Analysis of Metabolite Tandem Mass Spectra

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Bioinformatiker

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik

eingereicht von Florian Rasche
geb. am 4. Mai 1983 in Bielefeld

Betreuer: Prof. Dr. Sebastian Böcker

Jena, 10. Januar 2008

Deutsche Zusammenfassung

Massenspektrometrie ist eine Hochdurchsatztechnik um Proteine und Metabolite zu analysieren. Um den ganzheitlichen Ansatz der Systembiologie verfolgen zu können, braucht man solche Techniken, um die Gesamtheit aller Proteine oder Metabolite in einer Probe zu bestimmen. Die manuelle Interpretation von Massenspektren ist mühsam und man kann nicht mit dem hohen Durchsatz von Massenspektrometern mithalten. Daher sind Methoden für eine computergestützte Analyse notwendig. Diese Methoden können Datenbanken verwenden. In letzter Zeit wurden aber immer mehr "de-novo"-Analyseansätze entwickelt, da für die meisten Anwendungen und Organismen keine Datenbanken verfügbar sind.

Im Gegensatz zu Protein-Massenspektren wurden für die Analyse von Metabolit-Spektren bisher nur wenige Ansätze entwickelt. Zur Analyse von Metabolit-Spektren kann man Isotopenmuster verwenden. Um diese Muster mit hoher Genauigkeit zu bestimmen ist ein teures Fouriertransformation-Ionenzyklotronresonanz-Massenspektrometer nötig. In dieser Arbeit werden Tandem-Massenspektren analysiert, die mit einem Quadrupol-time-of-flight-Massenspektrometer gemessen wurden. In diesen Geräten wird der Analyt fragmentiert bevor die Spektren gemessen werden. Deswegen werden auch Fragmentmassen bestimmt.

Wir berechnen nun die möglichen Summenformeln für alle Fragmente und konstruieren einen Graphen mithilfe dieser Formeln. Aus diesem Graph berechnen wir den wahrscheinlichsten Fragmentierungsbaum. Wir zeigen, dass das zugrundeliegende informatische Problem NP-schwer ist. Daher entwickeln wir einen festparameterhandhabbaren Algorithmus und Heuristiken um das Problem zu lösen. Zusätzlich werden Techniken zum Bewerten von möglichen Summenformeln und Fragmentierungsschritten entwickelt. Diese orientieren sich an der Wahrscheinlichkeit, dass dieser Schritt stattgefunden hat. Dieser wahrscheinlichkeitsbasierte Ansatz ist strikten Filtern, wie sie Kind und Fiehn [KF07] verwenden, vorzuziehen, da keine Schwellwerte nötig sind.

Testläufe auf gemessenen Spektren ergaben, das der exakte Algorithmus schnell ist und gute Ergebnisse liefert. Bei allen 31 Testmetaboliten wurde die richtige Lösung unter den ersten fünf Vorschlägen gefunden, bei 25 Molekülen war sogar der erste Vorschlag korrekt. Die Heuristiken produzierten ebenfalls gute Ergebnisse. Sie sind vergleichbar mit dem ähnlichen Tool FFP, das von Zhang et al. [ZGC+05] entwickelt wurde. Aber im Gegensatz zu diesem Tool haben wir noch keine Informationen aus Isotopenmustern einbezogen.

Am Ende der Arbeit werden Möglichkeiten vorgestellt, wie man den Ansatz weiter verbessern und auf Spektren anderer biochemischer Stoffe anwenden könnte.

Abstract

Mass spectrometry is a high-throughput technology for the analysis of proteins and metabolites. The integrative approach of systems biology depends on such a technology to be able to analyse the abundance of all proteins and metabolites in a sample. Since the manual interpretation of mass spectra is tedious, methods for a computer-based analysis are necessary. These methods may use databases. But because no databases are available for most applications and species, bioinformaticians have developed "de-novo" interpretation methods recently.

In this work, we will analyse tandem mass spectra obtained from an quadrupole time-of-flight mass spectrometer. In these devices the analyte is fragmented before the spectra are measured, therefore the fragment masses are also detected.

We calculate the elemental decompositions for all fragments and construct a graph using these decompositions. From this graph we will calculate the most likely fragmentation tree. As the computer theoretical problem behind this is NP-hard, we develop a fixed-parameter tractable algorithm as well as heuristics to solve the problem. Additionally, we propose scoring concepts, which indicate the likelihood that a certain decomposition or fragmentation step is real.

Tests on real spectra indicate that the proposed exact algorithm runs fast and produces good results. For all 31 test compounds the correct solution was among the top five suggestions, for 25 compounds the first suggestion was correct. The heuristics also showed good results.

Finally, we give an outlook on the possibilities to further improve our tool and propose other areas of application for the algorithms developed.

Contents

Chapter 1

Introduction

Mass spectrometry (MS) is among the most widely used technologies to analyse microbiological samples to date. In its simplest form a mass spectrometer is nothing else than a very exact scale. Still, by different techniques for sample preparation and various combinations of the parts of a mass spectrometer, it is often possible to identify the molecules in the sample [Das01].

To achieve this identification, either a skilled biologist or chemist or a software program has to analyse the measured spectra. As mass spectrometry is a high-throughput technology, it can produce more data than any expert can analyse. Thus software programs for mass spectrometry analysis are inevitable if mass spectrometry shall develop its full potential.

For protein identification, one of the first ideas was to build reference mass spectral databases or to calculate reference mass spectra from protein sequence databases [PPCC99, EMY94]. The analysis then consists of comparing the measured spectra to the references. Thus the task was and still is to develop a kind of distance measure for mass spectra. These algorithms have to take the typical measurements errors of mass spectra into account. Computer scientists also developed concepts to efficiently calculate significance values [BK07, KNKA02].

Unfortunately, databases do not (and will probably never) contain all biological molecules that are analysed. For proteins, at least the genome of the species under investigation has to be known. Even then the analysis remains difficult due to RNA splicing and post-translational modifications.

Therefore experimental pipelines and software for the de-novo-sequencing of proteins have been developed in recent years. Mass spectrometrists avoid the use of a database by fragmenting the analyte molecules and thus increasing the available information. This fragmentation is usually achieved by conducting the analyte through an inert gas. The collisions with the gas fragment the analyte. This technol-

ogy is therefore called collision induced dissociation (CID) [WM05]. A mass spectral analysis unit afterwards measures the analyte fragments. Because two analysers are used for measurement, the whole setup is called tandem mass spectrometry. A third analysis unit is used as collision cell. For details, refer to Section 2.3.3 on page 24. Chen, Fischer and many others developed software to analyse these tandem mass spectra [CKT+01, FRR+05]. Although successful with proteins, nobody has transferred this concept to metabolites up to now, because in contrast to proteins the fragmentation of metabolites is not completely predictable and metabolites have a more complex, non-linear structure.

The database approach to identify metabolites is even more limited than the one to identify proteins, because the genome is not of great help when analysing metabolites [Fie02]. The metabolite itself has to be known to re-identify it. Whereas the metabolites involved in growth, development and reproduction are well known, only few of the metabolites not participating in these three areas are known [Ini00]. These secondary metabolites are important to understand, e.g., in plants, where they serve as main signalling molecules [JCB+00].

Note that mass spectrometry is not able to find structures of unknown metabolites, because the molecular weight measured by mass spectrometry only depends on the sum formula. Therefore to identify a metabolite in this work means to identify its sum formula.

This work develops a concept how to analyse tandem mass spectra of metabolites, despite the fragmentation process being not completely understood and difficult to predict [Wil02]. The concept of this work accounts for this missing comprehension by rating no fragmentation step impossible. Of course some knowledge exists on the typical fragments and the properties of typical metabolites are known. From this, we derive a scoring as presented in Chapter 4 of this thesis.

We transform the problem instance into a weighted graph and calculate the best scoring fragmentation tree from this graph. This calculation is a computationally hard problem. Therefore, we use heuristics as well as fixed-parameter tractability to solve this problem. We apply different algorithmic approaches to solve the problem exactly. The reduction rules as well as the branch and bound approach can not efficiently be applied to the graphs calculated from tandem mass spectra. The dynamic programming algorithm can determine a solution for a compound of our test set in a few seconds. But if certain conditions apply, it is outperformed by a brute force approach. Thus, the final implementation uses a combination of the dynamic programming and the brute force approaches.

We test this implementation by analysing metabolite tandem mass spectra. The

results are satisfactory regarding the results as well as running times. For all 45 compounds tested the correct sum formula was among the top five suggestions. The tool ranked the correct formula on first position for four of the eight compounds with a mass between 300 and 500 Da. For compounds below 300 Da, that was the case for twenty-two of twenty-four compounds. Our tool therefore performs as good as the similar tool FFP (Fragment ion Formula Prediction) by Zhang et al. [ZGC+05].

The structure of this work as follows. We define the required graph theoretical terms and describe the concept of fixed parameter tractability in the remainder of this chapter. This provides the computational science background for this work. Chapter 2 puts the research on metabolites into a greater biological context and presents the experimental technologies to acquire the data analysed here. Chapter 3 describes the basic idea of the analysis and sketches the procedures necessary for analysis. In Chapter 4 we develop a scoring scheme to assess the interpretations found. The transformation of the analysis idea into a graph theoretical problem is described in Chapter 5. We prove that this problem is NP-hard and develop exact algorithms as well as heuristics to solve the problem. Chapter 6 presents an implementation of these algorithms, and we use this implementation to analyse a testbed of spectra in Chapter 7. The results and running times are given and evaluated. Finally, in Chapter 8 we conclude this work by summing up the results and we give many areas for improvement and further investigation. We also mention other fields of application for our algorithms. As computer based data interpretation in this field has only just begun, there are plenty of possibilities for further research.

1.1 Graph theoretical notation

Graphs are the basis of the computational problems studied in this work. On a more application-based view graphs can be seen and are often referred to as networks. A *graph* is a pair of a vertex set and an edge set, $G = (V, E)$. As the graphs in this work are *directed*, an edge $e = (u, v) \in E$ with $u, v \in V$ is an ordered pair of vertices. We call u the *tail* and v the *head* of the edge e. For a given vertex v the edges $\{(v, w) \in E | w \in V\}$ are the *outgoing* edges of v, the *incoming* edges are $\{(w, v) \in E | w \in V\}$, respectively. The union of these two edge sets are the *incident* edges of v. Following the convention in graph theory, the number of vertices $|V|$ is denoted by n, the number of edges $|E|$ is m.

A *subgraph* $G' = (V', E')$ of $G = (V, E)$ is a graph for which it holds that $V' \subseteq V$ and $E' \subseteq E \cap V' \times V'$. The subgraph *induced* by $V' \subseteq V$ is the graph $G[V'] = (V', E')$ with $E' = E \cap V' \times V'$.

The graphs occurring in this work are directed acyclic graphs (DAGs) meaning that they do not contain directed cycles. Vertices of the DAG that do not possess incoming edges are source vertices or sources of the DAG.

A *vertex-coloured* graph is a graph with an additional colour function $c : V \mapsto C \subseteq \mathbb{N}$ that assigns a colour from the set of colours C to every vertex. (Of course, there are edge-coloured graphs, too, but these are not relevant to this work.) To enable scoring of the results, it is also necessary to introduce edge-weighted graphs. Those possess a *weight function* $w : E \mapsto \mathbb{R}$ giving each edge a score or weight. In this work, we look at edge-weighted vertex-coloured DAGs. We define the weight of a graph $G = (V, E)$ as $w(G) = \sum_{e \in E} w(e)$.

Trees are defined as connected graphs containing no cycles at all, whereas DAGs may contain cycles when disregarding the direction of the edges. We call directed trees *arborescences*, if there exists a vertex (called *root*) from which exactly one directed path to every other vertex exists. More intuitively speaking, it means that all edges point away from the root. Any tree we consider here is also an arborescence, so for the rest of this thesis whenever we write "tree" we in fact mean "arborescence tree".

In the algorithms developed and analysed in this work the Maximum Spanning Tree problem is often used. In literature it is commonly called Minimum Spanning Tree: Whether the spanning tree with maximum or minimum weight is calculated does not influence the general concept. Because a score is maximised here, the maximisation problem is more relevant. First we need to define the spanning trees of a graph:

Definition 1 (Spanning tree)**.** A spanning tree S of a connected graph G is a subtree of G that connects all vertices of G.

Now the the Maximum Spanning Tree problem can be defined:

Definition 2 (Maximum Spanning Tree)**.** Input: An edge-weighted connected graph G. Task: Find a spanning tree of G with maximum weight.

This problem has been known for nearly one hundred years, [Bor26, Kru56, Pri57] present algorithms to solve it.

The last graph theoretical concept relevant in this work is the transitivity of graphs. A graph is transitive if the following holds: $(u, v) \in E \wedge (v, w) \in E \implies (u, w) \in E$. That is, every vertex has edges to the children of its children, its great-grandchildren and so on.

This concludes the introduction of graph types and terms, and we now have a look at the main algorithm design concept applied in this work.

1.2 Fixed-parameter tractability

The problem studied in this thesis, determining the most likely fragmentation process, is formally defined in Section 5.1 and proven to be NP-hard in Section 5.2. That means it is very unlikely that a polynomial time algorithm exists to solve this problem. Thus the running time of any algorithm for this problem will increase exponentially with input size, rendering even the fastest computers useless for large inputs.

A possibility to tackle NP-hard problems is accepting non-optimal solutions. We can then use heuristics, randomised or approximation algorithms. They might perform well in practice, but only rarely guarantees can be given about how well the solution approximates the optimum or how long the calculation will run.

Due to these drawbacks, we apply another strategy in this work, named "fixed-parameter algorithms". This technique delivers exact solutions in acceptable time for NP-hard problems with a special problem structure. For this technique it is necessary to derive a parameter from the input. This parameter is usually a non-negative integer, but that is not a requirement of the underlying theory. In coloured graphs, a typical parameter is the number of colours. The algorithm developer tries to restrict the unavoidable combinatorial explosion, that is, the exponential growth to this parameter. If the parameter is small, calculating an exact solution for the problem can be done in acceptable running time, regardless of the problem size. To formulate it more exact from a theoretical point of view: If the parameter is fixed, and we consider it as a constant, the algorithm computes the solution in polynomial time. Hence this technique is called "fixed-parameter tractability".

1.2.1 Formal definition of fixed-parameter tractability

We formally define the basic concepts of fixed-parameter tractability here, beginning with the special problem structure necessary for fixed-parameter tractability to be applied.

Definition 3 (Parametrized language)**.** A *parametrised language* (or problem) L is a language $L \subset \Sigma^* \times \Sigma^*$, with Σ as finite alphabet. We call the second component of an instance of $\Sigma^* \times \Sigma^*$ the *parameter*.

In which cases can we consider a parametrised problem fixed-parameter tractable? We define this property by the running time of the algorithm that solves the corresponding decision problem:

Definition 4 (Fixed-parameter tractability)**.** A parametrised language L is *fixed-parameter tractable* if the question "Is (x, k) contained in L?" can be decided in $f(k) \cdot |x|^{O(1)}$ time with f being a computable function only depending on k. The corresponding complexity class containing all problems that are fixed-parameter tractable is called FPT.

It is now possible to define a concept of parametrised reducibility to compare the hardness of two parametrised problems, and a class of all problems parametrised reducible to the k-Step Halting problem. We call this class W[1]. Problems in this class are most likely not fixed-parameter tractable. As the problem covered in this work is in FPT, the details of W[1]-hardness are not discussed. The interested reader is referred to [Nie06, DF99].

It is important to keep in mind that FPT is a theoretical concept. Although most FPT algorithms are useful in practice, that is not always the case. As the function $f(k)$ may be extraordinary fast growing, an algorithm still including the input size in the exponential part can be faster for practical instances of the problem, albeit the worse asymptotic running time.

1.2.2 Design approaches for fixed-parameter algorithms

According to [Nie04] three major concepts are typically applied to design fixed-parameter algorithms:

- Reduction of the input to a small problem kernel
- Constructing a depth-bounded search tree
- Using dynamic programming

Reduction rules and problem kernels. This is probably the most widely used concept in parametrised algorithmics. The idea is to find rules that reduce the problem instance, e.g., for a graph they decrease the number of nodes or edges either by merging them together, or simply deleting them. Of course, the modifications must preserve the optimal solution. For some problems, it can be shown that these reduction rules always reduce the problem to a size only dependent on the parameter k. The remaining, irreducible problem is in that case called the *problem kernel*. If the reduction rules can be applied in polynomial time, an algorithm with exponential running time can then be applied to the problem kernel. This procedure yields a fixed-parameter algorithm with running time $f(k) \cdot |x|^{O(1)}$.

Depth-bounded search trees. This approach is often applied if the parameter is the size of the expected output. The parameter then is a bound for the depth of the search tree, as every decision made increases the output by at least one unit. Thus it remains for the algorithms engineer to reduce the width of the search tree by intelligent decision rules. The width of the tree needs to be independent from the input size, except for the parameter, otherwise the approach will not yield an FPT-algorithm.

Dynamic programming. Well known for speeding up algorithms for some polynomial-time solvable problems, we can of course apply dynamic programming also to NP-hard problems. It will yield an FPT-algorithm if the parameter is the size of a set of elements, that may or may not be included in the solution, e.g., the set of colours in the algorithm that is presented in this work. The major disadvantage of this approach is that not only its running time, but also its memory consumption will grow exponentially in the parameter. Note that the last two concepts do not exclude the first one: on the contrary, there exist efficient combinations of these approaches.

This concludes the computer science introduction, presenting the theoretical concepts behind the major technique applied here. Let us now have a look at the biological application, again beginning with some background.

Chapter 2

Biological and experimental background

The aim of this work is to analyse data obtained from measurements of metabolites. Section 2.1 of this chapter describes what metabolites are and why they are a focus of research. The concept of systems biology and the need for high throughput methods in this field is explained in Section 2.2.

Afterwards in Section 2.3 we present the technologies to measure the data analysed in this thesis, namely high performance liquid chromatography and tandem mass spectrometry. We introduce the terms commonly used in mass spectrometry analysis in Section 2.3.4 and present some previous work on mass spectrometry analysis in Section 2.3.5.

2.1 Metabolites

Metabolites are the substrates and products of chemical reactions taking place in living cells. Although this definition includes all compounds found in a cell, biologists and biochemists usually restrict the term to small molecules, but small is not exactly defined. As a rule of thumb the products of polymerisations are no longer considered metabolites: Thus proteins and DNA are not metabolites, but amino acids and nucleotides are. Metabolites rarely have masses of more than 1000 Da, the majority has a mass below 400 Da (KEGG database [KGH$^+$06]).

Metabolites are commonly divided into two groups: Primary metabolites and secondary metabolites. The former are molecules directly involved in growth, development and reproduction. All others are classified as secondary metabolites. Secondary metabolites have various functions, e.g., serving as signalling molecules, defending against pathogens, facilitating reproduction as “attractive smells” or colour-

ing agents or protecting against abiotic stress, such as UV light or high salt concentrations in plants [JCB+00].

Whereas the primary metabolites are well investigated and are completely covered by databases such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) [KGH+06], the Wiley Registry, or the NIST 2005 Mass Spectral Library [Joh06], a huge portion of secondary metabolites is completely unknown. There exists a huge diversity of secondary metabolites in living organisms. As far as we currently know, no two species possess the same set of secondary metabolites [PG00]. Some are shared between species, e.g., species of the same family; others are unique to a single species. Secondary metabolites are especially abundant in plant signal transduction. Even in the model plant Arabidopsis thaliana where 200 secondary metabolites are already known [DG05] the number of genes coding for enzymes of the secondary metabolism suggests that there are a lot more metabolites to be found [Ini00]. The whole field of bio-prospection searches the rain forest and other unexplored and biologically diverse areas for molecules, often secondary metabolites, that might serve as pharmaceuticals. This illustrates the need of "de-novo"-identification of metabolites from mass spectrometry data as it is insufficient to rely on databases in this matter.

2.2 Systems biology and metabolomics

In systems biology, scientists consider the cell or even larger structures like tissue or organisms as systems, that is, as a set of components and their interactions. If looking at all components at once, it might be possible to find out how function and behaviour emerge from this system. Some biologists describe this as a complete change of philosophy, as in classical biology the concept was to dissect systems to gain insight into their structures. In research, the systems biology concept can only be used because high throughput methods became available.

The genome was the first subsystem available for analysis due to fast sequencing methods. But it is static in most cases and does not describe the state of a cell. For capturing this as well, it is necessary to take the variable components of a cell into account, too: the transscriptome, the collection of all mRNA transcripts in the cell, the proteome, the collection of all proteins, and the metabolome, the collection of all metabolites. For all four types high throughput methods are available and constantly improved. For the studies of the latter two mass spectrometry plays an important role.

In proteomics, experimentators can separate proteins using 2D-gel electrophore-

sis, dividing them by their iso-electric point and their mass. Afterwards they are digested by an enzyme or otherwise dissociated, then the fragments are detected by a mass spectrometer. An alternative is the separation by liquid chromatography (LC) and the fragmentation using "collision induced dissociation" (CID) in the mass spectrometer. For more details on mass spectrometry in proteomics, see the references in Section 2.3.5. For metabolomics, gas or liquid chromatography in combination with mass spectrometry is used. We will describe these methods in detail in the next section. In a systems biology approach, researchers have to study the metabolome since it represents the most direct view onto the state of a cell. They can, for example, deduce the nutritional status from the ATP level of a cell. The control function of metabolites is also not to be missed; metabolites often ensure that their own level is kept constant by a feedback interaction to enzymes, transcripts, or genes. Another very important point is that organisms usually interact with their environment via metabolites. So, a reaction of an organism to adjust to a changed environment often starts by sensing metabolites. The methods applied to measure metabolites are presented in the following section.

2.3 Methods

2.3.1 Chromatography

As mentioned in Section 2.2, chromatographic methods are widespread to separate metabolites. The idea is to separate a mixture of analytes by first solving it into a so-called mobile phase, increasing its mobility. It is then pressed along a stationary phase, which can bind to some molecules in the sample: For example, the stationary phase can have hydrophobic tails to ease the binding of hydrophobic molecules. The experimentator then changes the properties of the mobile phase gradually to be more hydrophobic. Thus first the slightly hydrophobic molecules are washed off and exit the apparatus and afterwards the more hydrophobic ones until all analyte molecules have left the device. To separate metabolites, gas chromatography (GC) and high performance liquid chromatography (HPLC) are commonly used. They differ in the state of the mobile phase. For historical reasons, the device in which the stationary phase is contained is called column although nowadays metal tubes are in use. Biologists usually use capillaries, that is, microscopically small columns, e.g., with 150 μm diameter. This small diameter ensures a high velocity of the mobile phase enabling longer columns. In such a way the experimentator can achieve better separation resulting in a higher resolution.

2.3.2 Mass spectrometry

Mass spectrometry is a technology to analyse chemical compounds. Essentially, a mass spectrometer is a very exact scale, determining the molecular weight of a chemical species. To achieve this, the ion source ionises the sample molecules and then accelerates them in an electromagnetic field. The mass analyser separates the ions according to their mass-to-charge ratio m/z before they are measured by a detector. This technology can analyse the contents of the sample both qualitatively and quantitatively. This section only introduces the main concepts, for details on mass spectrometry we refer to [Das01, HS01].

A mass spectrometer therefore consists of at least three parts: The ion source, the mass analyser, and the detector. Different variants of all these parts are in use. These variants of course change the properties of the spectra produced, so that the interpretation approaches differ for different mass spectrometer types.

There are three ionisation methods that are commonly used with biological samples: When using Electro Spray Ionisation (ESI), the sample is atomised into the vacuum and then ionised by an electric field. This leaves most of the molecules intact. Alternatively, the mass spectrometrist can use Matrix Assisted Laser Desorption/Ionisation (MALDI). In this case the sample is crystallised together with a matrix substance. A laser pulse then evaporates the matrix without damaging the embedded sample molecules. This ionises the molecules. The third ionisation method is Electron ionisation (EI). Here, electrons are accelerated towards the sample by an electric field. The collision with the electrons in this case fragments the sample molecules. The degree of fragmentation can be adjusted by the strength of the electric field. In this work we analyse spectra measured using Electro Spray Ionisation.

As a wide variety of mass analysers is in use, this paragraph will focus on the two types relevant for this work: Time-of-flight analysers make use of the fact that molecules of different weights gain different velocities, if accelerated by the same force ($\vec{a} = \frac{\vec{F}}{m}$). Because the acceleration is in practice carried out by an electric field the force acting upon the ion is proportional to its charge ($\vec{F} = z \cdot \vec{E}$). We obtain $\frac{m}{z} = \frac{\vec{E}}{\vec{a}}$. Since mass and charge are both unknown in this equation, mass spectrometry can only measure mass-to-charge ratios.

The other analysing technology relevant here is the quadrupole mass analyser. It consists of four parallel rods connected to an AC power source. The induced alternating electrical field forces the ions into a spiral trajectory. For a fixed frequency of the power source only ions with a distinct mass-to-charge ratio can pass without colliding with a rod. Therefore a quadrupole is a filter adjustable by the

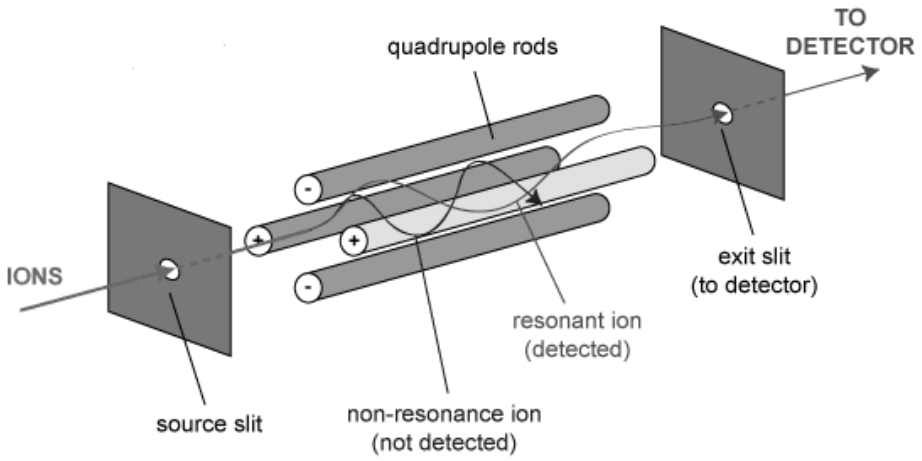


Figure 2.1: Schematic drawing of a quadrupole mass analyser. Picture by Paul J. Gates, University of Bristol.

AC frequency. Figure 2.1 shows a schematic drawing of a quadrupole analyser.

After passing the mass analyser unit a detector records the ions. Commonly used are the secondary electron multiplier and the Faraday cup. In the electron multiplier, the ions hit a series of metal plates. By a physical process called secondary emission, each ion or electron hitting a plate forces the emission of two or three electrons. Each plate hence doubles the amount of electrons, leading to a strong amplification. Alternatively a Faraday cup can be used. In its simplest form it is a metal cup, which is charged when hit by ions. This small change in charge can then be measured. Faraday cups lack sensitivity but are more accurate than electron multipliers, as the ions are directly measured.

Peak picking The raw data received from the detector is filtered to reduce noise. Afterwards the so called "baseline", the base activity of the detector, is subtracted. In tandem mass spectra, the technology relevant for this thesis, the maxima of the remaining data function are defined to be peaks. The environment of these maxima is then centroided, that is, the intensity-weighted mass average is calculated. These calculations result in peaklists containing mass, intensity and perhaps other properties of the peak, e.g., the width of the peak in the raw data. Software to perform these calculations is usually provided by the mass spectrometer vendor. Alternatively, open source software can be used for this task [SWO$^+$06, LGR$^+$06]. Common mass spectrometry analysis tools and the algorithms presented in this work use these peaklists as input.

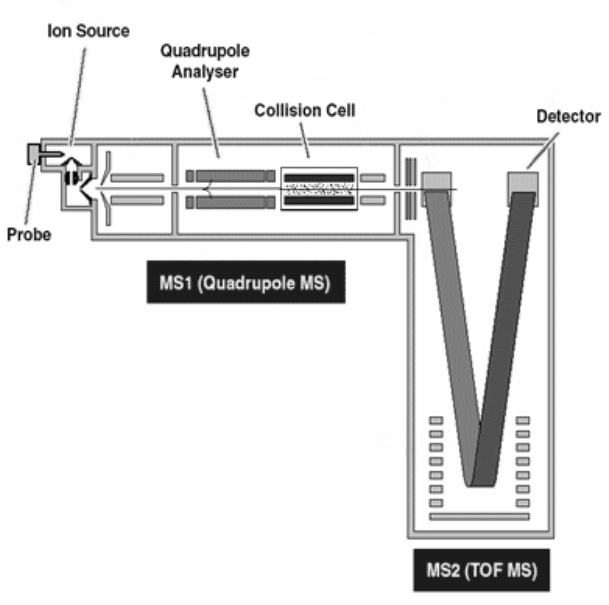


Figure 2.2: Basic layout of a QqTOF tandem mass spectrometer. Picture by the Protein Analysis Unit of the William Harvey Research Institute, London.

2.3.3 Tandem mass spectrometry and fragmentation

A mass spectrometer is not limited to having only one mass analyser. The data analysed in this work was acquired using as many as three coupled mass analysers. The first one is a quadrupole analyser used as mass filter to select one specific metabolite. The second one is again a quadrupole, but filled with nitrogen or argon. By this way, it can be used as a collision cell in which collision-induced dissociation occurs which fragments the analyte into smaller molecules [WM05]. Note that typically only one part of the analyst ion remains charged, the rest of the molecule is not detectable and therefore called neutral loss. Depending on the acceleration voltage applied before the collision cell, (the collision energy,) large fragments result from weak collisions or small ones from heavy colliding. A time-of-flight analyser separates the fragments produced and they finally reach the detector. This combination (in short QqTOF-MS) is a type of tandem mass spectrometry (MS-MS), as the second quadrupole is not used for analysis [CLT01]. Figure 2.2 shows the layout of such a tandem mass spectrometer and Figure 2.3 depicts an example set of spectra using different collision energies. The advantage of the technology is that one gets information not only about the mass of the molecule, but also masses of different fragments. This fragment information limits the explanations of the parent peak, because only explanations that can fragment into the ions found in the spectra are candidates for the real metabolite. Unfortunately, the fragmentation process is not

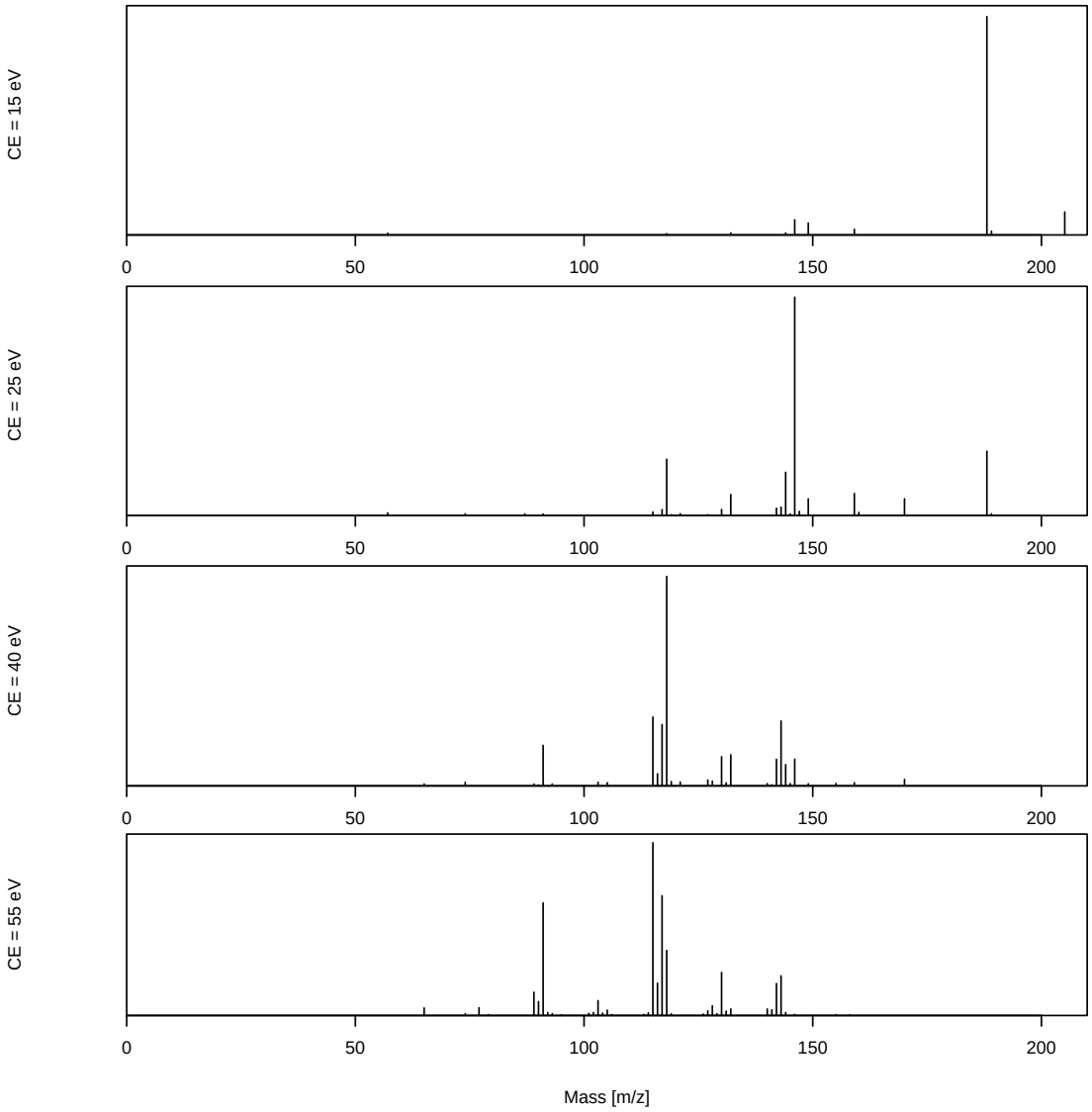


Figure 2.3: Tandem mass spectra of tryptophane at different collision energies (CE). The parent mass of tryptophane is appr. 205 Da.

too well understood.

2.3.4 Mass spectrometry terms

The following terms are commonly used when analysing mass spectra. Recall that in this work, we want to discover the sum formula of the metabolite only, and we ignore its structure. Note that in graph theory, the term parent is used differently than in mass spectrometry. We will retain the meaning of parent as used in mass spectrometry, and use the expression predecessor whenever meaning parent in its graph-theoretical sense. This leads to the following definitions:

The *parent ion* is the unfragmented ion which can pass the first quadrupole

analyser. The *parent peak* is the peak created by this ion and the *parent mass* analogously its mass. Usually, the peak with the highest mass in the spectra is the parent peak. It only occurs in spectra with a low collision energy, at higher energies it will be completely fragmented. All other peaks in the spectra are consequently *fragment peaks*. We calculate *decompositions* for the peak masses, that is sum formulas which have approximately the mass of the peak. Those decompositions then *explain* the peak. For a certain decomposition of a fragment peak, we call the decompositions of which this decomposition can be a fragment of its *predecessors*. The corresponding peaks are then *predecessor peaks*. These predecessors become important for the collision energy scoring in Section 4.4.2. Note that in graph theory, one usually calls them parents, but this term is occupied in mass spectrometry.

2.3.5 Previous work on mass spectrometry analysis

For the analysis of protein mass spectrometry data, many programs exist. MASCOT and SEQUEST are commercial tools widespread among biologists [PPCC99, EMY94]. VEMS poses a freely available alternative to these programs [MBS+04]. Whereas all these programs depend on a database for analysis, algorithms for protein de-novo sequencing from tandem MS data have been developed some years ago. Chen et al. proposed a concept that calculated the linear combinations of the amino acid masses that matched a mass difference between two peaks [CKT+01]. Bafna and Edwards improved this idea by finding a concept to assign confidence values to the peak explanations [BE03]. The most successful approach though was made by Fischer et al. They used hidden Markov models (HMM) to produce an amino acid-spectra mapping, which turned out to work well [FRR+05].

Up to now, significantly less effort has been spent on analysing metabolite mass spectrometry data. The classical approach is to use an extensive database, such as the Wiley or the KEGG database [Joh06, KGH+06], and compare the peaks with the masses of the molecules contained in this database. The tools for the comparison of protein spectra (MASCOT, SEQUEST and VEMS) also work for metabolite spectra. If analysing small molecules, there are usually only few matches [Fie02].

Unfortunately, this method is insufficient if larger metabolites have to be identified. The approach used in this case is to measure a tandem MS spectrum and to compare the fragment spectra with the ones in the database. Because spectra can differ a lot depending on the experimental conditions, this is rather inexact. There is another major drawback: As mentioned in Section 2.1 many secondary metabolites, especially in plants, are unknown. To depend on a database for identification is therefore not an option. De novo identification approaches are an alternative here,

as the software for proteomics analysis developed in the last years shows. Some of the approaches used to analyse peptides can be transferred to metabolites. For example, the approach of Zhang et al. exploits the isotopic patterns of the fragment ions [ZGC+05].

Finding all sum formula for a given mass leads to the long known Money Changing Problem [Wil78]. Interpreting the atom masses as coin values and the given mass as payable amount, the sum formula tells which combination of coins you may use. Böcker and Lipták present an efficient algorithm to solve this problem [BL07]. Unfortunately the number of combinations increases rapidly with mass. For k different atom types, it is in $O(m^{k-1})$. If high resolution mass spectrometry data is available, analysis software can use the isotopic pattern of the molecule to rank the possible solutions according to their ability to explain the pattern [BLLP06]. Unfortunately this only works well for spectra with less than 2 ppm mass deviation. Only expensive and complex spectrometers, such as Fourier-transform ion-cyclotron-resonance (FT-ICR-MS) or OrbiTrap spectrometers can measure spectra with this accuracy. Another possibility is to exploit the fragment spectra obtained by tandem MS, which is the aim of this work.

Chapter 3

Concept of the Analysis

As stated in the introduction, "de-novo"-identification of metabolites using mass spectrometry can only reveal sum formulas, since mass spectrometry only measures molecular weights. These masses do not give information about the structure, but identifying the sum formula is a first step preparing further investigation.

The idea for finding these sum formulas is to calculate all possible formulas for any peak present in the spectra. These formulas are connected, if one can be a fragment of the other. We score these connections with the likelihood that this fragmentation occurs. Chapter 4 describes the scoring scheme. In Section 3.3 we explain the transformation of the biological data into a graph in detail. From this graph the algorithm calculates the most probable fragmentation tree. The root of this tree is therefore our best explanation for the parent peak. We briefly describe concept of the algorithm in Section 3.4; the details follow in Chapter 5.

We perform two steps before calculating the input graph. First, we merge peaks of different spectra that have the same mass. Section 3.1 presents the details of this peak preprocessing. Afterwards we apply some filters to reduce the number of possible decompositions. These filters are described in Section 3.2.

3.1 Peak preprocessing

In this work series of spectra with different collision energies are analysed, but we shall treat the data as if it were only a single spectrum. Hence it is necessary to merge the peaks into one spectrum before the main analysis can take place. This step is done by simply applying a threshold, merging peaks from different spectra whose mass difference is smaller than the threshold. Another restriction is that the peaks have to be in adjacent spectra. For example, if peaks with similar mass occurred in the spectrum with 15 eV and with 35 eV, but not in the spectrum with 25 eV,

the program would not merge them, as they most likely have different explanations with incidentally the same mass: otherwise, a peak at 25 eV should also exist. The peaklists in Table 3.1 on page 31 illustrate the merging process. Peaks in the same row are merged, resulting in six peaks with distinct masses.

If peaks are to be merged, the intensity of the merged peak is the highest intensity of the original peak. There are two possibilities for the mass of the resulting peak: The standard is to calculate the intensity-weighted mean of the original masses, another possibility is to just keep the most intense peak. If peaks are merged it is stored in which spectra this peak occurred. Thus after merging, every peak has a range of collision energies (of the corresponding spectra) assigned, which becomes relevant for the scoring using collision energies in Section 4.4.2.

The next step is to calculate the decompositions of all these peaks. The software uses the Round Robin algorithm for this task [BL07]. We then need to filter these decompositions.

3.2 Filters

We apply only a few strict filters as long as the problem instance and hardware resources permit this. Not using filters avoids the necessity for thresholds and allows finding a solution even if it has an unexpected property. This property could prevent the solution from passing a strict filter. With scoring functions, as mainly used here, solutions with unexpected properties will just receive a lower score, which can be compensated by earning a higher score for other properties. There are only two strict filters used in this work, one of them is intrinsic in the algorithm for sum formula calculation. The other strict filter allows only decompositions passing the Senior rule filter (Section 3.2.2) to be analysed.

3.2.1 Mass deviation filter

Although it is not applied explicitly, this is one of the two strict filters being enforced, simply because the mass decomposition algorithm requires a mass range for which to calculate the decompositions. The mass deviation is usually given as parts-per-million (ppm) of the peak mass. The threshold here has to be set by the user, as it highly differs between mass spectrometers. Current Q-TOF-spectrometers can obtain a precision of 3 ppm, whereas old devices only manage between 10 and 20 ppm. The parent masses of the molecules measured also influence precision, as not only relative, but also absolute mass errors may occur, making it necessary to choose a higher relative deviation.

Energy	15 eV		25 eV		40 eV		55 eV	
	Mass	Int.	Mass	Int.	Mass	Int.	Mass	Int.
Peaks					144.02	3.07	144.02	54.66
					176.05	6.82	176.05	66.28
							205.05	11.27
	220.08	5.21	220.08	100.00	220.08	100.00	220.08	100.00
	441.21	100.00	441.21	39.29				

Table 3.1: Peaklist obtained by measuring hexosylferuloyl choline. The given intensities are raw intensities as described in Section 4.1.1. An input graph derived from these spectra is shown in Figure 3.1.

3.2.2 Senior rule filter

The second strict filter being applied is a Filter based on Senior's third theorem, that the sum of valences has to be greater than or equal to twice the number of atoms minus one [Sen51]. This rule is equivalent to restricting the RDBE (defined in Section 4.3.3) to non-negative values. Molecules violating Senior's third theorem are rare, especially in natural compounds. Kind and Fiehn find 64 substances violating the rule in the 45.000 entries of the Wiley mass spectral database [KF07]. Thus the filter has a sensitivity of 99.86 %. There are two examples with negative RDBE given by Kind and Fiehn, $C_{12}H_{36}F_6N_6O_4P_4Si_2$ and $CH_2F_{10}S_2$. The high amount of fluorine in both examples is not likely to occur in natural compounds.

3.3 Construction of the input graph

We construct the input graph from the remaining decompositions as follows: For every sum formula that passed through the filters we create a vertex representing it. We colour vertices that represent sum formulas explaining the same peak in the same colour. Thus the number of colours k is equal to the number of peaks. Now, we build a directed graph by applying the relation "can be fragment of". If sum formula a can be a fragment of sum formula b, we create the edge $b \rightarrow a$.[1] We now score those formulas with the intensity of the peak they explain, and the complementary error function of the mass difference to the peak. Then, we assign the scores of the formulas to all incoming edges of the corresponding vertex. It is possible to apply further scores on the edges, such as the likelihood that the neutral loss this edge implies occurs. This way, we build an edge-weighted, vertex-coloured directed acyclic graph (DAG). An example graph is shown in Figure 3.1.

[1]The direction is chosen arbitrarily, it only needs to be consistent during construction. In this work, we choose the direction in a way that the resulting fragmentation tree is an arborescence.

3.4 Concept of the algorithm

For the graph constructed above, the algorithm calculates the tree that does not use any colour twice and has maximum sum of edge weights. The root of this tree is claimed to be the most likely explanation for the analysed molecule, if the root is indeed a decomposition of the parent mass. The tree itself is a fragmentation graph. Whereas fragmentation graphs are not necessarily trees, this restriction is made to simplify computations as it avoids counting a score twice. The demand for the tree not to use a colour twice ensures that the algorithm selects only one explanation per peak. It is possible that a peak represents more than one ion, but this occurs so rarely, that it is ignored. If we allowed more than one explanation per peak, the algorithm would simply choose all explanations of a peak, which would certainly not represent reality.

The algorithm now chooses the explanation which allows for the highest scoring overall subtree. As the scores are chosen proportional to the probabilities that the corresponding fragmentation takes place, the parent peak explanation belonging to this subtree has a high probability to be the decomposition that generated this spectrum. Note that this tree does not necessarily represent the correct fragmentation tree, it is optimised to calculate a likely explanation of the parent peak.

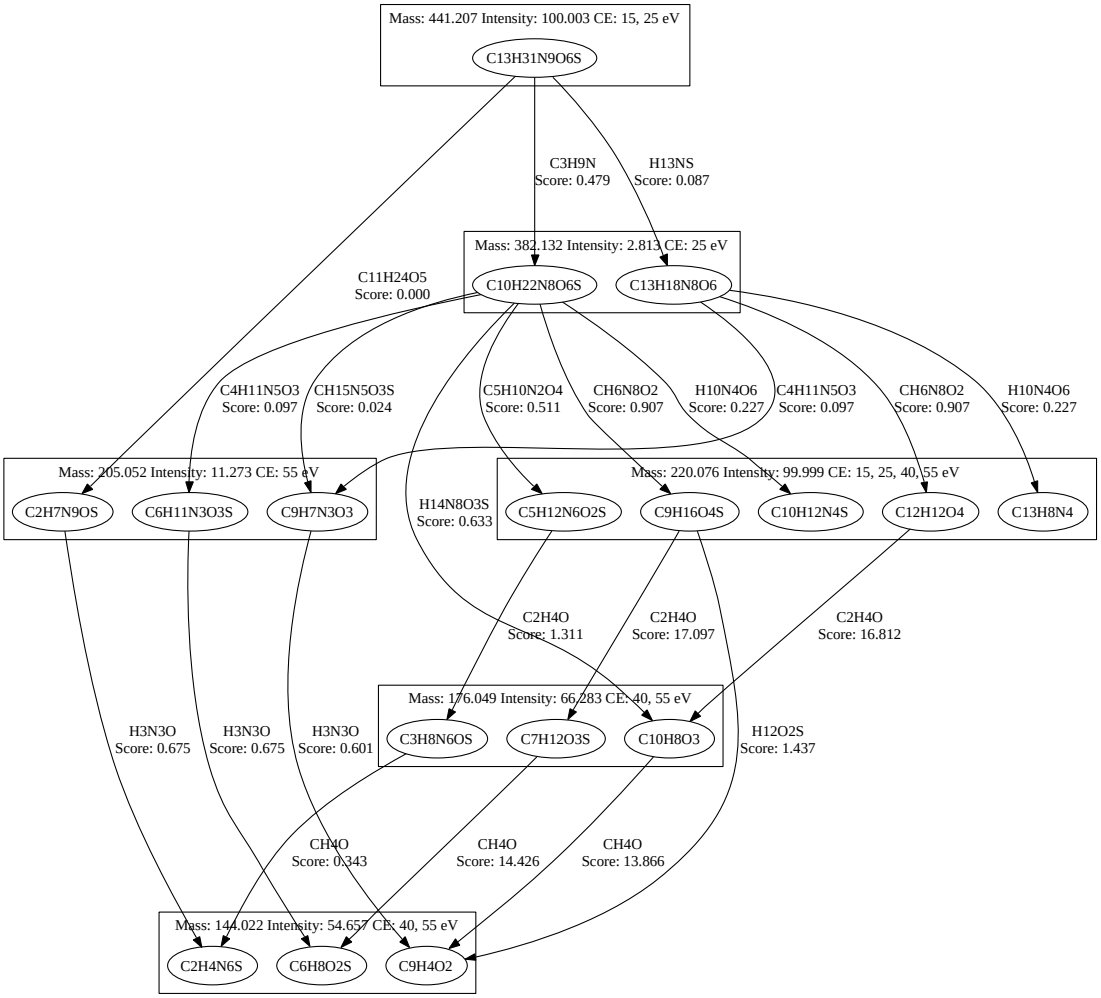


Figure 3.1: Extract of the graph derived from the spectra of hexosylferuloyl choline. For the sake of clarity, only one decomposition of the parent mass and its descendants are shown and the transitive edges have been removed. Each box represents a peak, thus vertices in the same box have the same colour. All peaks in the spectra of hexosylferuloyl choline are given in Table 3.1. Note that $C_{13}H_{31}N_9O_6S$ is not the correct decomposition. "CE": collision energy

Chapter 4

Scoring

As stated in the previous chapter, the score assigned to an edge roughly represents the likelihood that the corresponding fragmentation step is real. The simplest scheme would be to use a peak counting score, that is, to count every peak the suggested decomposition explains. This idea is too simple, since many decompositions manage to find an explanation for every peak in the spectra analysed. For example, take the wrong explanation $C_{13}H_{31}N_9O_6S$ for hexosylferuloyl choline in Figure 3.1. We can easily find a tree, that explains every peak exactly once. Thus, this explanation would, together with many others, receive the maximum peak counting score of 6. We need to use some properties of the peak, the decomposition, and the fragmentation step to score explanations.

The basis of these scores are peak intensities as Section 4.1 describes. The difference between the mass of the candidate formula and the peak is also taken into account. Afterwards, certain properties of the sum formula are exploited, but this has to be done with care, as we describe in Section 4.3. Finally, properties of the fragmentation step itself are used as Section 4.4 describes.

Because all scores represent likelihoods, we would multiply them with each other. Since multiplications are time-consuming and may produce underruns, we logarithmise the scores calculated in the following sections. This allows to add the scores, rather than multiplying them. The score of an edge consists of the following parts: The intensity score, the mass deviation score, the decomposition properties score and the fragmentation score. Peak and decomposition attributes are taken from the head of the edge, that is the vertex the edge points to.

4.1 Peak intensities for scoring

The first value we use for scoring is the peak intensity. It is easy to understand that a solution explaining stronger peaks should receive a higher score, thus, the intensity of the peak has to influence scoring. There is one problem in this concept, however: Peak intensities are usually normalised within a single spectrum. The strongest peak is set to a defined value and the other peaks are scaled relatively to this value. Therefore the intensities of two spectra are not comparable. There are two possibilities to overcome this restriction. We describe the possibilities in the following two sections.

4.1.1 Raw intensities

The idea here is to undo the previous scaling. To achieve this, it would be necessary to know the intensity of the most intense peak before scaling. Unfortunately, this is usually not stored, mass spectrometrists store the total ion current (TIC) instead. This is the total current measured by the spectrometer over the whole spectrum. We can derive pseudo raw intensities for the peak p in a spectrum S represented as set of peaks from this value by the following calculation:

$$\text{rawIntensity}(p) = \frac{\text{TIC}(S)}{\sum_{q \in S} \text{intensity}(q)} \cdot \text{intensity}(p)$$

These raw intensities are then logarithmised and used as score for all decompositions of this peak.

4.1.2 Smoothed intensities

Another possibility is to loosely adopt a concept by Wan et al. [WYC06]. They rank the peaks of a spectrum according to their intensities in descending order. Afterwards they divide the ranks by the highest rank in the spectrum, thus normalising them between 0 and 1. They find that the meaningful peaks are exponentially distributed on this scale. Therefore they can use the value of the corresponding probability density function as a score. This procedure requires annotated training data to confirm that the assumptions also hold for metabolite spectra and to derive the distribution parameters from. Unfortunately, no annotated data was available. Therefore in contrast to Wan et al., we sort the intensities in ascending order and normalise in the same way. These relative ranks are then used as score for the de-

compositions of the peak. Thus we avoid the scaling problem and deriving the score from the rank flattens large differences in the intensities. This effect and the advantages are similar to the Spearman rank order correlation. We believe the ranking of the intensities to be correct, although the actual intensities are random to a certain extent. It will be even better to fully adopt the idea as described above, if annotated data becomes available.

4.2 Scoring of decompositions

After scoring the peak intensities the decompositions are scored separately. How are decompositions represented as weighted vertices scored, when our algorithms are restricted to edge-weighted graphs? As we will see below, we also will score the fragmentation steps that are represented as edges. We then add the vertex scores to all incoming edges of the corresponding vertex, and discard the vertex scores. As we only allow one incoming edge per vertex, every vertex included in the tree is therefore scored exactly once. Decompositions of the parent peak have to be handled separately, though, because they do not have incoming edges; we omit the simple details.

4.2.1 Mass deviation

The most common approach in mass spectrometry analysis is to score the deviation between the mass of the calculated decomposition and the measured peak. Since mass spectrometrists assume that the measuring error of a device roughly is normally distributed, we add the evaluation of the logarithmised Gaussian probability density function at the value of the measuring error to the score. As standard deviation $\frac{1}{3}$ or $\frac{1}{2}$ of the relative mass error is used, assuming that 99.8% resp. 95% of all measured peaks have a mass error smaller than the given value. Mass spectrometry analysis software typically uses the complementary error function, but using the probability density function increased the sensitivity of the resulting scoring term. So, we measure the probability that a peak has a mass error of *at least* Δm instead of the probability that a mass error of *exactly* Δm occurs.

4.3 Scoring decomposition properties relative to the preceding decomposition

The following four scoring procedures share the problem that the properties scored are hereditary: As an example, assume that we want to penalise decompositions with unusually high hydrogen-to-carbon ratio. A decomposition with high hydrogen-to-carbon ratio will likely have fragments with a high ratio, too. Thus, the high ratio will be penalised in every fragmentation step, which is not desirable. Therefore the following scores are applied in a special manner: The scores of the decompositions are transferred to the edges as mentioned in Section 4.2. Then the scoring values of both decompositions connected to the edge are calculated. If the score of the fragment is better than the score of its predecessor, the score is not changed. Otherwise the logarithmised score of the fragment is added, but the log-score of the predecessor is subtracted from the score determined so far. Thus we decrease the score of the edge.

The following four concepts highly dependent on the parameters chosen and subsequently on the data these are derived from. The user has to make sure, whether the assumptions hold for the metabolite class of interest and only then enable the corresponding scoring. For example, for the test data presented in Section 7.1 the H/C-ratios (Section 4.3.1) were lower than usual, probably due to aromatic rings. Thus the H/C-ratio scoring was not helpful, in contrast to the C/Hetero-ratio scoring (Section 4.3.2) which improved the results.

4.3.1 Hydrogen to carbon ratio

Furthermore we improve a concept introduced by Kind and Fiehn [KF07]. They use a strict filter on the hydrogen/carbon ratio of the calculated sum formulas. Once again it is more suitable for the analysis to just derive a scoring scheme. We find that the hydrogen/carbon ratio of the 5100 molecules in the KEGG database to be normal distributed as Figure 4.1 shows. The parameters estimated from these molecules are $\hat{\mu} = 1.44$ and $\hat{\sigma} = 0.50$. The corresponding density function is also shown in Figure 4.1. Kind and Fiehn do not consider the use hydrogen/carbon ratio to be normally distributed. They use the more extensive Wiley mass spectral library [Joh06] for their studies, but this library is not restricted to biological compounds. Therefore data based on a purely biological database is more reliable if only biological substances are of interest.

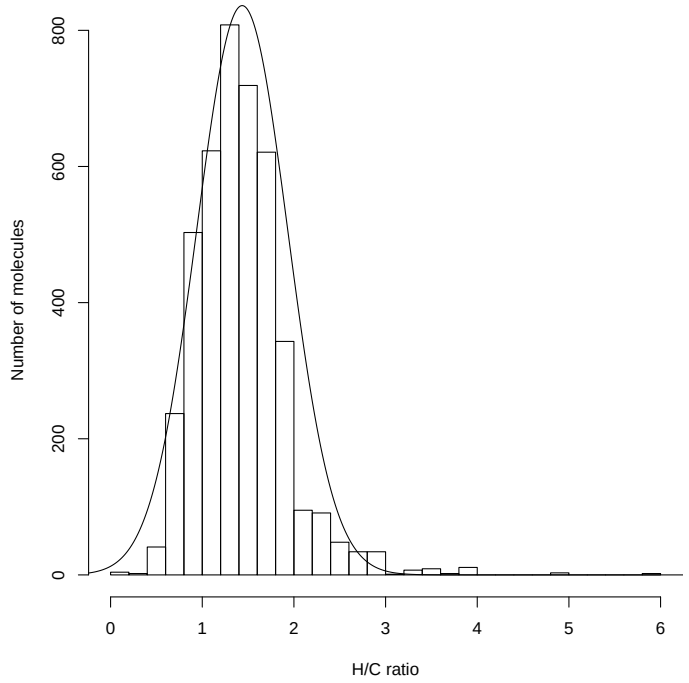


Figure 4.1: Frequency distribution of the H/C ratio of all molecules in the KEGG database and the corresponding scoring function.

4.3.2 Hetero atom to carbon ratio

In organic chemistry, all atoms not being carbon and hydrogen are called hetero atoms. The hetero to carbon ratio is as well a good measure for the likelihood of a sum formula to represent a really existing molecule as this ratio is typically between 0.25 and 1 in biologically relevant molecules. This scoring is again similar to Kind and Fiehn, but in contrast to their approach all hetero atoms are added together here.[1] We find the hetero to carbon ratio again to be normally distributed (see Figure 4.2). The parameter determined in this case are $\hat{\mu} = 0.59$ and $\hat{\sigma} = 0.56$. Statistics on the KNApSAcK database [SNAUA$^+$06] focusing on plant metabolites and the AraCyc database [MZR03] containing data only from Arabidopsis thaliana yielded roughly the same results. Thus the density function of this distribution is used for scoring. The parameter determined for this and the previous distribution of course depend on the molecule set used to calculate them.

[1]Kind and Fiehn consider the element to carbon ratios of eight frequent elements separately from each other. They examine nitrogen, oxygen, phosphorous, sulphur, three halogens and silicon.

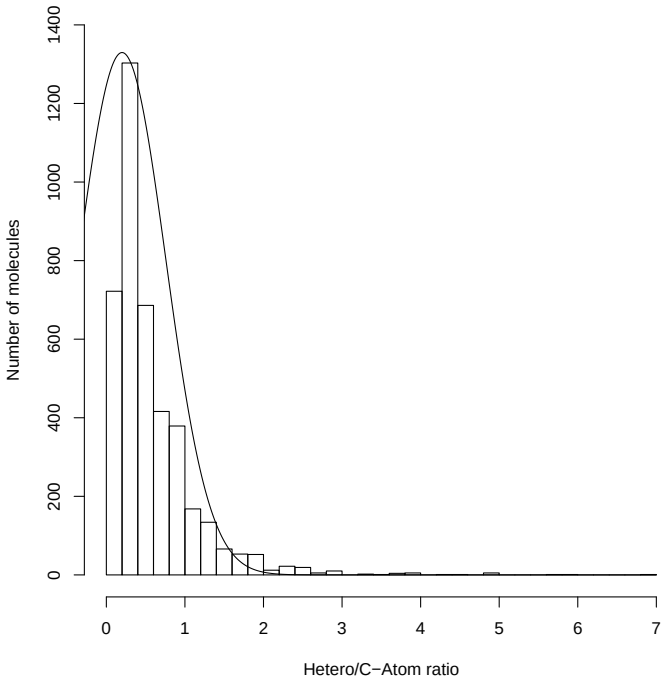


Figure 4.2: Frequency distribution of the Hetero/C-Atom ratio values of the molecules extracted from the KEGG database and the derived normal distribution.

4.3.3 Scoring using the RDBE distribution

The so called rings-plus-double-bonds equivalent (RDBE) measures the number of rings and double bonds in the molecule. We calculate it as follows:

$$n_v = \text{Number of atoms with valence } v \text{ in the molecule}$$
$$\text{RDBE} = 1 + \sum_{v=0}^{6} (\frac{1}{2}v - 1)n_v$$

It is not possible to use this formula to calculate the number of rings and double bonds from a given sum formula, as the valences of nitrogen, phosphorous and sulphur and other elements may vary. But the RDBE values calculated using the most common valences for these elements remain within a certain range for biologically relevant molecules.

By visual inspection of Figure 4.3 we assume that the RDBE values of molecules in the KEGG database [KGH$^+$06] are gamma distributed. The parameters of the distribution shown are the shape parameter $k = 2.6$ and the scale parameter $\theta = 3.39$. As the values between -1 and 2 are not well approximated by the distribution, we use counting statistics to assign the scores in this interval.

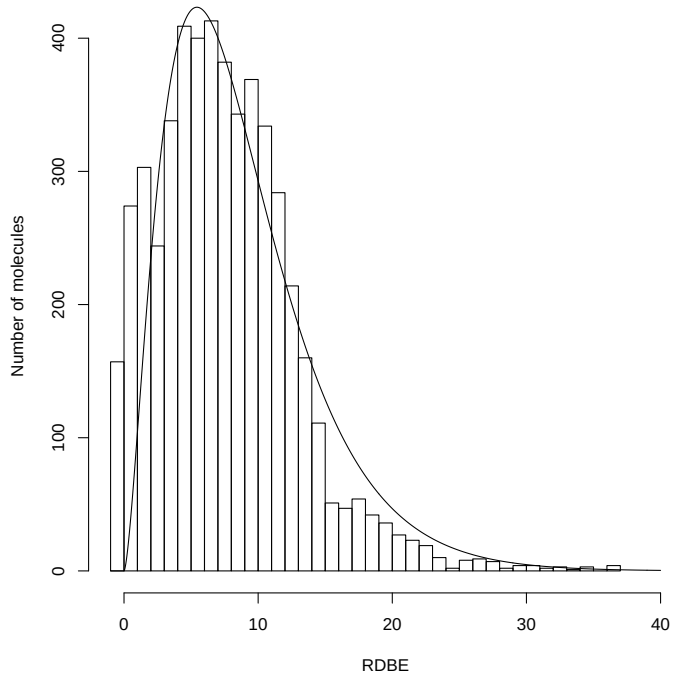


Figure 4.3: Frequency distribution of the RDBE values of 5100 molecules extracted from the KEGG database and the estimated gamma distribution used for scoring.

4.3.4 Bounds for element counts in the formula

Often researchers know which elements are likely or unlikely to appear in their sample. Therefore, the user should have the possibility to limit the occurrence of certain elements in the solutions to a certain range. Again we do not apply a strict filter, but derive a scoring function from the values given by the user. This function has a value of one in the given range and then decreases behind both boundaries. We suggest the following formula, where $N_{\mu,\sigma}$ denotes the density function of the normal distribution, $[a, b]$ is the interval chosen by the user, and x is the abundance of the element in question:

$$f(x) = \begin{cases} \sqrt{2\pi} N_{a,1}(x) & \text{for } x < a \\ 1 & \text{for } a \leq x \leq b \\ \sqrt{2\pi} N_{b,1}(x) & \text{for } x > b \end{cases}$$

The factor $\sqrt{2\pi}$ is necessary for scaling. Again, we logarithmise the value of $f(x)$. Although this concept might override the users choice, it prevents good solutions to be excluded only because they do not meet the requirements set by the user.

4.4 Scoring of the fragmentation process

The last scoring term rates the fragmentation steps. Since a fragmentation step is represented by an edge, these scores are of course applied to the edges. Often we exploit properties of the so-called neutral loss for these scores. The neutral loss is the fragment that is created during a fragmentation, but is not detected as it is uncharged. We calculate the neutral loss of a potential fragmentation step by determining the difference between the sum formula of the predecessor ion and the fragment ion.

4.4.1 Mass of the neutral loss

We add the logarithm of the complement of the ratio between the mass of the neutral loss represented by the current edge and the parent mass (In formula: $1 - \frac{\text{mass(neutral loss)}}{\text{parent mass}}$) to the score. This scoring does not make sense chemically, because large neutral losses are as likely to occur as smaller ones. Without this restriction the calculated fragmentation tree would often be a star, that is a tree where every vertex is connected directly to the root, because all fragments might as well be explained as direct fragments of the parent mass decomposition. To avoid this effect small neutral losses are favoured and therefore assigned a higher score.

4.4.2 Collision energies

Because the spectra are measured using different collision energies, we can deduce that some peaks can not represent direct fragments of other peaks if they either appeared at a lower energy or at a high energy where the predecessor peak has long disappeared. The ideal situation would be that there is a collision energy where both peaks appear. This will be given full score. It is highly unlikely that the fragment peak appears before the predecessor peak, therefore $\log(0.1)$ is added to the score. That effectively reduces the score by about 2.3. It is possible though, that the mass spectrometer did not detect the predecessor peak in the relevant spectrum, thus the score is not zero. Another possibility is that the predecessor peak ceases, there is one spectrum where it can not be found, and then at the next higher collision energy, the fragment peak starts to emerge. Then, the fragment did most likely not directly emerge from the predecessor, this connection is as well given a small score. If there is no spectrum in which both peaks can be found, but neither a spectrum containing none of the peaks in question, it is possible that the molecules are direct fragments, but there might as well exist another fragment between them.

Name	Condition	Score
Overlap	Highest energy of predecessor peak is larger than lowest energy of fragment peak	100%
Sequence	Highest energy of predecessor peak is directly followed by lowest energy of fragment peak	80%
Break up	Highest energy of predecessor peak is smaller than lowest energy of fragment peak, but they are not directly adjacent.	10%
Precedence	Lowest energy of fragment peak is smaller than lowest energy of predecessor peak	10%

Table 4.1: The situations occurring when scoring according to collision energy ranges and their corresponding unlogarithmised scores.

Therefore, in this situation log(0.8) is added to the score. Bear in mind that peaks may appear in a range of spectra. A visualisation of the situations described here can be found in Table 4.1. The values of log(0.1) and log(0.8) are initial guesses, that appear reasonable to score the situation. These values, as well as others in the next sections, should be optimised by a training process as soon as a larger training set and an independent test data are available.

4.4.3 Integer RDBE scoring

In addition to the scoring according to the RDBE value (Section 4.3.3), we can use the value to determine that a molecule is neither a radical nor an ion. For ions and radicals the RDBE value is not integer. Unfortunately, a radical ion has again an integer RDBE. Nevertheless, we can exploit the fact as follows: The neutral loss is uncharged, as the name implies, since the parent ion has passed its charge to the detected fragment. Therefore it needs to have an integer RDBE or be a radical. Radicals are rare, thus we reduce the score of the corresponding edge heuristically by log(4), if the RDBE of the neutral loss is not an integer.

4.4.4 Common neutral losses

Certain neutral losses occur often during fragmentation, especially in biological compounds. Chemists even rely on those to classify analytes. A list of these fragments can be found in Table 4.2. If a neutral loss is among this list, or is a combination of the list entries, its score is increased by log(2). We could reward the combinations a little less than the real entries, but because combinations are heavier than simple entries, the combinations are already lower scored due to the mass of the neutral loss scoring.

Name	Formula
Methyl	CH_3
Methane	CH_4
Oxy	O
Hydroxyl	H_2O
Carbonmonooxide	CO
Nitrogen	N_2
Ammonia	NH_3
Ethyl	C_2H_4
Formaldehyde	CH_2O
Isobutene	C_4H_8
Isopentene	C_5H_8
Formic acid	CH_2O_2
Malonic acid	$C_3H_2O_3$
Xylose	$C_5H_8O_4$
Rhamnose	$C_6H_{10}O_4$
Hexose	$C_6H_{10}O_5$
Glucuronic acid	$C_6H_8O_6$

Table 4.2: The common neutral losses used by the default scoring scheme.

Chapter 5

Algorithms for the Maximum Colourful Tree problem

After constructing and scoring the input graph in the previous chapters, we need calculate the most likely fragmentation process and thus the most likely explanation of the parent mass. As already mentioned in Section 3.4, certain restrictions are assumed for the fragmentation process: It must have a tree-structure and every peak should only have one explanation assigned to it. In Section 5.1, the problem is defined formally on vertex-coloured edge-weighted graphs. Section 5.2 shows that this problem is NP-hard and therefore not efficiently computable.

The following sections then present concepts how to address this problem. Section 5.6 presents the most successful concept with exact results. It is a fixed-parameter algorithm, as introduced in Section 1.2. The algorithms described in Section 5.8 do not solve the problem exactly, but have the advantage to run fast. These heuristics nevertheless produce good identification results.

5.1 Formal problem definition

Based on the definitions given in Section 1.1, we define a colourful tree as follows:

Definition 5 (Colourful tree)**.** A colourful tree $T = (V_T, E_T)$ of a vertex-coloured DAG G is a subtree of G which uses every colour in C at most once:

$$\text{for all } d \in C : |\{v \in V_T | c(v) = d\}| \leq 1$$

This combines both restrictions postulated in the introduction. As we are interested in the most probable fragmentation tree, it is necessary to look for the

colourful tree with maximum weight. Therefore the computational problem can be defined:

Definition 6 (MAXIMUM COLOURFUL TREE)**.** Input: A vertex-coloured edge-weighted directed acyclic graph G
Task: Find the colourful tree of G that has maximal weight.

As we shall see in the next section, this problem is NP-hard. Therefore it is very unlikely that an polynomial-time algorithm exists to solve the problem.

5.2 Proof of NP-hardness

Theorem 1. MAXIMUM COLOURFUL TREE *is NP-hard, even if G is a tree.*

Proof. We prove NP-hardness by reduction from the SAT problem that is known to be NP-complete [GJ79]. An algorithm solves the SAT problem if it can decide whether a given Boolean expression in conjunctive normal form (CNF) is satisfiable. This proof is analogous to the proof that the GRAPH MOTIF problem on vertex coloured graphs is NP-hard [FFHV07].

Given an instance of SAT as a CNF formula $\Phi = c_1 \wedge \cdots \wedge c_m$ over variables $x_1, \ldots, x_n$ one can construct an instance of MAXIMUM COLOURFUL TREE as follows: We shall construct a tree which possesses $m + n + 1$ distinct colours namely $r', x_1, \ldots, x_n, c_1, \ldots, c_m$. We define a root-vertex r of a tree G coloured r' and connect $2n$ children to it. Then we assign the every colour $x_i, 1 \leq i \leq n$, to two of these children. The two vertices in the same colour x_i each represent a different truth assignment for x_i. One vertex in the colour x_i represents $x_i = true$, the other one $x_i = false$. If a truth assignment to x_i satisfies clause c_j we connect a vertex coloured c_j to the vertex coloured x_i, that corresponds to this truth assignment. This assignment is done for all literals in all clauses. To complete the construction we assign a score of 1 to every edge of G. An example for the construction can be found in Figure 5.1.

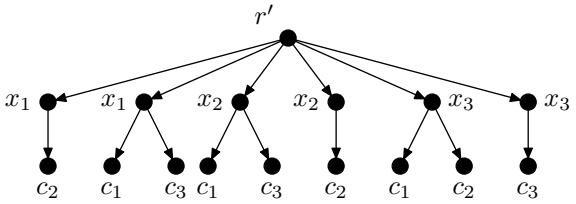


Figure 5.1: Example for the construction of G. The Boolean expression consists of the three clauses: $c_1 = (\overline{x}_1 \vee x_2 \vee x_3), c_2 = (x_1 \vee \overline{x}_2 \vee x_3), c_3 = (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$

As the constructed tree has as many leaves as there are literals in Φ, the construction is polynomial. We now claim that Φ is satisfiable if and only if a maximum colourful tree of G has a score equal to the number of clauses m plus the number of variables n of Φ. To prove the forward direction, assume a truth assignment ϕ that satisfies Φ. Define $A \subset V(G)$ to be the subset of vertices in the colours x_i that correspond to the assignment ϕ. Then, there exists at least one vertex coloured c_j in the neighbourhood of A. Add an arbitrary representative of these vertices coloured c_j to the set $B \subset V(G)$. $A \cup B \cup \{r\}$ form a colourful subtree T of G. It has a score of $m+n$, as for each clause and for each variable there is one edge in T. As there are only $m+n+1$ different colours in G, a colourful tree can contain at most $m+n+1$ vertices, thus consist of $m+n$ edges. As every edge has score 1 the maximal score is $m+n$. Therefore, T is a maximum colourful tree of G.

To prove the backward direction assume there is a maximum colourful tree T of G with score $m+n$. T then contains all colours of G. Let $A \subset V(T)$ denote the set of vertices coloured $x_i, 1 \geq i \geq n$. ϕ is the truth assignment corresponding to A. We constructed G in a way that a vertex coloured c_j is connected to a vertex coloured x_i if and only if assigning the value corresponding to this vertex to x_i satisfies clause c_j. As T contains all colours of G and vertices coloured $c_j, 1 \leq j \leq m$ in T are only connected to vertices in A, every clause of Φ is satisfied by the assignment ϕ. □

Note that it is possible to proof that Maximum Colourful Tree is NP-complete even if G is a binary tree. we can restrict the expression, such that every literal occurs at maximum in two clauses. Instead of linking all x_i coloured vertices to the root directly, we must construct a binary tree with $2n$ leaves. The proof presented here uses arbitrary trees to emphasise the main idea. Also a binary tree is unlikely to occur in any practical instance of the problem.

5.3 Splitting of the Input Graph

To produce a smaller input graph and therefore simplify the analysis, it is possible to split up the graph constructed in Section 3.3. This splitting is done by selecting one explanation of the parent peak and restricting the graph to all vertices reachable from the vertex representing this explanation. This vertex then is the only source vertex of the corresponding DAG. Through this restriction we can split up the graph into one graph per explanation of the parent peak. Therefore we obtain several smaller instances of the Maximum Colourful Tree problem, which can be feasible even if there are not enough resources to process the complete graph.

5.4 Reduction rules

As introduced in Section 1.2.2 reduction rules can be the base of efficient parametrised algorithms. Although these rules are far from reducing the input graph to a problem kernel, they make the input a bit smaller.

5.4.1 Minimal and maximal gain of a vertex

The minimal resp. the maximal gain of a vertex can be defined as the minimal or maximal score a vertex can contribute to the score of the maximum colourful tree. The contribution means the score lost if the vertex is not selected into the maximum colourful tree. It highly depends on the other vertices selected, therefore only the maximum and minimum gain may be determined based on local information. These values are calculated as follows:

$$\begin{aligned} minGain(v) = &\min\{score((u,v))|u \in V, (u,v) \in E\}+ \\ &\sum_{c \in C} \min\{score((v,u))|u \in V, colour(u) = c\} \\ \text{assuming } &score(e) = 0 \text{ if } e \notin E \\ maxGain(v) = &\max\{score((u,v))|u \in V, (u,v) \in E\}+ \\ &\sum_{c \in C} \max\{score((v,u))|u \in V, colour(u) = c\} \end{aligned}$$

Now it is possible to define the domination of vertices: A vertex v dominates vertex u, if u has the same colour as v and $minGain(v) \geq maxGain(u)$. If a vertex is dominated by another vertex, we can remove this vertex, here u, from the graph, as it will never be part of the solution, because a solution containing the dominating vertex v will always yield a better score. Recall that the graph is transitive. Therefore children of u can still be part of the solution as they are also connected to the parents of u. As the gains will change if a vertex is deleted, the rule has to be applied iteratively until no more vertices can be deleted.

5.4.2 The stronger predecessor rule

The following reduction rule can be applied to remove dispensable edges from the graph: The edge (v, w) can be removed if

$$score((v, w)) \leq score((u, w)) \text{ for all } u : (u, v) \in \text{ incoming edges of } v$$

The rule is not applicable to graphs constructed from tandem mass spectra, since these graphs are transitive. Thus, a vertex v has usually many incoming edges, which decreases the likelihood that all these edges have a larger score than (v, w). The situation is further impaired by the scoring according to the mass of the neutral loss as described in Section 4.4.1.

5.5 Branch and bound approach

To solve the Maximum Colourful Tree problem we can use the classical branch and bound approach by testing all combinations of edges that could define a maximum colourful tree. Depth first search is applied to find a good solution early during the search, which can then be used for bounding. To easily find an upper bound for the branches, the following relaxation is applied: The selection of edges is restricted by two constraints: First, two selected edges may not be incident to different vertices of the same colour, since this renders the result not colourful. Second, they may not be incoming edges for the same vertex, as it violates the tree properties. If a pair of edges violates these constraints, these edges *conflict*. As a relaxation the edges may conflict with each other during the search for an upper bound. Note that the edges chosen to determine an upper bound may not conflict with the edges which are part of our current solution at this point of the calculation.

As bounding cannot be applied in the worst case, this leads to a worst case running time of $O(\binom{m}{k}) = O(m^k)$. Recall that n is the number of vertices, m the number of edges and k the number of colours respectively peaks.

This approach is again not applicable to instances obtained from tandem mass spectra. Bounding is not successful if many edges have similar weights, which is the case in these instances. As the graphs obtained from spectra have many edges, the branch and bound approach cannot be applied to these graphs.

5.6 Dynamic programming

Although the branch and bound algorithm is sufficient for some instances of the problem, it scales badly, as expected by its exponential running time. An alternative is presented by Scott et al. [SIKS06]. They propose an algorithm to find colourful trees in so called protein-protein-interaction networks which they coloured randomly. The basis for this parametrised algorithm is dynamic programming over the vertices and all possible subsets of the colour set C. $W(v, S)$, where $S \subseteq C$ denotes the maximal score of a colourful tree with root v and using the colours in S. We derive the following recurrence:

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

with the initial condition $W(v, \{c(v)\}) = 0$. The first line extends a tree by just introducing v as new root, and adding the score of the edge (v, u) to the score of the tree. In the program, this is done by iterating over all outgoing edges of v. The second line merges two trees, which have nothing in common but their root. This is the expensive calculation, although in practice the implementation iterates over the defined values only. Not all entries of W are defined, as there does not necessarily exist a subtree of the input rooted in v using exactly the colours in S. The worst case running time for this algorithm is $O(3^k \cdot k \cdot m)$ and the necessary space is $O(2^k n)$. The factor of 3^k is needed to calculate the second line of the recurrence, where the k colours are divided into three groups: not contained in S, element of S_1 or element of S_2. Then, 3^k is the number of possibilities to perform this division. This yields an fixed-parameter algorithm as the exponential growth is restricted to the number of colours k, representing the number of peaks in the input spectra.

The major disadvantage of this method is its memory consumption. If the user is not interested in the fragmentation tree, it is possible to implement the colour sets S as bit strings, minimising the necessary space. To perform backtracking and thus construct the fragmentation tree it is necessary to store the order the colours were added to the sets. To retain the order of the colours it is necessary to store the colours explicitly. Although this optimisation only decreases memory demands by a constant factor, this often makes the difference between finding a solution and running out of memory. If the user is only interested in the fragmentation trees of the best f decompositions, we can optimise space demands as follows: First the best scoring decompositions of the parent ion are determined using bit strings. Afterwards a

graph is constructed containing only the best f parent mass decompositions and their fragments or, more graph theoretically speaking, their children.

5.7 Brute force

The brute force approach iterates over all combinations of vertices that can form a colourful tree. For every such combination the Maximum Spanning Tree (MST) is calculated using Kruskal's algorithm [Kru56] with some adaptions to ensure that it results in an arborescence. The MST with the highest weight is then returned as result. This leads to a worst case running time of $O((n/k)^k m \log m)$. For most instances this is of course infeasible. But if n/k is smaller than 3 and $\log m \leq k$, this yields a better running time than the dynamic programming algorithm. In practice this difference is relevant, because of the memory consumption and overhead of the dynamic programming. Therefore our implementation uses the brute force algorithm if n/k is small.

5.8 Heuristics

This work focuses on exact algorithms, as they are necessary to asses the accuracy of a heuristic. As a new problem was presented here, good heuristics can only be developed if an exact algorithm is available. These heuristics are merely first guesses. Although they work well as can be seen in Section 7.2, most likely smarter heuristics could be designed. Since the exact fixed-parameter algorithm runs reasonably fast (see Section 7.3 for running times), the need for improved heuristics is limited.

5.8.1 Maximum spanning tree

The simplest concept is to remove the restriction, that the resulting tree has to be colourful. This simplification reduces the problem to finding the Maximum Spanning Tree of the input graph. Kruskal's algorithm is used [Kru56] like by the brute-force approach. Tests (Section 7.2) showed that the results of this heuristic are worthless. They depend only on the fact how many sum formulas with the given peak masses are a subset of the parent mass decomposition in question, because every of these formulas is scored.

5.8.2 Greedy

The greedy algorithm is a simplification of the branch and bound approach. The algorithm sorts the edges according to their weights resp. scores in descending order. It then picks the first in this list. Afterwards the next edge from the list that does not conflict with the previously picked edges is selected. The algorithm continues until $k - 1$ edges have been selected. Recall that 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 colour. Thus we receive the first guess of the branch and bound approach as result here.

5.8.3 Top-down

This is another greedy concept, but this time the algorithm always tries to find paths away from the root. The algorithm starts at the root and follows the best scoring outgoing edge. To follow an edge means to add it to the solution set and continue from the vertex at its end. At the next vertex, it again follows the best scoring outgoing edge that does not conflict with already selected edges. If no such edge exists, the algorithm moves back to the root. It terminates if no edge at the root can be selected. This way, all colours are present in the resulting tree because the input graph is transitive.

Chapter 6

Software

In this chapter we describe the implementation of our algorithms. The details of input and output as well as the options offered are explained. The algorithms developed in this work have been implemented in Java. It is planned to integrate them into a framework for the analysis of metabolite mass spectra currently under development. Although all the concepts in this work have been implemented for testing, only those yielding the best results have been included in the final software package.

No reduction rules are used, because they are not applied often enough to improve the running time. Either the dynamic programming algorithm (Section 5.6) or the splitting (Section 5.3) followed by the brute-force algorithm (Section 5.7) are used. To decide which algorithm is used, a vertices-per-colour ratio (n/k) of four has proven to be a good change-over point to optimise running times. If the ratio is greater than four, dynamic programming is used. In this case the calculation is first done on the complete graph without storing backtracking information and afterwards the graph is reduced to the vertices connected to or being one of the high scoring parent-mass decompositions, and the backtracking information is gathered by analysing this reduced graph.

Concerning the scoring all concepts presented in Chapter 4 are included and can be enabled or disabled via command-line switches.

6.1 Command-line switches

There are three types of switches: Those setting special input files, the ones specifying output options and the majority changes scoring parameters. All switches can be found in Table 6.1.

Switch	Value type	Description	Default
`-et`	File name	Read alphabet masses and valences from specified file.	CHNOPS
`-n`	File name	Sets the file containing the common neutral losses.	Table 4.2
`-e`	Double	Sets the mass deviation in ppm.	20 ppm
`-p`	Double	Determines the precision of the decomposition.	10^{-5}
`-m`	Double	Specify the merging distance	0.1
`-mi`	None	If two peaks are merged, keep the mass of the most intense peak.	Average mass
`-gh`	None	Use the greedy heuristic for calculation.	Disabled
`-raw`	`on` or `off`	Enable or disable calculation of raw intensities as described in Section 4.1.1.	Enabled
`-rel`	`on` or `off`	Enable or disable calculation of smoothed intensities as in Section 4.1.2.	Disabled
`-md`	Double	Sets how many multiples of standard deviation, are regarded to be within the mass deviation error. See Section 4.2.1 for details.	3
`-d`	Two doubles	Enables scoring by DBE distribution, as in Section 4.3.3. Optionally mean and standard deviation of the underlying Gaussian distribution may be given, otherwise $\mu = 8.14$ and $\sigma = 5.53$ are used.	Disabled
`-hc`	Two doubles	Enables scoring by hydrogen carbon ratio. For details see Section 4.3.1. Mean defaults to 1.43, the standard deviation to 0.50	Disabled
`-he`	Two doubles	Enables scoring by hetero atom carbon ratio, described in Section 4.3.2. If not given, the mean is set to 0.59 and the deviation to 0.56.	Disabled
`-nc`	Integer	Combinations of how many likely neutral losses are treated as likely neutral losses, too.	3
`-f`	Integer	Number of fragment trees to be printed in the output file.	10
`-g`	File name	Writes the fragment trees as dot-files.	Disabled
`-o`	File name	Name of the output file.	`<Input file>.out`

Table 6.1: The command-line switches available for the tool developed in this thesis.

Keyword	Value type	Description
`compound`	String	The name of the compound, if known.
`formula`	String	The correct sum formula of the compound, if known. The Program will then compare this formula to its own results.
`charge`	Integer	The charge you expect the parent ion to possess.
`collision`	Integer	Starts a new spectrum. The value gives the collision energy used for that spectrum.
`tic`	Number	This gives the total ion current of the current spectrum. If raw scoring is used (the default), a TIC has to be supplied for every spectrum.

Table 6.2: The keywords that are recognised in the input file, their data types and their effects.

6.2 Input and output

The input file is an ASCII-file containing the peaklists of the spectra to be analysed. Keywords may be used to give further information. Keywords are followed by a value. The allowed keywords can be found in Table 6.2. The keyword `collision` is essential; it begins a new spectrum. All following peaks are added to that spectrum until the next `collision` or the end of the file is reached.

Other optional input files are the element table and the neutral loss list. The element table is given in a text file describing one element per line. Each line contains three values: The letter code of the element, its mass and its typical valence state. The values are separated by a space. The neutral loss list is as simple: It contains just the sum formula of one neutral loss per line. Figure 6.1 shows examples for the input files.

The output file is a simple text file, too. It will contain the best scoring interpretations and the edges of their fragmentation trees, as well as a list of all parent mass decompositions and their scores. Via the option `-f` the user can determine how many trees are calculated and shown. If the option `-g` is specified, `dot`-files are created containing the best scoring fragmentation trees. These files can be converted to images by the graph visualisation tool Graphviz [GN00]. An example output file is shown in Figure 6.2.

compound 4-hexosylferuloyl choline formula C21H32NO9 charge 1 collision CE 15 eV // DP1 50 V tic 105.21 221.085 520.651 442.215 10000 collision CE 25 eV // DP1 50 V tic 142.10 221.082 10000 383.139 281.344 442.213 3928.75 collision CE 40 eV // DP1 50 V tic 109.89 145.029 307.048 177.057 682.162 221.082 10000 collision CE 55 eV // DP1 50 V tic 232.21 145.029 5465.78 177.055 6628.4 206.059 1127.3 221.082 10000	C 12.0 4 H 1.007825 1 N 14.003074 3 O 15.994915 0 P 30.973762 3 S 31.972071 0	CH3 CH4 O H2O CO N2 C2H4 CH2O C4H8 C5H8 CH2O2 C3H2O3 C5H8O4 C6H10O4 C6H10O5 C6H8O6
File containing the spectrum of hexosylferuloyl choline	An example element file	An example neutral loss file

Figure 6.1: Examples for the three types of input files possible. Only the spectrum file is mandatory.

```
Analysed spectra of 4-hexosylferuloyl choline
Parent Peak:  442.214 Da Intensity:  10000.0
Number of parent mass decompositions:  140
Vertices:  300 Colours:  6

Best Decomposition:  C18H35NO9S Score:  100,750
Correct at position:  2 of 140
Correct decomposition:  C21H31NO9 Score:  97,656
Decomposition time:  185.618ms Preprocessing time:  224.195ms
Algorithm time:  347.662ms Backtracking time:  61.27ms

2 best sum formulas with their fragmentation trees:
1) C18H35NO9S 100,750
C18H35NO9S -> C14H7NO 0,016
C18H35NO9S -> C15H26O9S 0,694
C7H12O3S -> C6H8O2S 14,426
C9H16O4S -> C7H12O3S 17,097
C18H35NO9S -> C9H16O4S 14,311

2) C21H31NO9 97,656
C21H31NO9 -> C14H7NO 0,016
C21H31NO9 -> C18H22O9 0,698
C10H8O3 -> C9H4O2 13,866
C12H12O4 -> C10H8O3 16,812
C21H31NO9 -> C12H12O4 13,650

10 best parent mass decompositions with scores:
1) C18H35NO9S 100,750
2) C21H31NO9 97,656
3) C19H31N5O5S 47,135
4) C17H37N3O4P2S 47,120
5) C17H36N3O6PS 47,109
6) C16H35N5O5S2 46,924
7) C19H39NO4S3 46,870
8) C22H36NO4PS 46,638
9) C22H35NO4S2 46,569
10) C17H35N3O8S 46,408
```

Figure 6.2: The output file of hexosylferuloyl choline. The two best scoring fragmentation trees are shown, as well as the ten best scoring decompositions.

Chapter 7

Experimental results

In this chapter we perform some tests on real mass spectrometry data. Section 7.1 introduced the test data. In the Sections 7.2 and 7.3, the results and running times of the proposed software are evaluated and compared to other approaches presented in this work. Section 7.4 describes the results achieved with spectra obtained from an online database and Section 7.5 discusses the calculation of fragmentation trees by the software.

7.1 Test data set

To test the algorithms and scoring functions, the Leibniz Institute of Plant Biochemistry in Halle provided 194 tandem mass spectra of 51 compounds with about four spectra at different collision energies per compound. 45 of these compounds were known beforehand or have been identified manually, so that the correct sum formulas were available for comparison with the results of the program.

The compounds were either reference compounds or extracted from the seed of Arabidopsis thaliana plants. Separation was done using a capillary HPLC system with a GROM-SIL 120 Å ODS-4 HE 3 μm column, which separates the metabolites depending on their hydrophobicity. The mass spectrometry measurement was performed with an API QSTAR Pulsar Hybrid Quadrupole TOF instrument by Applied Biosystems. The preprocessing of the raw data was performed using the AnalystQS software supplied with the instrument.

The compounds with a mass of over 400 Da were most interesting, as they yield more than 100 decompositions of the parent mass, assuming that a spectrum without fragmentation and isotope pattern was measured at 20 ppm. There were six compounds in the test bed fulfilling this criterion: Hexosyloxyphenyl-propanoyl choline, hexosylferuloyl choline, hexosyloxybenzoyl choline, hexosyloxycinnamoyl choline,

hexosylvanilloyl choline and an unknown compound identified as $C_{23}H_{31}NO_8$.

7.2 Results of the analysis

The test bed was analysed with the following options: Masses were decomposed using a relative mass error of 20 ppm, a precision of 10^{-5} and the standard CHNOPS-Alphabet containing the six most abundant elements in living organisms. Raw intensities have been calculated for the peaks, and afterwards peaks closer than 0.1 Da have been merged. Except for the H/C ratio and the DBE distribution scoring (Sections 4.3.1 and 4.3.3), all scoring schemes mentioned in Chapter 4 have been applied with their default values as given in Table 6.1.

The majority of the compounds available were aromatic natural compounds, possessing a lower H/C ratio than the average. The aromatic structure containing many double bonds also increases the DBE value of the compounds. Therefore these two scoring techniques were disabled. We achieved good results without these scoring concepts. Ideally the user restricts the compounds used for calculation of the parameters to the metabolite class he is interested in and passes these parameters to the program.

For nine compounds in the test bed the correct sum formula was not found within a 20 ppm range around the parent peak. Eight of these nine molecules had a parent mass below 200 Da. This inaccuracy arises because mass spectrometers produce a small absolute mass error additional to the relative error, which is not covered by small molecules. All these molecules can be analysed well when using a 50 ppm mass error, but were excluded from the following analysis as inexact data. Four small compounds yielded only one parent mass decomposition. These were excluded, too, as no analysis of the fragmentation spectra is necessary.

7.2.1 Results of the exact algorithm

The identification results of the exact algorithm can be found in Table 7.1. The identification achieves good results. For the majority of the compounds the correct sum formula is ranked first, even for such large compounds as 4-hexosylvanilloyl choline (416 Da). All correct formulas can be found among the first five solutions enabling researchers to restrict further analysis to the top five candidates.

The number of compounds in the data set of Section 7.1 was unfortunately too small to perform a statistical analysis. Zhang et al. present performance values for their Fragment ion Formula Prediction tool (FFP) in [ZGC+05]. Table 7.2 shows that the tool developed in this work performs as well as FFP. FFP uses the

Compound	Mass	# dec.	Rank of the correct formula			
			Exact	Greedy	Top-Down	MST
Acetyl choline	146.12	2	1	1	1	1
Arginine	174.11	2	1	1	1	1
Asparagine	132.05	3	1	1	1	1
Aspartic acid	133.04	3	1	1	1	1
Benzoyl choline	208.13	6	1	1	1	1
Cafeoyl choline	266.14	15	1	1	1	2
Cinnamoyl choline	234.15	7	1	1	1	2
Citrulline	175.10	3	1	1	1	1
Coumaroyl choline	250.14	11	2	2	1	2
Cysteine	121.02	3	1	1	1	1
Cystine	240.02	35	1	1	1	2
Dopamine	153.08	5	1	1	1	1
Feruloyl choline	280.16	18	4	4	3	6
Glutamic acid	147.05	3	1	1	1	1
Glutamine	146.07	2	1	1	1	1
Hexosylferuloyl choline	442.21	140	2	2	1	35
Hexosyloxybenzoyl choline	386.18	87	1	1	1	10
Hexosyloxycinnamoyl choline	412.20	113	1	1	1	28
Hexosyloxyphenylpropan. choline	414.21	100	1	1	1	19
Hexosylvanilloyl choline	416.20	129	1	1	1	29
Hydroxybenzoyl choline	224.13	9	1	1	1	2
Histidine	155.07	4	1	1	1	1
Methionine	149.05	4	1	1	1	1
Phenylalanine	165.08	4	1	1	1	1
Sinapoyl choline	310.17	29	2	2	2	5
Syringoyl choline	284.15	27	3	3	2	5
Threonine	119.06	2	1	1	1	1
Tryptophane	204.09	8	1	1	1	1
Tyramine	137.18	2	1	1	1	1
Tyrosine	181.07	6	1	1	1	1
Vanilloyl choline	254.14	16	1	1	1	2
$C_{23}H_{31}NO_8$	449.20	182	5	5	1	59

Table 7.1: The identification results of the exact algorithm and the three heuristics applied. The third column gives the number of parent mass decompositions.

Mass range	rates using our tool				rates using FFP			
	# ions	Top 1	Top 5	Top 20	# ions	Top 1	Top 5	Top 20
0–300 Da	24	92%	100%	100%	237	83%	97%	99%
300–500 Da	8	50%	100%	100%	135	50%	95%	96%

Table 7.2: The identification rates of our concept and the FFP tool by Zhang et al. [ZGC+05]. Note that different spectra were used and Zhang et al. had more data available.

isotope distribution of fragments, which we do not exploit. Including the analysis of isotope patterns probably further improves our tool, as presented in the outlook (Section 8.2.2).

Although the results here are satisfactory, they are sensitive to the scoring applied. If, for example, the H/C ratio and the DBE scoring are enabled, all correct formulas can only be found among the top twenty, not the top five.

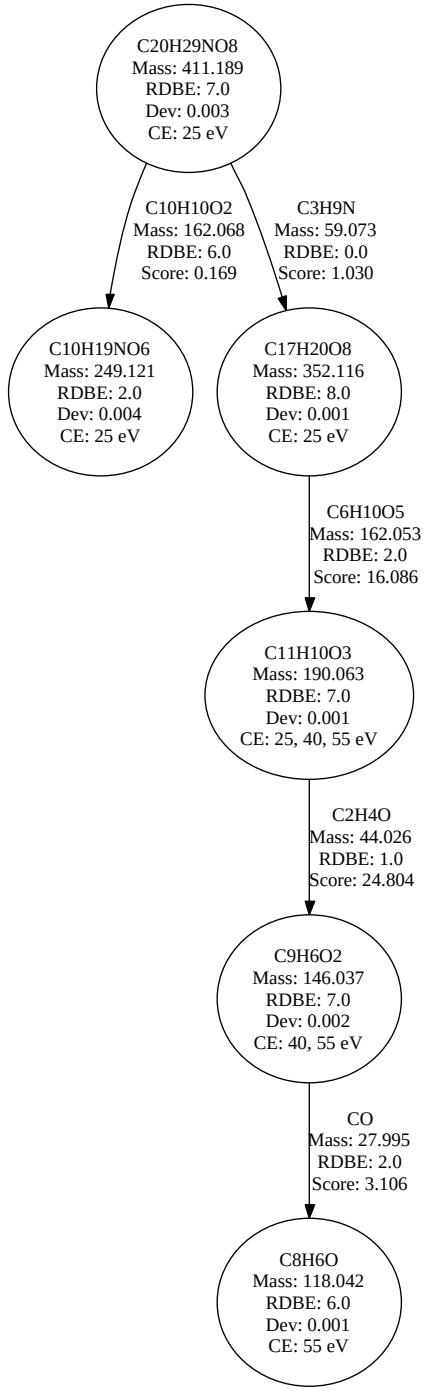
Examples for the calculated fragmentation trees are shown in Figure 7.1. One problem of the technique presented here can be seen when comparing the two trees: the three nodes at the lower right are identical. These remain identical in the graphs of the 26 best scoring decompositions. Of course, parent mass decompositions that do not allow for this interpretation of the three lightest peaks do not receive a high rank. Thus for the remaining 26 decompositions, only three peaks remain to distinguish them. It is even more difficult to distinguish the two examples, as the neutral loss C_3H_9N occurs in both fragmentation trees. The wrong decomposition can only be ruled out here, because hexosyl ($C_6H_{10}O_5$) is separated in a fragmentation step of the correct formula, whereas an unknown loss occurs in the corresponding step of the wrong decomposition. Therefore the scoring using the common neutral losses of Table 4.2 improves the results significantly.

7.2.2 Results of the heuristics

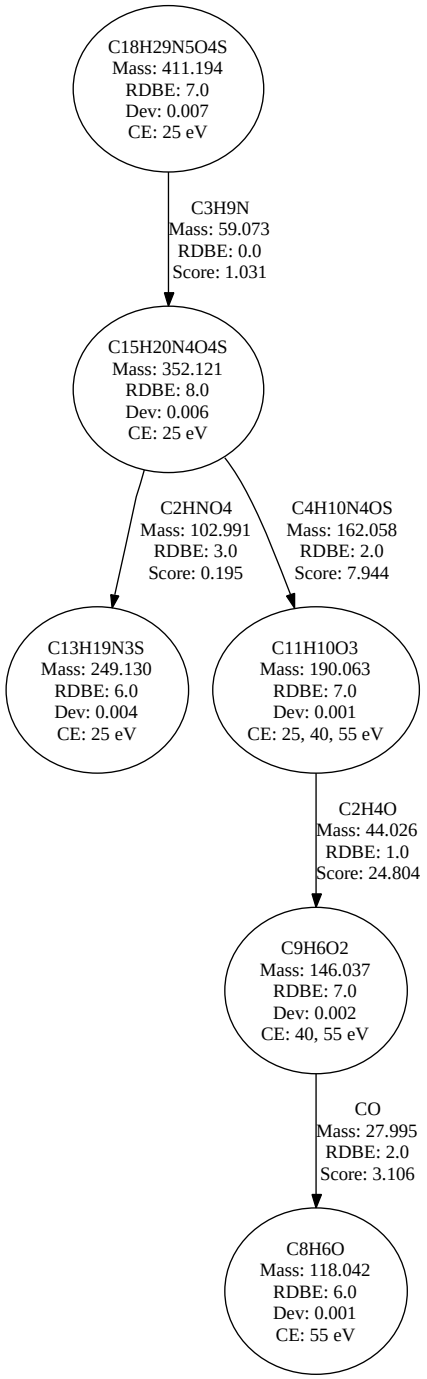
The heuristics as described in Section 5.8 have also been applied to the test data. The results can be found in Table 7.1. The same scoring scheme as with the exact algorithms has been applied.

First, the need for the coloured approach is shown by the results of the MST heuristic. Recall that this heuristic simply ignored the colours in the graph. The accuracy degrades dramatically if the colours are not regarded. Therefore it is necessary to force the analysis algorithm to select only one explanation per peak, as it is done with the colours in this work.

The results of the two other heuristics are excellent. They do not determine the optimal scores for the parent mass decompositions, but the ranks remain identical.



(a) Fragmentation tree of the correct sum formula ranked at first position.



(b) Fragmentation tree of an incorrect sum formula ranked at seventh position.

Figure 7.1: Two fragmentation trees calculated from the spectra of hexosyloxycinnamoyl choline. Dev: Mass deviation between the sum formula and the peak it explains. CE: Collision Energies during which the peak occurred.

The difference in the score is systematic. The results of hexosylferuloyl choline illustrate this fact: The best decomposition receives a score of 100.75 and the second best and correct one receives a score of 97.66 with the exact algorithm. With the greedy heuristic (Section 5.8.2), the same decompositions get a score of 54.21 and 52.62 respectively, but are still ranked at the same positions.

The greedy heuristic achieves the same results as the exact algorithm and the top-down-heuristic even improves the results, although we consider this an effect intrinsic to the test data for the following reason: The fewer vertices of the same colour exist in the input graph the better the heuristic performs because there are fewer possibilities to select from. For some reason with this test data the input graphs containing the correct decompositions contain fewer vertices with the same colour than the graphs of the wrong decompositions.

Therefore the heuristics must be more thoroughly tested if more data becomes available. If a random effect can be ruled out, the heuristics should be used for analysis in future, as they save time and memory. To perform these tests it is nevertheless necessary to have a suitably fast exact algorithm available. In the next section we compare the running times of the algorithms to identify which algorithms are fast in practice.

7.3 Running time comparisons

The algorithms were implemented in Java and compiled with the Sun Java Standard Edition compiler version 1.5. They were run on an Intel Pentium 4 running at 1.80 GHz with 512 KB cache and 512 MB main memory. The virtual machine corresponding to the compiler was used. No special options were passed to the virtual machine.

The task was to analyse the spectra of all 51 compounds described in Section 7.1. The resulting running times can be found in Table 7.3. The times without overhead are the core running times. Input and output, mass decomposition, graph construction and scoring are not included in these times.

As can be seen, the branch and bound approach is magnitudes slower than the other algorithms and thus of no practical use. Interestingly, the brute force approach is significantly faster than the dynamic programming concept in the current testbed. The reason for this is the large number of small compounds with a small vertices-per-colour ratio in the data set. Thus the brute force algorithm possesses a better worst-case running time and even saves lot of additional work, e.g., by allocating less memory. The combination of both strategies achieves the fastest running time as it

Algorithm	Total running time	Running time w/o overhead
Branch and bound	26 h	26 h
Brute force	312 s	226 s
Dynamic programming	4358 s	4066 s
Combination of DP and BF	88 s	19 s
Reduction rules	94 s	24 s
Greedy heuristic	90 s	1 s

Table 7.3: The running times of the algorithms with and without overhead. Mass decomposition, graph construction and scoring as well as I/O operation were considered as overhead.

selects the presumably faster algorithm based on graph size and number of colours. If the testbed contained larger molecules such as the one described in Section 7.4, the brute force algorithm would take a lot longer. The combination runs fast in all cases and therefore is the best choice if the user is interested in exact results. As expected, the heuristic provides results even faster, but has the disadvantage of being inexact. If we applied the reduction rules presented in Section 5.4 to the input graph, before executing the combination of dynamic programming and brute force. This does not speed up running times. On the contrary, the application of the rules takes more time than the main algorithm saves by only processing the reduced graphs.

7.4 Tests with other spectra

To test the tool also with spectra obtained in another laboratory, as well as with larger metabolites, the Q-TOF tandem mass spectra of Glycyrrhizate were downloaded from the mass spectrometry database MassBank [Mas06]. Glycyrrhizate was chosen as it has the highest molecular weight, namely 822 Dalton. The spectra contained much less intense peaks than the previously analysed spectra. As it was not possible to find a proper explanation for these weak peaks, any peak which had a relative intensity below 100 was removed from the spectrum. The relative intensities in MassBank are determined by normalising the most intense peak to a value of 999. As no total ion counts were available for these spectra, no raw intensities were calculated.

There were 2277 parent mass decompositions for the molecule, the complete graph contained appr. 4000 vertices, but had only four colours. Due to the small number of colours the dynamic programming algorithm ran only four seconds. The time consuming steps here were the decomposition, taking eight seconds and the construction of the graph, which took 21 seconds, yielding a total time of about

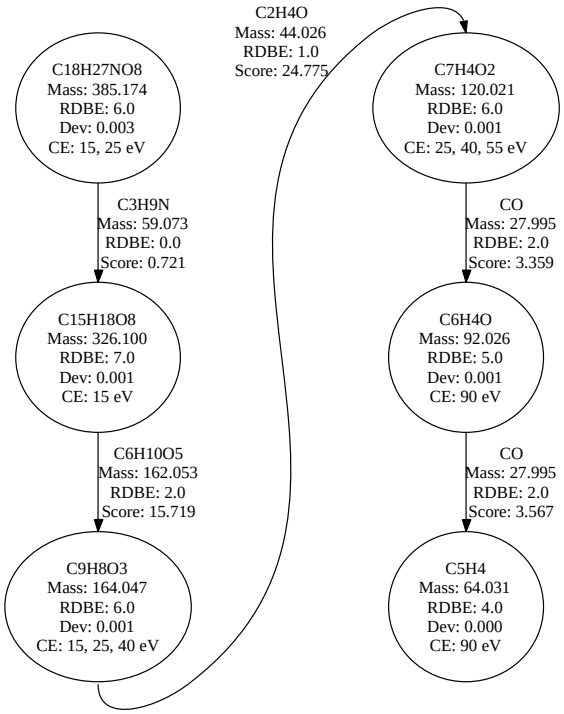


Figure 7.2: The calculated and manually confirmed fragmentation tree of hexosyloxybenzoyl choline.

one minute. The correct formula was ranked at position 18. Considering the poor quality of the spectra, this result is acceptable. It shows that the analysis concept is not restricted to data of a single lab, although a certain quality standard concerning the mass error and the noise reduction has to be met. It also shows that the dynamic programming algorithm is suitably fast even for large metabolites, thanks to fixed-parameter tractability.

7.5 Prediction of fragmentation

The scoring and the algorithms were designed to identify metabolites. The maximum colourful tree of the correct parent mass decomposition is also a prediction of the fragmentation process. However, relying on these predictions is not suggested. The fragmentation tree of one example, hexosyloxybenzoyl choline, was manually constructed. In this case, the calculated fragmentation tree shown in Figure 7.2 exactly matched the manually constructed one. Unfortunately no more manually constructed fragmentation trees were available for comparison.

The fragmentation tree is linear in case of the hexosyloxybenzoyl choline spectra. First, a part of the choline separates, then hexose is lost. The compound then further

dissolves, emitting an ethanol-like structure.

The approach to calculate fragmentation trees from the decompositions of the fragment peak has the advantage that no prior knowledge about fragmentation mechanisms is necessary. Common commercial tools such as Mass Frontier and the ACD Fragmenter make use of this knowledge [Wil02], although the mechanism is only little understood.

Heinonen et al. [HRM+06] also present an approach to determine the fragmentation tree without prior knowledge. As they focus on finding the fragmentation tree, their concept and scoring is probably better suited for the task, especially because they make use of the compound's structure. Interestingly, their formalism can be easily transformed into a weighted coloured DAG and their calculation is equivalent to finding the maximum colourful tree of this graph. They solve the problem by applying integer linear programming (ILP). Since ILP is a time-consuming process, their program needs about twelve hours to solve the problem for the graph of glycyrrhizate mentioned in Section 7.4. The algorithms presented here can probably solve the problem more efficiently. Thus a combination of both techniques appears to be promising.

Chapter 8

Conclusion

This chapter subsumes the main ideas and results presented in this work. Many possibilities to improve and extend the concept are given in Section 8.2. Finally, Section 8.3 presents other possible applications.

8.1 Summary

We have developed a concept for the analysis of metabolite tandem mass spectra. It is based on how the spectra are generated by collision induced dissociation. We do not restrict possible fragmentation steps by prior knowledge, but assign scores that correspond to the likelihood that this fragmentation occurs. To generate the scores, we use a small amount of prior knowledge. We apply statistics on metabolite databases to derive scores from key properties of the fragment sum formulas.

To make the spectrum and its interpretations accessible to computer science techniques, we transform them into a graph. We then define a formal problem that includes all the restrictions necessary for our concept. This problem is NP-hard, but different algorithms allow to compute a solution after an acceptable running time, among them a fixed-parameter tractable algorithm and different heuristics.

We tested these algorithms using real-world data and successfully identified many metabolites: For all 45 compounds, six of them with a mass over 400 Da, the correct sum formula was among the first five of the suggested list. The quality of results depends on the scoring scheme chosen. Improvements in scoring candidate will be required to increase the robustness of results. The exact fixed-parameter algorithm as well as the heuristics proved to be fast, they analysed all compounds in about 1.5 minutes total running time. The fragmentation trees calculated by the concept are also acceptable first guesses. As identifying the correct fragmentation tree was not the focus of this work, adapting our concept could to this task improve the

fragmentation prediction.

8.2 Future Work

As the preceding paragraph indicates, there is plenty of room for improvements and modifications in many areas of our approach. We briefly mentioned several ideas in the previous chapters. All these open tasks are gathered in this section to provide an overview over the potential of the technique.

8.2.1 More and improved training data

To derive statistics and perform a well-founded training and test of the identification program, large amounts of data are needed. As performing mass spectral measurements is still a tedious and resource consuming task, the required data can not easily be obtained. Perhaps in future, the following requirements for more and even annotated data can be met.

More training data is needed to optimise the scoring parameters and thus, to improve the robustness of identification results. Unfortunately, the only publicly available metabolite mass spectra, which can be obtained from the database MassBank [Mas06], are not of the required quality. Parameter estimations that would benefit from more training data are: The standard deviation of mass deviation scoring, the parameters of collision energy scoring, and the reward for known neutral losses.

Of course, training data with annotations whether a peak is noise or not, would further improve our technique, as annotated data allows us to take full advantage of the concept developed by Wan et al. [WYC06], as sketched in Section 4.1.2. This offers the advantage of using real probabilities as scores, not only some values expected to be proportional to a probability.

More specific metabolite data sets can also help to tune parameters of decomposition properties more exactly for the type of metabolites the user is interested in. In particular, the ability to restrict data sets to secondary metabolites of a certain group of organisms would help, as there is no need for de-novo identification of primary metabolites, and the organism that metabolites were extracted from is usually known.

The list of common neutral losses should also be extended and improved. For example, one can classify these neutral losses into three groups according to the frequency they occur. Then, a more frequent neutral loss could receive a higher

score than one which is more rare. Of course, all entries of this list will still receive a better score than a neutral loss not present at all.

8.2.2 Change of experimental parameters

To improve results one can also change the experimental procedures. This might lead to data that is difficult to interpret manually, but a computer might be able to analyse it. To be more precise, the following ideas might provide such data: In the mass spectrometry procedure used here, a quadrupole mass analyser is used as a filter, ensuring that only ions of one specific parent mass reach the collision cell. The properties of this filtering quadrupole might now be changed in two ways.

One is to enlarge the filtering range. This will allow not only the parent ion but also the first isotope species to pass. This will create small isotope patterns following fragment peaks. These patterns can then be analysed as described by Zhang et al. [ZGC+05]. The combination of Zhang's technique with the one presented in this thesis seems promising. The only obvious disadvantage is that the isotope peaks might overlap with other fragment peaks. This would lead to a wrong assessment of the isotope pattern and the fragment peak would be lost, as all the isotope peaks have to be removed before the main analysis. In reality, this situation will rarely occur and, if so, can be detected easily by the program.

The second possibility to modify the filtering quadrupole is to select a different ion. Typically, hydrogen adducts are measured, that is, the neutral metabolite molecule has absorbed a proton to become charged. Additionally, sodium adducts occur during ionisation. The spectra from sodium adducts are difficult to interpret by manual inspection but might provide another source of information to the identification program.

8.2.3 Further ideas

Another information available from the experiment is the retention time of the molecule. The retention time is the time the molecule stayed on the column during chromatography before measuring the mass spectra. Retention times can be predicted for oligonucleotides [SQHK07]. Although the retention times highly depend on molecular structure, it might be possible to roughly predict them from sum formulas, too.

It might even be possible to elucidate the structure of a de-novo identified molecule using molecular structure generators. Structure generators produce many suggestions for a single sum formula. Because we can additionally provide the sum

formulas of fragments, the number of suggested structures will be reduced. In certain cases only a small number of suggestions remain that can be presented to the user.

Whereas all the aforementioned ideas focus on biological aspects, the following is a purely theoretical idea: Björklund et al. [BHKK07] present a technique called "Fast Subset Convolution". This would reduce the running time of the dynamic programming algorithm from $O(3^k km)$ to $O(2^k km)$. It remains to be shown whether its complex calculations improve running times in practice.

8.3 Other fields of application

In addition to to metabolite analysis, algorithms of this work could be applied to tandem mass spectra of other molecule types. This of course will require completely new scoring concepts and probably also some changes in the algorithms.

One could apply the technique to peptide mass spectra. These spectra would need the same accuracy as the metabolite spectra discussed in this work. The alphabet would then consist of the amino acids and the construction of the input graph would need some modifications, too.

Another application would be to determine glycan structures from tandem mass spectra. Glycans are trees of covalently bonded sugar molecules, which are often attached to proteins. Here, it might be necessary to make major changes to the concept. The maximum colourful tree could resemble the tree structure of the glycan analysed instead of modelling the fragmentation process.

Thus the algorithms and basic ideas presented could be of use for a broader range of tandem mass spectra based on molecule fragmentation.

Bibliography

[BE03] V. Bafna and N. Edwards. On de novo interpretation of tandem mass spectra for peptide identification. *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 9–18, 2003.

[BHKK07] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets möbius: fast subset convolution. *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 67–74, 2007.

[BK07] S. Böcker and H.M. Kaltenbach. Mass spectra alignments and their significance. *Journal of Discrete Algorithms*, 5(4):714–728, 2007.

[BL07] S. Böcker and Z. Lipták. A Fast and Simple Algorithm for the Money Changing Problem. *Algorithmica*, 48(4):413–432, 2007.

[BLLP06] S. Böcker, M.C. Letzel, Z. Lipták, and A. Pervukhin. Decomposing metabolomic isotope patterns. *Proc. of the 6th Workshop on Algorithms in Bioinformatics (WABI)*, 3:12–23, 2006.

[Bor26] O. Boruvka. O jistém problému minimálním (On a certain minimal problem). *Práce Moravské Prírodovedecké Spolecnosti*, 3:37–58, 1926.

[CKT+01] T. Chen, M.Y. Kao, M. Tepel, J. Rush, and G.M. Church. A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, 8(3):325–337, 2001.

[CLT01] I.V. Chernushevich, A.V. Loboda, and B.A. Thomson. An introduction to quadrupole-time-of-flight mass spectrometry. *Journal of Mass Spectrometry*, 36(8):849–865, 2001.

[Das01] Chhabil Dass. *Principles and practice of biological mass spectrometry.* John Wiley and Sons, 2001.

[DF99] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer New York, 1999.

[DG05] J.C. D'Auria and J. Gershenzon. The secondary metabolism of Arabidopsis thaliana: growing like a weed. *Current Opinion in Plant Biology*, 8(3):308–316, 2005.

[EMY94] J.K. Eng, A.L. McCormack, and J.R. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrometry*, 5(11):976–989, 1994.

[FFHV07] M.R. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp Tractability Borderlines for Finding Connected Motifs in Vertex-Colored Graphs. *Lecture Notes In Computer Science*, 4596:340, 2007.

[Fie02] O. Fiehn. Metabolomics-the link between genotypes and phenotypes. *Plant Molecular Biology*, 48(1):155–171, 2002.

[FRR+05] B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J.M. Buhmann. NovoHMM: A Hidden Markov Model for de Novo Peptide Sequencing. *Anal. Chem*, 77(22):7265–7273, 2005.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. WH Freeman & Co. New York, NY, USA, 1979.

[GN00] E.R. Gansner and S.C. North. An open graph visualization system and its applications to software engineering. *Software Practice and Experience*, 30(11):1203–1233, 2000.

[HRM+06] M. Heinonen, A. Rantanen, T. Mielikainen, E. Pitkanen, J. Kokkonen, and J. Rousu. Ab Initio prediction of molecular fragments from tandem mass spectrometry data. *German Conference on Bioinformatics*, pages 40–53, 2006.

[HS01] Edmond de Hoffman and Vincent Stroobant. *Mass Spectrometry: Principles and Applications*. John Wiley and Sons, 2nd edition, 2001.

[Ini00] The Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature*, 408(6814):796–815, 2000.

[JCB+00] H. Jin, E. Cominelli, P. Bailey, A. Parr, F. Mehrtens, J. Jones, C. Tonelli, B. Weisshaar, and C. Martin. Transcriptional repression by AtMYB4 controls production of UV-protecting sunscreens in Arabidopsis. *The EMBO Journal*, 19:6150–6161, 2000.

[Joh06] John Wiley and Sons Inc. *Wiley Registry/NIST 2005 Mass Spectral Library*. John Wiley and Sons, 8th edition, 2006.

[KF07] T. Kind and O. Fiehn. Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*, 8(1):105, 2007.

[KGH+06] 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 Research*, 34:D354–357, 2006.

[KNKA02] 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.

[Kru56] J.B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[LGR+06] Eva Lange, Clemens Gröpl, Knut Reinert, Oliver Kohlbacher, and Andreas Hildebrandt. High Accuracy Peak-Picking of Proteomics Data using Wavelet Techniques. In *Proceedings of the 11th Pacific Smyposium on Biocomputing (PSB-06)*, pages 243–254, 2006.

[Mas06] MassBank. `www.massbank.jp`, 2006.

[MBS+04] R. Matthiesen, J. Bunkenborg, A. Stensballe, O.N. Jensen, K.G. Welinder, and G. Bauw. Database-independent, database-dependent, and extended interpretation of peptide mass spectra in VEMS V2. 0. *Proteomics*, 4(9):2583–2593, 2004.

[MZR03] L.A. Mueller, P. Zhang, and S.Y. Rhee. AraCyc: A Biochemical Pathway Database for Arabidopsis. *Plant Physiology*, 132(2):453–460, 2003.

[Nie04] R. Niedermeier. Ubiquitous parameterization — invitation to fixed-parameter algorithms. In *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS 2004)*, number 3153 in Lecture Notes in Computer Science, pages 84–103. Springer, August 2004.

[Nie06] R. Niedermeier. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, 2006.

[PG00] E. Pichersky and D.R. Gang. Genetics and biochemistry of secondary metabolites in plants: an evolutionary perspective. *Trends in Plant Science*, 5(10):439–445, 2000.

[PPCC99] D.N. Perkins, D.J. Pappin, D.M. Creasy, and J.S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[Pri57] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401, 1957.

[Sen51] J.K. Senior. Partitions and Their Representative Graphs. *American Journal of Mathematics*, 73(3):663–689, 1951.

[SIKS06] J. Scott, T. Ideker, R.M. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J Comput Biol*, 13:133–144, 2006.

[SNAUA+06] 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. *Biotechnology in Agriculture and Forestry*, 57:165, 2006.

[SQHK07] M. Sturm, S. Quinten, C.G. Huber, and O. Kohlbacher. A statistical learning approach to the modeling of chromatographic retention of oligonucleotides incorporating sequence and secondary structure data. *Nucleic Acids Res*, 35(12):4195–4202, 2007.

[SWO+06] C.A. Smith, E.J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.*, 78(3):779–787, 2006.

[Wil78] H.S. Wilf. A Circle-Of-Lights Algorithm for the "Money-Changing Problem". *The American Mathematical Monthly*, 85(7):562–565, 1978.

[Wil02] A. Williams. Applications of computer software for the interpretation and management of mass spectrometry data in pharmaceutical science. *Curr Top Med Chem*, 2(1):99–107, 2002.

[WM05] J.M. Wells and S.A. McLuckey. Collision-induced dissociation (CID) of peptides and proteins. *Methods Enzymol*, 402:148–85, 2005.

[WYC06] Y. Wan, A. Yang, and T. Chen. PepHMM: A Hidden Markov Model Based Scoring Function for Mass Spectrometry Database Search. *Anal Chem*, 78(2):432–437, 2006.

[ZGC+05] J. Zhang, W. Gao, J. Cai, S. He, R. Zeng, and R. Chen. Predicting molecular formulas of fragment ions with isotope patterns in tandem mass spectra. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):217–230, 2005.

List of Tables

List of Figures

Selbstständigkeitserklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, den 10. Januar 2008

Florian Rasche