Soc Mass Spectrom*, 23(4):753–63, 2012. 17

[22] G. M. Cragg, P. G. Grothaus, and D. J. Newman. Impact of natural products on developing new anti-cancer agents. *Chem Rev*, 109:3012–3043, 2009. 1

[23] Q. Cui, I. A. Lewis, A. D. Hegeman, M. E. Anderson, J. Li, C. F. Schulte, W. M. Westler, H. R. Eghbalnia, M. R. Sussman, and J. L. Markley. Metabolite identification via the Madison Metabolomics Consortium Database. *Nat Biotechnol*, 26(2):162–164, 2008. 7

[24] J. C. D'Auria and J. Gershenzon. The secondary metabolism of Arabidopsis thaliana: growing like a weed. *Curr Opin Plant Biol*, 8(3):308–316, 2005. 1

[25] W. Demuth, M. Karlovits, and K. Varmuza. Spectral similarity versus structural similarity: Mass spectrometry. *Anal Chim Acta*, 516(1-2):75–85, 2004. 2, 16, 67, 68, 69

[26] P. D'haeseleer. How does gene expression clustering work? *Nat Biotechnol*, 23(12):1499–1501, 2005. 55

[27] F. Dieterle, B. Riefke, G. Schlotterbeck, A. Ross, H. Senn, and A. Amberg. NMR and MS methods for metabonomics. *Methods Mol Biol*, 691:385–415, 2011. 1

[28] R. Dondi, G. Fertin, and S. Vialette. Maximum motif problem in vertex-colored graphs. In *Proc. of Symposium on Combinatorial Pattern Matching (CPM 2009)*, volume 5577 of *Lect Notes Comput Sci*, pages 221–235. Springer, Berlin, 2009. 28, 72

[29] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1972. 28

[30] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci*, 42(6):1273–1280, 2002. 61

[31] I. Eidhammer, K. Flikka, L. Martens, and S.-O. Mikalsen. *Computational Methods for Mass Spectrometry Proteomics*. Wiley, 2007. 5, 14, 15

[32] M. Elyashberg, A. Williams, and G. Martin. Computer-assisted structure verification and elucidation tools in NMR-based structure elucidation. *Prog Nucl Magn Reson Spectrosc*, 53(1–2):1–104, 2007. 1

[33] M. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP 2007)*, volume 4596 of *Lect Notes Comput Sci*, pages 340–351. Springer, Berlin, 2007. 72

[34] M. R. Fellows, J. Gramm, and R. Niedermeier. On the parameterized intractability of motif search problems. *Combinatorica*, 26(2):141–167, 2006. 28, 72

[35] A. R. Fernie, R. N. Trethewey, A. J. Krotzky, and L. Willmitzer. Metabolite profiling: From diagnostics to systems biology. *Nat Rev Mol Cell Biol*, 5(9):763–769, 2004. 1

[36] O. Fiehn. Extending the breadth of metabolite profiling by gas chromatography coupled to mass spectrometry. *Trends Analyt Chem*, 27(3):261–269, 2008. 9

[37] D. Goldberg, M. W. Bern, B. Li, and C. B. Lebrilla. Automatic determination of O-glycan structure from fragmentation spectra. *J Proteome Res*, 5(6):1429–1434, 2006. 23

[38] J. Gross. *Mass Spectrometry: A textbook*. Springer, Berlin, Berlin, 2004. 5, 9, 13

[39] S. Guillemot and F. Sikora. Finding and counting vertex-colored subtrees. In *Proc. of Mathematical Foundations of Computer Science (MFCS 2010)*, volume 6281 of *Lect Notes Comput Sci*, pages 405–416. Springer, Berlin, 2010. 29

[40] M. Heinonen, A. Rantanen, T. Mielikäinen, J. Kokkonen, J. Kiuru, R. A. Ketola, and J. Rousu. FiD: A software for ab initio structural identification of product ions from tandem mass spectrometric data. *Rapid Commun Mass Spectrom*, 22(19):3043–3052, 2008. 19, 71, 74

[41] D. W. Hill, T. M. Kertesz, D. Fontaine, R. Friedman, and D. F. Grant. Mass spectral metabonomics beyond elemental formula: Chemical database querying by matching experimental with computational fragmentation spectra. *Anal Chem*, 80(14):5574–5582, 2008. 18, 31, 34, 37, 46, 65, 66, 71

[42] M. Holčapek, R. Jirásko, and M. Lísa. Basic rules for the interpretation of atmospheric pressure ionization mass spectra of small molecules. *J Chromatogr A*, 1217(25):3908–3921, 2010. 38

[43] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima, Y. Oda, Y. Kakazu, M. Kusano, T. Tohge, F. Matsuda, Y. Sawada, M. Y. Hirai, H. Nakanishi, K. Ikeda, N. Akimoto, T. Maoka, H. Takahashi, T. Ara, N. Sakurai, H. Suzuki, D. Shibata, S. Neumann, T. Iida, K. Tanaka, K. Funatsu, F. Matsuura, T. Soga, R. Taguchi, K. Saito, and T. Nishioka. MassBank: A public repository for sharing mass spectral data for life sciences. *J Mass Spectrom*, 45(7):703–714, 2010. 16, 52

[44] F. Hufsky, K. Dührkop, F. Rasche, M. Chimani, and S. Böcker. Fast alignment of fragmentation trees. *Bioinformatics*, 28:i265–i273, 2012. Proc. of *Intelligent Systems for Molecular Biology* (ISMB 2012). ix, 50

[45] F. Hufsky, M. Rempt, F. Rasche, G. Pohnert, and S. Böcker. De novo analysis of electron impact mass spectra using fragmentation trees. *Anal Chim Acta*, 739:67–76, 2012. ix, 9, 81

[46] N. Jaitly, M. E. Monroe, V. A. Petyuk, T. R. W. Clauss, J. N. Adkins, and R. D. Smith. Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline. *Anal Chem*, 78(21):7397–7409, 2006. 23

[47] T. Jiang, L. Wang, and K. Zhang. Alignment of trees: An alternative to tree edit. *Theor Comput Sci*, 143(1):137–148, 1995. 48, 49, 50

[48] L. Käll, J. D. Storey, M. J. MacCoss, and W. S. Noble. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J Proteome Res*, 7(1):29–34, 2008. 19

[49] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: New developments in KEGG. *Nucleic Acids Res*, 34:D354–D357, 2006. 7, 24

[50] M. Karas and F. Hillenkamp. Laser desorption ionization of proteins with molecular masses exceeding 10,000 Daltons. *Anal Chem*, 60:2299–2301, 1988. 9

[51] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal Chem*, 74(20):5383–5392, 2002. 19

[52] S. Kim, N. Gupta, and P. A. Pevzner. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J Proteome Res*, 7(8):3354–3363, 2008. 20

[53] T. Kind and O. Fiehn. Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*, 8:105, 2007. 34

[54] T. Kind and O. Fiehn. Advances in structure elucidation of small molecules using mass spectrometry. *Bioanal Rev*, 2(1-4):23–60, 2010. 13

[55] I. Koo, X. Zhang, and S. Kim. Wavelet- and fourier-transform-based spectrum similarity approaches to compound identification in gas chromatography/mass spectrometry. *Anal Chem*, 83(14):5631–5638, 2011. 65

[56] J. Kopka, N. Schauer, S. Krueger, C. Birkemeyer, B. Usadel, E. Bergmüller, P. Dörmann, W. Weckwerth, Y. Gibon, M. Stitt, L. Willmitzer, A. R. Fernie, and D. Steinhauser. GMD@CSB.DB: The Golm Metabolome Database. *Bioinformatics*, 21(8):1635–1638, 2005. 17

[57] I. Koutis and R. Williams. Limits and applications of group algebras for parameterized problems. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP 2009)*, volume 5555 of *Lect Notes Comput Sci*, pages 653–664. Springer, Berlin, 2009. 29

[58] E. Lange, C. Gröpl, K. Reinert, O. Kohlbacher, and A. Hildebrandt. High-accuracy peak picking of proteomics data using wavelet techniques. In *Proc. of Pacific Symposium on Biocomputing (PSB 2006)*, pages 243–254, 2006. 16

[59] R. L. Last, A. D. Jones, and Y. Shachar-Hill. Towards the plant metabolome and beyond. *Nat Rev Mol Cell Biol*, 8:167–174, 2007. 1

[60] A. R. Leach and V. J. Gillet. *An Introduction to Chemoinformatics.* Springer, Berlin, Dordrecht, The Netherlands, 2005. 61

[61] J. W.-H. Li and J. C. Vederas. Drug discovery and natural products: End of an era or an endless frontier? *Science*, 325(5937):161–165, 2009. 1, 8

[62] I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. Solving the prize-collecting Steiner tree problem to optimality. In *Proc. of Algorithm Engineering and Experiments (ALENEX 2005)*, pages 68–76. SIAM, 2005. 28, 30

[63] R. E. March, X.-S. Miao, and C. D. Metcalfe. A fragmentation study of a flavone triglycoside, kaempferol-3-o-robinoside-7-o-rhamnoside. *Rapid Commun Mass Spectrom*, 18:931–934, 2004. 39

[64] F. W. McLafferty and F. Tureček. *Interpretation of Mass Spectra.* University Science Books, Mill valley, California, fourth edition, 1993. 38

[65] M. Meringer. Structure enumeration and sampling. In J.-L. Faulon and A. Bender, editors, *Handbook of Chemoinformatics Algorithms*, Mathematical & Computational Biology, pages 233–267. Chapman and Hall/CRC, 2010. 82

[66] C. E. Mortimer. *Chemistry.* Wadsworth Pub. Co., 6. edition, 1986. 5

[67] G. Moss, P. Smith, and D. Tavernier. Glossary of class names of organic compounds and reactive intermediates based on structure. *Pure and applied chemistry*, 67:1307–1375, 1995. 7

[68] S. J. Nelson, W. D. Johnston, and B. L. Humphreys. Relationships in medical subject headings. In C. A. Bean and R. Green, editors, *Relationships in the organization of knowledge*, pages 171–184. Kluwer Academic Publishers, 2001. 7

[69] A. I. Nesvizhskii, O. Vitek, and R. Aebersold. Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nat Methods*, 4(10):787–797, 2007. 19

[70] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and inter-laboratory transferability of a tandem mass spectral reference library: 1. Results of an Austrian multicenter study. *J Mass Spectrom*, 44(4):485–493, 2009. 16

[71] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and the inter-laboratory transferability of a tandem mass spectral reference library: 2. Optimization and characterization of the search algorithm. *J Mass Spectrom*, 44(4):494–502, 2009. 16

[72] J. V. Olsen, B. Macek, O. Lange, A. Makarov, S. Horning, and M. Mann. Higher-energy c-trap dissociation for peptide modification analysis. *Nat Methods*, 4(9):709–712, 2007. 12

[73] J. V. Olsen, J. C. Schwartz, J. Griep-Raming, M. L. Nielsen, E. Damoc, E. Denisov, O. Lange, P. Remes, D. Taylor, M. Splendore, E. R. Wouters, M. Senko, A. Makarov, M. Mann, and S. Horning. A dual pressure linear ion trap orbitrap instrument with very high sequencing speed. *Mol Cell Proteomics*, 8:2759–2769, 2009. 10

[74] G. J. Patti, O. Yanes, and G. Siuzdak. Innovation: Metabolomics: The apogee of the omics trilogy. *Nat Rev Mol Cell Biol*, 13(4):263–269, 2012. 2, 14, 16

[75] T. Pluskal, T. Uehara, and M. Yanagida. Highly accurate chemical formula prediction tool utilizing high-resolution mass spectra, MS/MS fragmentation, heuristic rules, and isotope pattern matching. *Anal Chem*, 84(10):4396–4403, 2012. 17

[76] F. Rasche, K. Scheubert, F. Hufsky, T. Zichner, M. Kai, A. Svatoš, and S. Böcker. Identifying the unknowns by aligning fragmentation trees. *Anal Chem*, 84(7):3417–3426, 2012. ix, 52, 53

[77] F. Rasche, A. Svatoš, R. K. Maddula, C. Böttcher, and S. Böcker. Computing fragmentation trees from tandem mass spectrometry data. *Anal Chem*, 83(4):1243–1251, 2011. ix, 31, 37, 52

[78] I. Rauf, F. Rasche, F. Nicolas, and S. Böcker. Finding maximum colorful subtrees in practice. In *Proc. of Research in Computational Molecular Biology (RECOMB 2012)*, volume 7262 of *Lect Notes Comput Sci*, pages 213–223. Springer, Berlin, 2012. ix, 31

[79] D. J. Rogers and T. T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960. 62

[80] S. Rogers, R. A. Scheltema, M. Girolami, and R. Breitling. Probabilistic assignment of formulas to mass peaks in metabolomics experiments. *Bioinformatics*, 25(4):512–518, 2009. 34

[81] M. Rojas-Chertó, P. T. Kasper, E. L. Willighagen, R. J. Vreeken, T. Hankemeier, and T. H. Reijmers. Elemental composition determination based on MS^n. *Bioinformatics*, 27:2376–2383, 2011. 18

[82] M. Rojas-Chertó, J. E. Peironcely, P. T. Kasper, J. J. J. van der Hooft, R. C. H. de Vos, R. J. Vreeken, T. Hankemeier, and T. H. Reijmers. Metabolite identification using automated comparison of high-resolution multistage mass spectral trees. *Anal Chem*, 84(13):5524–5534, 2012. 18

[83] K. Scheubert, F. Hufsky, F. Rasche, and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. In *Proc. of Research in Computational Molecular Biology (RECOMB 2011)*, volume 6577 of *Lect Notes Comput Sci*, pages 377–391. Springer, Berlin, 2011. ix

[84] K. Scheubert, F. Hufsky, F. Rasche, and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. *J Comput Biol*, 18(11):1383–1397, 2011. ix, 28, 81

[85] J. Senior. Partitions and their representative graphs. *Amer J Math*, 73(3):663–689, 1951. 22

[86] P. Y. I. Shek, J. Zhao, Y. Ke, K. W. M. Siu, and A. C. Hopkinson. Fragmentations of protonated arginine, lysine and their methylated derivatives: Concomitant losses of carbon monoxide or carbon dioxide and an amine. *J Phys Chem A*, 110:8282–8296, 2006. 38

[87] M. T. Sheldon, R. Mistrik, and T. R. Croley. Determination of ion structures in structurally related compounds using precursor ion fingerprinting. *J Am Soc Mass Spectrom*, 20(3):370–376, 2009. 18

[88] Y. Shinbo, Y. Nakamura, M. Altaf-Ul-Amin, H. Asahi, K. Kurokawa, M. Arita, K. Saito, D. Ohta, D. Shibata, and S. Kanaya. KNApSAcK: A comprehensive species-metabolite relationship database. In K. Saito, R. A. Dixon, and L. Willmitzer, editors, *Plant Metabolomics*, volume 57 of *Biotechnology in Agriculture and Forestry*, pages 165–181. Springer-Verlag, 2006. 7

[89] F. Sikora. An (almost complete) state of the art around the graph motif problem. Technical report, Université Paris-Est, France, 2010. Available from `http://www-igm.univ-mlv.fr/~fsikora/pub/GraphMotif-Resume.pdf`. 28, 72

[90] F. Sikora. *Aspects algorithmiques de la comparaison d'éléments biologiques*. PhD thesis, Université Paris-Est, 2011. 28

[91] C. A. Smith, G. O'Maille, E. J. Want, C. Qin, S. A. Trauger, T. R. Brandon, D. E. Custodio, R. Abagyan, and G. Siuzdak. METLIN: A metabolite mass spectral database. *Ther Drug Monit*, 27(6):747–751, 2005. 7, 16

[92] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958. 55

[93] C. M. Statham, R. Crowden, and J. Harborne. Biochemical genetics of pigmentation in pisumsativum. *Phytochemistry*, 11(3):1083–1088, 1972. 1

[94] T. Steijger. Automated annotation of fragmentation patterns. Diplomarbeit, Friedrich-Schiller-Universität Jena, 2009. 74

[95] S. E. Stein and D. R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *J Am Soc Mass Spectrom*, 5(9):859–866, 1994. 16, 65

[96] C. Steinbeck, C. Hoppe, S. Kuhn, M. Floris, R. Guha, and E. L. Willighagen. Recent developments of the Chemistry Development Kit (CDK) - an open-source Java library for chemo- and bioinformatics. *Curr Pharm Des*, 12(17):2111–2120, 2006. 62

[97] M. Sud, E. Fahy, D. Cotter, A. Brown, E. A. Dennis, C. K. Glass, A. H. Merrill, R. C. Murphy, C. R. H. Raetz, D. W. Russell, and S. Subramaniam. Lmsd: Lipid maps structure database. *Nucleic Acids Res*, 35(Database issue):D527–D532, Jan 2007. 8

[98] U.S. Department of Commerce. *NIST/EPA/NIH Mass Spectral Library 2011*. John Wiley & Sons, 2011. 17

[99] K. Varmuza and W. Werther. Mass spectral classifiers for supporting systematic structure elucidation. *J Chem Inf Comp Sci*, 36(2):323–333, 1996. 19

[100] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant. PubChem: A public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res*, 37(Web Server issue):W623–W633, 2009. 8, 18, 62

[101] J. Watrous, P. Roach, T. Alexandrov, B. S. Heath, J. Y. Yang, R. D. Kersten, M. van der Voort, K. Pogliano, H. Gross, J. M. Raaijmakers, B. S. Moore, J. Laskin, N. Bandeira, and P. C. Dorrestein. Mass spectral molecular networking of living microbial colonies. *Proc Natl Acad Sci U S A*, 109(26):E1743–E1752, 2012. 81

[102] W. Weckwerth. *Metabolomics — Methods and Protocols*. Humana Press, 2007. 5

[103] S. Wernicke and F. Rasche. FANMOD: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006. ix

[104] S. Wernicke and F. Rasche. Simple and fast alignment of metabolic pathways by exploiting local diversity. In *Proc. of Asia-Pacific Bioinformatic Conference (APBC 2007)*, Advances in Bioinformatics and Computational Biology, pages 353–362. Imperial College Press, Jan. 2007. ix

[105] S. Wernicke and F. Rasche. Simple and fast alignment of metabolic pathways by exploiting local diversity. *Bioinformatics*, 23(15):1978–1985, 2007. ix

[106] C. Whitehouse, R. Dreyer, M. Yamashita, and J. Fenn. Electrospray interface for liquid chromatographs and mass spectrometers. *Anal Chem*, 57:675–679, 1985. 9

[107] D. S. Wishart, C. Knox, A. C. Guo, R. Eisner, N. Young, B. Gautam, D. D. Hau, N. Psychogios, E. Dong, S. Bouatra, R. Mandal, I. Sinelnikov, J. Xia, L. Jia, J. A. Cruz, E. Lim, C. A. Sobsey, S. Shrivastava, P. Huang, P. Liu, L. Fang, J. Peng, R. Fradette, D. Cheng, D. Tzur, M. Clements, A. Lewis, A. D. Souza, A. Zuniga, M. Dawe, Y. Xiong, D. Clive, R. Greiner, A. Nazyrova, R. Shaykhutdinov, L. Li, H. J. Vogel, and I. Forsythe. HMDB: A knowledgebase for the human metabolome. *Nucleic Acids Res*, 37:D603–D610, 2009. 7, 16

[108] S. Wolf, S. Schmidt, M. Müller-Hannemann, and S. Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinformatics*, 11:148, 2010. 19, 71, 81

[109] K. Xu and W. Li. Many hard examples in exact phase transitions. *Theor Comput Sci*, 355(3):291–302, 2006. 31

[110] C. Yang, Z. He, and W. Yu. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. *BMC Bioinformatics*, 10(4), 2009. 15

[111] R. Zubarev and M. Mann. On the proper use of mass accuracy in proteomics. *Mol Cell Proteomics*, 6(3):377–381, 2007. 23, 65

A Datasets for Fragmentation Tree Evaluation

Compound name	*m/z*[a]	molecular formula[b]	measured peaks[c]	rank using isotopes[d]	rank using fragmentation[d]	rank combined[d]
Adenosine	268,092	C10H13N5O4	4	1	1	1
Anisic acid	153,055	C8H8O3	2	1	1	1
Apomorphine	268,134	C17H17NO2	10	1	2	1
Armentoflavone	539,098	C30H18O10	18	1	19	1
Berberine	335.116[e]	C20H17NO4$^+$	11	1	4	1
Bergapten	217.050	C12H8O4	73	1	1	1
Bicuculline	368,113	C20H17NO6	55	1	4	1
Biochanin A	285.076	C16H12O5	74	2	2	1
Chelidonine	354.134	C20H19NO5	69	1	2	1
Cinchonine	295.181	C19H22N2O	29	3	1	1
Emetine	481.307	C29H40N2O4	62	1	5	1
(-)-Epicatechine	291.087	C15H14O6	11	2	1	1
Erythromycin	734.469	C37H67NO13	2	2	18	1
Genistein	271.061	C15H10O5	36	1	1	1
Harmane	183.092	C12H10N2	4	1	1	1
IAA-Val	275.140	C15H18N2O3	6	1	1	1
Indol-3-carboxylic acid	162.056	C9H7NO2	3	1	1	1
Kaempherol	287.056	C15H10O6	47	1	1	1
Kinetin	216.089	C10H9N5O	8	1	2	1
Laudanosin	358.202	C21H27NO4	7	1	1	1
Methylumbelliferrylglucoronide	353.087	C16H16O9	4	1	1	1
(S,R)-Noscapine	414.155	C22H23NO7	1	2	3	1
Phenylalanine	166.087	C9H11NO2	5	1	1	1
Phlorizin	437.145	C21H24O10	10	1	3	1
Quercetin	303.050	C15H10O7	45	1	1	1
Reserpine	609.281	C33H40N2O9	31	1	7	1
Resveratrol	229.086	C14H12O3	20	1	1	1
Rotenone	395.149	C23H22O6	83	2	5	1
Rutine	611.161	C27H30O16	3	1	30	1
Safranin	315.161	C20H18N4	22	1	3	1
Salsolinol	180.102	C10H13NO2	5	1	1	1
Sinapine	310.166[e]	C16H24NO5$^+$	5	1	1	1
Tetrahydropapaveroline	288.124	C16H17NO4	8	1	2	1
3,4,5-Trimethoxycinnamic acid	239.092	C12H14O5	9	1	1	1
Tryptophan	205.098	C11H12N2O2	2	1	2	1
Vitexin-2-O-rhamnoside	579.171	C27H30O14	15	1	58	1
Xanthohumol	355.155	C21H22O5	8	2	1	1

Table A.1: Molecular formula identification for the Orbitrap dataset: Compound, [a]m/z value for [M+H]+ adduct precursor; [b]molecular formula of the compounds; [c]number of peaks in the merged spectra; [d]rank of molecular formula identification using isotope patterns, using fragmentation patterns, and combined identification; [e]Value for M+, as quaternary nitrogen in the compound.

Compound name	*m/z*[a]	molecular formula[b]	collision energies[c]	measured peaks[d]	rank using isotopes[e]	rank using fragmentation[e]	rank combined[e]
3-(4-Hexosyloxyphenyl)propanoyl choline	414.214[f]	C20H32NO8+	25, 40, 55	5	1	1	1
4-Coumaroyl choline	250.145[f]	C14H20NO3+	15, 25, 40	5	1	1	1
4-Hexosylferuloyl choline	442.209[f]	C21H32NO9+	15, 25, 40, 55	7	1	3	1
4-Hexosyloxybenzoyl choline	386.182[f]	C18H28NO8+	15, 25, 40, 55, 90	7	4	1	1
4-Hexosyloxycinnamoyl choline	412.198[f]	C20H30NO8+	25, 40, 55	6	3	2	1
4-Hexosylvanilloyl choline	416.193[f]	C19H30NO9+	15, 25, 40, 55, 70	5	7	3	1
4-Hydroxybenzoyl choline	224.130[f]	C12H18NO3+	15, 25, 40, 55	5	1	1	1
5-Hydroxyferuloyl choline	296.151[f]	C15H22NO5+	15, 25, 40, 55	13	2	5	1
6-Aminocapronic acid	132103.000	C6H13NO2	15, 20, 30, 40	29	1	1	1
Acetyl choline	146.119[f]	C7H16NO2+	10, 20,30	4	1	1	1
Alanine	90.056	C3H7NO2	10, 15, 20	2	1	1	1
Arginine	175.120	C6H14N4O2	20, 25, 30	17	1	1	1
Asparagine	133.062	C4H8N2O3	10, 15, 20, 30, 40	26	1	1	1
Aspartic acid	134.046	C4H7NO4	10, 15, 20, 30	13	3	1	1
Benzoyl choline	207.126[f]	C12H18NO2	15, 25, 40, 55	4	1	1	1
Cafeoyl choline	266.140[f]	C14H20NO4+	15, 25, 40, 55	10	1	1	1
Cinnamoyl choline	234.150[f]	C14H20NO2+	15, 25, 40, 55	4	1	1	1
Citrulline	176.104	C6H13N3O3	10, 15, 20, 25, 30	25	1	1	1
Cysteine	122.028	C3H8NO2S	10, 15, 20, 30	10	1	1	1
Cystine	241.033	C6H12N2O4S2	10, 15, 20, 30, 40	55	1	1	1
Dopamine	154.088	C8H11NO2	10, 20, 30, 40, 50	19	1	1	1
Feruloyl choline	280.156[f]	C15H22NO4+	15, 25, 40	9	1	3	1
Glutamic acid	148.062	C5H9NO4	10, 15, 20, 30	8	2	1	1
Glutamine	147.078	C5H10N2O3	10, 15, 20, 30	10	1	1	1
Histidine	156.078	C6H9N3O2	15, 25, 35, 45	17	1	1	1
Isoleucine	132.103	C6H13NO2	10, 15, 25, 40,	18	1	1	1
Leucine	132.103	C6H13NO2	15, 25, 40	19	1	1	1
Methionine	150.060	C5H11NO2S	10, 15, 20, 30	13	1	1	1
Nicotinic acid choline ester	209.130[f]	C11H17N2O2+	15, 25, 40, 55	4	1	1	1
Phenylalanine	166.088	C9H11NO2	15, 25, 40	15	1	1	1
Proline	116.072	C5H9NO2	10, 15, 55	9	1	1	1
Serine	106.051	C3H7NO3	10, 15, 20, 30	7	1	1	1
Sinapoyl choline	310.166[f]	C16H24NO5+	15, 25, 40	6	2	1	1
Spermidine	146.167	C7H19N3	15, 25, 35, 45	21	1	1	1
Spermine	203.224	C10H26N4	15, 25, 35, 45	13	1	1	1
Syringoyl choline	284.151[f]	C14H22NO5+	15, 25, 40, 55	17	2	1	1
Threonine	120.067	C4H9NO3	10, 15, 20, 30	9	1	1	1
Tryptophane	205.099	C11H12N2O2	15, 25, 40, 55	33	1	1	1
Tyramine	138.093	C8H12NO	15, 20, 30, 40, 50	21	1	1	1
Tyrosine	182.083	C9H11NO3	10, 15, 25, 30, 40	25	1	1	1
Valine	118.088	C5H11NO2	10, 25, 40, 55	15	1	1	1
Vanilloyl choline	254.140[f]	C13H20NO4+	15, 25, 40, 55	10	1	1	1

Table A.2: Molecular formula identification for the QSTAR dataset: Compound, [a]m/z value for [M+H]+ adduct precursor; [b]molecular formula of the compounds; [c]collision energies at which spectra have been recorded in eV; [d]number of peaks in the merged spectra; [e]rank of molecular formula identification using isotope patterns, using fragmentation patterns, and combined identification; [f]Value for M+.

Compound	PubChem ID	molecular formula	monoisotopic mass	measured peaks
6a-Methylprednisolone	4159	$C_{22}H_{30}O_5$	374.209	192
Acepromazine	6077	$C_{19}H_{22}N_2OS$	326.145	44
Acetophenazine	441185	$C_{23}H_{29}N_3O_2S$	411.198	47
Adenosine Diphosphate	197	$C_{10}H_{15}N_5O_{10}P_2$	427.029	16
Adiphenine	2031	$C_{20}H_{25}NO_2$	311.189	15
Albuterol	2083	$C_{13}H_{21}NO_3$	239.152	45
Alfentanil	51263	$C_{21}H_{32}N_6O_3$	416.254	60
Amfenac	2136	$C_{15}H_{13}NO_3$	255.090	59
Aminophylline	2153	$C_7H_8N_4O_2$	180.065	36
Ampicillin	2174	$C_{16}H_{19}N_3O_4S$	349.110	58
Anileridine	8944	$C_{22}H_{28}N_2O_2$	352.215	16
Antipyrine	2206	$C_{11}H_{12}N_2O$	188.095	71
Antipyrine-4-amino	2151	$C_{11}H_{13}N_3O$	203.106	57
Apomorphine	2215	$C_{17}H_{17}NO_2$	267.126	16
Apramycin	71428	$C_{21}H_{41}N_5O_{11}$	539.280	105
Betaxolol	2369	$C_{18}H_{29}NO_3$	307.215	95
Boldenone Undecylenate	25702	$C_{30}H_{44}O_3$	452.329	45
Bumetanide	2471	$C_{17}H_{20}N_2O_5S$	364.109	73
Buprenorphine	2476	$C_{29}H_{41}NO_4$	467.304	241
Buspirone	2477	$C_{21}H_{31}N_5O_2$	385.248	39
Cholesterol	304	$C_{27}H_{46}O$	386.355	25
Cromolyn	2882	$C_{23}H_{16}O_{11}$	468.069	51
Cymarin	539061	$C_{30}H_{44}O_9$	548.299	114
Daunorubicin	2958	$C_{27}H_{29}NO_{10}$	527.179	35
Dextromethorphan	3008	$C_{18}H_{25}NO$	271.194	62
Dihydroergotamine	3066	$C_{33}H_{37}N_5O_5$	583.279	51
Dimefline	3078	$C_{20}H_{21}NO_3$	323.152	16
Diphenoxylate	13505	$C_{30}H_{32}N_2O_2$	452.246	91
Dobutamine	36811	$C_{18}H_{23}NO_3$	301.168	16
Doxorubicin	1691	$C_{27}H_{29}NO_{11}$	543.174	72
Drofenine	3166	$C_{20}H_{31}NO_2$	317.235	19
Enalapril	3222	$C_{20}H_{28}N_2O_5$	376.200	22
Enalaprilat	5362033	$C_{18}H_{24}N_2O_5$	348.169	21
Ephedrine	5032	$C_{10}H_{15}NO$	165.115	30
Ergocristine	98255	$C_{35}H_{39}N_5O_5$	609.295	50
Ergoloid Mesylate	592735	$C_{33}H_{45}N_5O_5$	591.342	16
Etamiphylline	28329	$C_{13}H_{21}N_5O_2$	279.170	62
Etodolac	3308	$C_{17}H_{21}NO_3$	287.152	66
Fenbendazole	3334	$C_{15}H_{13}N_3O_2S$	299.073	38
Fenoterol	3343	$C_{17}H_{21}NO_4$	303.147	15
Folic Acid	3405	$C_{19}H_{19}N_7O_6$	441.140	19
Gallamine	3450	$C_{30}H_{60}N_3O_3$	510.463	24
Gingerol	3473	$C_{17}H_{26}O_4$	294.183	34
Hematoporphyrin I	11103	$C_{34}H_{38}N_4O_6$	598.279	79
Hydrocortisone	3640	$C_{21}H_{30}O_5$	362.209	174
Hydroxybutorphanol	3064246	$C_{21}H_{29}NO_3$	343.215	101
Hydroxyphenethylamine	5610	$C_8H_{11}NO$	137.084	26
Isoxsuprine	3783	$C_{18}H_{23}NO_3$	301.168	51
Ketorolac	3826	$C_{15}H_{13}NO_3$	255.090	18
Leucine Enkephalin	3903	$C_{28}H_{37}N_5O_7$	555.269	53

Table A.3: Compounds of the Micromass QTOF dataset: Name, PubChem ID, molecular formula, mass, and number of measured peaks.

Compound	PubChem ID	molecular formula	monoisotopic mass	measured peaks
Mebeverine	4031	$C_{25}H_{35}NO_5$	429.252	12
Mefenamic Acid	4044	$C_{15}H_{15}NO_2$	241.110	28
Meprobamate	4064	$C_9H_{18}N_2O_4$	218.127	13
Methionine Enkephalin	42785	$C_{27}H_{35}N_5O_7S$	573.226	62
Methotrexate	4112	$C_{20}H_{22}N_8O_5$	454.171	15
Methylergonovine	4140	$C_{20}H_{25}N_3O_2$	339.195	53
Morphine-3-Glucuronide	4318740	$C_{23}H_{27}NO_9$	461.169	56
Naltrexone	4428	$C_{20}H_{23}NO_4$	341.163	138
Nandrolone	9904	$C_{18}H_{26}O_2$	274.193	80
Nimesulide	4495	$C_{13}H_{12}N_2O_5S$	308.047	42
Norpropoxyphene	18804	$C_{21}H_{27}NO_2$	325.204	10
Noscapine	4544	$C_{22}H_{23}NO_7$	413.147	165
Ormetoprim	23418	$C_{14}H_{18}N_4O_2$	274.143	94
Oxaprozin	4614	$C_{18}H_{15}NO_3$	293.105	23
Oxybutynin	4634	$C_{22}H_{31}NO_3$	357.230	64
Oxycodone	4635	$C_{18}H_{21}NO_4$	315.147	146
Oxytetracycline	5280972	$C_{22}H_{24}N_2O_9$	460.148	152
Perindopril	107807	$C_{19}H_{32}N_2O_5$	368.231	17
Piperacetazine	19675	$C_{24}H_{30}N_2O_2S$	410.203	22
Poldine	11018	$C_{21}H_{26}NO_3$	340.191	34
Prazosin	4893	$C_{19}H_{21}N_5O_4$	383.159	71
Prednisolone	4894	$C_{21}H_{28}O_5$	360.194	172
Prednisolone Tebutate	4898	$C_{27}H_{38}O_6$	458.267	161
Prednisone	4900	$C_{21}H_{26}O_5$	358.178	194
Prolintane	14592	$C_{15}H_{23}N$	217.183	8
Pyrilamine	4992	$C_{17}H_{23}N_3O$	285.184	11
Remifentanil	60815	$C_{20}H_{28}N_2O_5$	376.200	55
Reserpine	5052	$C_{33}H_{40}N_2O_9$	608.273	122
Rolitetracycline	6420073	$C_{27}H_{33}N_3O_8$	527.227	17
Salmeterol	5152	$C_{25}H_{37}NO_4$	415.272	71
Spectinomycin	2021	$C_{14}H_{24}N_2O_7$	332.158	122
Streptomycin	19649	$C_{21}H_{39}N_7O_{12}$	581.266	147
Strychnine	5304	$C_{21}H_{22}N_2O_2$	334.168	148
Strychnine N-oxide	73393	$C_{21}H_{22}N_2O_3$	350.163	181
Sufentanil	41693	$C_{22}H_{30}N_2O_2S$	386.203	34
Sulfadimethoxine	5323	$C_{12}H_{14}N_4O_4S$	310.074	54
Sulfasalazine	5384001	$C_{18}H_{14}N_4O_5S$	398.068	76
Taurocholate	8959	$C_{26}H_{45}NO_7S$	515.292	134
Tenoxicam	5282194	$C_{13}H_{11}N_3O_4S_2$	337.019	30
Terbutaline	5403	$C_{12}H_{19}NO_3$	225.136	35
Terfenadine	5405	$C_{32}H_{41}NO_2$	471.314	101
Testosterone Propionate	5701990	$C_{22}H_{32}O_3$	344.235	69
Tetracaine	5411	$C_{15}H_{24}N_2O_2$	264.184	30
Tetracycline	5353990	$C_{22}H_{24}N_2O_8$	444.153	149
Tetramisole	3913	$C_{11}H_{12}N_2S$	204.072	42
Theobromine	5429	$C_7H_8N_4O_2$	180.065	45
Thiethylperazine	5440	$C_{22}H_{29}N_3S_2$	399.180	33
Thioridazine	5452	$C_{21}H_{26}N_2S_2$	370.154	25
Thiothixene	941651	$C_{23}H_{29}N_3O_2S_2$	443.170	89
Thonzide	5456	$C_{32}H_{55}N_4O$	511.438	6
Tripelennamine	5587	$C_{16}H_{21}N_3$	255.174	11
Vecuronium	39765	$C_{34}H_{57}N_2O_4$	557.432	58

Table A.3: Compound of the Micromass QTOF dataset (continued)

	annotated	evaluation by experts			
Compound name	NL[a]	correct[b]	unclear[b]	wrong[b]	note
Adenosine	1	1	0	0	
Anisic acid	1	1	0	0	
Apomorphine	6	5	1	0	
Armentoflavone	13	11	2	0	
Berberine	3	3	0	0	radical loss, pull-up
Bergapten	12	8	3	1	
Bicuculline	34	22	9	3	
Biochanin A	36	27	3	6	
Chelidonine	19	15	4	0	radical loss, pull-up
Cinchonine	23	18	4	1	
Emetine	36	31	2	3	
(-)-Epicatechine	6	5	1	0	pull-up
Erythromycin	1	1	0	0	
Genistein	31	23	2	6	radical loss, pull-up
Harmane	2	2	0	0	
IAA-Val	3	2	0	1	pull-up
Indol-3-carboxylic acid	2	2	0	0	
Kaempherol	38	24	5	9	
Kinetin	7	5	0	2	pull-up
Laudanosin	6	6	0	0	pull-up
Methylumbelliferrylglucoronide	1	1	0	0	
(S,R)-Noscapine	7	5	2	0	radical loss
Phenylalanine	3	3	0	0	
Phlorizin	7	7	0	0	
Quercetin	36	26	3	7	
Reserpine	19	13	6	0	pull-up
Resveratrol	19	15	0	4	
Rotenone	45	34	8	3	
Rutine	2	2	0	0	
Safranin	7	5	0	2	
Salsolinol	4	4	0	0	
Sinapine	1	1	0	0	
Tetrahydropapaveroline	6	6	0	0	pull-up
3,4,5-Trimethoxycinnamic acid	5	4	1	0	
Tryptophan	1	1	0	0	
Vitexin-2-O-rhamnoside	12	11	1	0	
Xanthohumol	3	2	0	1	pull-up
	458	352	57	49	

Table A.4: Results for the Orbitrap dataset, expert evaluation: Compound, [a]number of annotated neutral losses (edges) in hypothetical fragmentation trees, [b]number of neutral losses marked “correct”, “unclear”, or “wrong” by an MS expert. “Radical loss” denotes that MS experts have identified a radical loss in the MS data not annotated by the program, and “pull-up” indicates that neutral losses may be inserted too deep in the fragmentation tree.

	annotated	evaluation by experts			
Compound name	**NI[a]**	**correct[b]**	**unclear[b]**	**wrong[b]**	**note**
3-(4-Hexosyloxyphenyl)propanoyl choline	4	4	0	0	
4-Coumaroyl choline	4	4	0	0	
4-Hexosylferuloyl choline	4	4	0	0	
4-Hexosyloxybenzoyl choline	5	5	0	0	
4-Hexosyloxycinnamoyl choline	4	4	0	0	
4-Hexosylvanilloyl choline	3	3	0	0	
4-Hydroxybenzoyl choline	4	4	0	0	
5-Hydroxyferuloyl choline	11	8	0	3	radical loss
6-Aminocapronic acid	17	13	4	0	
Acetyl choline	3	3	0	0	
Alanine	1	1	0	0	
Arginine	14	13	1	0	
Asparagine	20	14	3	3	
Aspartic acid	7	7	0	0	
Benzoyl choline	3	3	0	0	
Cafeoyl choline	8	7	0	1	
Cinnamoyl choline	3	3	0	0	
Citrulline	11	11	0	0	
Cysteine	6	6	0	0	
Cystine	16	8	8	0	
Dopamine	14	10	4	0	
Feruloyl choline	5	5	0	0	
Glutamic acid	4	4	0	0	
Glutamine	8	8	0	0	
Histidine	13	12	0	1	
Isoleucine	9	7	2	0	
Leucine	10	8	2	0	
Methionine	10	9	1	0	
Nicotinic acid choline ester	3	3	0	0	
Phenylalanine	12	8	4	0	
Proline	5	5	0	0	
Serine	4	4	0	0	
Sinapoyl choline	5	5	0	0	
Spermidine	13	10	3	0	
Spermine	12	7	4	1	pull-up
Syringoyl choline	9	6	3	0	
Threonine	5	5	0	0	
Tryptophane	22	15	4	3	radical loss
Tyramine	10	7	3	0	
Tyrosine	13	11	1	1	
Valine	10	7	3	0	
Vanilloyl choline	6	5	1	0	
	350	286	51	13	

Table A.5: Results for the QSTAR dataset, expert evaluation: Compound, [a]number of annotated neutral losses (edges) in hypothetical fragmentation trees, [b]number of neutral losses marked “correct”, “unclear”, or “wrong” by an MS expert. “Radical loss” denotes that MS experts have identified a radical loss in the MS data not annotated by the program, and “pull-up” indicates that neutral losses may be inserted too deep in the fragmentation tree.

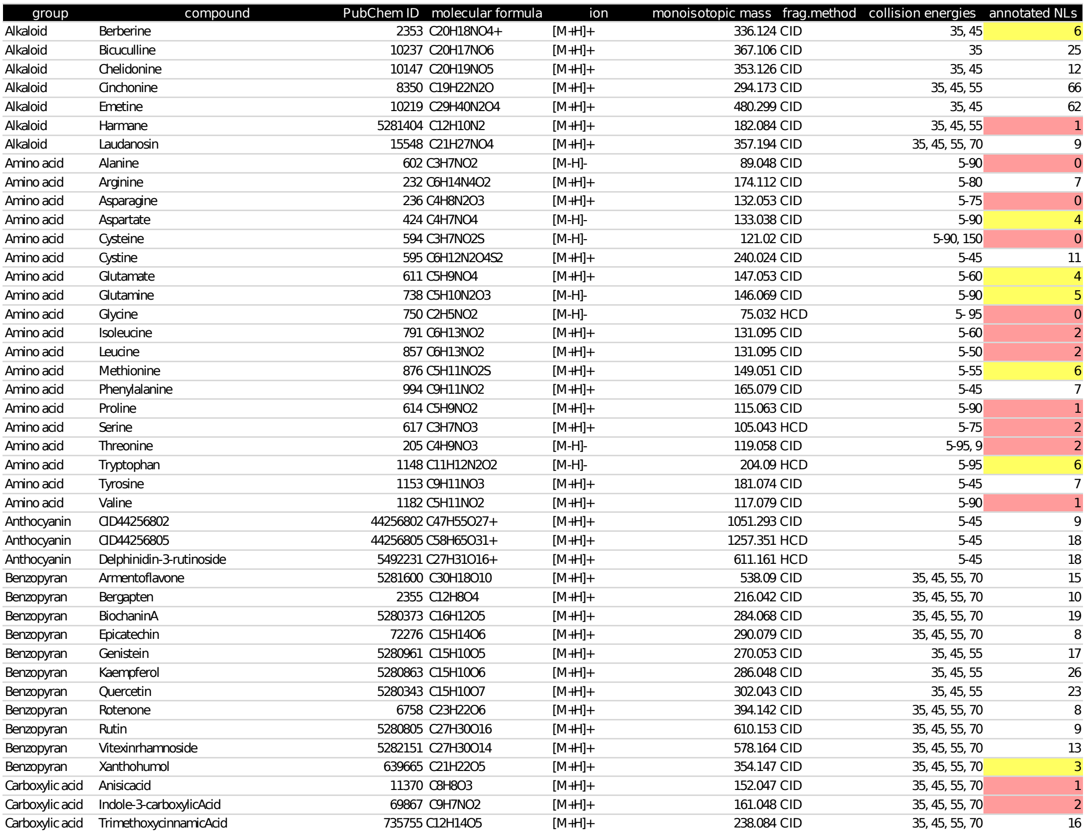
	Mass Frontier prediction			our method	common	non-trivial	non-matching
Compound	sensitivity	specificity	F-value	sensitivity	peaks	common peaks	explanations
6a-Methylprednisolone	0.135	0.268	0.180	0.479	17	0	0
Acepromazine	0.182	0.421	0.254	0.591	8	4	0
Acetophenazine	0.404	0.250	0.309	0.660	18	14	1
Adenosine Diphosphate	0.313	0.238	0.270	0.750	5	5	1
Adiphenine	0.333	0.076	0.123	0.733	5	0	0
Albuterol	0.133	0.133	0.133	0.644	6	1	0
Alfentanil	0.350	0.119	0.178	0.783	19	11	0
Amfenac	0.119	0.350	0.177	0.695	7	0	0
Aminophylline	0.139	0.357	0.200	0.556	5	3	0
Ampicillin	0.328	0.066	0.110	0.845	18	14	3
Anileridine	0.188	0.046	0.074	0.813	3	0	0
Antipyrine	0.085	0.500	0.145	0.592	6	0	0
Antipyrine-4-amino	0.105	0.207	0.140	0.702	6	0	0
Apomorphine	0.063	0.077	0.069	0.438	1	0	0
Apramycin	0.410	0.139	0.207	0.857	42	40	4
Betaxolol	0.400	0.380	0.390	0.674	32	4	0
Boldenone Undecylenate	0.311	0.132	0.185	0.867	11	0	0
Bumetanide	0.068	0.135	0.091	0.808	4	1	0
Buprenorphine	0.012	0.030	0.018	0.598	3	1	0
Buspirone	0.436	0.142	0.214	0.846	14	11	0
Cholesterol	0.120	0.088	0.102	0.480	2	0	0
Cromolyn	0.333	0.293	0.312	0.824	17	1	0
Cymarin	0.219	0.116	0.152	0.649	16	5	0
Daunorubicin	0.200	0.035	0.060	0.943	7	3	0
Dextromethorphan	0.097	0.222	0.135	0.645	6	0	0
Dihydroergotamine	0.216	0.039	0.066	0.922	11	8	0
Dimefline	0.188	0.136	0.158	0.625	1	0	0
Diphenoxylate	0.176	0.229	0.199	0.593	16	1	0
Dobutamine	0.500	0.178	0.262	0.938	8	1	0
Doxorubicin	0.208	0.068	0.103	0.972	15	7	0
Drofenine	0.474	0.143	0.220	0.947	9	0	0
Enalapril	0.636	0.046	0.085	0.909	14	5	0
Enalaprilat	0.619	0.053	0.098	0.952	13	4	0
Ephedrine	0.267	0.348	0.302	0.700	8	0	0
Ergocristine	0.340	0.059	0.101	0.960	17	15	4
Ergoloid Mesylate	0.250	0.011	0.022	0.938	4	3	1
Etamiphylline	0.194	0.203	0.198	0.726	12	3	0
Etodolac	0.197	0.151	0.171	0.773	13	7	0
Fenbendazole	0.053	0.222	0.085	0.632	2	0	0
Fenoterol	0.467	0.117	0.187	0.867	7	1	0
Folic Acid	0.368	0.040	0.073	1.000	7	7	1
Gallamine	0.167	0.060	0.088	0.625	4	3	0
Gingerol	0.265	0.180	0.214	0.794	9	0	0
Hematoporphyrin I	0.038	0.056	0.045	0.949	3	2	0
Hydrocortisone	0.161	0.246	0.194	0.477	20	0	0
Hydroxybutorphanol	0.198	0.190	0.194	0.743	19	1	0
Hydroxyphenethylamine	0.077	0.400	0.129	0.615	2	0	0
Isoxsuprine	0.373	0.279	0.319	0.706	18	1	0
Ketorolac	0.278	0.125	0.172	0.722	4	0	0
Leucine Enkephalin	0.811	0.088	0.159	0.943	39	36	0

Table A.6: Results for the evaluation against MassFrontier based on the Micromass QTOF dataset: Compound sensitivity, specificity, and F-value of the Mass Frontier prediction; sensitivity of our prediction; number of common peaks, number of non-trivial common peaks, and number of non-matching explanations. Zero non-matching explanations indicate perfect agreement.

	Mass Frontier prediction			our method	common	non-trivial	non-matching
Compound	**sensitivity**	**specificity**	**F-value**	**sensitivity**	**peaks**	**common peaks**	**explanations**
Mebeverine	0.500	0.052	0.094	0.833	6	2	0
Mefenamic Acid	0.036	0.071	0.048	0.643	1	0	0
Meprobamate	0.154	0.250	0.190	1.000	2	0	0
Methionine Enkephalin	0.710	0.085	0.153	0.968	44	42	5
Methotrexate	0.200	0.024	0.000	0.800	3	3	3
Methylergonovine	0.302	0.131	0.183	0.698	16	0	0
Morphine-3-Glucuronide	0.071	0.033	0.045	0.732	4	4	0
Naltrexone	0.087	0.128	0.103	0.587	12	1	0
Nandrolone	0.225	0.419	0.293	0.650	13	0	0
Nimesulide	0.000	0.000	0.000	0.714	0	0	0
Norpropoxyphene	0.500	0.051	0.093	0.600	4	0	0
Noscapine	0.055	0.155	0.081	0.600	7	3	0
Ormetoprim	0.011	0.125	0.020	0.660	1	1	0
Oxaprozin	0.087	0.095	0.091	0.609	2	0	0
Oxybutynin	0.328	0.206	0.253	0.859	20	5	0
Oxycodone	0.068	0.169	0.098	0.541	9	1	0
Oxytetracycline	0.092	0.125	0.106	0.914	13	7	3
Perindopril	0.706	0.042	0.079	0.882	11	2	0
Piperacetazine	0.409	0.184	0.254	0.909	9	8	0
Poldine	0.118	0.125	0.121	0.471	4	0	0
Prazosin	0.169	0.375	0.233	0.915	12	12	0
Prednisolone	0.140	0.253	0.180	0.483	17	0	0
Prednisolone Tebutate	0.106	0.106	0.106	0.516	12	0	0
Prednisone	0.124	0.235	0.162	0.500	20	0	0
Prolintane	0.500	0.129	0.205	0.875	4	0	0
Pyrilamine	0.182	0.105	0.133	0.909	2	1	0
Remifentanil	0.400	0.125	0.190	0.891	22	6	0
Reserpine	0.164	0.096	0.121	0.877	19	19	0
Rolitetracycline	0.294	0.029	0.053	1.000	5	5	0
Salmeterol	0.282	0.171	0.213	0.789	19	2	0
Spectinomycin	0.393	0.251	0.307	0.672	33	18	0
Streptomycin	0.184	0.088	0.119	0.755	25	24	0
Strychnine	0.014	0.051	0.021	0.520	2	0	0
Strychnine N-oxide	0.011	0.069	0.019	0.630	2	0	0
Sufentanil	0.441	0.097	0.160	0.882	14	5	0
Sulfadimethoxine	0.037	0.333	0.067	0.778	1	1	0
Sulfasalazine	0.053	0.364	0.092	0.908	4	3	1
Taurocholate	0.060	0.052	0.055	0.806	7	3	0
Tenoxicam	0.233	0.318	0.269	0.900	7	7	0
Terbutaline	0.229	0.242	0.235	0.771	6	1	0
Terfenadine	0.129	0.171	0.147	0.653	13	0	0
Testosterone Propionate	0.232	0.213	0.222	0.826	14	0	0
Tetracaine	0.267	0.136	0.180	0.667	8	0	0
Tetracycline	0.060	0.074	0.066	0.799	8	5	1
Tetramisole	0.310	0.481	0.377	0.690	11	5	0
Theobromine	0.111	0.333	0.167	0.556	3	1	0
Thiethylperazine	0.212	0.175	0.192	0.576	8	7	0
Thioridazine	0.280	0.241	0.259	0.600	7	1	0
Thiothixene	0.213	0.613	0.317	0.764	18	10	0
Thonzide	0.333	0.080	0.129	0.667	2	1	0
Tripelennamine	0.364	0.222	0.276	0.727	4	0	0
Vecuronium	0.155	0.040	0.064	0.966	9	5	0

Table A.6: Results for the MassFrontier evaluation (continued)

B Datasets for Fragmentation Tree Alignment



group	compound	PubChem ID	molecular formula	ion	monoisotopic mass	frag.method	collision energies	annotated NLs
Alkaloid	Berberine	2353	C20H18NO4+	[M+H]+	336.124	CID	35, 45	6
Alkaloid	Bicuculline	10237	C20H17NO6	[M+H]+	367.106	CID	35	25
Alkaloid	Chelidonine	10147	C20H19NO5	[M+H]+	353.126	CID	35, 45	12
Alkaloid	Cinchonine	8350	C19H22N2O	[M+H]+	294.173	CID	35, 45, 55	66
Alkaloid	Emetine	10219	C29H40N2O4	[M+H]+	480.299	CID	35, 45	62
Alkaloid	Harmane	5281404	C12H10N2	[M+H]+	182.084	CID	35, 45, 55	1
Alkaloid	Laudanosin	15548	C21H27NO4	[M+H]+	357.194	CID	35, 45, 55, 70	9
Amino acid	Alanine	602	C3H7NO2	[M-H]-	89.048	CID	5-90	0
Amino acid	Arginine	232	C6H14N4O2	[M+H]+	174.112	CID	5-80	7
Amino acid	Asparagine	236	C4H8N2O3	[M+H]+	132.053	CID	5-75	0
Amino acid	Aspartate	424	C4H7NO4	[M-H]-	133.038	CID	5-90	4
Amino acid	Cysteine	594	C3H7NO2S	[M-H]-	121.02	CID	5-90, 150	0
Amino acid	Cystine	595	C6H12N2O4S2	[M+H]+	240.024	CID	5-45	11
Amino acid	Glutamate	611	C5H9NO4	[M+H]+	147.053	CID	5-60	4
Amino acid	Glutamine	738	C5H10N2O3	[M-H]-	146.069	CID	5-90	5
Amino acid	Glycine	750	C2H5NO2	[M-H]-	75.032	HCD	5- 95	0
Amino acid	Isoleucine	791	C6H13NO2	[M+H]+	131.095	CID	5-60	2
Amino acid	Leucine	857	C6H13NO2	[M+H]+	131.095	CID	5-50	2
Amino acid	Methionine	876	C5H11NO2S	[M+H]+	149.051	CID	5-55	6
Amino acid	Phenylalanine	994	C9H11NO2	[M+H]+	165.079	CID	5-45	7
Amino acid	Proline	614	C5H9NO2	[M+H]+	115.063	CID	5-90	1
Amino acid	Serine	617	C3H7NO3	[M+H]+	105.043	HCD	5-75	2
Amino acid	Threonine	205	C4H9NO3	[M-H]-	119.058	CID	5-95, 9	2
Amino acid	Tryptophan	1148	C11H12N2O2	[M-H]-	204.09	HCD	5-95	6
Amino acid	Tyrosine	1153	C9H11NO3	[M+H]+	181.074	CID	5-45	7
Amino acid	Valine	1182	C5H11NO2	[M+H]+	117.079	CID	5-90	1
Anthocyanin	CID44256802	44256802	C47H55O27+	[M+H]+	1051.293	CID	5-45	9
Anthocyanin	CID44256805	44256805	C58H65O31+	[M+H]+	1257.351	HCD	5-45	18
Anthocyanin	Delphinidin-3-rutinoside	5492231	C27H31O16+	[M+H]+	611.161	HCD	5-45	18
Benzopyran	Armentoflavone	5281600	C30H18O10	[M+H]+	538.09	CID	35, 45, 55, 70	15
Benzopyran	Bergapten	2355	C12H8O4	[M+H]+	216.042	CID	35, 45, 55, 70	10
Benzopyran	BiochaninA	5280373	C16H12O5	[M+H]+	284.068	CID	35, 45, 55, 70	19
Benzopyran	Epicatechin	72276	C15H14O6	[M+H]+	290.079	CID	35, 45, 55, 70	8
Benzopyran	Genistein	5280961	C15H10O5	[M+H]+	270.053	CID	35, 45, 55	17
Benzopyran	Kaempferol	5280863	C15H10O6	[M+H]+	286.048	CID	35, 45, 55	26
Benzopyran	Quercetin	5280343	C15H10O7	[M+H]+	302.043	CID	35, 45, 55	23
Benzopyran	Rotenone	6758	C23H22O6	[M+H]+	394.142	CID	35, 45, 55, 70	8
Benzopyran	Rutin	5280805	C27H30O16	[M+H]+	610.153	CID	35, 45, 55, 70	9
Benzopyran	Vitexinrhamnoside	5282151	C27H30O14	[M+H]+	578.164	CID	35, 45, 55, 70	13
Benzopyran	Xanthohumol	639665	C21H22O5	[M+H]+	354.147	CID	35, 45, 55, 70	3
Carboxylic acid	Anisicacid	11370	C8H8O3	[M+H]+	152.047	CID	35, 45, 55, 70	1
Carboxylic acid	Indole-3-carboxylicAcid	69867	C9H7NO2	[M+H]+	161.048	CID	35, 45, 55, 70	2
Carboxylic acid	TrimethoxycinnamicAcid	735755	C12H14O5	[M+H]+	238.084	CID	35, 45, 55, 70	16

Table B.1: Compound list for the Orbitrap dataset: Compound class, compound name, PubChem ID, molecular formula, ion type, monoisotopic mass (Da), fragmentation technique, collision energies, and number of annotated losses (NLs) in hypothetical fragmentation trees. Collision energies are given in electron volt for CID and arbitrary units for HCD fragmentation. If a range is given, we used a step size of 5 units within this range. Compounds with less than three (seven) annotated losses are colored red (yellow).

group	compound	PubChem ID	molecular formula	ion	monoisotopic mass	frag.method	collision energies	annotated NLs
Glucosinolate	3-Hydroxypropyl-Glucosinolate	25245521	C10H17NO10S2	[M-H]-	375.029	HCD	5-90	9
Glucosinolate	3-Methylthiopropyl-Glucosinolate	25244538	C11H19NO9S3	[M-H]-	405.022	HCD	5-90	13
Glucosinolate	4-Methoxy-3-indolylmethyl glucosinolate	656562	C17H20N2O10S2	[M-H]-	476.056	HCD	5-90	19
Glucosinolate	7-Methylthioheptyl glucosinolate	44237368	C15H27NO9S3	[M-H]-	461.085	HCD	5-90	18
Glucosinolate	8-Methylthiooctyl glucosinolate	44237373	C16H29NO9S3	[M-H]-	475.1	HCD	5, 15-55, 65-90	21
Glucosinolate	Glucoalyssin	656523	C13H25NO10S3	[M-H]-	451.064	HCD	5, 15-50, 60	4
Glucosinolate	Glucoerucin	656538	C12H21NO9S3	[M-H]-	419.038	HCD	5-90	19
Glucosinolate	Glucohirsutin	44237257	C16H29NO10S3	[M-H]-	491.095	HCD	5-90	24
Glucosinolate	Glucoibarin	44237203	C15H27NO10S3	[M-H]-	477.08	HCD	5-90	28
Glucosinolate	Glucoiberin	9548621	C11H19NO10S3	[M-H]-	421.017	HCD	55-90	30
Glucosinolate	Glucomalcommin	25244201	C17H21NO11S2	[M-H]-	479.056	HCD	5-90	25
Glucosinolate	Glucoraphanin	9548633	C12H21NO10S3	[M-H]-	435.033	HCD	5-90	8
Glucosinolate	Glucoraphenin	6443008	C12H21NO11S3	[M-H]-	451.028	HCD	5-90	16
Glucosinolate	Indolylmethyl glucosinolate	25244590	C16H18N2O9S2	[M-H]-	446.045	HCD	5-90	22
Lipid	DErySphinganine	91486	C18H39NO2	[M-H]-	301.298	CID	25	12
Lipid	DErySphingosine	5280335	C18H37NO2	[M+H]+	299.282	CID	10	1
Lipid	Phosphatidylcholine	129900	C25H54NO6P	[M+H]+	495.369	HCD	30	3
Lipid	Phosphatidylethanolamine	46891780	C39H74NO8P	[M-H]-	715.515	CID	20	6
Sugar	Cellobiose	294	C12H22O11	[M+H]+	342.116	HCD	4	10
Sugar	DP5		C30H52O26	[M+Na]+	828.275	HCD	45	16
Sugar	DP7		C42H72O36	[M+H]+	1152.38	HCD	12	17
Sugar	Fucose	17106	C6H12O5	[M+Na]+	164.068	CID	46	2
Sugar	Galactose	6036	C6H12O6	[M+NH4]+	180.063	HCD	12	4
Sugar	Gentiobiose	441422	C12H22O11	[M+Na]+	342.116	CID	20	6
Sugar	Lactose	6134	C12H22O11	[M+H]+	342.116	HCD	4	10
Sugar	Mannitol	6251	C6H14O6	[M+H]+	182.079	HCD	20	12
Sugar	Mannose	18950	C6H12O6	[M+H]+	180.063	CID	15	6
Sugar	Rhamnose	19233	C6H12O5	[M+Na]+	164.068	CID	46	2
Sugar	Sorbitol	5780	C6H14O6	[M+H]+	182.079	CID	20	14
Sugar	Trehalose	7427	C12H22O11	[M+Na]+	342.116	CID	20	2
Zeatin	Cis-Zeatin	449093	C10H13N5O	[M+H]+	219.112	CID	44	7
Zeatin	Cis-Zeatin-9-glucoside	9842892	C16H23N5O6	[M+H]+	381.165	CID	17	5
Zeatin	Cis-Zeatin-o-glucoside	25244165	C16H23N5O6	[M+H]+	381.165	CID	19	6
Zeatin	Cis-Zeatin-riboside	6440982	C15H21N5O5	[M+H]+	351.154	CID	11	4
Zeatin	Cis-Zeatin-riboside-O-glucoside	11713250	C21H31N5O10	[M+H]+	513.207	CID	20	4
Zeatin	D5-Cis-Zeatin-riboside	6440982	C15H21N5O5	[M+H]+	351.154	CID	15	15
Zeatin	D5-Trans-Zeatin	449093	C10D5H8N5O	[M+H]+	224.143	CID	15	8
Zeatin	D5-Trans-Zeatin-7-glucoside		C16D5H18N5O6	[M+H]+	386.196	CID	14	8
Zeatin	D5-Trans-Zeatin-9-glucoside	9842892	C16D5H18N5O6	[M+H]+	386.196	CID	14	10
Zeatin	D5-Trans-Zeatin-riboside	6440982	C15H21N5O5	[M+H]+	351.154	CID	13	8
Zeatin	D5-Trans-Zeatin-riboside-o-glucoside	11713250	C21H31N5O10	[M+H]+	513.207	CID	23	15
Zeatin	D6-isopentenyl-Adenine		C10D6H7N5	[M+H]+	209.155	CID	27	4
Zeatin	D6-isopentenyl-Adenine-7-glucoside	330023	C16D6H17N5O5	[M+H]+	371.208	CID	30	1
Zeatin	D6-isopentenyl-Adenine-9-glucoside	23197432	C16D6H17N5O5	[M+H]+	371.208	CID	15	6
Zeatin	D6-isopentenyl-Adenosine	24405	C15D6H15N5O4	[M+H]+	341.197	CID	22	4
Zeatin	Isopentenyl-Adenine		C10H13N5	[M+H]+	203.117	CID	35	2
Zeatin	Isopentenyl-Adenine-7-glucoside	330023	C16H23N5O5	[M+H]+	365.17	CID	14	4
Zeatin	Isopentenyl-Adenine-9-glucoside	23197432	C16H23N5O5	[M+H]+	365.17	CID	14	5
Zeatin	Isopentenyl-Adenosine	24405	C15H21N5O4	[M+H]+	335.159	CID	13	3
Zeatin	Trans-Zeatin	449093	C10H13N5O	[M+H]+	219.112	CID	47	6
Zeatin	Trans-Zeatin-9-glucoside	9842892	C16H23N5O6	[M+H]+	381.165	CID	28	5
Zeatin	Trans-Zeatin-o-glucoside	25244165	C16H23N5O6	[M+H]+	381.165	CID	28	9
Zeatin	Trans-Zeatin-riboside	6440982	C15H21N5O5	[M+H]+	351.154	CID	24	1
Zeatin	Trans-Zeatin-riboside-O-glucoside	11713250	C21H31N5O10	[M+H]+	513.207	CID	12	5

Table B.1: Compound list for the Orbitrap dataset (continued)

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLs
Aldehyde	1-Methoxy-3-carbaldehyde	398554	C10H9NO2	175.063	Ramp 5-60	2
Aldehyde	4-Hydroxy-3-methoxycinnamaldehyde	5280536	C10H10O3	178.063	Ramp 5-60	2
Aldehyde	Indole-3-acetaldehyde	800	C10H9NO	159.068	Ramp 5-60	2
Aldehyde	Indole-3-carboxyaldehyde	10256	C9H7NO	145.053	30, Ramp 5-60	6
Aldehyde	Syringaldehyde	8655	C9H10O4	182.058	Ramp 5-60	3
Amino acid	1-Aminocyclopropane-1-carboxylic_acid	535	C4H7NO2	101.048	Ramp 5-60	0
Amino acid	2-Aminoisobutyric_acid	6119	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	3-Hydroxy-DL-kynurenine	89	C10H12N2O4	224.08	Ramp 5-60	6
Amino acid	3-Methyl-L-histidine	64969	C7H11N3O2	169.085	Ramp 5-60	3
Amino acid	5-Aminovaleric_acid	138	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	Alpha-Methyl-DL-histidine	4396761	C7H11N3O2	169.085	Ramp 5-60	7
Amino acid	Alpha-Methyl-DL-serine	439656	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	Carbamoyl-DL-aspartic_acid	93072	C5H8N2O5	176.043	Ramp 5-60	3
Amino acid	Creatine	586	C4H9N3O2	131.069	Ramp 5-60	1
Amino acid	Cystathionine	834	C7H14N2O4S	222.067	Ramp 5-60	2
Amino acid	D-Alloisoleucine	94206	C6H13NO2	131.095	Ramp 5-60	0
Amino acid	D-beta-homophenylalanine	102530	C10H13NO2	179.095	Ramp 5-60	2
Amino acid	D-beta-homoserine	779	C4H9NO3	119.058	Ramp 5-60	4
Amino acid	Delta-Aminolevulinic_acid	137	C5H9NO3	131.058	Ramp 5-60	1
Amino acid	DL-2-Aminobutyric_acid	80283	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	DL-5-Hydroxylysine	1029	C6H14N2O3	162.1	Ramp 5-60	3
Amino acid	DL-alpha-epsilon-Diaminopimelic_acid	865	C7H14N2O4	190.095	Ramp 5-60	4
Amino acid	DL-threo-beta-Methylaspartic_acid	852	C5H9NO4	147.053	Ramp 5-60	3
Amino acid	D-Pantothenic_acid	6613	C9H17NO5	219.111	Ramp 5-60	4
Amino acid	Folic_acid	6037	C19H19N7O6	441.14	Ramp 5-60	4
Amino acid	Glutathione_(oxidized_form)	65359	C20H32N6O12S2	612.152	Ramp 5-60	13
Amino acid	Glycocyamine	763	C3H7N3O2	117.054	Ramp 5-60	1
Amino acid	Glycyl-L-proline	3013625	C7H12N2O3	172.085	Ramp 5-60	3
Amino acid	Gly-Gly	11163	C4H8N2O3	132.053	Ramp 5-60	2
Amino acid	L-(-)-Phenylalanine	6140	C9H11NO2	165.079	Ramp 5-60	4
Amino acid	L(+)-Arginine	6322	C6H14N4O2	174.112	Ramp 5-60	1
Amino acid	L-(+)-Lysine	5962	C6H14N2O2	146.106	Ramp 5-60	0
Amino acid	L-2-Aminobutyric_acid	80283	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	L-allo-threonine	99289	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	L-Anserine	112072	C10H16N4O3	240.122	Ramp 5-60	3
Amino acid	L-Arginine	6322	C6H14N4O2	174.112	Ramp 5-60	1
Amino acid	L-beta-Homoisoleucine	16211048	C7H15NO2	145.11	Ramp 5-60	0
Amino acid	L-beta-homoleucine	2761525	C7H15NO2	145.11	Ramp 5-60	0
Amino acid	L-beta-homolysine	2761529	C7H16N2O2	160.121	Ramp 5-60	0
Amino acid	L-beta-homomethionine	5706673	C6H13NO2S	163.067	Ramp 5-60	2
Amino acid	L-beta-Homophenylalanine	2761537	C10H13NO2	179.095	Ramp 5-60	2
Amino acid	L-beta-homoproline	2761541	C6H11NO2	129.079	Ramp 5-60	0
Amino acid	L-beta-homoserine	1502076	C4H9NO3	119.058	Ramp 5-60	4
Amino acid	L-beta-homothreonine	5706676	C5H11NO3	133.074	Ramp 5-60	3
Amino acid	L-beta-homotryptophan	2761550	C12H14N2O2	218.106	Ramp 5-60	3
Amino acid	L-beta-homotyrosine	2761554	C10H13NO3	195.09	Ramp 5-60	2
Amino acid	L-beta-homovaline	2761558	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Carnosine	439224	C9H14N4O3	226.107	Ramp 5-60	5
Amino acid	L-Citrulline	9750	C6H13N3O3	175.096	Ramp 5-60	1
Amino acid	L-Ethionine	25674	C6H13NO2S	163.067	Ramp 5-60	1
Amino acid	Leucylleucyltyrosine	88513	C21H33N3O5	407.242	Ramp 5-60	6
Amino acid	Leupeptin	439527	C20H38N6O4	426.295	Ramp 5-60	4
Amino acid	L-Glutamic_acid	33032	C5H9NO4	147.053	Ramp 5-60	2
Amino acid	L-Histidine	6274	C6H9N3O2	155.069	Ramp 5-60	8
Amino acid	L-Homocarnosine	89235	C10H16N4O3	240.122	Ramp 5-60	3
Amino acid	L-Homoserine	12647	C4H9NO3	119.058	Ramp 5-60	3
Amino acid	L-Leucine	6106	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Methionine_sulfone	445282	C5H11NO4S	181.041	Ramp 5-60	2
Amino acid	L-Norleucine	21236	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Norvaline	65098	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	L-Proline	145742	C5H9NO2	115.063	Ramp 5-60	0
Amino acid	L-saccharopine	160556	C11H20N2O6	276.132	Ramp 5-60	4
Amino acid	L-Threonine	6288	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	L-Tryptophane	6305	C11H12N2O2	204.09	Ramp 5-60	4
Amino acid	L-Tyrosine	6057	C9H11NO3	181.074	Ramp 5-60	4
Amino acid	L-Valine	6287	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	N-Acetyl-DL-aspartic_acid	65065	C6H9NO5	175.048	Ramp 5-60	6
Amino acid	N-Acetyl-DL-glutamic_acid	70914	C7H11NO5	189.064	Ramp 5-60	6
Amino acid	N-acetyl-DL-serine	352294	C5H9NO4	147.053	Ramp 5-60	2
Amino acid	N-Acetylglycine	10972	C4H7NO3	117.043	Ramp 5-60	2
Amino acid	N-alpha-Acetyl-L-ornithine	439232	C7H14N2O3	174.1	Ramp 5-60	2
Amino acid	N-Formyl-L-methionine	439750	C6H11NO3S	177.046	Ramp 5-60	3
Amino acid	N-N-Dimethylglycine	673	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	N-Tigloylglycine	6441567	C7H11NO3	157.074	Ramp 5-60	4
Amino acid	O-Phospho-L-serine	68841	C3H8NO6P	185.009	Ramp 5-60	2
Amino acid	S-Adenosyl-L-homocysteine	439155	C14H20N6O5S	384.122	Ramp 5-60	1
Amino acid	S-Lactoylglutathione	440018	C13H21N3O8S	379.105	Ramp 5-60	18
Amino acid	S-Sulfocysteine	115015	C3H7NO5S2	200.977	Ramp 5-60	6
Benzimidazole	Thiabendazole	5430	C10H7N3S	201.036	Ramp 5-60	3
Bile acid	Cholate	221493	C24H40O5	408.288	30, Ramp 5-60	10
Bile acid	Deoxycholate	440355	C24H40O4	392.293	30, Ramp 5-60	6
Capsaicinoid	Capsaicin	1548943	C18H27NO3	305.199	Ramp 5-60	1
Capsaicinoid	Dihydrocapsaicin	107982	C18H29NO3	307.215	Ramp 5-60	1
Carboxylic acid	(-)-Citramalic_acid	439766	C5H8O5	148.037	Ramp 5-60	4
Carboxylic acid	(-)Shikimic_acid	8742	C7H10O5	174.053	Ramp 5-60	7
Carboxylic acid	(+-)-Alpha-Lipoic_acid	864	C8H14O2S2	206.044	Ramp 5-60	5
Carboxylic acid	(R)-(-)-mandelic_acid	11914	C8H8O3	152.047	Ramp 5-60	1
Carboxylic acid	(S)-(+)-Citramailc_acid	441696	C5H8O5	148.037	Ramp 5-60	3
Carboxylic acid	16-Hydroxyhexadecanoic_acid	10466	C16H32O3	272.235	Ramp 5-60	0
Carboxylic acid	1-O-b-D-glucopyranosyl_sinapate	5280406	C17H22O10	386.121	Ramp 5-60	10
Carboxylic acid	2-5-Dihydroxy_benzoic_acid	3469	C7H6O4	154.027	Ramp 5-60	2
Carboxylic acid	2-Aminoethylphosphonic_acid	339	C2H8NO3P	125.024	Ramp 5-60	2
Carboxylic acid	2-Hydroxyisobutyric_acid	11671	C4H8O3	104.047	30, Ramp 5-60	2
Carboxylic acid	2-Hydroxyisocaproic_acid	439960	C6H12O3	132.079	Ramp 5-60	2
Carboxylic acid	2-Isopropylmalic_acid	5280523	C7H12O5	176.068	Ramp 5-60	5
Carboxylic acid	2-Methylglutaric_Acid	12046	C6H10O4	146.058	Ramp 5-60	2
Carboxylic acid	2-Oxobutyrate	58	C4H6O3	102.032	Ramp 5-60	0
Carboxylic acid	2-Oxovaleric_acid	74563	C5H8O3	116.047	Ramp 5-60	0
Carboxylic acid	3-4-Dihydroxybenzoic_acid	72	C7H6O4	154.027	Ramp 5-60	2
Carboxylic acid	3-Guanidinopropionic_acid	67701	C4H9N3O2	131.069	Ramp 5-60	1
Carboxylic acid	3-Hydroxy-3-methylglutarate	1662	C6H10O5	162.053	Ramp 5-60	3
Carboxylic acid	3-Hydroxymandelic_acid	86957	C8H8O4	168.042	Ramp 5-60	2

Table B.2: Compound list for the MassBank dataset: Compound class, compound name, PubChem ID, molecular formula, monoisotopic mass (Da), collision energies (eV), and number of annotated losses (NLs) in hypothetical fragmentation trees. The ion type of all compounds is $[M-H]^-$ or M^-. Compounds with less than three (seven) annotated losses are colored red (yellow).

Carboxylic acid	3-Indoleacetic_acid	802	C10H9NO2	175.063	Ramp 5-60	2
Carboxylic acid	4-Coumaric_acid	637542	C9H8O3	164.047	30, Ramp 5-60	2
Carboxylic acid	4-Hydroxy-3-methoxycinnamic_acid	445858	C10H10O4	194.058	Ramp 5-60	3
Carboxylic acid	4-Hydroxy-benzoate	135	C7H6O3	138.032	Ramp 5-60	1
Carboxylic acid	6-Hydroxynicotinic_Acid	72924	C6H5NO3	139.027	Ramp 5-60	1
Carboxylic acid	Anthranilic_acid	227	C7H7NO2	137.048	Ramp 5-60	1
Carboxylic acid	Caffeic_acid	689043	C9H8O4	180.042	Ramp 5-60	2
Carboxylic acid	Cis-Aconitic_Acid	643757	C6H6O6	174.016	Ramp 5-60	3
Carboxylic acid	Citraconic_Acid	643798	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Citric_acid	311	C6H8O7	192.027	Ramp 5-60	6
Carboxylic acid	D-(-)-Quinic_acid	6508	C7H12O6	192.063	Ramp 5-60	1
Carboxylic acid	D(+)-Galacturonic_acid	439215	C6H10O7	194.043	Ramp 5-60	11
Carboxylic acid	D-(+)-Glyceric_acid	439194	C3H6O4	106.027	Ramp 5-60	2
Carboxylic acid	D-(+)-Malic_acid	92824	C4H6O5	134.022	Ramp 5-60	4
Carboxylic acid	D-Gluconic_acid	10690	C6H12O7	196.058	Ramp 5-60	8
Carboxylic acid	D-Glucuronic_acid	94715	C6H10O7	194.043	Ramp 5-60	10
Carboxylic acid	DL-2-Hydroxyvaleric_acid	98009	C5H10O3	118.063	Ramp 5-60	1
Carboxylic acid	DL-3-4-Dihydroxymandelic_acid	85782	C8H8O5	184.037	Ramp 5-60	2
Carboxylic acid	DL-3-Aminoisobutyric_acid	64956	C4H9NO2	103.063	Ramp 5-60	1
Carboxylic acid	DL-4-Hydroxy-3-methoxymandelic_acid	1245	C9H10O5	198.053	Ramp 5-60	1
Carboxylic acid	DL-beta-Aminobutyric_acid	2761506	C4H9NO2	103.063	Ramp 5-60	0
Carboxylic acid	DL-beta-Hydroxybutyric_acid	441	C4H8O3	104.047	Ramp 5-60	1
Carboxylic acid	DL-Glyceric_acid	439194	C3H6O4	106.027	Ramp 5-60	2
Carboxylic acid	DL-Lactic_acid	107689	C3H6O3	90.032	Ramp 5-60	0
Carboxylic acid	DL-mandelic_acid	1292	C8H8O3	152.047	Ramp 5-60	1
Carboxylic acid	DL-p-Hydroxyphenyllactic_acid	9378	C9H10O4	182.058	Ramp 5-60	5
Carboxylic acid	DL-Pipecolinic_acid	439227	C6H11NO2	129.079	Ramp 5-60	0
Carboxylic acid	D-tartaric_acid	439655	C4H6O6	150.016	Ramp 5-60	4
Carboxylic acid	Gamma-Linolenic_acid	5280933	C18H30O2	278.225	Ramp 5-60	1
Carboxylic acid	Gibberellin_A4	443457	C19H24O5	332.162	Ramp 5-60	8
Carboxylic acid	Glutaric_acid	743	C5H8O4	132.042	Ramp 5-60	2
Carboxylic acid	Homogentisic_acid	780	C8H8O4	168.042	Ramp 5-60	3
Carboxylic acid	Indole-3-carboxylic_acid	69867	C9H7NO2	161.048	Ramp 5-60	1
Carboxylic acid	Isoguvacine	3765	C6H9NO2	127.063	Ramp 5-60	1
Carboxylic acid	Isonicotinic_acid	5922	C6H5NO2	123.032	Ramp 5-60	1
Carboxylic acid	Itaconic_acid	811	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Kynurenic_acid	3845	C10H7NO3	189.043	Ramp 5-60	1
Carboxylic acid	L(+)-Tartaric_acid	444305	C4H6O6	150.016	Ramp 5-60	2
Carboxylic acid	L-2-Aminoadipic_Acid	92136	C6H11NO4	161.069	Ramp 5-60	3
Carboxylic acid	L-Pyroglutamic_acid	7405	C5H7NO3	129.043	Ramp 5-60	0
Carboxylic acid	Maleic_acid	444266	C4H4O4	116.011	Ramp 5-60	1
Carboxylic acid	Mesaconic_acid	638129	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Methylsuccinic_acid	10349	C5H8O4	132.042	30, Ramp 5-60	1
Carboxylic acid	Mucic_acid	3037582	C6H10O8	210.038	Ramp 5-60	5
Carboxylic acid	N-acetylneuraminic_acid	439197	C11H19NO9	309.106	Ramp 5-60	3
Carboxylic acid	Nicotinic_Acid	938	C6H5NO2	123.032	Ramp 5-60	1
Carboxylic acid	Orotic_acid	967	C5H4N2O4	156.017	Ramp 5-60	1
Carboxylic acid	Phosphoenolpyruvic_Acid	1005	C3H5O6P	167.982	Ramp 5-60	1
Carboxylic acid	Prostaglandin_E1	5280723	C20H34O5	354.241	Ramp 5-60	6
Carboxylic acid	Rosmarinic_acid	639655	C18H16O8	360.085	Ramp 5-60	8
Carboxylic acid	Sebacic_acid	5192	C10H18O4	202.121	Ramp 5-60	3
Carboxylic acid	Sinapic_acid	637775	C11H12O5	224.068	Ramp 5-60	10
Carboxylic acid	Sinapoyl_malate	11953815	C15H16O9	340.079	Ramp 5-60	12
Carboxylic acid	Succinic_acid	1110	C4H6O4	118.027	Ramp 5-60	2
Carboxylic acid	Trans-4-Hydroxy-L-proline	5810	C5H9NO3	131.058	Ramp 5-60	2
Carboxylic acid	Trans-Cinnamic_acid	444539	C9H8O2	148.052	Ramp 5-60	1
Carboxylic acid	Urocanic_acid	736715	C6H6N2O2	138.043	Ramp 5-60	1
Coumarin	4-Methylumbelliferone	5280567	C10H8O3	176.047	Ramp 5-60	5
Coumarin	6-7-Dihydroxycoumarin	5281416	C9H6O4	178.027	30, Ramp 5-60	19
Coumarin	7-Hydroxy-4-methylcoumarin	5280567	C10H8O3	176.047	30, Ramp 5-60	10
Coumarin	Daphnetin	5280569	C9H6O4	178.027	30, Ramp 5-60	12
Coumarin	Esculin	5281417	C15H16O9	340.079	Ramp 5-60	4
Coumarin	Scopoletin	5280460	C10H8O4	192.042	Ramp 5-60	4
Ethanolamine	O-Phosphorylethanolamine	1015	C2H8NO4P	141.019	Ramp 5-60	1
Flavonoid	(-)-Epicatechin	72276	C15H14O6	290.079	Ramp 5-60	25
Flavonoid	(-)-Riboflavin	493570	C17H20N4O6	376.138	Ramp 5-60	4
Flavonoid	(+)-Catechin	9064	C15H14O6	290.079	Ramp 5-60	13
Flavonoid	(+)-Epicatechin	182232	C15H14O6	290.079	Ramp 5-60	13
Flavonoid	7-Methylquercetin-3-Galactoside-6-Rhamnoside-3-Rhamnoside	44259338	C34H42O20	770.227	30, Ramp 5-60	4
Flavonoid	Apigenin	5280443	C15H10O5	270.053	Ramp 5-60	2
Flavonoid	Apigenin-7-O-glucoside	5280704	C21H20O10	432.106	Ramp 5-60	7
Flavonoid	Baicalin	64982	C21H18O11	446.085	Ramp 5-60	3
Flavonoid	Daidzein	5281708	C15H10O4	254.058	30, Ramp 5-60	18
Flavonoid	Daidzin	107971	C21H20O9	416.111	Ramp 5-60	10
Flavonoid	Datiscin	5883291	C27H30O15	594.158	30, Ramp 5-60	14
Flavonoid	Eriodictyol	440735	C15H12O6	288.063	Ramp 5-60	5
Flavonoid	Eriodictyol-7-O-glucoside	5319853	C21H22O11	450.116	Ramp 5-60	7
Flavonoid	Flavanomarein	101781	C21H22O11	450.116	Ramp 5-60	4
Flavonoid	Formononetin	5280378	C16H12O4	268.074	Ramp 5-60	7
Flavonoid	Fortunellin	5317385	C28H32O14	592.179	Ramp 5-60	2
Flavonoid	Gossypin	5281621	C21H20O13	480.09	Ramp 5-60	7
Flavonoid	Hesperidin	10621	C28H34O15	610.19	Ramp 5-60	5
Flavonoid	Homoorientin	114776	C21H20O11	448.101	Ramp 5-60	13
Flavonoid	Hyperoside	5281643	C21H20O12	464.095	Ramp 5-60	8
Flavonoid	Isorhamnetin	5281654	C16H12O7	316.058	Ramp 5-60	3
Flavonoid	Isorhamnetin-3-Galactoside-6-Rhamnoside	44259338	C28H32O16	624.169	30, Ramp 5-60	8
Flavonoid	Isorhamnetin-3-O-glucoside	5318645	C22H22O12	478.111	30, Ramp 5-60	13
Flavonoid	Isorhamnetin-3-O-rutinoside	5481663	C28H32O16	624.169	30, Ramp 5-60	8
Flavonoid	Kaempferide	5281666	C16H12O6	300.063	Ramp 5-60	10
Flavonoid	Kaempferol	5280863	C15H10O6	286.048	Ramp 5-60	3
Flavonoid	Kaempferol-3-7-O-bis-alpha-L-rhamnoside	5323562	C27H30O14	578.164	30, Ramp 5-60	10
Flavonoid	Kaempferol-3-Galactoside-6-Rhamnoside-3-Rhamnoside	5281693	C33H40O19	740.216	30, Ramp 5-60	4
Flavonoid	Kaempferol-3-Glucoside-2-p-coumaroyl	25245527	C30H26O13	594.137	Ramp 5-60	6
Flavonoid	Kaempferol-3-Glucoside-2-Rhamnoside-7-Rhamnoside	25202803	C33H40O19	740.216	30, Ramp 5-60	7
Flavonoid	Kaempferol-3-Glucoside-3-Rhamnoside	25202803	C27H30O15	594.158	Ramp 5-60	4
Flavonoid	Kaempferol-3-Glucoside-6-p-coumaroyl	5320686	C30H26O13	594.137	30, Ramp 5-60	11
Flavonoid	Kaempferol-3-Glucuronide	5318759	C21H18O12	462.08	Ramp 5-60	3
Flavonoid	Kaempferol-3-O-alpha-L-arabinoside	5481882	C20H18O10	418.09	Ramp 5-60	7
Flavonoid	Kaempferol-3-O-alpha-L-rhamnopyranosyl(1-2)-beta-D-glucopyranoside-7-O-alpha-L-rhamnopyranoside	44258837	C33H40O19	740.216	30, Ramp 5-60	8
Flavonoid	Kaempferol-3-O-alpha-L-rhamnoside	5316673	C21H20O10	432.106	Ramp 5-60	9
Flavonoid	Kaempferol-3-O-beta-D-galactoside-7-O-alpha-L-rhamnoside	5281693	C27H30O15	594.158	30, Ramp 5-60	13
Flavonoid	Kaempferol-3-O-beta-glucopyranosyl-7-O-alpha-rhamnopyranoside	25203808	C27H30O15	594.158	30, Ramp 5-60	11
Flavonoid	Kaempferol-3-O-glucoside	5282102	C21H20O11	448.101	30, Ramp 5-60	13

Table B.2: Compound list for the MassBank dataset (continued)

Flavonoid	Kaempferol-3-O-rutinoside	5318767	C27H30O15	594.158	30, Ramp 5-60	6
Flavonoid	Kaempferol-3-Rhamnoside-4-Rhamnoside-7-Rhamnoside	44259005	C33H40O18	724.221	Ramp 5-60	6
Flavonoid	Kaempferol-7-O-alpha-L-rhamnoside	5316673	C21H20O10	432.106	30, Ramp 5-60	28
Flavonoid	Kaempferol-7-O-neohesperidoside	5483905	C27H30O15	594.158	30, Ramp 5-60	3
Flavonoid	Linarin	5317025	C28H32O14	592.179	Ramp 5-60	2
Flavonoid	Luteolin	5280445	C15H10O6	286.048	30, Ramp 5-60	19
Flavonoid	Luteolin-3-7-di-O-glucoside	5490298	C27H30O16	610.153	Ramp 5-60	3
Flavonoid	Luteolin-4-O-glucoside	5319116	C21H20O11	448.101	Ramp 5-60	6
Flavonoid	Luteolin-7-O-glucoside	5280637	C21H20O11	448.101	Ramp 5-60	8
Flavonoid	Marein	6441269	C21H22O11	450.116	Ramp 5-60	8
Flavonoid	Maritimein	6450184	C21H20O11	448.101	Ramp 5-60	3
Flavonoid	Myricetin-3-Galactoside	5491408	C21H20O13	480.09	Ramp 5-60	11
Flavonoid	Myricetin-3-Rhamnoside	5281673	C21H20O12	464.095	Ramp 5-60	12
Flavonoid	Myricetin-3-Xyloside	5281673	C20H18O12	450.08	Ramp 5-60	9
Flavonoid	Myricitrin	5281673	C21H20O12	464.095	Ramp 5-60	11
Flavonoid	Naringenin-7-O-glucoside	92794	C21H22O10	434.121	Ramp 5-60	7
Flavonoid	Neodiosmin	44258230	C28H32O15	608.174	Ramp 5-60	2
Flavonoid	Ononin	442813	C22H22O9	430.126	30, Ramp 5-60	5
Flavonoid	Peltatoside	5484066	C26H28O16	596.138	30, Ramp 5-60	18
Flavonoid	Poncirin	442456	C28H34O14	594.195	30, Ramp 5-60	5
Flavonoid	Procyanidin_B1	11250133	C30H26O12	578.142	Ramp 5-60	15
Flavonoid	Procyanidin_B2	122738	C30H26O12	578.142	Ramp 5-60	16
Flavonoid	Puerarin	5281807	C21H20O9	416.111	Ramp 5-60	6
Flavonoid	Quercetin	5280343	C15H10O7	302.043	Ramp 5-60	8
Flavonoid	Quercetin-3-(6-malonyl)-Glucoside	5282159	C24H22O15	550.096	Ramp 5-60	8
Flavonoid	Quercetin-3-4-O-di-beta-glucopyranoside	5320835	C27H30O17	626.148	30, Ramp 5-60	9
Flavonoid	Quercetin-3-7-O-alpha-L-dirhamnopyranoside	44259217	C27H30O15	594.158	30, Ramp 5-60	10
Flavonoid	Quercetin-3-Arabinoside	5481224	C20H18O11	434.085	Ramp 5-60	8
Flavonoid	Quercetin-3-D-xyloside	5320863	C20H18O11	434.085	Ramp 5-60	9
Flavonoid	Quercetin-3-Glucuronide	5274585	C21H18O13	478.075	Ramp 5-60	8
Flavonoid	Quercetin-3-O-alpha-L-rhamnopyranoside	5280459	C21H20O11	448.101	Ramp 5-60	12
Flavonoid	Quercetin-3-O-alpha-L-rhamnopyranosyl(1-2)-beta-D-glucopyranoside-7-O-alpha-L-rhamnopyranoside	5489459	C33H40O20	756.211	30, Ramp 5-60	8
Flavonoid	Quercetin-3-O-beta-D-galactoside	5281643	C21H20O12	464.095	Ramp 5-60	9
Flavonoid	Quercetin-3-O-beta-glucopyranoside	5280804	C21H20O12	464.095	Ramp 5-60	6
Flavonoid	Quercetin-3-O-beta-glucopyranosyl-7-O-alpha-rhamnopyranoside	5280805	C27H30O16	610.153	30, Ramp 5-60	11
Flavonoid	Quercetin-3-O-glucose-6-acetate	5280804	C23H22O13	506.106	Ramp 5-60	8
Flavonoid	Quercetin-7-O-rhamnoside	5748601	C21H20O11	448.101	Ramp 5-60	9
Flavonoid	Rhamnetin	5281691	C16H12O7	316.058	Ramp 5-60	6
Flavonoid	Rhoifolin	5282150	C27H30O14	578.164	30, Ramp 5-60	3
Flavonoid	Robinin	5281693	C33H40O19	740.216	30, Ramp 5-60	7
Flavonoid	Spiraeoside	5320844	C21H20O12	464.095	Ramp 5-60	6
Flavonoid	Syringetin-3-O-galactoside	5321576	C23H24O13	508.122	30, Ramp 5-60	17
Flavonoid	Syringetin-3-O-glucoside	5321577	C23H24O13	508.122	Ramp 5-60	14
Flavonoid	Tiliroside	5320686	C30H26O13	594.137	30, Ramp 5-60	9
Flavonoid	Vitexin	5280441	C21H20O10	432.106	Ramp 5-60	4
Flavonoid	Vitexin-2-O-rhamnoside	5282151	C27H30O14	578.164	Ramp 5-60	5
Glucosinolate	4-Methylsulfinylbutyl_glucosinolate	9548634	C12H23NO10S3	437.048	Ramp 5-60	6
Glucosinolate	4-Methylthiobutyl_glucosinolate	9548895	C12H23NO9S3	421.053	Ramp 5-60	4
Glucosinolate	Sinigrin	6911854	C10H17NO9S2	359.034	Ramp 5-60	4
Indole	3-Indoxylsulfate	10258	C8H7NO4S	213.01	Ramp 5-60	3
Indole	Harmaline	5280951	C13H14N2O	214.111	Ramp 5-60	4
Isoprenoid	Glycyrrhizic_acid	14982	C42H62O16	822.404	30, Ramp 5-60	3
Isoprenoid	Glycyrrhizin	14982	C42H62O16	822.404	30, Ramp 5-60	3
Nucleotide	1-3-Dimethylurate	70346	C7H8N4O3	196.06	30, Ramp 5-60	10
Nucleotide	1-7-Dimethylxanthine	4687	C7H8N4O2	180.065	Ramp 5-60	5
Nucleotide	2-Deoxyadenosine-5-monophosphate	12599	C10H14N5O6P	331.068	Ramp 5-60	5
Nucleotide	2-Deoxycytidine	13711	C9H13N3O4	227.091	Ramp 5-60	3
Nucleotide	2-Deoxycytidine-5-diphosphate	150855	C9H15N3O10P2	387.023	Ramp 5-60	6
Nucleotide	2-Deoxyguanosine_5-monophosphate	65059	C10H14N5O7P	347.063	Ramp 5-60	4
Nucleotide	2-Deoxyguanosine-5-diphosphate	439220	C10H15N5O10P2	427.029	Ramp 5-60	2
Nucleotide	2-Deoxyinosine-5-monophosphate	91531	C10H13N4O7P	332.052	Ramp 5-60	6
Nucleotide	2-Deoxyuridine-5-monophosphate	65063	C9H13N2O8P	308.041	Ramp 5-60	5
Nucleotide	3-Hydroxypyridine	7971	C5H5NO	95.037	30, Ramp 5-60	1
Nucleotide	3-Methylxanthine	70639	C6H6N4O2	166.049	Ramp 5-60	5
Nucleotide	4-Pyridoxate	6723	C8H9NO4	183.053	Ramp 5-60	2
Nucleotide	5-Aminoimidazole-4-carboxamide-1-beta-D-ribofuranosyl_5-monophosphate	65110	C9H15N4O8P	338.063	Ramp 5-60	3
Nucleotide	5-Deoxy-5-Methylthioadenosine	439176	C11H15N5O3S	297.09	Ramp 5-60	2
Nucleotide	6-(Gamma-gamma-Dimethylallylamino)purine	92180	C10H13N5	203.117	Ramp 5-60	6
Nucleotide	6-(Gamma-gamma-Dimethylallylamino)purine_riboside	24405	C15H21N5O4	335.159	Ramp 5-60	4
Nucleotide	Adenine	190	C5H5N5	135.054	30, Ramp 5-60	4
Nucleotide	Adenosine	60961	C10H13N5O4	267.097	Ramp 5-60	2
Nucleotide	Adenosine_3-monophosphate	41211	C10H14N5O7P	347.063	Ramp 5-60	4
Nucleotide	Adenosine_5-diphosphate	6022	C10H15N5O10P2	427.029	Ramp 5-60	3
Nucleotide	Adenosine_5-diphospho-glucose	16500	C16H25N5O15P2	589.082	Ramp 5-60	9
Nucleotide	Adenosine_5-monophosphate	6083	C10H14N5O7P	347.063	Ramp 5-60	3
Nucleotide	Beta-Nicotinamide_adenine_dinucleotide	5893	C21H27N7O14P2	663.109	Ramp 5-60	10
Nucleotide	Cytidine	6175	C9H13N3O5	243.086	Ramp 5-60	4
Nucleotide	Cytidine_5-diphosphocholine	13804	C14H26N4O11P2	488.107	Ramp 5-60	8
Nucleotide	Cytidine-3-5-cyclicmonophosphate	19236	C9H12N3O7P	305.041	Ramp 5-60	6
Nucleotide	Cytidine-3-monophosphate	66535	C9H14N3O8P	323.052	Ramp 5-60	4
Nucleotide	Cytidine-5-diphosphate	6132	C9H15N3O11P2	403.018	Ramp 5-60	5
Nucleotide	Cytidine-5-monophosphate	6131	C9H14N3O8P	323.052	Ramp 5-60	3
Nucleotide	Guanine	764	C5H5N5O	151.049	Ramp 5-60	3
Nucleotide	Guanosine	6802	C10H13N5O5	283.092	Ramp 5-60	3
Nucleotide	Guanosine_5-diphosphate-D-mannose	18396	C16H25N5O16P2	605.077	Ramp 5-60	6
Nucleotide	Guanosine_5-diphospho-beta-L-fucose	10918995	C16H25N5O15P2	589.082	Ramp 5-60	9
Nucleotide	Guanosine_5-diphosphoglucose	439225	C16H25N5O16P2	605.077	Ramp 5-60	7
Nucleotide	Guanosine_5-monophosphate	6804	C10H14N5O8P	363.058	Ramp 5-60	5
Nucleotide	Guanosine-3-5-cyclic_monophosphate	24316	C10H12N5O7P	345.047	Ramp 5-60	6
Nucleotide	Inosine	6021	C10H12N4O5	268.081	Ramp 5-60	3
Nucleotide	Inosine-5-diphosphate	6831	C10H14N4O11P2	428.013	Ramp 5-60	7
Nucleotide	Inosine-5-monophosphate	8582	C10H13N4O8P	348.047	Ramp 5-60	6
Nucleotide	N-6-(delta-2-Isopentenyl)adenosine	24405	C15H21N5O4	335.159	Ramp 5-60	5
Nucleotide	Oxypurinol	4644	C5H4N4O2	152.033	Ramp 5-60	1
Nucleotide	Pyridoxal	1050	C8H9NO3	167.058	Ramp 5-60	3
Nucleotide	Pyridoxal_5-phosphate	1051	C8H10NO6P	247.025	Ramp 5-60	2
Nucleotide	Pyridoxamine	1052	C8H12N2O2	168.09	Ramp 5-60	9
Nucleotide	Pyridoxine	1054	C8H11NO3	169.074	Ramp 5-60	7
Nucleotide	Thiamine	1130	C12H17N4OS	265.112	Ramp 5-60	5
Nucleotide	Thymidine-5-diphosphate	164628	C10H16N2O11P2	402.023	Ramp 5-60	8
Nucleotide	Thymidine-5-monophosphate	9700	C10H15N2O8P	322.057	Ramp 5-60	5
Nucleotide	Thymine	1135	C5H6N2O2	126.043	Ramp 5-60	0
Nucleotide	Trans-Zeatin	449093	C10H13N5O	219.112	Ramp 5-60	8

Table B.2: Compound list for the MassBank dataset (continued)

Nucleotide	Trans-Zeatin-riboside	6440982	C15H21N5O5	351.154	Ramp 5-60	6
Nucleotide	UDP-beta-L-rhamnose	23724469	C15H24N2O16P2	550.06	Ramp 5-60	13
Nucleotide	UDP-Galactose	23724458	C15H24N2O17P2	566.055	Ramp 5-60	13
Nucleotide	UDP-xylose	23724459	C14H22N2O16P2	536.044	Ramp 5-60	15
Nucleotide	Uracil	1174	C4H4N2O2	112.027	Ramp 5-60	0
Nucleotide	Uridine	6029	C9H12N2O6	244.07	Ramp 5-60	5
Nucleotide	Uridine_5-diphosphate	6031	C9H14N2O12P2	404.002	Ramp 5-60	5
Nucleotide	Uridine_5-diphospho-D-glucose	8629	C15H24N2O17P2	566.055	Ramp 5-60	13
Nucleotide	Uridine_5-diphosphoglucuronic_acid	17473	C15H22N2O18P2	580.034	Ramp 5-60	14
Nucleotide	Uridine_5-diphospho-N-acetylgalactosamine	23724461	C17H27N3O17P2	607.082	30, Ramp 5-60	17
Nucleotide	Uridine_5-diphospho-N-acetylglucosamine	445675	C17H27N3O17P2	607.082	30, Ramp 5-60	17
Nucleotide	Uridine_5-monophosphate	6030	C9H13N2O9P	324.036	Ramp 5-60	4
Nucleotide	Xanthine	1188	C5H4N4O2	152.033	Ramp 5-60	1
Nucleotide	Xanthosine	64959	C10H12N4O6	284.076	Ramp 5-60	2
Nucleotide	Xanthosine-5-monophosphate	73323	C10H13N4O9P	364.042	Ramp 5-60	6
Organosulfonic acid	2-Mercaptoethanesulfonic_acid	598	C2H6O3S2	141.976	Ramp 5-60	2
Organosulfonic acid	Hypotaurine	107812	C2H7NO2S	109.02	Ramp 5-60	2
Organosulfonic acid	S-Sulforaphene	6433206	C6H9NOS2	175.013	Ramp 5-60	4
Penicillin	Piperacillin	6604563	C23H27N5O7S	517.163	Ramp 5-60	5
Phenol	4-Nitrophenol	980	C6H5NO3	139.027	Ramp 5-60	0
Phenol	4-Nitrophenyl_phosphate	378	C6H6NO6P	218.993	30, Ramp 5-60	1
Phenol	Catechol	289	C6H6O2	110.037	30, Ramp 5-60	3
Polyketide	Zearalenone	5281576	C18H22O5	318.147	Ramp 5-60	8
Stilbene	E-3-4-5-trihydroxy-3-glucopyranosylstilbene	5281712	C20H22O9	406.126	Ramp 5-60	5
Sugar	2-Deoxyribose-5-phosphate	439288	C5H11O7P	214.024	Ramp 5-60	2
Sugar	Alpha-D-(+)-mannose-1-phosphate	439279	C6H13O9P	260.03	Ramp 5-60	2
Sugar	Alpha-D-Galactose-1-phosphate	123912	C6H13O9P	260.03	Ramp 5-60	4
Sugar	Alpha-D-Glucose-1-6-diphosphate	82400	C6H14O12P2	339.996	Ramp 5-60	6
Sugar	Alpha-D-glucose-1-phosphate	439165	C6H13O9P	260.03	Ramp 5-60	4
Sugar	D(-)-Gulono-gamma-lactone	165105	C6H10O6	178.048	Ramp 5-60	9
Sugar	D-(+)-Cellotriose	440950	C18H32O16	504.169	Ramp 5-60	22
Sugar	D-(+)-Melezitose	92817	C18H32O16	504.169	Ramp 5-60	12
Sugar	D-(+)-Raffinose	439242	C18H32O16	504.169	Ramp 5-60	9
Sugar	D-(+)-Trehalose	7427	C12H22O11	342.116	Ramp 5-60	10
Sugar	D-Arabinose-5-phosphate	230	C5H11O8P	230.019	Ramp 5-60	3
Sugar	D-Erythrose-4-phosphate	697	C4H9O7P	200.009	Ramp 5-60	3
Sugar	D-Fructose-6-phosphate	439160	C6H13O9P	260.03	Ramp 5-60	2
Sugar	D-Glucosamine-6-phosphate	439217	C6H14NO8P	259.046	Ramp 5-60	3
Sugar	D-Glucose-6-phosphate	5958	C6H13O9P	260.03	Ramp 5-60	3
Sugar	D-Mannose-6-phosphate	65127	C6H13O9P	260.03	Ramp 5-60	4
Sugar	D-Ribose-5-phosphate	439167	C5H11O8P	230.019	Ramp 5-60	3
Sugar	D-Ribulose-5-phosphate	439184	C5H11O8P	230.019	Ramp 5-60	2
Sugar	L-(+)-Rhamnose	25310	C6H12O5	164.068	Ramp 5-60	0
Sugar	Maltotriose	439586	C18H32O16	504.169	Ramp 5-60	25
Sugar	Palatinose	439559	C12H22O11	342.116	Ramp 5-60	14
Sugar	Sucrose	5988	C12H22O11	342.116	Ramp 5-60	11
Sugar alcohol	1-2-Dilauroyl-sn-Glycero-3-Phosphate	9547171	C27H53O8P	536.348	Ramp 5-60	5
Sugar alcohol	1-Lauroyl-2-Hydroxy-sn-Glycero-3-Phosphocholine	460605	C20H42NO7P	439.27	Ramp 5-60	1
Sugar alcohol	1-Myristoyl-2-Hydroxy-sn-Glycero-3-Phosphate	9547180	C17H35O7P	382.212	Ramp 5-60	3
Sugar alcohol	D-(-)-Mannitol	6251	C6H14O6	182.079	Ramp 5-60	9
Sugar alcohol	DL-Glyceraldehyde_3-phosphate	729	C3H7O6P	169.998	Ramp 5-60	3
Sugar alcohol	D-Sorbitol	5780	C6H14O6	182.079	Ramp 5-60	9
Sugar alcohol	D-Sorbitol-6-phosphate	152306	C6H15O9P	262.045	Ramp 5-60	2
Sugar alcohol	Dulcitol	11850	C6H14O6	182.079	Ramp 5-60	11
Sugar alcohol	Galactinol	439451	C12H22O11	342.116	Ramp 5-60	14
Sugar alcohol	Glycerol-2-phosphate	2526	C3H9O6P	172.014	Ramp 5-60	2
Sugar alcohol	L-Iditol	5460044	C6H14O6	182.079	Ramp 5-60	5
Sugar alcohol	Maltitol	493591	C12H24O11	344.132	Ramp 5-60	10
Sugar alcohol	Rac-Glycerol_3-phosphoate	439162	C3H9O6P	172.014	Ramp 5-60	2
	2-Hydroxyphenylacetic_acid	11970	C8H8O3	152.047	Ramp 5-60	1
	Hinokitiol	3611	C10H12O2	164.084	30, Ramp 5-60	0
	Methyl_Salicylate	4133	C8H8O3	152.047	Ramp 5-60	1

Table B.2: Compound list for the MassBank dataset (continued)

group	compound	molecular formula	monoisotopic mass	collision energies	annotated NLs
Amine	Dopamine	C8H11NO2	153.079	10, 20, 30, 40, 50	19
Amine	Spermidine	C7H19N3	145.158	15, 25, 35, 45	17
Amine	Spermine	C10H26N4	202.216	15, 25, 35, 45	12
Amine	Tyramine	C8H12NO+	138.092	15, 20, 30, 40, 50	23
Amino acid	Alanine	C3H7NO2	89.048	10	1
Amino acid	Arginine	C6H14N4O2	174.112	20, 25, 30	15
Amino acid	Asparagine	C4H8N2O3	132.053	10, 15, 20, 30, 40	15
Amino acid	Aspartic acid	C4H7NO4	133.038	10, 15, 20, 30	8
Amino acid	Citrulline	C6H13N3O3	175.096	10, 15, 20, 25, 30	22
Amino acid	Cysteine	C3H8NO2S+	122.028	10, 15, 20, 30	7
Amino acid	Cystine	C6H12N2O4S2	240.024	10, 15, 20, 30, 40	40
Amino acid	Glutamic acid	C5H9NO4	147.053	10, 15, 20, 30	7
Amino acid	Glutamine	C5H10N2O3	146.069	10, 15, 20, 30	8
Amino acid	Histidine	C6H9N3O2	155.069	15, 25, 35, 45	18
Amino acid	Isoleucine	C6H13NO2	131.095	10, 15, 25, 40	18
Amino acid	Leucine	C6H13NO2	131.095	15, 25, 40	9
Amino acid	Lysine	C6H14N2O2	146.106	10, 15, 20, 30, 40	23
Amino acid	Methionine	C5H11NO2S	149.051	10, 15, 20, 30	10
Amino acid	Phenylalanine	C9H11NO2	165.079	15, 25, 40	15
Amino acid	Proline	C5H9NO2	115.063	10, 15, 55	7
Amino acid	Serine	C3H7NO3	105.043	10, 15, 20, 30	5
Amino acid	Threonine	C4H9NO3	119.058	10, 15, 20, 30	6
Amino acid	Tryptophane	C11H12N2O2	204.09	15, 25, 40, 55	38
Amino acid	Tyrosine	C9H11NO3	181.074	10, 15, 25, 30, 40	22
Amino acid	Valine	C5H11NO2	117.079	10, 25, 40, 55	15
Carboxylic acid	6-Aminocapronic acid	C6H13NO2	131.095	15, 20, 30, 40	29
Choline	3-(4-Hexosyloxyphenyl)propanoyl choline	C20H32NO8+	414.213	25, 40, 55	4
Choline	4-Coumaroyl choline	C14H20NO3+	250.144	15, 25, 40	4
Choline	4-Hexosylferuloyl choline	C21H32NO9+	442.208	15, 25, 40, 55	5
Choline	4-Hexosyloxybenzoyl choline	C18H28NO8+	386.181	15, 25, 40, 55, 90	5
Choline	4-Hexosyloxycinnamoyl choline	C20H30NO8+	412.197	25, 40, 55	4
Choline	4-Hexosylvanilloyl choline	C19H30NO9+	416.192	15, 25, 40, 55, 70	3
Choline	4-Hydroxybenzoyl choline	C12H18NO3+	224.129	15, 25, 40, 55	4
Choline	5-Hydroxyferuloyl choline	C15H22NO5+	296.15	15, 25, 40, 55	11
Choline	Acetyl choline	C7H16NO2+	146.118	20	3
Choline	Benzoyl choline	C12H18NO2+	208.134	15, 25, 40, 55	3
Choline	Cafeoyl choline	C14H20NO4+	266.139	15, 25, 40, 55	8
Choline	Choline with Arylglycerol-arylether backbone	C23H32NO8+	450.213	50	3
Choline	Cinnamoyl choline	C14H20NO2+	234.149	15, 25, 40, 55	3
Choline	Feruloyl choline	C15H22NO4+	280.155	15, 25, 40	7
Choline	Nicotinic acid choline ester	C11H17N2O2+	209.129	15, 25, 40, 55	3
Choline	Sinapoyl choline	C16H24NO5+	310.165	15, 25, 40	4
Choline	Syringoyl choline	C14H22NO5+	284.15	50	19
Choline	Vanilloyl choline	C13H20NO4+	254.139	15, 25, 40, 55	10

Table B.3: Compound list for the QSTAR dataset: Compound class, compound name, molecular formula, monoisotopic mass (Da), collision energies (eV), and number of annotated losses (NLs) in hypothetical fragmentation trees. The ion type of all compounds is $[M+H]^+$ or M^+. Compounds with less than three (seven) annotated losses are colored red (yellow).

Ehrenwörtliche Erklärung

Hiermit erkläre ich

- dass mir die Promotionsordnung der Fakultät bekannt ist,
- dass ich die Dissertation selbst angfertigt habe, keine Textabschnitte oder Ergebnisse eines dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönliche Mitteilungen und Quellen in meiner Arbeit angegeben habe,
- dass ich die Hilfe eines Promotionsberaters nicht in Anspruch genommen habe und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts haben mich folgende Personen unterstützt:
Sebastian Böcker, Christoph Böttcher, Franziska Hufsky, Marco Kai, Ravi Maddula, François Nicolas, Imran Rauf, Kerstin Scheubert, Tamara Steijger, Aleš Svatoš, Thomas Zichner

Ich habe weder die gleiche, noch eine ähnliche oder eine andere Arbeit an einer anderen Hochschule als Dissertation eingereicht.

Jena, den 15. August 2012

Florian Rasche