Entwicklung des Annotationsprogramms ChemFrag zur Aufklärung von Fragment-Ionen in Massenspektren und Fragmentierungswegen sowie eines Verfahrens zum schnellen Testen von Molekül-Äquivalenzen (MET)

Dissertation

zur Erlangung des
Doktorgrades der Naturwissenschaften (Dr. rer. nat.)

der

Naturwissenschaftlichen Fakultät III
Agrar- und Ernahrungswissenschaften,
Geowissenschaften und Informatik
der Martin-Luther-Universität Halle-Wittenberg,

vorgelegt

von Frau Jördis-Ann Schüler
geb. am 04.02.1991 in Dessau

Gutachter:
Prof. Dr. Matthias Müller-Hannemann
Prof. Dr. Sebastian Böcker

Tag der Verteidigung: 15.12.2022

Abstract

The identification and structure determination of small molecules by mass spectrometry is an important step in chemistry, biochemistry or doping analysis. However, the chemically correct annotation of a fragment ion spectrum can be a difficult challenge. Therefore, the development of the programme `ChemFrag` for the detection of fragmentation pathways and the annotation of fragment ions with chemically plausible structures is carried out in this thesis. `ChemFrag` combines a quantum chemical approach with a rule-based approach. From the quantum chemical outputs, the head of formation of the fragment ions and the strength of the bonds can be derived. Weak bonds can then be homolytically and heterolytically cleaved by implemented rules. For more complex cleavages and rearrangements, there are a further 51 rules in `ChemFrag`. Experiments on various doping substances show that `ChemFrag` can annotate the fragment ions in a chemically meaningful way. In most cases, the predicted fragment ions are chemically more realistic than those of purely combinatorial approaches or approaches based on machine learning. The annotations generated by `ChemFrag` often coincide with spectra manually annotated by experts. Furthermore, `ChemFrag` can be used to annotate mass spectra of natural substances in a chemically meaningful way. Among other things, it was possible to predict fragmentation pathways for several substances for which none were previously known. Thus, `ChemFrag` offers a major advance in fragment ion annotation and enables more precise automatic annotation of mass spectra.

Another important question in chemoinformatics is the equivalence recognition of molecules, which must also be solved in `ChemFrag`. Mathematically, the equivalence recognition of molecules corresponds to the isomorphism problem of labelled graphs. The second part of the thesis presents the `MET` (`Molecule Equivalence Tester`) programme for solving this problem. `MET` converts the molecules into labelled graphs and exploits chemical properties and the local neighbourhood of atoms to obtain node labels that are as unique as possible. These characteristic labels are the key to clever partitioning of molecules into molecular equivalence classes and effective equivalence testing. Based on extensive computational experiments, we show that the algorithm is significantly faster than existing implementations in `SMSD`, `CDK` and `RDKit`. In addition, `MET` can take into account more chemical properties than its competitors and can thus distinguish molecules that are considered equivalent by the other programmes. Furthermore, the experiments show that `MET` is slightly more efficient, for molecules up to 400 atoms than the algorithm of `VF2++`. `MET` thus offers a useful enhancement for the correct determination of the molecule equivalence.

Zusammenfassung

Die Identifizierung und Strukturbestimmung kleiner Moleküle durch die Massenspektrometrie ist ein wichtiger Schritt in der Chemie, Biochemie oder Doping-Analyse. Allerdings kann die chemisch korrekte Annotation eines Fragment-Ionen-Spektrums eine enorme Herausforderung darstellen. Daher erfolgt im Rahmen dieser Arbeit die Entwicklung des Programms `ChemFrag` für die Detektion von Fragmentierungswegen und der Annotation von Fragment-Ionen mit chemisch plausiblen Strukturen. `ChemFrag` kombiniert dafür einen quantenchemischen mit einem regelbasierten Ansatz. Aus den quantenchemischen Ausgaben lassen sich die Bildungsenthalpien der Fragment-Ionen und die Stärke der Bindungen ableiten. Schwache Bindungen können anschließend durch implementierte Regeln homolytisch und heterolytisch gespalten werden. Für komplexere Spaltungen und Umlagerungen existieren in `ChemFrag` weitere 51 Regeln. In Experimenten an verschiedenen Dopingsubstanzen zeigt sich, dass `ChemFrag` die Fragment-Ionen chemisch sinnvoll annotieren kann. In den meisten Fällen sind die vorhergesagten Fragment-Ionen chemisch realistischer als die von rein kombinatorischen Ansätzen oder Ansätzen, die auf maschinellem Lernen basieren. Die von `ChemFrag` erzeugten Annotationen decken sich oft mit Spektren, die von Experten manuell annotiert wurden. Weiterhin können mit `ChemFrag` Massenspektren von Naturstoffen chemisch sinnvoll annotiert werden. Unter anderem gelang es für mehrere Substanzen Fragmentierungswege vorherzusagen, für die bisher noch keine bekannt waren. Damit bietet `ChemFrag` einen großen Fortschritt bei der Fragment-Ionen-Annotation und ermöglicht eine präzisere automatische Annotation von Massenspektren.

Eine weitere wichtige Fragestellung in der Chemoinformatik ist die Äquivalenzerkennung von Molekülstrukturen, die auch in `ChemFrag` gelöst werden muss. Mathematisch entspricht die Äquivalenzerkennung von Molekülstrukturen dem Isomorphieproblem gelabelter Graphen. Der zweite Teil der Arbeit stellt das Programm `MET` (`Molecule Equivalence Tester`) zur Lösung diese Problems vor. `MET` wandelt die Molekülstrukturen in gelabelte Graphen um und nutzt chemische Eigenschaften und die lokale Nachbarschaft von Atomen aus, um möglichst eindeutige Knotenbeschriftungen zu erlangen. Diese charakteristischen Label sind der Schlüssel für eine geschickte Partitionierung von Molekülstrukturen in Molekül-Äquivalenzklassen und einem effektiven Äquivalenztest. Auf der Grundlage umfangreicher Berechnungsexperimente zeigen wir, dass der Algorithmus deutlich schneller ist als bestehende Implementierungen in `SMSD`, `CDK` und `RDKit`. Zusätzlich kann `MET` mehr chemische Eigenschaften als die Konkurrenz berücksichtigen und kann damit Molekülstrukturen unterscheiden, die von den anderen Programmen als äquivalent erachtet werden. Weiterhin zeigen die Experimente, dass `MET` für Molekülstrukturen bis 400 Atomen etwas effizienter ist als der Algorithmus von `VF2++`. `MET` bietet damit eine nützliche Weiterentwicklung für die korrekte Bestimmung der Molekül-Äquivalenz.

Inhaltsverzeichnis

Abkürzungsverzeichnis

CASMI	Critical Assessment of Small Molecule Identification
CDK	Chemistry Development Kit
CML	Chemical Markup Language
CSV	Comma-separated values
DFT	Dichtefunktionaltheorie
EI	Elektronenstoßionisation
ESI	Elektrosprayionisation
InChI	International Chemical Identifier
IUPAC	International Union of Pure and Applied Chemistry
MOPAC	Molecular Orbital PACkage
m/z	Masse-zu-Ladungs-Verhältnis
NMR	Nuclear Magnetic Resonance
PDB	Protein Data Bank
SARMs	Selective Androgen Receptor Modulators
SMARTS	SMILES arbitrary target specification
SMILES	Simplified Molecular Input Line Entry Specification

1. Einleitung

Die Analytik kleiner Moleküle ist in den vergangen Jahrzehnten ein wichtiger Bestandteil in den Bereichen Biologie, Biochemie, Chemie oder auch Pharmazie geworden [1, 2]. Aber nicht nur in diesen Wissenschaftsbereichen möchte man die Bestandteile einer Probe analysieren und die Struktur der enthaltenen Stoffe aufklären und charakterisieren, sondern auch in der Archäologie oder in der Doping-Analyse ist dies von hoher Wichtigkeit. Eine Methode der Analytik ist die Massenspektrometrie. Der Fokus dieser Arbeit liegt deshalb auf der Annotation von Massenspektren. Speziell werden wir uns mit MS/MS-Spektren beschäftigen und deren Fragment-Ionen annotieren. Mit Hilfe der Massenspektrometrie können die molekularen Massen kleiner Moleküle bestimmt werden und unter bestimmten Voraussetzungen, wie die Nutzung der NMR-spektroskopischen Methoden, strukturell charakterisiert werden. Als eine große Herausforderung in diesem Gebiet gilt nicht nur die Aufklärung und somit Annotation der Fragment-Ionen, sondern auch deren chemische Plausibilität sowie die Herleitung von Fragmentierungswegen. An dieser Stelle existiert ein großer Unterschied zwischen den existierenden *in-silico*-Methoden [3] und Methoden auf Basis quantenchemischer Berechnungen [4], die wir in einem späteren Teil der Arbeit im Detail vorstellen. Mit Hilfe der *in-silico*-Methoden lassen sich mit kurzer Rechenzeit Strukturen für die Fragment-Ionen im Massenspektrum bestimmen. Nachteilig ist hier, dass diese in vielen Fälle chemisch nicht sinnvolle Vorhersagen treffen oder nur Summenformeln vorhersagen. Im Vergleich dazu benötigen die quantenchemischen Ansätze intensive Rechenleistung mit hoher Laufzeit. Jedoch ermöglichen sie damit die Vorhersage chemisch sinnvoller Annotationen und Fragmentierungswege. Ziel dieser Arbeit ist es, die Lücke zwischen diesen beiden Ansätzen zu schließen. Dafür beinhaltet diese Arbeit die Präsentation einer Methode, die einen regelbasierten und einen quantenchemischen Ansatz kombiniert. Damit soll die chemisch sinnvolle Annotation der Fragment-Ionen und die Bestimmung der Fragmentierungswege mit einer kurzen Rechenzeit ermöglicht werden. Das dafür entwickelte Programm trägt den Namen `ChemFrag`. In der Implementation von `ChemFrag` ergab sich das Problem der Moleküläquivalenzbestimmung. Diese wurde im Rahmen dieser Arbeit durch die Entwicklung des Algorithmus `MET` gelöst, der in seiner Schnelligkeit und korrekten Zuordnung äquivalenter Moleküle und deren Atome vorhandene chemoinformatische Methoden verbessert.

Um die Motivation und die Grundlagen von `ChemFrag` und `MET` zu verstehen, werden wir dazu anfangs notwendige Grundlagen der Chemoinformatik und der Massenspektrometrie einführen. Die vorliegende Arbeit untergliedert sich daher wie folgt:

- Das **Kapitel 2** erläutert die Grundlagen der Chemoinformatik. Dabei geht es

auf die verschiedenen Repräsentationsarten von Molekülstrukturen, wie beispielsweise SMILES, InChI oder graphbasierte Darstellung, ein. Es stellt zunächst den Aufbau der String-basierten Moleküldarstellungen durch SMILES und InChI an Beispielen vor und diskutiert deren Vor- und Nachteile sowie die Anwendungsmöglichkeiten. Anschließend erläutert es die Möglichkeit der Graphrepräsentation von Molekülstrukturen. Dazu zählen zum einen der Molekülgraph, der bereits im Jahr 1874 von Cayley [5] erstmals Erwähnung fand, und die Darstellung als einfacher Graph. Diese beiden Repräsentationsarten unterscheiden sich in der Verwendung des Labelings von Knoten und Kanten. Während ein einfacher Graph die Atom- und Bindungseigenschaften im Molekül durch zusätzliche Dummy-Atome und Kanten zwischen Dummy-Atomen abbildet, verwendet ein Molekülgraph Knotenlabel, um die Atomeigenschaften zu symbolisieren und Kantenlabel für die Bindungstypen. Diese drei Repräsentationsarten spielen beim Einlesen von Molekülstrukturen sowie im Bereich des Äquivalenzvergleichs von Molekülen in dieser Arbeit eine große Rolle.

Im zweiten Abschnitt führt das **Kapitel 2** die in dieser Arbeit verwendeten Chemoinformatik-Bibliotheken ein. Die Grundlage der Implementierungen dieser Arbeit bildet die Java-basierte Bibliothek `Chemical Development Kit` (`CDK`), die es unter anderem ermöglicht Molekülstrukturen ein- und auszulesen, zu bearbeiten und algorithmische Fragestellungen, wie die Substruktursuche, darauf zu untersuchen. Einen weiteren Bestandteil bildet das Python-basierte Programm `RDKit`, welches ebenfalls Molekülstrukturen ein- und auslesen sowie bearbeiten kann und in dieser Arbeit die Funktion der Generierung der 3D-Koordinaten übernimmt. Abschließend beschäftigt sich das Kapitel mit quantenchemischen Methoden, die für den quantenchemischen Ansatz von `ChemFrag` verwendet werden können. Besonderes Augenmerk legt es dort auf die semiempirische Methode PM7, aus dem Programm `Molecular Orbital PACkage` (`MOPAC`), welches Ausgaben zu den Bindungsenthalpien und Bindungsordnungen in einem Molekül zurückliefert.

- Das **Kapitel 3** stellt die Grundlagen der Massenspektrometrie vor. Im ersten Schritt gibt es eine Intuition zum Aufbau eines Massenspektrometers und die dortige Erzeugung der Daten eines Massenspektrums. Besonderes Augenmerk legt das Kapitel anschließend auf die Ionisierungsmethoden Elektrosprayionisation und Elektronenstoßionisation, die in `ChemFrag` integriert sein werden.

  Einen großen Abschnitt widmet das Kapitel den existierende Methoden zur Aufklärung von MS/MS-Spektren. Es werden der Spektrenvergleich, die regelbasierte Fragmentierung, die kombinatorische Fragmentierung sowie Fragmentierungsbäume und Ansätze auf Basis des Maschinellen Lernens [3] vorgestellt

und diskutiert. Ihre Erläuterungen sind notwendig, um die Herangehensweise des regelbasierten Ansatzes in `ChemFrag` nachvollziehen zu können. Insbesondere erläutert das Kapitel die Programme `MetFrag`, `CFM-ID` und `SIRIUS` detaillierter, da sie später dem Vergleich von `ChemFrag` dienen. Weiterhin legt das Kapitel die Grundlage zum Verständnis des quantenchemischen Ansatzes, indem es die Ansätze von Mayer und Gömöry [6] sowie Alex *et al.* [7] beschreibt.

- Im **Kapitel 4** wird der regel-und quantenchemisch basierte Ansatz des Annotierungsprogramms `ChemFrag` vorgestellt. Zunächst beginnt das Kapitel, den allgemeinen Aufbau von `ChemFrag` zu beschreiben. Anschließend geht es im Detail auf die Schritte der Ionisierung, der Bindungsspaltung und Umlagerungen von Atomen und Atomgruppen durch den regelbasierten und den quantenchemischen Ansatz ein. Darin inbegriffen sind Erläuterungen zu den implementierten Regeln. Daran schließen sich Erklärungen zur Berechnung der 3D-Koordinaten mittels `RDKit` sowie der Bindungsenthalpie und der Bindungsordnungen durch `MOPAC` an. Den Abschluss bildet die Verifikation der Annotation.

  Um die Einsetzbarkeit von `ChemFrag` mit dem quantenchemischen Ansatz zu ermöglichen, stellt das Kapitel ebenfalls Methoden zur Reduktion quantenchemischer Berechnungen vor. Daran schließt sich der erste Abschnitt zu den Ergebnissen an. Darin zeigt das Kapitel, dass durch den Vergleich berechneter Protonenaffinitäten durch `MOPAC` mit experimentellen Werten die Verwendung von `MOPAC` zur Berechnung der Bindungsenthalpie eines Moleküls möglich ist. Weiterhin vergleicht es die Änderungen von Bindungsordnungen mit sich ändernden Bindungslängen, womit der Nachweis erbracht wird, dass Bindungsordnungen die Stabilität von Bindungen charakterisieren. Damit unterstreicht das Kapitel, dass `MOPAC` für den quantenchemischen Einsatz verwendet werden kann.

  Um anschließend die korrekte Vorhersage von Fragment-Ionen und Fragmentierungswegen durch `ChemFrag` zu bewerten, stellt das Kapitel die Ergebnisse für Ephedrin und Kokain von `ChemFrag` im Vergleich zu `MetFrag`, `CFM-ID`, `SIRIUS` und publizierten Fragmentierungswegen aus der Literatur vor. Für beide Substanzen bildet `ChemFrag` die gleichen Ergebnisse wie Thevis [8] ab, welche chemisch plausibler sind als die Annotation der Fragment-Ionen von `MetFrag` und `CFM-ID`. In einem zweiten Ergebnis-Abschnitt zeigt das Kapitel die Anwendung von `ChemFrag` auf einen Datensatz mit über 200 Molekülen. Auf Basis dieses Datensatzes führen wir eine Parameterermittlung durch. Wir ermitteln die optimale Fragmentierungstiefe sowie Eingabeparameter, die einen Kompromiss zwischen der Laufzeit und Anzahl annotierter Fragment-Ionen ermöglicht. Weiterhin beleuchten wir, welche Regeln besonders häufig von `ChemFrag` an-

gewendet werden und welche an der Annotation von Fragment-Ionen beteiligt sind.

Auf Basis der ermittelten Parameter führen wir ein letztes Experiment zur Anwendung von `ChemFrag` auf zwei verschiedene Naturstoffklassen durch. Auch hier zeigt sich, dass `ChemFrag` Fragmentierungswege erzeugt, die bereits in dieser Form veröffentlicht sind und eine hohe Anzahl Fragment-Ionen chemisch sinnvoll annotieren kann. Des Weiteren stellt das Kapitel Fragmentierungswege für Substanzen vor, zu denen noch keine Annotationen publiziert sind.

- Den Äquivalenzvergleich für Moleküle stellt das **Kapitel 5** vor. Diese Thema leitet sich aus dem notwendigen Molekülvergleich in `ChemFrag` ab. Um die Rechenzeit von `ChemFrag` essentiell zu minimieren, müssen redundante quantenchemische Rechnungen von generierten Molekülen verhindert werden. Dafür ist die Einbindung eines korrekten Molekülvergleichs notwendig. Im ersten Teil stellt das Kapitel die Begrifflichkeit Graphisomorphie und etablierte Algorithmen, wie den `nauty`-Algorithmus oder den Backtracking-Algorithmus von Ullmann [9] und die `VF2++`-Bibliothek [10] dar. Anschließend erläutert das Kapitel ausführlich den Aufbau von `MET`, der sich in zwei Phasen untergliedert. Für die erste Phase erklärt das Kapitel die Umwandlung des Moleküls in einen gelabelten Graphen, wo mehrere notwendige chemische und strukturelle Eigenschaften für die Umwandlung vorgestellt werden. Die zweite Phase besteht aus einem Backtracking-Algorithmus zur Bestimmung der Äquivalenz zweier Moleküle und anschließender Zuordnung der Atome.

  Anschließend wird `MET` experimentell evaluiert. Es erfolgt eine Einschätzung der Kandidatenreduktion, um deren Effizienz in `MET` zu bewerten. Weiterhin widmet sich das Kapitel dem Vergleich von `MET` mit dem Programm `VF2++`. Darin inbegriffen ist die Bestimmung der optimalen Nachbarschaftstiefe als Schlüsseleigenschaft zur Charakterisierung von Atomen. Neben `VF2++` vergleicht das Kapitel die Korrektheit und die Laufzeit von `MET` mit den Ergebnissen der Bibliotheken `RDKit` und `SMSD` sowie eines String-basierten Vergleichs durch SMILES und InChI‘s. Das Kapitel stellt durch die Experimente heraus, dass `MET` in der Korrektheit den String basierten Methoden sowie `SMSD` und `RDKit` überlegen ist. Weiterhin verdeutlicht der Laufzeitvergleich, dass `MET` besonders für Moleküle bis 400 Atome der effizienten und schnellen Bibliothek `VF2++` überlegen ist. Ein weiterer Schwerpunkt dieses Kapitels ist die Repräsentation der Moleküle als Fingerprints und deren Einordnung in Äquivalenzklassen gleicher Moleküle.

- Das abschließende **Kapitel 6** fasst die wesentlichen Ergebnisse von `ChemFrag` und `MET` zusammen und weist kurz auf offene Problemstellungen hin.

2. Grundlagen der Chemoinformatik

Dieses Kapitel wird einen kurzen Überblick zu den Grundlagen der Chemoinformatik liefern, die in dieser Arbeit Anwendung finden. Der erste Abschnitt enthält eine Aufzählung von verschiedenen Arten der Molekülrepräsentation. Daran schließt sich die Vorstellung von zwei häufig verwendeten Bibliotheken zur Chemoinformatik an. Abschließend erläutert das Kapitel Möglichkeiten zur Energieminimierung von Molekülen.

2.1. Molekülrepräsentation in der Chemoinformatik

Um wissenschaftliche Fragestellungen aus den Bereichen Chemie, Biochemie oder Pharmazie beantworten zu können, müssen Moleküle zunächst maschinenlesbar kodiert werden. Die Strukturen kleiner Moleküle können in verschiedenen Formaten, wie SDF, MOL oder Chemical Markup Language (CML) eingelesen und ausgegeben werden. Alle Formate ermöglichen die 2D- und 3D-Repräsentation der Strukturen. Die folgenden Unterkapitel gehen auf eine Auswahl dieser Methoden ein. Abhängig von den Formaten ist dabei die interne Molekülrepräsentation verschieden und von den verfügbaren Datenstrukturen abhängig. Zwei häufig verwendete Chemoinformatik-Bibliotheken werden im nachfolgenden Kapitel erläutert.

2.1.1. Etablierte Datei-Formate für Moleküle

In der Chemoinformatik existieren eine Vielzahl von Datei-Formaten zur Speicherung von Molekülstrukturen. Besonders haben sich das MOL/SDF- und das PDB-Format durchgesetzt [11]. Den Aufbau dieser Dateien möchten wir nun kurz beschreiben. MOL/SDF- und PDB sind Textdateien, wie ein Ausschnitt in Abbildung 1 zeigt. Diese Textdateien enthalten notwendige Informationen der Molekülstruktur, die für die Anwendung von `MET` in Kapitel 5 von Bedeutung sind. Eine Datei besteht dabei aus Blöcken für Atome und Bindungen. In den Atomblöcken sind die einzelnen Atome unter anderem mit Elementsymbol und Koordinaten aufgeführt. Die Bindungs-Blöcke enthalten die Art der Bindung und die Indizes der an der Bindung beteiligten Atome. Zusätzlich können in weiteren Blöcken Informationen über Ladungen, Radikale, Isotope und Stereochemie aufgeführt sein. Werden mehrere Moleküle in einer Datei gespeichert, wird das SDF-Format bevorzugt. In diesem ist jedes Molekül im MOL-Format kodiert. Viele der existierenden Chemoinformatik-Programme ermöglichen das Einlesen und Ausgeben dieses Formats.
Neben der Kodierung als Textdatei mit kartesischen Koordinaten existiert ebenfalls noch die Darstellung als String, die wir in den zwei folgenden Unterkapiteln genauer betrachten möchten.

```
triethylamin
  MOE2020           3D

  6  5  0  0  0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.4280    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.9040    1.3004   -0.3485 N   0  0  0  0  0  0  0  0  0  0  0  0
    1.4280    2.2524    0.6035 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.4280    1.6489   -1.6489 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.9040    2.9494   -1.9974 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  3  5  1  0  0  0  0
  5  6  1  0  0  0  0
M  END
$$$$
```

Abbildung 1: Darstellung von Triethylamin aus Abbildung 2 im MOL-Format.

2.1.2. SMILES und SMARTS

David Weininger entwickelte 1988 die Zeichenketten-Notation Simplified Molecular Input Line Entry Specification (SMILES) [12]. In Abbildung 2 sehen wir die Strukturen und die SMILES für Triethylamin und 3-Ethylpentan. Die Notation repräsentiert Atome durch ihre Elementsymbole. Großbuchstaben stehen dabei für aliphatische Atome, wohingegen aromatische Atome mit Kleinbuchstaben kodiert sind. Die Strukturen aus Abbildung 2 enthalten beispielsweise nur aliphatische Atome. Im Vergleich dazu bildet der SMILES `c1ccccc1` einen Benzenring ab. Sollen die SMILES explizit Wasserstoffe angeben, dann werden die Atomsymbole in eckigen Klammern mit der Anzahl der Wasserstoffe geschrieben. Weiterhin kann diese Klammer auch die Formladung enthalten. Beispielsweise codiert `CH3+` ein Methyl-Ion. Eine implizite Wasserstoffdarstellung ist durch SMILES ebenfalls möglich. Verbunden werden die Elementsymbole durch Bindungsnotationen. Einzelbindungen sind durch einen Bindestrich symbolisiert, die optional gesetzt werden können. Doppelbindungen müssen durch = und Dreifachbindungen mit # kodiert werden. Ein Doppelpunkt visualisiert aromatische Bindungen. SMILES kodiert Ringe mit gleichen Zahlen an den Positionen des Ringschlusses und Abzweigungen durch runde Klammern [13].

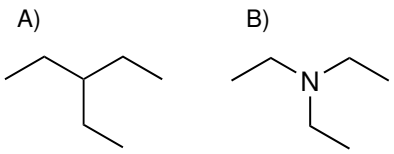


Abbildung 2: Darstellung der Molekülstruktur von 3-Ethylpentan (A), kodiert mit dem SMILES `C(CC)(CC)CC`, und Triethylamin (B), kodiert mit dem SMILES `N(CC)(CC)CC` .

Eine Erweiterung von SMILES sind SMARTS (SMILES arbitrary target specification), die Substrukturen definieren können. Der Einsatz von SMARTS liegt in der Suche von Unter- oder Teilstrukturen. Jeder SMILES ist damit automatisch ein gültiger SMARTS. Die Umkehrrichtung ist jedoch nicht gültig. Um in einem SMARTS eine beliebige Bindung zwischen zwei Atomen auszudrücken, wird die Bindung als Tilde gesetzt. Das Erlauben verschiedener Atome an einer Position erfolgt durch das Setzen der zulässigen Elementsymbole in eckigen Klammern. Die Elementsymbole sind innerhalb der Klammern durch Kommas getrennt. Der SMARTS `CC[N,C](CC)CC` repräsentiert durch diese einzige Notation die Moleküle Triethylamin und 3-Etyhlpentan. Dieser SMARTS zeigt, dass an der dritten Position entweder ein Kohlenstoffatom oder ein Stickstoffatom positioniert sein kann.
Eine Herausforderung der SMILES und SMARTS Generierung ist die Kanonisierung. Um für identische Moleküle den gleichen SMILES zu erzeugen, müssen die Atome in einer kanonischen Form ausgegeben werden [14]. Für die SMILES und SMARTS Generierung existieren verschiedene Algorithmen, die beispielsweise von DayLight [15], CDK [16], RDKit [17] bereit gestellt werden. Aufgrund der verschiedenen Implementierungen können sich die generierten SMILES der Bibliotheken leider unterscheiden. Beispielsweise ist es möglich, Ethanol durch `CCO` oder durch `OCC` zu repräsentieren. Somit kann ein Molekül nicht eindeutig durch einen SMILES abgebildet werden. Eine eindeutige Abbildung zwischen einem SMILES und einer Molekülstruktur findet daher meist nur innerhalb eines Programms statt. Ebenfalls ist Algorithmen abhängig, ob Isotope und Stereochemie kodiert werden können [18]. Beispielsweise zeigt O'Boyle [14] diese Kodierungen in seiner kanonischen SMILES Generierung, die in der ursprünglichen Implementierung von Weininger *et al.* nicht enthalten war.

2.1.3. InChI

Um eine einheitliche kanonische Repräsentation zu ermöglichen, begann 2001 unter Aufsicht der IUPAC (International Union of Pure and Applied Chemistry) die Entwicklung eines neuen Standards zur Beschreibung chemischer Strukturen [19]. In dem dreistufigen Prozess, der Normalisierung, Kanonisierung und Serialisierung, wandelt der Algorithmus die Strukturinformationen in eine Zeichenkette um.
Ein InChI (International Chemical Identifier) kann aus sechs Hauptebenen, die wiederum Subebenen haben können [18], bestehen. Diese werden jeweils durch ein / voneinander getrennt. Zu den sechs Ebenen zählen:

- Hauptebene
- Ladungsebene
- Stereochemieebene

- Isotopenebene
- fixierte-H-Ebene
- „Reconnected Layer“

Die in Abbildung 2 dargestellten Strukturen haben für die Hauptebene die folgenden InChI-Kodierungen:

Triethylamin: InChI=1S/C6H15N/c1-4-7(5-2)6-3/h4-6H2,1-3H3

3-Etyhlpentan: InChI=1S/C7H16/c1-4-7(5-2)6-3/h7H,4-6H2,1-3H3

Im Vergleich zu der SMILES Darstellung ist ein Molekül durch einen InChI repräsentiert eindeutig und ist daher nicht abhängig von dem verwendeten Programm.

Die bereits genannten Repräsentationsarten haben alle den Vorteil, dass sie die Strukturen von Molekülen kurz und leicht interpretierbar repräsentieren können. Weiterhin dienen sie einem einfachen Transfer zwischen verschiedenen Programmen, da ihre Kodierungen etabliert sind. Ein Nachteil dieser Darstellungen ist, dass Informationen über 2D- oder 3D-Strukturen darin nicht kodiert sind. Als weiterer Nachteil ist zu nennen, dass sie nicht zur direkten Weiterverarbeitung und Bearbeitung von Molekülen geeignet sind. Die eingelesenen Moleküle müssen zunächst in eine andere Datenstruktur umgewandelt werden. Eine Möglichkeit dafür ist die Graph-Darstellung, die wir uns nachfolgend ansehen.

2.1.4. Graphrepräsentation

Um mit computergestützten Methoden die Struktur der Moleküle zu bearbeiten, ist deren Umwandlung in Graphen von Vorteil. Dabei unterscheiden wir zwischen zwei verschiedenen Varianten. Das sind zum einen die Molekülgraphen und die einfachen Graphen.

Molekülgraph Im Jahr 1874 wurde ein erstes Konzept zu Molekülgraphen vorgestellt [20]. Als Molekülgraph bezeichnet man einen ungerichteten Graphen, bei dem die Atome durch Knoten und die Bindungen durch Kanten dargestellt werden. In der Terminologie unterscheidet man zwischen „Plerogram“ und „Kenogram“. Ein „Plerogram“ ist dabei ein Wasserstoff einschließender Molekülgraph und ein „Kenogram“ ein Wasserstoff ausschließender Molekülgraph [20]. Ein Plerogram bildet daher die Wasserstoffe aus einem Molekül in einen Graphen ab.

Bei der Überführung einer Molekülstruktur in einen Molekülgraphen bilden die Atome des Moleküls die Knoten des Graphens. Um die Atomeigenschaften zu erhalten,

werden die Knoten mit diesen Eigenschaften gelabelt. Dazu gehören beispielsweise die Ordnungszahl, die Formalladung, die Anzahl der Wasserstoff sowie die Anzahl an freien Elektronen (Radikale). Die Kanten im Graphen stehen für zwei verbundene Atome. Hierbei ist zu unterscheiden, ob der Graph mit Mehrfachkanten oder mit Einfachkanten generiert werden soll. Die Mehrfachkanten können sowohl Doppel- als auch Dreifachbindungen darstellen. Handelt es sich um einfache Kanten, müssen diese mit einem Label versehen werden, die die Wertigkeit der Bindungen anzeigen. Zusätzlich sind sie notwendig, um die Information zur Aromatizität der Bindung in den Graphen zu überführen.

Einfacher Graph Im Vergleich zum Molekülgraphen handelt es sich bei einem einfachen Graphen um einen ungerichteten Graphen ohne Mehrfachkanten, Schleifen und Labeln, wie in der untersten in Grafik (SIMPLE GRAPH) aus Abbildung 3 zu erkennen ist. Zunächst wird für jedes Atom ein Knoten erzeugt. Abhängig vom seinem Elementtyp wird dieser Knoten mit weiteren Dummy-Knoten verbunden, die den Elementtyp abbilden sollen. Durch die reine Dummy-Erweiterung ist es jedoch schwer möglich Eigenschaften zur Ladung, Radikalen oder Isotopen zu kodieren. Existiert zwischen zwei Atomen eine Einfachbindung, wird diese als Kante in dem Graphen zwischen den Knoten dargestellt. Handelt es sich um Mehrfachbindungen werden in der Kante ein bis zwei Dummy-Knoten eingefügt [21].
Abbildung 3 zeigt die schrittweise Umwandlung der Molekülstruktur eines Benzen-Derivates in einen einfachen Graphen.
Der Vorteil dieser Darstellung ist, dass nur die Struktur der Graphen mit seinen Knoten und Kanten überprüft werden muss, ohne zusätzliche Überprüfungen der Atom- und Bindungseigenschaften in den Labels durchzuführen. Ein großer Nachteil ist, dass durch die Einbindung der Dummy-Atome sich die Größe der Graphen extrem vergrößert, wodurch sich die Laufzeit erhöhen kann. Weiterhin ist es schwer möglich alle Eigenschaften durch Dummy-Atome zu repräsentieren, wie beispielsweise die Aromatizität von Atomen und Bindungen oder Unterscheidung der Isotope. Je mehr Eigenschaften durch die Dummy-Atome kodiert werden, desto leichter kann es passieren, dass durch die Vielzahl der Dummy-Atome ein Graph nicht mehr eindeutig ein Molekül repräsentiert.

2.2. Programme und Bibliotheken der Chemoinformatik

Nach der Auflistung der verschiedenen Möglichkeiten zur Molekülrepräsentation, betrachten wir jetzt zwei Programme bzw. Programmier-Bibliotheken, die die Arbeit mit Molekülstrukturen ermöglichen. Zu den bekanntesten Bibliotheken zählen das `Chemistry Development Kit` (`CDK`), auf Java-Basis, sowie das Python-basierende `RDKit`.

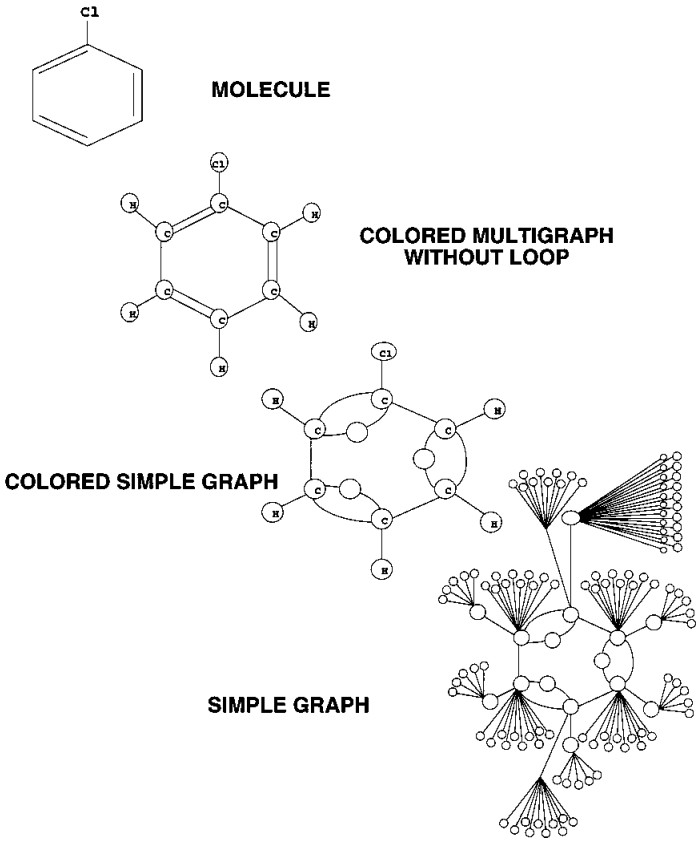


Abbildung 3: Schrittweise Umwandlung eines Benzen-Derivates in einen Einfachen Graphen (adaptiert von [22]).

2.2.1. `Chemistry Development Kit`

Die Java-Bibliothek Chemistry Development Kit ist eine seit über zehn Jahren existierende Open-Source Bibliothek, die der Kodierung und Bearbeitung von Molekülstrukturen dient [23]. Die in `CDK` implementierten Datenstrukturen ermöglichen das Aus- und Einlesen von Molekülstrukturen aus verschiedenen Formaten. Dazu zählen beispielsweise CML-Format, SDF-Format, PDB-Fomat sowie die SMILES- und InChI-Kodierung. Weiterhin bietet `CDK` die Möglichkeit Molekülstrukturen zu verändern sowie beispielsweise die Suche nach Ringstrukturen, Substruktur- und Ähnlichkeitssuche in einer Molekülstruktur durchzuführen. Zusätzlich ist es durch `CDK` möglich, einfache chemische Eigenschaften von Molekülen zu bestimmen. Hier sind unter anderem die Bestimmung der verschiedenen Massen, die Ermittlung der Atomtypen oder die Berechnung der Summenformel zu nennen [16].

2.2.2. `RDKit`

Das frei verfügbare Chemoinformatik Programm `RDKit` wurde ab 2006 mit dem ursprünglichen Ziel entwickelt, prädiktive Modelle für toxikologische und biologische

Aktivitäten zu erstellen [17]. Aktuell ist es ein häufig genutztes Programm zur Analyse chemoinformatischer Fragen. `RDKit` stellt ebenso das Einlesen und Ausgeben von Molekülstrukturen, deren Untersuchung (z.B. Ringsuche) und Generierung von 2D- und 3D-Koordinaten bereit. Ebenso kann `RDKit` Substruktursuche, Maximum Common Substruktur Bestimmung und Fingerprint-Methoden durchführen. Weiterhin ist der Umgang mit chemischen Reaktionen möglich. Die Algorithmen dazu sind in C++ implementiert. Zusätzlich existieren Wrapper für Python und Java [17].

2.3. Energieminimierung von Molekülstrukturen

Ein anderes Feld der Chemoinformatik, welches im weiterem Kontext der Arbeit von Bedeutung ist, ist die Molekülstrukturoptimierung. Ziel dieser Optimierung ist die energetische Minimierung der Molekülstruktur. Diese benötigen wir für das Annotationsprogramm `ChemFrag`, aus Kapitel 4. `ChemFrag` nutzt in seinem quantenchemischen Ansatz energie-minimierte Strukturen, um korrekte Bindungsordnungen und Bindungsenthalpien zu bestimmen. Bei der Optimierung unterscheiden wir zwischen empirischen Methoden auf Basis von Kraftfeldern, ab-inito-Methoden, wie das Lösen der Hartree-Fock-Gleichung oder die Dichtefunktionaltheorie (DFT) und semiempirischen Methoden. Die ab-initio Methoden sind von den genannten Methoden die genausten, da sie ohne empirische Näherung auskommen. Ihr großer Nachteil liegt darin, dass sie nur für kleine Moleküle gelöst werden können, da nur die Gleichungen für einige wenige Elektronensysteme analytisch gelöst werden können. Im Vergleich dazu dienen die Kraftfeld-basierten Methoden der schnellen Geometrieoptimierung großer Molekülstrukturen. Einen vertiefenden Einblick dazu liefert das Lehrbuch „Einführung in die Quantenchemie“ von Lechner [24]. Im folgenden Abschnitt betrachten wir nur die in `ChemFrag` verwendete semi-empirische Methode.

2.3.1. Semi-Empirische Methode

Die semi-empirischen Methoden bilden einen Ansatz, der die Vorteile von ab-initio-Methoden und empirischen Methoden kombiniert. Ziel ist die schnelle und trotzdem präzise Vorhersage des Energieminimus einer Molekülstruktur [25]. Dazu verwenden semi-empirische Methoden ab-initio Methoden, wie die Hartree-Fock-Gleichung, und vereinfachen diese durch das Einsetzen experimentell bestimmter Werte. Dadurch kommt es zu einer Beschleunigung der Berechnung. Zusätzlich erfolgt eine Reduzierung der Rechenzeit durch die Beschränkung auf Valenzelektronen während der Berechnung. Dadurch können semi-empirische Methoden auch für größere Molekülstrukturen angewendet werden. Stewart entwickelte das Programm `MOPAC` [26], welches unter anderem die semi-empirische Methode PM7 [27] enthält. Ein Problem der semi-empirischen Methode ist, dass randomisierte Algorithmen verwendet werden,

die ein nicht deterministisches Verhalten zeigen. Das führt dazu, dass in mehreren Berechnungen leicht abweichende Werte für die Bindungsenthalpie bestimmt werden können.

3. Grundlagen der Massenspektrometrie und Fragmentierung

Die Identifikation und strukturelle Charakterisierung von kleinen Molekülen ist in verschiedenen Wissenschaftsbereichen von großer Bedeutung. Beispielsweise können wir dafür die Chemie, Pharmazie und die Biochemie oder auch die Archäologie benennen [28, 29]. Aber auch im Bereich der Metabolitenidentifikation [30], der Diagnostik sowie der Dopinganalyse [8] ist die Aufklärung kleiner Moleküle von starkem Interesse. Als Schlüsseltechnologien für die Detektion und Identifikation dieser Moleküle zählen die Massenspektrometrie (MS) sowie die NMR-Spektroskopie [31].

Im Blickpunkt der vorliegenden Arbeit liegt die Massenspektrometrie als Identifikationsmethode. Die Massenspektrometrie zeichnet sich durch eine hohe Sensitivität und Schnelligkeit, geringer Detektionsbeschränkungen sowie vielfältiger Einsatzmöglichkeiten aus, da selbst die Analyse von Molekülen in äußerst geringen Mengen möglich ist [31].
In den folgenden Kapiteln lernen wir die Methode der Massenspektrometrie sowie deren Verwendung zur Charakterisierung von Molekülen genauer kennen.

3.1. Prinzip und Entwicklung der Massenspektrometrie

Das Ziel der Massenspektrometrie ist die Molekülidentifikation anhand des Masse-zu-Ladungs-Verhältnisses (m/z) eines Stoffes und dessen charakteristischen (Fragment)-Ionen. Grundlage dafür bildet ein vom Massenspektrometer generiertes Massenspektrum. In diesem sind Informationen zur Masse und Häufigkeit von Molekül- und Fragment-Ionen enthalten. Beispielhaft zeigt die Abbildung 5 das Massenspektrum für Triethylamin mit einem Masse-zu-Ladungsverhältnis von 101 Da und einer relativen Intensität von 20 % zum Basispeak für das positiv geladene Molekül-Ion.

Historisch betrachtet, legte J.J. Thomson mit der Identifikation der Neon-Isotope ^{20}Ne und ^{22}Ne den Grundstein für die Massenspektrometrie. Thomson konnte in einem Experiment die Neon-Isotope ^{20}Ne und ^{22}Ne massenspektrometrisch darstellen und bildete damit einen ersten Meilenstein in der Molekülidentifikation [32]. Einen ersten großen Durchbruch verzeichnete die Massenspektrometrie als Analysemethode im Jahr 1960 als Elementzusammensetzungen und Strukturzusammensetzungen von Gemischen aufgeklärt werden konnten.
Grundlage der Massenspektrometrie ist stets die Ionisierung des Moleküls im ersten Schritt, wobei positiv oder negativ geladene Molekül-bzw. Fragment-Ionen gebildet werden. Um ein Molekül zu ionisieren, wurde anfangs die Elektronenstoßionisation verwendet. Der limitierende Faktor der Elektronenstoßionisation (EI) ist die geringe

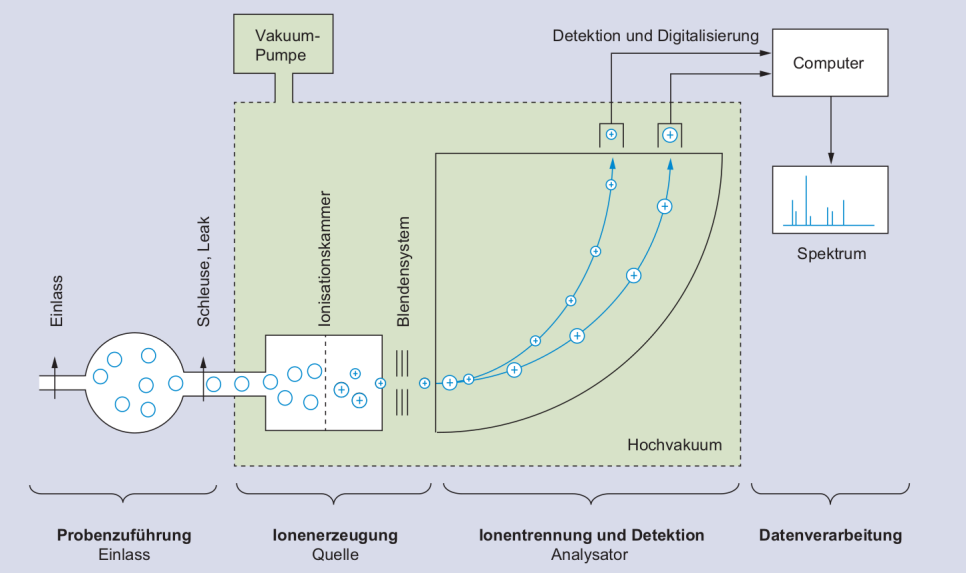


Abbildung 4: Darstellung der Funktionsweise eines Massenspektrometers [34].

Masse zu analysierender Moleküle, sodass die Einsatzmöglichkeiten der Massenspektrometrie anfangs noch begrenzt waren. Durch die Entwicklung der Elektrosprayionisation (ESI) als weiche Ionisation im Jahr 1984 [33] konnten dann Moleküle bis 1.000.000 Da direkt aus Lösungen untersucht werden. Diese Ionisationsmethode verhalf der Massenspektrometrie zu einer der wichtigsten Analysemethode in verschiedensten Wissenschaftsbereichen.
Im Folgenden sehen wir uns die Funktionsweise eines Massenspektrometers sowie die zwei häufig verwendeten Ionisationsmethoden EI und ESI im Detail an.

3.1.1. Aufbau eines Massenspektrometers

Ein Massenspektrometer besteht aus vier Bereichen, wie wir in Abbildung 4 erkennen. Im ersten Abschnitt findet die Probenzufuhr (Einlass) statt. Nach dem Einlass erfolgt die Ionenerzeugung, da nur geladene Molekül-Ionen später detektiert werden können. Zur Erzeugung von Fragment-Ionen wird eine weitere Einheit zwischen die Ionenerzeugung und den Analysator eingefügt. Diese Funktionsweise ist als MS/MS bekannt. Im sich anschließenden Analysator herrscht ein elektrisches/magnetisches Feld vor. Dieses kann die Ionen nach ihrem Masse-zu-Ladungs-Verhältnis selektieren und stellt damit sicher, dass die Ionen die gleiche Richtung und Geschwindigkeit aufweisen. Das ist Voraussetzung, um die korrekte Masse ermitteln zu können. Nachdem die Ionen nach ihrer geraden Flugbahn sortiert sind, gelangen sie in den Detektor. Dort herrscht ein weiteres Magnetfeld vor, dass die Ionen in einen Halbkreis ablenkt. Je größer die Masse eines Ions ist, desto größer ist auch der Radius des Halbkreises. Basierend auf dieser Ausgabe erfolgt im letzten Schritt die Verarbeitung der Da-

ten und die Spektrenerzeugung [34]. Das Masse-zu-Ladungs-Verhältnis stellt dann die massenabhängige Bewegung der Ionen im elektrischen/magnetischen Feld dar. Anhand der unterschiedlichen Bewegung der Ionen treten diese zu unterschiedlichen Zeitpunkten an einem Ort zur Detektion auf. Abbildung 5 zeigt ein erläuterndes Spektrum.

Dieses Spektrum ist das Ergebnis einer massenspektrometrischen Analyse und enthält die Informationen zu Masse und Häufigkeit der detektierten Ionen. Im Massenspektrum ist auf der Abszisse der m/z-Wert des jeweils detektierten Ions erkennbar und dazugehörig auf der Ordinate dessen Intensität. Bei der Angabe der Intensität handelt es sich in den meisten Fällen um die relative Intensität. Sie stellt die Häufigkeit der Ionen im Verhältnis zum Basispeak dar. Der Basispeak ist das am häufigsten detektierte Ion und hat die Intensität von 100%. Ein weiterer wichtiger Peak ist der des ionisierten Ausgangsmoleküls mit der Bezeichnung Molekül-Ion-Peak. Alle weiteren Peaks stammen bei MS/MS-Spektren von Fragment-Ionen. Während des Fragmentierungsprozess bilden sich nicht nur die Fragment-Ionen sondern auch die Neutralverluste. Sie entstehen während der Fragmentierung, wenn ein Fragment-Ion in ein kleineres Fragment-Ion und ein neutrales Molekül zerfällt. Die Neutralverluste lassen sich anhand der Abstände zwischen den Peaks errechnen. Sie tauchen jedoch nicht explizit im Spektrum auf, da sie neutral geladen sind und damit im elektrischen/magnetischen Feld nicht beeinflusst werden.

3.1.2. Ionisationsmethoden

Wie bereits erwähnt, existieren verschiedene Möglichkeiten, den zu untersuchenden Stoff zu ionisieren. Zu den beiden häufigsten Methoden zählen dabei die Elektronenstoßionisation und die Elektrosprayionisation, die wir nachfolgend detaillierter erläutern. Weitere Methoden sind beispielsweise die chemische Ionisierung oder Matrix Assisted Laser Desorption Ionisierung (MALDI) [36].

Elektronenstoßionisation

Die Elektronenstroßionisation (EI) tritt vielfach in Kombination mit der Gaschromatografie auf und ist eine Verdampfungsmethode. Sie zählt als älteste eingesetzte Ionisierungsmethode der Massenspektrometrie. Bei der Elektronenstroßionisation bildet sich nach der Ionisation ein positiv geladenes Radikal. Dieses Radikal-Kation entsteht, wenn durch eine hohe Energiezufuhr Elektronen mit dem Analyten kollidieren und dem Analyten dadurch Elektronen entrissen werden. Häufig wird dafür eine Ionisierungsenergie von 70 eV verwendet [37]. Eine solche Reaktion zeigt Abbildung 33. Das Freisetzen eines Elektrons ist am leichtesten bei Atomen mit nicht vollbesetzten Außenschalen. Dazu gehören unter anderem Sauerstoff und Stickstoff.

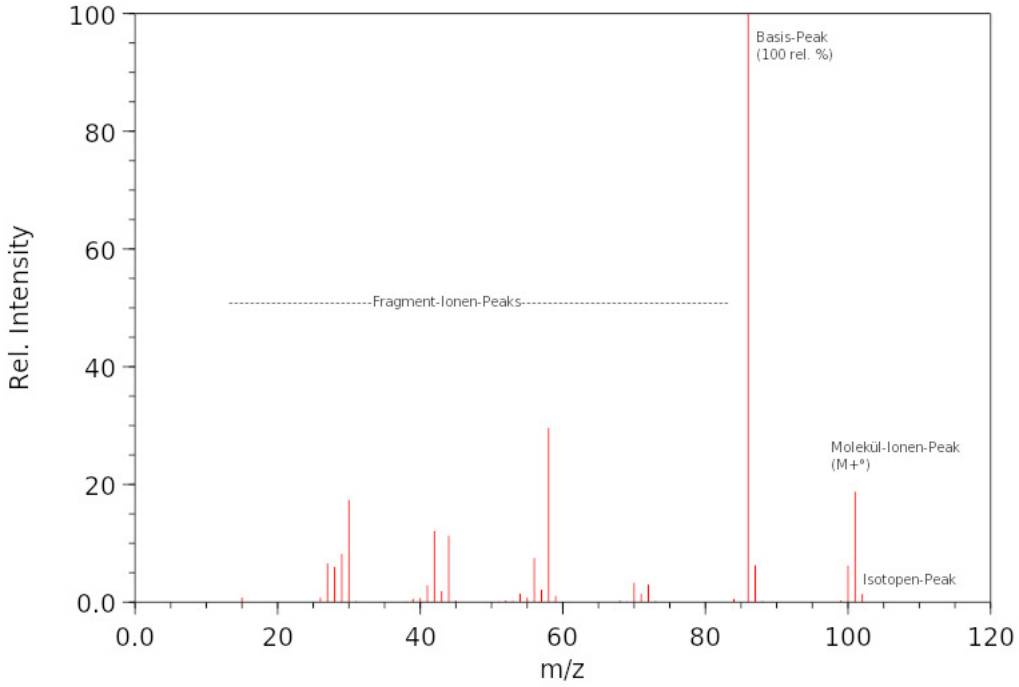


Abbildung 5: Darstellung des EI-Massenspektrums für Triethylamin (Abbildung 2). Das Spektrum ist der NIST-Datenbank [35] entnommen.

Ein Vorteil der Elektronenstoßionisation ist die hohe Reproduzierbarkeit der Spektren. Diese Spektren werden in Datenbanken, z.B. NIST [35] oder Wiley [38] gespeichert und können für Spektrenvergleiche herangezogen werden [39]. Auf Basis der reproduzierbaren Spektren konnten im Laufe der Entwicklung gute Kenntnisse über das Verhalten von Radikal-Kationen gewonnen werden. Als Nachteil dieser Methode benennt Bienz in [34], dass nur Moleküle mit geringen Massenzahlen ($\leq$ 600 Da) analysiert werden können. Bei größeren Molekülen erfolgt durch eine höhere Anzahl an funktionellen Gruppen eine stärkere thermische Zersetzung.

Elektrosprayionisation

Die Elektrosprayionisation (ESI) ist eine Zerstäubungsmethode und wird häufig mit der Flüssigkeitschromatografie kombiniert. Sie wurde zur Untersuchung von Makromolekülen entwickelt, die eine Masse von größer als 600 Da besitzen. Durch die ESI können selbst Biopolymere, wie Proteine oder Nukleinsäuren, mit Massen bis zu 1.000.000 Da direkt aus Lösungen untersucht werden. Diese Ionisierungmethode findet außerdem Anwendung im Bereich der Analytik kleiner polarer Moleküle und zur Untersuchung von Metall-Komplexen [40].

Bienz zeigt in seinem Lehrbuch [34], dass bei ESI der Probenstrom durch eine Kapillare geführt wird, an deren Ende sich eine Spannung befindet und wodurch die

Lösung in ein elektrisches Feld gesprüht wird. Durch die angelegte Spannung an der Kapillare und an der Gegenelektrode zerstäubt sich die Probenlösung und es kommt zur Ionisierung. Bei positiver Spannung entstehen positiv geladene Ionen und bei negativer Spannung negative Ionen.
Da diese Ionisierungsmethode in verschiedenen Kombinationen vorkommen kann, entstehen geräteabhängig und energieabhängig verschiedene Spektren. Die Reproduktion der Spektren ist dadurch erschwert. Aus diesem Grund wurden neue Datenbanken entwickelt. Beispielsweise sind die Datenbanken METLIN [41], GNPS [42] und MassBank [43] zu nennen, in denen zusätzlich zum Spektrum auch die Gerätekonfiguration angegeben wird. Besonders gut ist Elektrosprayionisation für die Ionisierung polarer Verbindungen geeignet. Polare oder basische Verbindungen werden üblicherweise leicht protoniert oder lagern leicht Alkali- oder andere Kationen an, sodass Ionen entstehen [34]. Diese Ionen können entweder positiv oder negativ geladen sein. Das Beispiel aus Abbildung 6 zeigt, dass für einzelne Atomgruppen sowohl die Bildung negativer als auch positiver Ionen möglich ist.

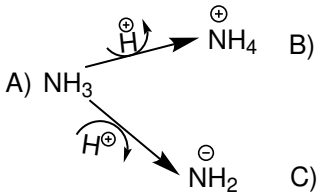


Abbildung 6: Darstellung der Protonierungs- und Deprotonierungsreaktion bei Ammoniak.

3.1.3. Massengenauigkeit

Für die Zuordnung von Fragment-Ionen zu den Peaks in einem Spektrum ist unter anderem die Art der Massenangabe und damit der Massengenauigkeit entscheidend. Im Bereich der Massenspektrometrie treten mehrere Massenbegriffe auf. Laut dem Lehrbuch von Hesse *et al.* [1] unterscheidet man zwischen der nominalen Masse und der monoisotopischen Masse, auch exakte Masse genannt. Die nominale Masse eines Ions oder Moleküls ist die Summe der nominalen Massen der daran beteiligten Elemente, wobei die nominale Masse eines Elementes definiert ist als die ganzzahlige Masse des natürlich am häufigsten vorkommenden, stabilen Isotops dieses Elementes. Als monoisotopische Masse eines Ions definieren Hesse *et al.* [1] die isotopische Masse, auch exakte Masse genannt, desjenigen Isotopen-Ions, das ausschließlich aus den häufigsten natürlichen Isotopen der einzelnen beteiligten Elementen zusammengesetzt ist. Die monoisotopische Masse ermöglicht durch die Nachkommastellen eine genauere Unterscheidung zwischen den chemischen Verbindungen. Beispielweise haben Benzol (C_6H_6) und 3-Fluoropropan-1-ol (C_3H_7FO) die gleiche Nominalmasse

von 78 Da. Jedoch besitzen sie unterschiedliche exakte Massen mit 78,11 Da und 78,09 Da.

Betrachten wir den Molekül-Ionen-Peak aus Abbildung 5, hat dieser eine Masse von 101 Da. Die monoisotopische Masse des einfach positiv geladenen Triethyamin-Radikal-Ions ist 101,19 Da. Um diesen Massenunterschied von 0,1 Da auszugleichen, wird der Begriff der Massengenauigkeit verwendet. Dieser gibt die Abweichung der gemessenen Masse zur berechneten exakten Masse an. Die relative Massengenauigkeit wird in ppm (parts per million) und die absolute in Dalton (Da) angeben. In dem gegebenen Beispiel ist der Unterschied -0,19 Da und -1881 ppm.

3.2. Algorithmische Ansätze zur Annotation von Fragment-Ionen-Spektren

Nachdem wir uns in den vorangegangen Kapiteln mit der Funktionsweise und dem Aufbau eines Massenspektrometers beschäftigt haben, möchten wir in diesem Kapitel das Augenmerk auf die verschiedenen Ansätze zur Annotation und damit Aufklärung der Fragment-Ionen-Spektren legen. Für die Annotation ist stets das Massenspektrum und manchmal auch die Struktur des Analyten bekannt.

Die Annotation eines Massenspektrums ist eine große Herausforderung. Ist der direkte Vergleich eines Spektrums mit einem in der Datenbank gespeicherten Referenzspektrum nicht möglich, können verschiedenste computergestützte Ansätze zur *in-silico* Fragmentierung verwendet werden. Ziel der *in-silico* Fragmentierung ist es, den Fragmentierungsprozess zu simulieren und ein Massenspektrum oder die Annotation der Fragment-Ionen vorherzusagen. Abbildung 7 zeigt einen Überblick über fünf häufig verwendete Methoden, die wir nachfolgend im Detail verstehen und erläutern werden.

Spektrenvergleich

Bei einem Spektrenvergleich wird das gegebene Spektrum in einer Spektrenbibliothek gesucht. Ziel ist es bei einem erfolgreichen Auffinden des Spektrums den Analyten zuordnen zu können. Die Grundlage eines Spektrenvergleich ist, dass das analysierte Molekül bereits zuvor massenspektrometrisch identifiziert und in der Datenbank gespeichert wurde. Dieses Verfahren kommt hauptsächlich bei der Interpretation von Spektren mittels EI zum Einsatz, da diese Methode reproduzierbare und vergleichbare Spektren liefert [44]. Hufsky *et al.* benannte bereits 2014 als Nachteil, dass die Anzahl an Substanzen (rund 111 Millionen in der PubChem) deutlich höher ist als die Anzahl gespeicherter Spektren in den zugehörigen Datenbanken (NIST:200.000, Wiley: 600.000). Daher gibt es für eine Mehrheit an Substanzen keine zugehörigen

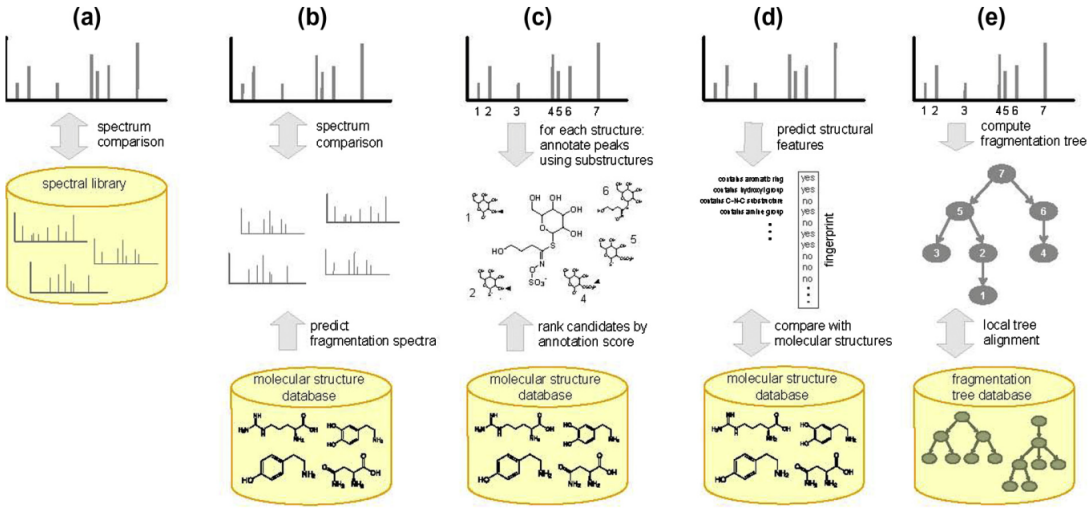


Abbildung 7: Überblick zu fünf Ansätzen der *in-silico* Fragmentierung [3].
(a) Suche in Spektrenbibliotheken,
(b) Vorhersage von Spektren,
(c) kombinatorische Fragmentierung,
(d) Vorhersage von Struktureigenschaften,
(e) Fragmentierungsbäume

Spektren in den Datenbanken. Aufgrund dieser Differenz wird es häufig dazu kommen, dass zu einem generierten Spektrum kein Ergebnis für den Spektrenvergleichs vorliegt und der Analyt nicht bestimmt werden kann.

Regelbasierte Fragmentierung

Ist kein Vergleichsspektrum in einer Datenbank hinterlegt, bietet der regelbasierte Ansatz eine neue Möglichkeit zur Analyse. Dieser Ansatz ist ein zweistufiger Prozess. Der erste Schritt ist die Vorhersage des Massenspektrums für die Substanz. Diese Vorhersage basiert auf der Simulation des Fragmentierungsprozesses durch implementierte Fragmentierungsregeln. Im zweiten Schritt erfolgt der Vergleich des simulierten Spektrums mit dem Spektrum der analysierten Substanz. `MassFrontier` [45], `ACD/MS Fragmenter` [46] und `MOLGEN-MS(F)` [47] zählen zu etablierten Software-Paketen, die dieses Vorgehen verwenden [48]. Ein Nachteil dieses Ansatzes ist, dass nur Regeln zur Fragmentierung angewendet werden können, die bereits implementiert sind. Diese Regeln werden zu großen Teilen aus der Literatur aber auch durch Data Mining Ansätze gelernt. Aufgrund der gelernten Regeln, lassen sich spontane Umlagerungen oder Spaltungen, die in quantenchemischen Rechnungen erkannt werden würden, nicht durch die vorgeschriebenen Regeln simulieren. Die Folge ist, dass unterschiedliche Spektren entstehen und ein erfolgreicher Vergleich nicht möglich ist.

Kombinatorische Fragmentierung Ziel der kombinatorischen Fragmentierung ist die Annotation der Fragment-Ionen eines Massenspektrums. Dieser Ansatz generiert durch eine Kombination von Bindungsspaltungen die Fragment-Ionen, wodurch Teilstrukturen abgespalten werden und neue Teil-Fragment-Ionen entstehen. Für die Stabilität der Bindungen und der Fragment-Ionen zieht dieser Ansatz die Bindungsdissoziationsenergie oder die Bindungsenergie heran [44]. Da hier nur Bindungen gespalten und nicht neu verknüpft werden oder sogar Atome, wie Protonen, nicht verschoben werden können, unterliegt dieser Ansatz der Annahme, dass sich Fragment-Ionen ohne Umlagerungen bilden. `EPIC`[49] und `FiD`[50] waren die ersten Vertreter mit diesem Ansatz [51]. In ihren Ansätzen bestimmen sie alle möglichen Bindungsspaltungen. Diese Bestimmungen hatten eine hohe Laufzeit zur Folge, wodurch ein Einsatz für Datensätze mit vielen Kandidaten nicht möglich war. Im Jahr 2010 veröffentlichten Wolf *et al.* das Programm `MetFrag` [51] mit einem heuristischen kombinatorischen Ansatz [51]. `MetFrag` kann für eine gegebene Masse alle Kandidaten aus einer Datenbank mit dieser Masse bestimmen und die kombinatorische Fragmentierung durchführen. Anschließend werden die Kandidaten entsprechend der erklärten Peaks sortiert. Für die kombinatorische Fragmentierung bestimmt `MetFrag` die Kosten zum Spalten der Bindungen. Die Kostenfunktion enthält unter anderem die Bindungsdissoziationsenergie, um schwache und starke Bindungen zu unterscheiden. Anschließend erklärt `MetFrag` die Peaks mit den Fragment-Ionen, die die geringsten Kosten haben. Zur Reduzierung der Laufzeit bestimmt `MetFrag` nur Fragment-Ionen bis zur Fragmentierungstiefe von zwei. `MetFrag` wurde in den Folgejahren unter anderem durch eine Optimierung der Kandidatenauswahl aus der Datenbank verbessert [30, 52]. Ein Nachteil des Ansatzes der kombinatorischen Fragmentierung ist, dass Umlagerungen von Atomen oder Atomgruppen nicht oder nur teilweise beachtet werden. Es kann daher nicht sicher gestellt sein, dass die Annotation durch chemisch sinnvolle und stabile Fragment-Ionen erfolgt. Die chemische Korrektheit oder Plausibilität bezieht sich beispielsweise darauf, dass ein Atom eine zulässige Anzahl an Bindungen besitzt oder, dass die Formalladung eines Atoms mit der Anzahl seiner Elektronen übereinstimmt. Weiterhin besteht eine große Herausforderung bei der Kostenbestimmung und beim Ranking der Kandidaten.

Fragmentierungsbäume Eine weitere Möglichkeit den Fragmentierungsprozess darzustellen, ist über Fragmentierungsbäume. Hier ist der Ausgangspunkt das Masse-zu-Ladungs-Verhältnis (m/z) des Molekül-Ions. Basierend auf dessen m/z-Verhältnis können Summenformeln für das Ausgangsmolekül bestimmt werden, welche in der Wurzeln mehrerer Fragmentierungsbäume gespeichert werdem [28]. Alle weiteren Knoten im entsprechenden Fragmentierungsbaum enthalten Summenformeln, die vom Wurzelknoten bzw. den Elternknoten abgeleitet werden. Die aus den Sum-

menformeln errechneten Massen der Knoten entsprechen dann den m/z-Werten der Peaks. Die Kanten des Baums zeigen den Neutralverlust zwischen zwei Knoten. Nach der Generierung der Fragmentierungsbäume für ein gegebenes Massenspektrum können diese mit den generierten Bäumen aus einer Datenbank verglichen werden. Ist die analysierte Substanz nicht in der Datenbank enthalten, kann durch den Vergleich trotzdem auf die Substanzklasse geschlossen werden [53]. Ist die analysierte Substanz bekannt, kann deren Summenformel direkt als Eingabe verwendet werden, sodass nur ein Fragmentierungsbaum generiert wird. Diesen Ansatz nutzt beispielsweise das Programm `SIRIUS` [54, 55]. Ein Nachteil der Fragmentierungsbäume ist, dass eine Vielzahl von Fragmentierungsbäumen ein Spektrum beschreiben können. Hintergrund ist, dass auf Grundlage der Masse mehrere Summenformeln als Ausgangsmolekül zur Verfügung stehen. Dieses Problem tritt nach Hufsky *et al.* besonders stark auf, wenn eine Molekülmasse von über 500 Da vorliegt oder das Molekül neben den Atomen Kohlenstoff, Wasserstoff, Stickstoff, Sauerstoff, Phosphor andere Atome enthält [3].

Maschine-Learning basierender Ansatz Die Bestimmung der Fragment-Ionen kann neben den bereits erläuterten Ansätzen ebenfalls über Maschine-Learning basierende Ansätze erfolgen. Einen solchen Ansatz verwendet das Programm `FingerID` von Heinonen *et al.* [56]. Im ersten Schritt der Entwicklung steht eine Trainingsphase. In dieser wird das Massenspektrum in einen Eigenschafts-Vektor überführt. Zusätzlich wird für das Molekül, das das Spektrum repräsentiert, ein Fingerprint-Vektor erstellt. Ziel ist es, durch das Training eines Klassifikators mit dem Eigenschaftsvektor auf den Fingerprint-Vektor schließen zu können. Im zweiten Schritt, der Vorhersage-Phase, kommt der trainierte Klassifikator zur Anwendung. Auf Basis des Massenspektrums wird der Eigenschafts-Vektor erstellt und der trainierte Klassifikator sagt den Fingerprint-Vektor vorher. Dieser Fingerprint kann dann beispielsweise in einer Datenbank gesucht werden. Bei einer erfolgreichen Suche ist die gemessene Substanz ermittelt. Ist die Substanz noch nicht vermerkt, kann zumindest auf die Eigenschaften des Moleküls geschlossen werden.

Eine Kombination aus den Fragmentierungsbäumen und der Vorhersage der Fingerprints bildet das Programm `CSI:FingerID` [57]. Statt direkt aus dem Massenspektrum den Eigenschaftsvektor vorherzusagen, wird zunächst ein Fragmentierungsbaum bestimmt. Nach der Anwendung von Lernverfahren ist die Bestimmung eines probabilistischen Fingerprints der abschließende Schritt. `CSI:FingerID` ist in `SIRIUS 4` integriert [54].

Ein weiteres Maschine-Learning-basierendes Programm ist `CFM-ID`. Dieses ist unter

anderem webbasiert verfügbar und enthält drei Anwendungsszenarien [58, 59, 60]. Das sind die Spektrenvorhersage, die Peakzuweisung und die Substanzidentifikation. Ziel des CFM-ID Ansatzes ist es, ein Model zur Beschreibung des Fragmentierungsprozesses zu lernen. Ein stochastisches Markov Modell modelliert die ESI MS/MS Fragmentierung. Um den Übergang von einem Fragment zu einem anderen Fragment zu simulieren, werden Wahrscheinlichkeiten in einem Übergangsmodell definiert. Zur Erzeugung der Fragment-Ionen sind verschiedene Regeln implementiert. Dazu zählen beispielsweise die McLafferty Umlagerung, aber auch die systematische Bindungsspaltung, ähnlich zu MetFrag [61].

Quantenchemische Fragmentierung Der Vorteil der bereits erläuterten Methoden ist die Schnelligkeit. Jedoch werden nicht immer chemisch plausible Annotation der Fragment-Ionen vorhergesagt. Den Gegensatz dazu bilden die quantenchemischbasierten Verfahren zu Fragment-Ionen-Vorhersage. Sie ermöglichen eine Simulation der Fragmentierung und damit eine präzise Kalkulation der Fragment-Ionen. Bereits 1993 stellten Mayer und Gömöry [6] einen ersten Ansatz vor, der die Bindungsspaltung auf Basis von Energiewerten auswählt. Alex *et al.* [7] verwenden einen ähnlichen Ansatz. In diesem werden die Bindungsspaltungen auf Grundlage der Thermodynamik ausgewählt. Auf Grundlage der DFT können geometrisch optimierte Strukturen bestimmt und Änderungen der Bindungslängen mit resultierenden Spaltungen vorhergesagt werden. Einen anderen Ansatz verfolgen Bauer *et al.* [62] und Grimme *et al.* [4], die den Fragmentierungsprozess über Born-Oppenheimer Quanten-Molekül-Dynamik (MD) simulieren. Dafür sind jedoch mehrere Hundert MDs notwendig. Die Folge sind hoher Zeit- und Rechenaufwand. Zu weiteren Programmen, die auf Basis der Quantenchemie die Fragmentierung bestimmen, zählen `QCMS`^{2} von Cautereels *et al.* [63, 64] und `QC-FPT` von Janesko *et al.* [65].

4. Annotationsprogramm ChemFrag

Im vorherigen Kapitel betrachteten wir bereits entwickelte Programme zur *in-silico*-Fragmentierung. Blicken wir dort genauer auf die Programme der kombinatorischen Fragmentierung (`MetFrag`) oder des Maschine-Learning-Ansatzes (`CFM-ID`), zeichnen sich diese durch hohe Schnelligkeit aus. Nachteil dieser Programme ist die oftmals chemisch nicht sinnvolle strukturelle Vorhersage der Fragment-Ionen. Im Gegensatz dazu bieten die quantenchemisch-basierenden Ansätze diese strukturelle Korrektheit. Um diese zu erreichen sind hohe Laufzeiten (mehrere tausend CPU Stunden pro Molekül) notwendig. Damit ist ein Einsatz für die Berechnung mehrerer hundert bis tausend Moleküle, wie beispielsweise im Bereich der Metabolomik benötigt, nicht günstig.

Um die chemisch strukturelle Annotation von ESI-MS/MS-Spektren in kurzer Zeit zu ermöglichen, entstand im Rahmen dieser Arbeit das Programm `ChemFrag`. `ChemFrag` kombiniert einen regelbasierten und einen quantenchemischen Ansatz. Für den quantenchemischen Ansatz verwendet es die semi-empirische Methode PM7 aus `MOPAC`. Die Implementierungen basieren auf Java 12 mit der Bibliothek `CDK` 2.3.

Im Folgenden werden wir der Aufbau vom `ChemFrag` im Detail betrachten. Weiterhin werden wir die Ergebnisse von `ChemFrag` mit existierenden Programmen und experimentellen Auswertungen vergleichen. Die Inhalte des Kapitels basieren auf der Veröffentlichung

> Jördis-Ann Schüler, Steffen Neumann, Matthias Müller-Hannemann, Wolfgang Brandt: *ChemFrag: Chemically meaningful annotation of fragment ion mass spectra*. Journal of Mass Spectrometry **53** (2018), 1104-1115

sowie weiteren Experimenten und Analysen für verschiedene Stoffklassen.

4.1. Methodik von `ChemFrag`

Der Ablauf von `ChemFrag` gliedert sich in vier Bereiche, die wir in Abbildung 8 erkennen. Auf diese vier wichtigen Hauptschritte legen wir in den folgenden Abschnitten unser Augenmerk. Darin inbegriffen werden Erläuterungen zur Berechnung der 3D-Koordinaten und der Energie sein.

4.1.1. Ionisierung und Bestimmung der Protonenaffinität (Schritt a)

Der erste Schritt einer massenspektrometrischen Analyse einer chemischen Substanz ist die Ionisierung des Moleküls. Dafür stehen bei der Elektrospray-Ionisiation die

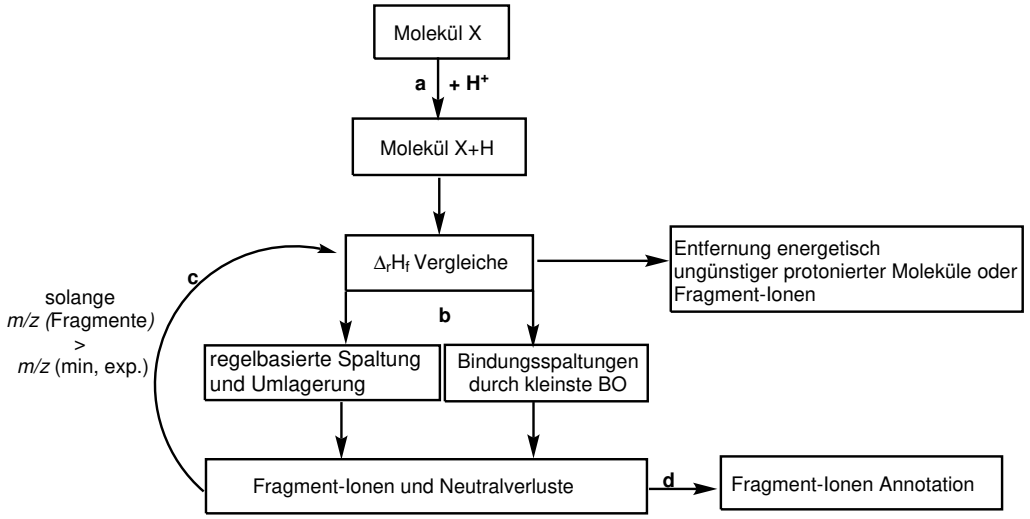


Abbildung 8: Überblick zum Ablauf von `ChemFrag` im positiven Ionisierungs-Modus. Die Eingabe für `ChemFrag` ist eine Struktur im Datenformat, die ein Molekül X repräsentiert und eine Datei mit den m/z Werten des zugehörigen Fragment-Ionen-Spektrums. Die Ausgabe von `ChemFrag` ist die Annotierung der Fragment-Ionen aus dem Massenspektrum. In Schritt a) bestimmt `ChemFrag` alle möglichen Protonierungspositionen und wählt chemisch realistische protonierte Moleküle für den nächsten Fragmentierungsschritt aus. In Schritt b) werden neue Fragment-Ionen durch zwei verschiedene Ansätze generiert, der regelbasierten Fragmentierung und der quantenchemischen Fragmentierung. Für die regelbasierte Fragmentierung wendet `ChemFrag` aus der Literatur bekannte Spaltung- und Umlagerungsregeln auf die Fragment-Ionen an. In der quantenchemischen Fragmentierung spaltet `ChemFrag` alle Bindungen, bei denen die Bindungsordnungen unterhalb eines vorberechneten Schwellwertes liegen. Alle neu erstellten Fragment-Ionen werden mit dem Fragment-Ionen-Spektrum in Schritt d) verglichen, wodurch die Fragment-Ionen annotiert werden. Der Prozess zur Fragment-Generierung endet (Schritt c), wenn kein Fragment-Ion mehr existiert, das eine größere Masse als der kleinste experimentelle m/z-Wert hat oder die vorgegebenen Fragmentierungstiefe erreicht ist (angelehnt an [66]).

Protonierung und die Deprotonierung zur Verfügung. Beide Ionisierungsmöglichkeiten sind in `ChemFrag` verfügbar. Für die Erläuterung der Ionisierung gehen wir beispielhaft nur auf die Protonierung ein. `ChemFrag` simuliert die Ionisierung durch das Hinzufügen eines Protons an Heteroatome, Doppelbindungen oder Dreifachbindungen. Dabei betrachtet `ChemFrag` alle möglichen Protonierungspositionen und generiert diese, wie in Abbildung 9 zu erkennen ist. Für jede Position berechnet `ChemFrag` die zugehörige Protonenaffinität, die später zur qualitativen Beurteilung der Reaktion verwendet wird. Treten bei Molekülen konjugierte π-Systeme auf, erkennt `ChemFrag`

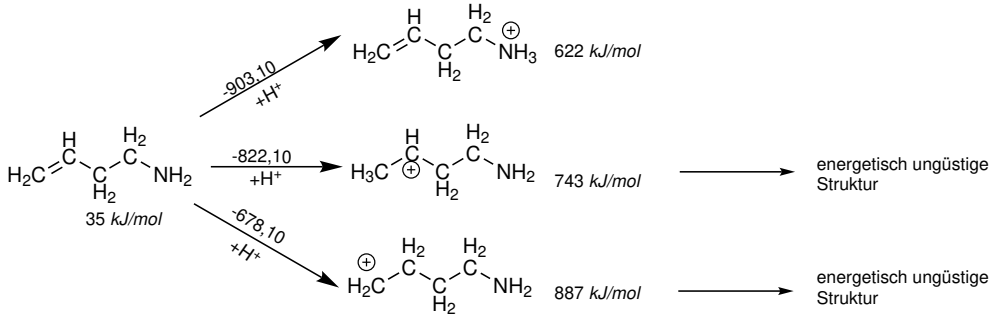


Abbildung 9: Drei mögliche und unterschiedliche Ergebnisse der Protonierung von But-3-en-1-amin. Das erlaubte Intervall der Reaktionsenergie ist hier [-903.10, -853.10]. Dadurch wird nur die oberste Struktur für den nächsten Schritt beibehalten (angelehnt an [66]).

diese und protoniert die Mehrfachbindung. Dazu spaltet es die Mehrfachbindung und erstellt ein neues positiv geladenes Molekül.
Die Protonenaffinität eines Moleküls berechnet sich durch die Formel 1, wobei m' das protonierte Moleküle und m das neutrale Eingabe-Molekül ist. Der Wert $\Delta H_f(H^+)$, die Bildungsenergie eines Protons, beträgt 1530,10 kJ/mol.

$$\Delta_r H_f = \Delta H_f(m') - \Delta H_f(m) - \Delta H_f(H^+) \tag{1}$$

Nach der Berechnung der Reaktionsenergie aller protonierten Varianten, entfernt `ChemFrag` alle energisch nicht sinnvollen Moleküle. Beispielsweise können alle Reaktionen entfernt werden, die in ihrer Reaktionsenergie mehr als 50 kJ/mol über der minimalen Reaktionsenergie im Protonierungsschritt liegen. In Abbildung 9 sehen wir ein Beispiel dafür.
Nach der Protonierung erfolgt in den nächsten Iterationen die Generierung der Fragment-Ionen. Die Methodik dazu stellt das folgende Unterkapitel vor.

4.1.2. Bindungsspaltungen und Umlagerungen (Schritt b)

`ChemFrag` bildet iterativ neue Fragment-Ionen durch das Spalten von Bindungen mit kleiner Bindungsordnung (BO) sowie durch die Anwendung von Fragmentierungsregeln und durch das Umlagern chemischer Gruppen. Dabei generiert `ChemFrag` nicht nur Fragment-Ionen, die die Peaks erläutern sondern auch Zwischenprodukte des Fragmentierungsprozesses (Abbildung 16). Nachfolgend führen wir uns die beiden Fragmentierungsansätze im Detail vor Augen.

Quantenchemisch basierte Fragmentierung Wie in Abbildung 8 dargestellt, erstellt `ChemFrag` neue Fragment-Ionen durch Bindungsspaltungen. Dafür nutzt es eine quantenchemische Methode, um schwache Bindungen zu detektieren, die spal-

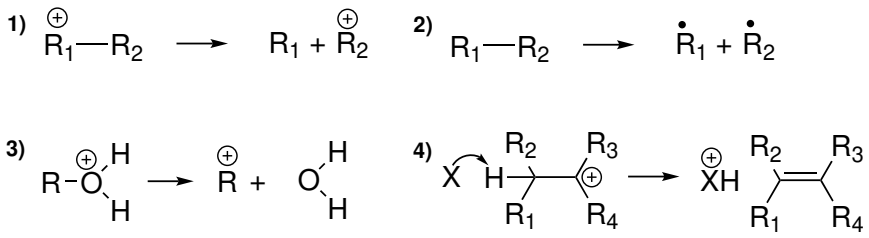


Abbildung 10: Darstellung der heterolytischen Spaltung (1), der homolytischen Spaltung (2), der Abspaltung von Wasser (3) sowie des Proton Shifts (4), die in `ChemFrag` implementiert sind.

tungsfähig sind. Im Gegensatz zu existierenden Programmen, wie `FID` [50] oder `MetFrag` [30, 51], die die Standard-Bindungsenergie oder die feste Bindungsdissoziationsenergie verwenden, nutzt `ChemFrag` Bindungsordnungen zur Bestimmung der Bindungsstärke. Dass dieser Ansatz gerechtfertigt ist, quantifiziert das Kapitel 4.4.1, indem wir in einem Experiment die Bindungslängen eines Moleküls aus Alex *et al.* [7] mit den Bindungsordnungen vergleichen werden. Basierend auf diesem Experiment können wir schlussfolgern, dass die Berechnung der Bindungsordnung tatsächlich (zumindest qualitativ) eine schnelle Abschätzung der Bildungsstärke zulässt. Das hat zur Konsequenz, dass Bindungen durch ihren tatsächlichen chemischen Kontext beschrieben werden und nicht durch einen festen Wert. Eine kleine Bindungsordnung charakterisiert eine schwache Bindung [67].

Um die Bindungsordnung jeder Bindung sowie die Bildungsenthaltpie (ΔH_f) zu berechnen, verwendet `ChemFrag` die semi-empirische Methode PM7 aus der Software `MOPAC` (MOPAC2016). `MOPAC` benötigt als Eingabe die 3D-Koordinaten jedes Atoms. Um diese zu generieren, verwendet `ChemFrag` das Kraftfeld `MMFF94` aus `RDKit`. Treten bei einem Moleküle mehrere Konformere auf, erstellt `ChemFrag` nur eines davon. Die Ausgabe von `MOPAC` repräsentiert die Bindungsordnungen als nxn-Matrix, wobei n die Atomanzahl ist. Jeder Eintrag der Matrix beschreibt die Bindungsordnung eines Atompaares. Hierbei werden auch Bindungsordnungen von Atompaaren angegeben, die im tatsächlichen Molekül nicht durch eine Bindung verbunden sind. Um diese und sehr stabile Bindungen zu entfernen, wird der Wert ihrer Bindung mit dem vordefinierten Schwellwert $\theta_{\text{bond order}}$ verglichen. Nur Bindungen deren Bindungsordnung kleiner als der Schwellwert sind, basierend auf dem Wert des Parameters $\theta_{\text{bond order}}$, vermerkt `ChemFrag` in einer Liste für den nächsten Fragmentierungsschritt. Diese schwachen Bindungen können durch heterolytische (Abbildung 10(1)) oder homolytische Spaltung (Abbildung 10 (2)) gespalten werden. Im Fall einer heterolytischen Spaltung berechnet `ChemFrag` jeweils die Reaktionsenergie für die beiden möglichen positiv geladenen Fragment-Ionen. Anschließend prüft `ChemFrag` wieder die Reakti-

onsenergien, ob sie im zulässigen Energie-Intervall liegen und damit für den nächsten Fragmentierungsschritt selektiert werden können. Durch diesen Ansatz generiert `ChemFrag` erste neue Fragment-Ionen für den nächsten Fragmentierungsschritt, solange bis die Abbruchbedingungen erreicht sind.

Regelbasierte Fragmentierung Die Simulation von strukturellen Umlagerungen, wie beispielsweise Protonen-Shifts, ist durch die Verwendung quantenchemischer Methoden zeitaufwändig. Um diesen Zeitaufwand zu reduzieren, nutzt `ChemFrag` einen regelbasierten Ansatz für die Generierung solcher zusätzlicher Fragment-Ionen. Ein weiterer Vorteil des regelbasierten Ansatzes ist, dass der Generierung instabilder Moleküle vorgebeugt werden kann. Die Implementation von `ChemFrag` beinhaltet 31 Spaltungsregeln und 20 für Umlagerungen, die typischerweise bei der Elektrosprayionisation auftreten. Sie gehen unter anderem auf die Auflistung von Weissberg und Dagan [68] zurück. Eine Teilauflistung der wichtigsten Regeln visualisieren die Abbildungen 10, S-1 und S-2. Unter anderem sind in den Regeln Proton-Shifts, Ally-Shifts und Zyklisierungen enthalten. Jede Regel ist durch einen SMARTS definiert, der eine Substruktur repräsentiert, die im Edukt enthalten sein muss, um die Regel anwenden zu dürfen. Für die Überprüfung wendet `ChemFrag` die SMARTS Matching-Funktion aus `CDK` an. Diese bestimmt gleichzeitig die gematchten Positionen [23]. Abbildung 11 zeigt als Beispiel das Molekül 1-Propenol mit der protonierten Hydroxylgruppe. Durch die Protonierung ist die Kohlenstoff-Sauerstoff-Bindung die schwächste Bindung und lässt das Molekül instabil werden. Um der Energieberechnung eines instabilen Moleküls vorzubeugen, wendet `ChemFrag` die Regel zur Wasser-Abspaltung (Vergleich Abbildung 10(3)) an. Dadurch entsteht ein stabiles Molekül, für das die Bildungsenthalpie und die Bindungsordnungen bestimmt werden können. Basierend auf diesem zweiten Ansatz erhöht sich die Anzahl an potentiellen Fragment-Ionen für den nächsten Fragmentierungsschritt. Es ist an dieser Stelle noch ein Mal besonders darauf hinzuweisen, dass die Regeln zur Umlagerung auch auf die Fragment-Ionen aus dem quantenchemisch-basierenden Ansatz angewendet werden.

4.1.3. Berechnung der 3D-Koordinaten und der Energie (Schritt c)

Um die Stabilität der Fragment-Ionen sowie die Reaktion der Fragmentierung zu bestimmen, ermittelt `ChemFrag` für neue Fragment-Ionen und Neutralverluste die Bindungsenthalpie. Diese Bindungsenthalpien und Bindungsordnungen sind, wie im vorhergehenden Kapitel bereits ausführlich beschrieben, Voraussetzung für die weitere Fragmentierung. Um eine korrekte Berechnung durchführen, müssen im ersten Schritt 3D-Koordinaten für die Fragment-Ionen erzeugt und optimiert werden. Daher betrachtet dieser Abschnitt die Erzeugung der Koordinaten und die Berechnung der Reaktionsenergien.

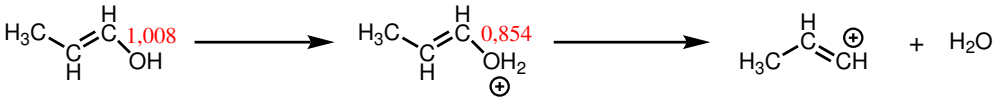


Abbildung 11: Visualisierung des Protonierungs-Prozesses und der Wasserabspaltung bei 1-Propenol. Die Bindungsordnung zwischen dem Kohlenstoff und dem Sauerstoffatom ist rot markiert. In diesem Beispiel ist der SMILES für 1-Propenol `[OH2+]C=CC`. `ChemFrag` prüft, ob die Regel zur Wasserabspaltung (Abbildung 10(3)), durch die SMARTS Matching-Funktion mit dem SMARTS `[OH2+]`, anwendbar ist. Nach dem erfolgreichen Matching, wird die Regel zur Bildung des neuen Fragments angewendet und simuliert die Reaktion, die hier in der Abbildung zu sehen ist (angelehnt an [66]).

Berechnung der 3D-Koordinaten Um die Bildungsenergie und die Bindungsordnungen eines Fragment-Ions zu berechnen, benötigt `MOPAC` die 3D-Koordinaten der Molekülstruktur. Das erfolgt in einem zweistufigen Prozess. Zunächst generiert `ChemFrag` mittels `CDK` 2D-Koordinaten und speichert diese in einer MOL-Datei. Für die 3D-Koordinaten-Generierung ist in `ChemFrag` der Aufruf des Programms `RDKit` integriert. `RDKit` liest die Datei mit den 2D-Koordinaten ein.
Im ersten Schritt berechnet `RDKit` dann zufällige 3D-Koordinaten mit der Funktion `AllChem.EmbedMolecule` [69]. Anschließend optimiert `RDKit` die Koordinaten mit dem Kraftfeld `MMFF94` der Funktion `AllChem.MMFFOptimizeMolecule`. Die optimierten Koordinaten liest `ChemFrag` im nächsten Schritt ein und nutzt sie für die Erzeugung der Eingabe-Datei von `MOPAC`, um die abschließende Energieoptimierung durchzuführen.
Die Bestimmung der 3D-Koordinaten erfolgt dabei nur für ein Konformer, aus folgenden Gründen:

- Die Energieunterschiede zwischen den Konformeren sind gering im Vergleich zu generellen Unterschieden in den Reaktionsenergien aller Fragment-Ionen.
- Die Konformationssuche ist zeitaufwändig.
- Die Bindungsordnungen hängen hauptsächlich von den Bindungseigenschaften (z.B. induktive Effekte) ab und sind nur sehr minimal von sphärischen Interaktionen beeinflusst.

Diese Gründen lassen die Annahme zu, dass die Konformationssuche zum Vorteil der Laufzeit vernachlässigt werden kann.
Nachdem die 3D Koordinaten erzeugt sind, generiert `ChemFrag` die Eingabe-Datei für `MOPAC`. Anschließend ruft es `ChemFrag` damit auf.

Energieberechnung `MOPAC` benötigt für seinen Aufruf mehrere Schlüsselworte (Keywords), um die korrekte Ausführung der Optimierung zu ermöglichen. Dabei unterscheiden wir zwischen Rechnungen mit Radikalen, wo das Schlüsselwort `UHF` übergeben wird und Rechnungen ohne Radikale. Beide Varianten enthalten das Schlüsselwort `BFGS` zur Auswahl des Optimierungsverfahrens. Die Optimierung endet, wenn 99999 Durchläufe beendet sind oder der Algorithmus selbst eine optimale Konformation gefunden hat. Weiterhin ist das Schlüsselwort `BONDS` angegeben, welches später die Bindungsordnungen angibt und damit die Voraussetzung für die homolytische und heterolytische Spaltung bildet. Während der Strukturoptimierung mit `MOPAC` kann es bei instabilen Molekülen zu deren Zerfall kommen. Das tritt besonders dann auf, wenn ein einfach-gebundenes Sauerstoffatom protoniert wurde. Um diesem Zerfall in `MOAPC` vorzubeugen, enthält `ChemFrag` dafür spezielle Spaltungs- und Umlagerungsregeln (Abbildung 10 und S-2). Wenn trotz dieser Regeln auffällige Energiewerte erkannt werden, wird die Struktur überprüft. Im Anschluss werden Regeln abgeleitet und implementiert, die die Generierung des instabilen Fragment-Ions verhindern.
Unter der Annahme einer erfolgreichen Energieoptimierung liest `ChemFrag` die Bildungsenergie aus der Ausgabe-Datei von `MOPAC` aus. Anschließend bestimmt es die Reaktionsenergien aller Fragment-Ionen, indem es die Differenz der Bindungsenthalpien zwischen den Produkten und dem Edukt berechnet. Ist die Reaktionsenthalpie eines Fragment-Ions größer als ein bestimmter Schwellwert übernimmt `ChemFrag` es nicht in die Liste der Kandidaten des nächsten Fragmentierungsschritts. Dieser Prozess wird solange wiederholt bis nur noch Fragment-Ionen mit Massen kleiner als der kleinste Peak existieren oder die Fragmentierungstiefe erreicht ist.

4.1.4. Annotation der Fragment-Ionen (Schritt d)

Nachdem `ChemFrag` neue Fragment-Ionen generiert hat, überprüft es welche Peaks des Spektrums durch die neuen Fragment-Ionen annotiert werden können. Dabei sind Instrument abhängige Massenabweichungen zwischen der berechneten Masse des Fragment-Ions und des Massen-Peak erlaubt. Es gilt, dass Moleküle mit einer Masse von

$$m/z_i \pm mzabs + mzppm \cdot 10^{-6} \cdot m/z_i$$

dem i-ten Massen-Peak zugewiesen werden könnten. Dabei ist m/z_i das Masse-zu-Ladungs-Verhältnis des i-ten Peaks, und $mzabs$ und $mzppm$ beschreiben die erlaubte absolute und relative Abweichung, die vom Nutzer definiert werden kann.

Bewertungsfunktion Um die Annotation des Massenspektrums zu bewerten, berechnet `ChemFrag` am Ende einen Wert (Score). Dafür existieren in `ChemFrag` zwei verschiedene Scores, deren Details wir uns nachfolgend ansehen. Eine Quantifizierung

der Scores führt das Kapitel 4.4.3 durch.

Ein einfacher Score ist das absolute Zählen, wieviele Fragment-Ionen erklärt werden können. In diesem Wert ist kein chemisches Hintergrundwissen enthalten. Die Berechnung erfolgt mit der Gleichung 2.

$$s = \sum_{f \in F} f \tag{2}$$

Die Menge F enthält hier alle Fragment-Ionen, die einem Peak zu geordnet werden können. Der errechnete Wert s kann anschließend mit dem maximalen Wert verglichen werden, der die Anzahl aller gemessenen Peaks abbildet.

Problematisch bei dieser absoluten Bewertungsmethode ist, dass Peaks, die als Rauschen zählen oder eine minimale Intensität besitzen, genauso gewichtet werden, wie Peaks mit einer starken Intensität, wie der Basispeak. Aus diesem Grund ist es das Ziel, die Intensität genauer in die Bewertung mit einfließen zu lassen. Dafür besitzt `ChemFrag` eine zweite Bewertungsmethode, die gewichtete. Diese bezieht die Intensität der Fragment-Ionen ($f \in F$) mit ein und ist in Gleichung 3 ersichtlich.

$$s = \sum_{f \in F} (intensitaet_f) \tag{3}$$

In der Gleichung wird jedes Fragment-Ion durch die Intensität bewertet. Dadurch ist es möglich, dass Fragment-Ionen mit einer hohen Intensität stärker gewichtet werden als Fragment-Ionen mit schwacher Intensität. Sie haben damit eine höhere Wichtigkeit und Bedeutung in der Aufklärung des Massenspektrums. Auch hier kann der Wert s mit dem maximalen Wert verglichen werden. Um diesen zu berechnen geht man davon aus, dass alle Peaks erklärt sind.

4.1.5. Zusammenfassung der Methodik von `ChemFrag`

Die Methodik von `ChenFrag` zeichnet sich durch die Kombination einer regelbasierten und einer auf Quantenchemie basierten Fragmentierung aus. Sie ermöglicht die Ionisierung durch Protonierung oder Deprotonierung. Die Stabilität von Fragment-Ionen wird durch ihre Bildungsenthalpie und für die Bindungen durch die Bindungordnungen aus der Ausgabe der semi-empirischen Methode PM7 in `MOPAC` quantifiziert. `ChemFrag` enthält 31 Spaltungsregeln und 20 Umlagerungsregeln. Die Spaltung von Bindungen im quantenchemischen Ansatz erfolgt durch die Simulation von heterolytischen und homolytischen Spaltungen. `ChemFrag` berechnet solange neue Fragment-Ionen, bis eine maximale Fragmentierungstiefe erreicht ist oder alle Fragment-Ionen

eine kleinere Masse als der niedrigste Peak aufweisen.

4.2. Umsetzung des regelbasierten Ansatzes

Aus dem Kapitel 4.1.2 ist uns bereits bekannt, dass die regelbasierte Fragmentierung ein Kernelement von `ChemFrag` ist. Die 31 implementierten Spaltungs- und 20 Umlagerungsregeln geben `ChemFrag` die Möglichkeit komplexe chemische Reaktionen zu simulieren und dadurch chemisch plausible Fragment-Ionen zu erzeugen. Wie diese Regeln implementiert sind, soll in diesem Kapitel kurz beleuchtet werden.

`ChemFrag` verwendet für die Implementierung der Regeln Java 12 und `CDK` 2.3. In seiner Architektur enthält `ChemFrag` zwei Pakete, in denen die Regeln implementiert sind. Das Paket `rearrangement` für die Umlagerungsregeln und `cleavage` für die Spaltungsregeln. In beiden Paketen sind die jeweiligen Klassen zur Umlagerung und Spaltung aufgeführt. Jede einzelne Klasse entspricht einer Regel. Der grundsätzliche Aufbau der Klassen ist für alle Klassen gleich. Als Eingabe erhalten die Klassen eine Kopie des Fragment-Ions sowie die Positionen der Atome des Fragment-Ions, auf welche die Regel angewendet werden soll. Die Ausgabe ist dann das neu fragmentierte oder umgelagerte Fragment-Ion. Um zu entscheiden, ob eine Regel auf einem Fragment-Ion ausgeführt werden kann, ist die Struktur des Fragment-Ions entscheidend. Daher werden über SMARTS die notwendigen Substrukturen definiert, die im Fragment-Ion für die jeweiligen Regeln vorhanden sein müssen. Um das Vorhandensein der Substrukturen zu überprüfen, verwendet `ChemFrag` das `SMARTSQueryTool` der `CDK`-Bibliothek. Findet das `SMARTSQueryTool` die durch den SMARTS definierte Teilstruktur, wählt `ChemFrag` die dazugehörige Regel aus und wendet sie auf die ermittelte Teilstruktur an. In allen Regeln werden die chemischen Reaktionen dann beispielsweise durch Änderungen der Ladungspositionen, der Anzahl der Wasserstoffe oder der Anzahl der Radikale sowie durch das Hinzufügen und Entfernen von Bindungen simuliert. Schematisch bildet die Klassenstruktur von `ChemFrag` die Abbildung 12 ab.
Als Beispiel sehen wir uns nun die Umsetzung der Spaltungsregeln *RemoveH2O*, für die Abspaltung von Wasser, genauer an.

Für die Regel ist als SMARTS `[OH2+]` definiert. Findet das `SMARTSQueryTool` diese Teilstruktur übergibt es die Position des positiv geladenen Sauerstoffatoms und an Kopie des Fragment-Ions an die Klasse *RemoveH2O*. Sie führt dann die nachfolgenden Schritte aus:

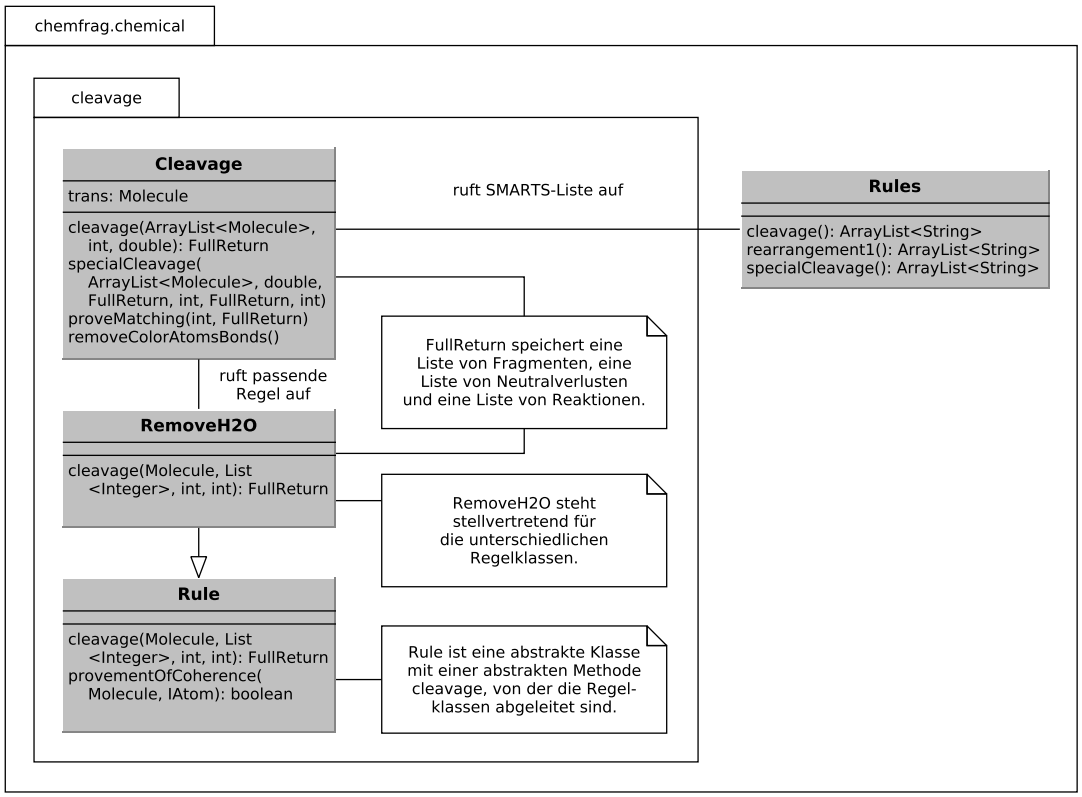


Abbildung 12: Klassendiagramm der direkt für die Spaltung relevanten Pakete und Klassen

1. Bestimme das benachbarte Atom (nachfolgend mit coAt abgekürzt) des Sauerstoffatoms.
2. Entferne die Bindung zwischen coAt und Sauerstoffatom.
3. Setze die Ladung des Sauerstoffatoms auf 0 mittels der Funktion *setFormalCharge()* aus `CDK`.
4. Setze die Ladung des coAt auf +1 mittels der Funktion *setFormalCharge()* aus `CDK`.
5. Erstelle eine Kopie von diesem geänderten Fragment-Ion.
6. Entferne aus dieser Kopie alle Atome außer das Wassermolekül und speichere dieses in der Liste der Neutralverluste.
7. Entferne aus dem geänderten Fragment-Ion das Wassermolekül und speichere das neue Fragment-Ion in der Liste der Fragment-Ion des aktuellen Fragmentierungsschrittes.

Durch diese Abfolge von Anweisungen simuliert `ChemFrag` die Reaktion aus Abbildung 11.

Nach der Generierung des Fragment-Ions und des Neutralverlustes berechnet `ChemFrag` durch `MOPAC` die Bildungsenthalpien und errechnet mit diesen die Reaktionsenergie.

4.3. Laufzeitoptimierung

Nachdem wir im vorherigen Kapitel den Aufbau von `ChemFrag` betrachtet haben, werden wir das Augenmerk auf die Berechnung der Schwellwerte der Reaktionsenergie und der Bindungsordnungen legen. Weiterhin stellen wir verschiedene integrierte Ansätze zur Reduzierung der Laufzeit von `ChemFrag` vor.

Die Laufzeit von ChemFrag hängt stark von der Größe und Anzahl der Moleküle ab. Bei der Molekülgröße spielt besonders die Elektronenanzahl eine entscheidende Rolle. Dadurch variiert die Laufzeit auf einer CPU zwischen wenigen Sekunden für kleine Moleküle und einigen Minuten für größere Moleküle. Daraus zeichnet sich ab, dass besonders die Laufzeit für größere Moleküle höher ist als bei *in-silico* Methoden wie `MetFrag` oder `CFM-ID`, jedoch auch deutlicher schneller als bei Methoden, die DFT-Berechnungen verwenden. Die Laufzeit-Analysen in Kapitel 4.3.3 zeigen, dass die Berechnung der 3D-Koordinaten mit `RDKit` und die anschließende Ausführung von `MOPAC` die meiste Laufzeit beanspruchen.
Daraus können wir schlussfolgern, dass ein wichtiges Ziel von `ChemFrag` sein muss, die Anzahl an quantenchemischen Berechnungen zu reduzieren. Dazu existieren zwei Möglichkeiten:

1. Vermeidung redundanter Berechnungen von Fragment-Ionen und Neutralverlusten
2. Reduzierung der Fragment-Ionen für den nächsten Fragmentierungsschritt. Dafür notwendig ist die Einschätzung und Auswahl chemisch sinnvoller Strukturen sowie die Bestimmung instabiler Bindungen auf Basis geeigneter Reaktions- und Bindungsordnungsintervalle.

Wie diese zwei Ansätze in `ChemFrag` integriert sind, werden wir nachfolgend nachvollziehen.

4.3.1. Schwellwerte

In diesem ersten Abschnitt betrachten wir die Reduzierung der Laufzeit durch die geeignete Auswahl von Fragment-Ionen. Um chemisch plausible Moleküle für die Fragmentierung zu simulieren, enthält `ChemFrag` zwei verschiedene Schwellwerte:

1. für die Auswahl chemisch plausibler Strukturen für den nächsten Fragmentierungsschritt
2. und für die Auswahl instabiler Bindungen für die quantenchemisch-basierende Fragmentierung.

Die Berechnung der beiden Schwellwerte sehen wir uns nun an.

1. Da ein Molekül mit geringer Energie als stabil und chemisch sinnvoll gilt, verwenden wir die Formel 4 zur Berechnung.

$$\theta_{\Delta_r H_f}(l) = \min_{m' \in l} \left(\Delta_f H_f(m')\right) + \epsilon \tag{4}$$

   Die Liste l enthält hierbei die Fragment-Ionen des aktuellen Fragmentierungsschrittes und ϵ ist ein vom Anwender einzustellender Parameter. Daraus leitet sich ein Intervall der Reaktionsenergie ab. Dieses ist begrenzt aus der unteren Schranken mit dem Wert von $\min_{m' \in l} (\Delta_f H_f(m'))$, der geringsten Reaktionsenergie der betrachteten Fragment-Ionen im aktuellen Fragmentierungsschritt, und aus der oberen Schranke mit $\theta_{\Delta_f H_f}(l)$. Alle noch nicht fragmentierten Strukturen aus diesem Intervall selektiert `ChemFrag` für den nächsten Fragmentierungsschritt. Ein hoher Wert von ϵ führt zu einem großen Intervallbereich und damit zu einer hohen Anzahl an generierten Molekülen, wohingegen ein niedriger Wert eine Reduzierung der Molekülanzahl bewirkt.

2. Eine ähnliche Idee nutzen wir bei der Auswahl an zu spaltenden Bindungen. Für jedes Produkt m' einer Reaktion wird der zugehörige Schwellwert $\theta_{\text{bond order}}(m')$ nach Formel 5 berechnet.

$$\theta_{\text{bond order}}(m') = \min_{b \in m'}(bond\ order(b)) + \kappa \tag{5}$$

   Hier steht b für die Bindungen aus dem Molekül m' und κ für einen vom Anwender zu wählenden Parameter. Auch hier entsteht ein Intervall der Bindungsordnungen für jedes Fragment-Ion. Dieses ist nach unten begrenzt durch den Wert von $\min_{b \in m'}(bond\ order(b))$, der kleinsten Bindungsordnung im Fragment-Ion, und nach oben begrenzt durch den Wert von $\theta_{\text{bond order}}(m')$. Nur Bindungen dieses Intervalls können in der quantenchemischen Fragmentierung gespalten werden.

Die Werte ϵ und κ können wie bereits erwähnt vom Anwender justiert werden und können die Kollisionsenergie aus dem Massenspektrometer widerspiegeln. Dadurch ist es möglich, den Fragmentierungsprozess abhängig von der Energie zu simulieren.

4.3.2. Optimierung der quantenchemischen Berechnungen

Die Auswahl von Fragment-Ionen, die eine weitere sinnvolle Fragmentierung widerspiegeln sollen, ist ein wichtiger Aspekt. Weitere Aspekte sind die Vermeidung redundanter Berechnungen sowie die Parallelisierung der Berechnungen.

1. Wie bereits erläutert, ist die Laufzeit von `MOAPC` sehr hoch und das Ziel ist, dessen Aufrufe zu minimieren. Oftmals werden Neutralverluste und Fragment-Ionen mehrfach über verschiedene Fragmentierungswege in `ChemFrag` generiert. Um die redundanten Berechnungen zu vermeiden, speichert `ChemFrag` die Neutralverluste und die Fragment-Ionen in Hashtabellen (Hash-Maps). Bevor `RDKit` und `MOPAC` aufgerufen werden, prüft `ChemFrag` durch einen Äquivalenztest, ob die Bindungsenthalpien und Bindungsordnungen eines Fragment-Ions oder Neutralverlustes bereits berechnet wurden. Der Molecule Equivalence Tester (`MET`), welcher im Rahmen dieser Arbeit entwickelt wurde und im Kapitel 5 beschrieben wird, führt diese Äquivalenztests durch. Ist ein Molekül bereits berechnet worden, wird die Energie aus dem in der Hash-Map gespeicherten Molekül übernommen. Gleichzeitig können durch die Isomorphie-Funktion aus `MET` die Bindungsordnungen des gespeicherten Fragment-Ions auf das zu berechnende Fragment-Ion abgebildet werden. Ist der Neutralverlust oder das Fragment-Ion noch nicht in der Hash-Map enthalten, wird es dort nach der Berechnung eingefügt.

2. Weiterhin ist es möglich, dass `ChemFrag` das selbe Fragment-Ion über verschiedene Fragmentierungswege erreicht. Um redundante Fragmentierungen zu vermeiden, verwendet `ChemFrag` ein Hash-Set. In diesem werden die Fragment-Ionen gespeichert, die bereits weiterfragmentiert wurden. Das Hash-Set basiert wieder auf `MET`, sodass ein Äquivalenztest das Vorhandensein eines Fragments effizient prüfen kann. Ein Fragment-Ion wird somit nur für den nächsten Fragmentierungsschritt ausgewählt, wenn es noch nicht in dem Hash-Set enthalten ist.

3. Zusätzlich zu den Optimierungen aus Punkt eins und zwei wird die Laufzeit durch die Parallelisierung der Fragmentgenerierung (Schritt b) aus Abbildung 8 reduziert. Für jedes Molekül führt `ChemFrag` zuerst auf die Quantenchemie basierende Fragmentierung aus. Anschließend prüft `ChemFrag` parallel die Anwendung der implementierten Regeln. Die Generierung der 3D-Koordinaten

mit RDKit und die Energieberechnung mit MOPAC ist für alle Fragment-Ionen und Neutralverluste in ChemFrag parallelisiert.

4.4. Ergebnisse zur chemischen Plausibilität von ChemFrag

Wie eben in der Laufzeitoptimierung gesehen, sind die Ergebnisse der quantenchemischen Berechnung von großer Bedeutung in ChemFrag. Deshalb werden wir in diesem Kapitel den quantenchemischen Ansatz evaluieren. Dafür werden wir zeigen, dass die gewählte semi-empirische Methode ausreichend ist, um die chemischen Eigenschaften eines Moleküls zu beschreiben. Um dies zu quantifizieren, werden wir die von MOPAC berechneten Protonenaffinitäten mit experimentell ermittelten Protonenaffinitäten vergleichen. Zusätzlich werden wir demonstrieren, dass die absoluten Werte der Bindungsordnungen für die Beschreibung der Bindungsstärke verwendet werden können. Der zweite Teil dieses Kapitel enthält die Vorhersage von Fragmentierungswegen für zwei exemplarische Stoffgruppen durch ChemFrag. Dabei vergleichen wir die Ergebnisse von ChemFrag mit den in der Literatur angegebenen Fragment-Ionen und denen durch MetFrag und CFM-ID generierten Fragment-Ionen. Im abschließenden dritten Teil werden die verschiedenen Optimierungsstrategien auf deren Wirksamkeit untersucht. Die sich daraus ableitende Umsetzung von ChemFrag werden wir auf mehrere Naturstoffe anwenden, um für diese erstmalig Fragmentierungswege vorherzusagen.

4.4.1. Validierung der Protonenaffinität und Bindungsordnungen mit MOPAC

Im ersten Schritt beginnen wir mit der Untersuchung des quantenchemischen Ansatzes. Um den Einsatz in ChemFrag zu bewerten, validieren wir die Berechnung der Protonenaffinitäten. Zusätzlich evaluieren wir, ob die Bindungsordnung die Stärke einer Bindung ausreichend beschreibt.

Experiment 4.1. Da ChemFrag die chemisch sinnvollen protonierten Moleküle selektieren muss, ist die Bestimmung der Protonierungspositionen der erste kritische Schritt der Fragment-Ionen-Annotation. ChemFrag berechnet die Energie aller protonierten Moleküle mit der semi-empirischen Methode PM7 aus MOPAC. Das Ziel des ersten Experiments ist daher die Validierung der Protonenaffinitäten. Dafür müssen wir die Verwendung der Methode PM7 rechtfertigen können. Im Experiment vergleichen wir 172 durch MOPAC (mittels PM7) berechnete Protonenaffinitäten mit den von Hunter *et al.* experimentell bestimmten Affinitäten aus der Gasphasen-Protonierung [70]. Bekannt ist, dass die berechneten Bildungsenergien aus MOPAC typischerweise um etwa 16 kJ/mol von experimentellen Werten abweichen. Diese Aussage unterliegt der Annahme, dass die Moleküle Wasserstoff, Kohlenstoff, Sauerstoff, Stickstoff oder Flur

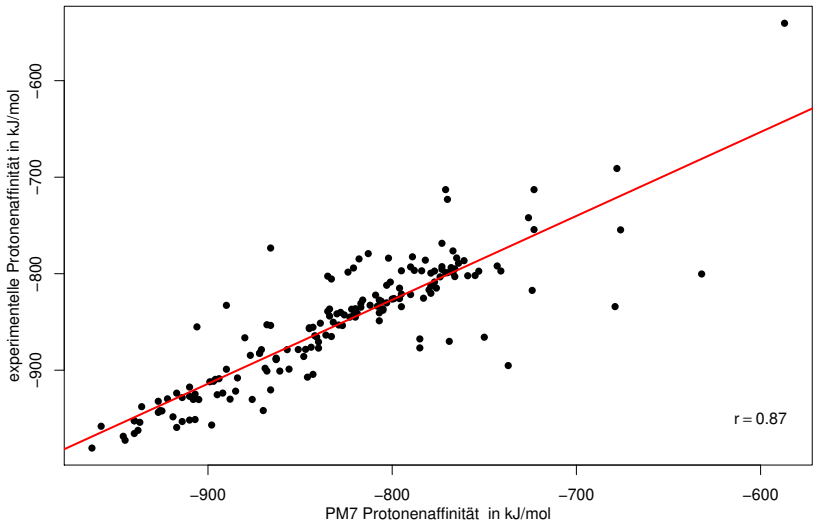


Abbildung 13: Die Abbildung vergleicht die berechneten und die experimentell bestimmten Protonenaffinitäten von 172 Molekülen, die in Hunter und Lias [70] dargestellt sind. Der Pearson Korrelationskoeffizient ist $r = 0.87$. Der Mittelwert und die mittlere absolute Abweichung sind 30 kJ/mol und 25 kJ/mol jeweils mit einer Standardabweichung von 26 kJ/mol (angelehnt an [66]).

als Elemente enthalten [27]. Abbildung 13 zeigt den Vergleich in graphischer Form.

Aus der Abbildung 13 erkennen wir, dass die verglichenen Werte nahe der rot eingezeichneten Geraden liegen. Es sind nur wenige große Abweichungen von der Geraden erkennbar. Zusätzlich erhalten wir einen Pearson Korrelationskoeffizienten von $r = 0.87$. Anhand der graphischen Auswertung und des Korrelationskoeffizienten können wir darauf schließen, dass die berechnete Protonenaffinität auf Basis von `MOPAC` und die experimentelle Protonenaffinität relativ stark korrelieren. Das lässt die Schlussfolgerung zu, dass `MOPAC` die Berechnung der Protonenaffinität und der Reaktionsenergien geeignet abbildet, um Rückschlüsse auf die Stabilität der Moleküle zu ziehen.
Betrachten wir zusätzlich die Protonenaffinitäten aus der Tabelle S-1 im Anhang im Detail erkennen wir, dass die berechneten Protonenaffinitäten die experimentellen Protonenaffinitäten in jedem Fall übersteigen. Da jedoch `ChemFrag` die Protonenaffinitäten und die Reaktionsenergien nur als qualitative Bewertung verwendet, ist dieser Fehler für die Verwendung in einem akzeptablen Rahmen.

Experiment 4.2. `ChemFrag` verwendet neben der Bildungsenthalpie aus der Ausgabe von `MOPAC` auch die Ausgabe der Bindungsordnungen, um die Stärke der Bindungen zu bestimmen. Bereits Mayer und Gömer [6, 71] diskutierten die Anwen-

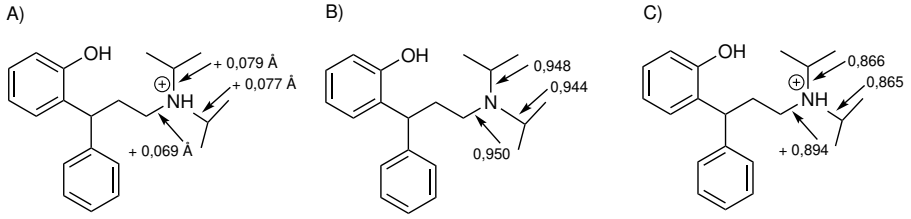


Abbildung 14: Vergleich der Unterschiede in den Bindungsordnungen und Bindungslängen zwischen dem neutralen Molekül 2-(3-diisopropyl-Amino-1-Phenyl-Propyl)Phenol und dem zugehörigen am Strickstoff protonierten Molekül. Die Änderung der Bindungslänge nach der Protonierung zeigt sich in A) und wurde durch Alex *et al.* mittel DFT berechnet. Die Bindungsordnungen in B) und C) sind durch die semi-empirische Methode PM7 aus `MOPAC` ermittelt (angelehnt an [66]).

dung der Bindungsordnungen für die Bewertung der Bindungsstärke durch Wiberg-Indizes [72]. Ebenfalls zeigte Improta [73] die Verwendbarkeit quantenchemisch berechneter Werte. Diese basierten auf DFT-Berechnungen, der „natural bond orbital theory“ sowie der „natural population analysis“. Um die Verwendbarkeit der Bindungsordnungen weiter zu unterstützen, untersuchen wir die Bindungsordnungen- und Längen, die sich während der Protonierung verändern, an einem Beispiel. Für den Vergleich nutzen wir die Werte der DFT-Berechnung von Alex *et al.* [7] (Abbildung 14). In dem Experiment berechnen wir die Änderung der Bindungsordnungen und der Bindungslängen zwischen dem neutralen und dem positiv geladenen Molekül. Dazu wählen wir das Molekül 2-(3-diisopropyl-Amino-1-Phenyl-Propyl)Phenol aus Alex *et al.* (Abbildung 14) und das zugehörige Ion mit der Stickstoffprotonierung als Vergleich. Alex *et al.* verwendete dieses Molekül, um den Unterschied der Bindungslängen durch die Protonierung mittels DFT-Berechnungen zu bestimmen. Die Berechnung der Bindungsordnungen führen wir wieder mit PM7 aus `MOPAC` durch.

Anhand der Abbildung 14 A) erkennen wir, dass die drei Bindungen des Stickstoffatom sich im protonierten Molekül um 0,069 bis 0,079 im Vergleich zum neutralen Molekül verlängern. Diese Verlängerungen führen zu einer geringen Stabilität der Bindungen, wodurch sie leichter homolytisch oder heterolytisch gespalten werden können. Beim Vergleich der Bindungsordnungen in B) und C) aus Abbildung 14 beobachten wir, dass sich die Bindungsordnungen ebenfalls verringern. Die geringeren Bindungsordnungen weisen auch hier wieder auf eine Schwächung der Bindungen hin. Zusätzlich stellen wir fest, dass die Bindungen zwischen dem Stickstoffatom und den zwei tertiären Kohlenstoffatomen schwächer sind als die Bindung des Stickstoffs zum sekundären Kohlenstoffatom. Dieser Vergleich und die Erkenntnisse aus Alex *et al.* sind ein explizites Beispiel dafür, dass die Unterschiede in den Bindungsordnun-

gen die Modifikation der Bindungen genauso gut reflektieren wie die Differenz der Bindungslängen. Für die Bestimmung der schwächsten Bindung verwendet `ChemFrag` die absoluten Werte der Bindungsordnungen eines Moleküls. Für die Implementierung wurde somit davon ausgegangen, dass die schwächste Bindung die kleinste Bindungsordnung hat. Dieses Experiment hat uns daher gezeigt, dass der Einsatz der Bindungsordnungen in `ChemFrag` chemisch gerechtfertigt ist.

4.4.2. Anwendung auf bekannte Doping-Substanzen

Nachdem wir im vorherigen Kapitel den Einsatz der quantenchemischen Methode in `ChemFrag` unterstreichen konnten, ist das Ziel dieses Kapitels die Fragmentierung durch `ChemFrag` an zwei Beispielen zu simulieren. Im Mittelpunkt steht dabei den regelbasierten Ansatz deutlicher zu visualisieren sowie die Anwendung der Energieintervalle zur Auswahl der Fragment-Ionen zu zeigen.

Um das Verhalten von `ChemFrag` im positiven Ionisierungs-Modus $[M+H]^+$ nachvollziehen zu können, wurde eine Auswahl an sieben Doping-Substanzen für das Experiment getroffen. Diese Gruppen enthalten Substanzen aus Stimulanzien, β_2-Agonisten, Narkotika, von Tetrahydrochinolin abgeleitete Selective Androgen Receptor Modulators (SARMs), Hypoxie-induzierte Faktor (HIF) Stabilisierer und Sirtuin Aktivatoren sowie Kalzium-Kanal Modulatoren [8]. Das Experiment wird zeigen, dass im Mittel 80 % der Fragment-Ionen durch `ChemFrag` annotiert werden können. Zunächst visualisieren wir in diesem Unterkapitel `ChemFrag`‘s Verhalten an den zwei Beispielen Ephedrin und Kokain im positiven Ionisierungsmodus. Anschließend gehen wir auf die weiteren Beispiele in kompakterer Form ein.

Beispiel 1: Ephedrin
Ephedrin gehört zu den Phenylethylaminen und gliedert sich in die chemische Gruppe der Alkaloide ein. Die Anwendung von Ephedrin bewirkt eine stimulierende Wirkung [74]. Abbildung 15 zeigt das Massenspektrum von Ephedrin, welches annotiert werden soll sowie dessen chemische Struktur.
In einem ersten Schritt erläutern wir, wieviele Fragment-Ionen `ChemFrag` während der Fragmentierung generiert hat und wie groß der Anteil an selektierten Fragment-Ionen für den nächsten Fragmentierungsschritt ist. Die Selektion basiert, wie im Kapitel 4.2.1 beschrieben, auf der Größe des Reaktions-Intervalls. Tabelle 1 gibt dazu einen ersten Überblick. Hervorzuheben ist dabei, dass `ChemFrag` nur drei Fragmentierungsschritte benötigt, um das Spektrum vollständig zu annotieren. Beim Vergleich der generierten und selektierten Fragment-Ionen sehen wir gut, dass durch die Wahl der Schwellwerte die selektierten Fragment-Ionen für den nächsten Fragmentierungsschritt deutlich geringer sind als die Gesamtanzahl der generierten Fragment-

Tabelle 1: Übersicht zur Anzahl an generierten und selektierten Fragment-Ionen von Ephedrin für den Protonierungsschritt und die ersten drei Fragmentierungsschritte (entnommen aus [66]).

Schritt	Fragmente		Umlagerungen		Umlagerungen mit Spaltungen	
	generiert	selektiert	generiert	selektiert	generiert	selektiert
0	7	2	-	-	3	3
1	12	5	27	6	8	5
2	38	6	27	5	13	4
3	21	2	5	2	0	0

Ionen. Erkennen können wir das beispielsweise im Protonierungsschritt, wo sieben protonierte Moleküle generiert werden, jedoch nur zwei für den nächsten Fragmentierungsschritt ausgewählt werden. Gleiches beobachten wir auch bei den generierten Fragment-Ionen im zweiten Fragmentierungsschritt. `ChemFrag` generiert 38 Fragment-Ionen, wovon nur sechs selektiert werden. Aus den genannten Erkenntnissen schlussfolgern wir, dass die energetische Bewertung die Auswahl chemisch stabiler Moleküle ermöglicht, die für den nächsten Fragmentierungsschritt geeignet sind.

Der zweite Teil des Experiments stellt den Fragmentierungsweg von Ephedrin dar und verdeutlicht die Anwendung des regelbasierten sowie quantenchemischen Ansatzes. Abbildung 16 visualisiert den vollständigen von `ChemFrag` erstellten Fragmentierungsweg. Vergleichend kann dafür der in der Literatur von Thevis [8] publizierte Weg in Abbildung S-3 im Anhang betrachtet werden. `ChemFrag` sagt zwei favorisierte Fragmentierungswege vorher. Dazu wählt es zwei der sieben verschiedenen Protonierungspositionen aus.

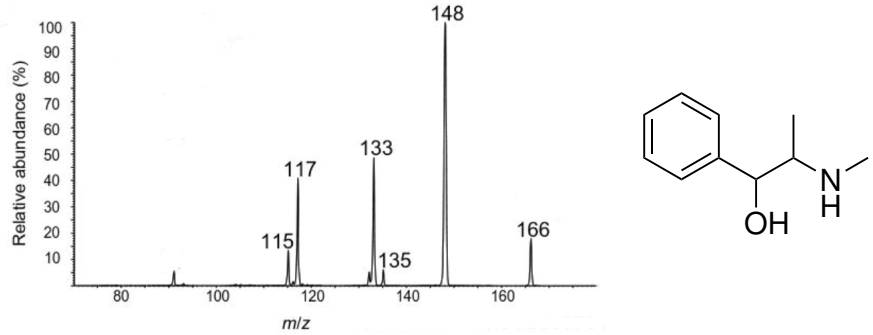


Abbildung 15: Positiv ionisiertes ESI CID Massenspektrum von Ephedrin aus Thevis [8] und die zugehörige Molekülstruktur (entnommen aus [66]).

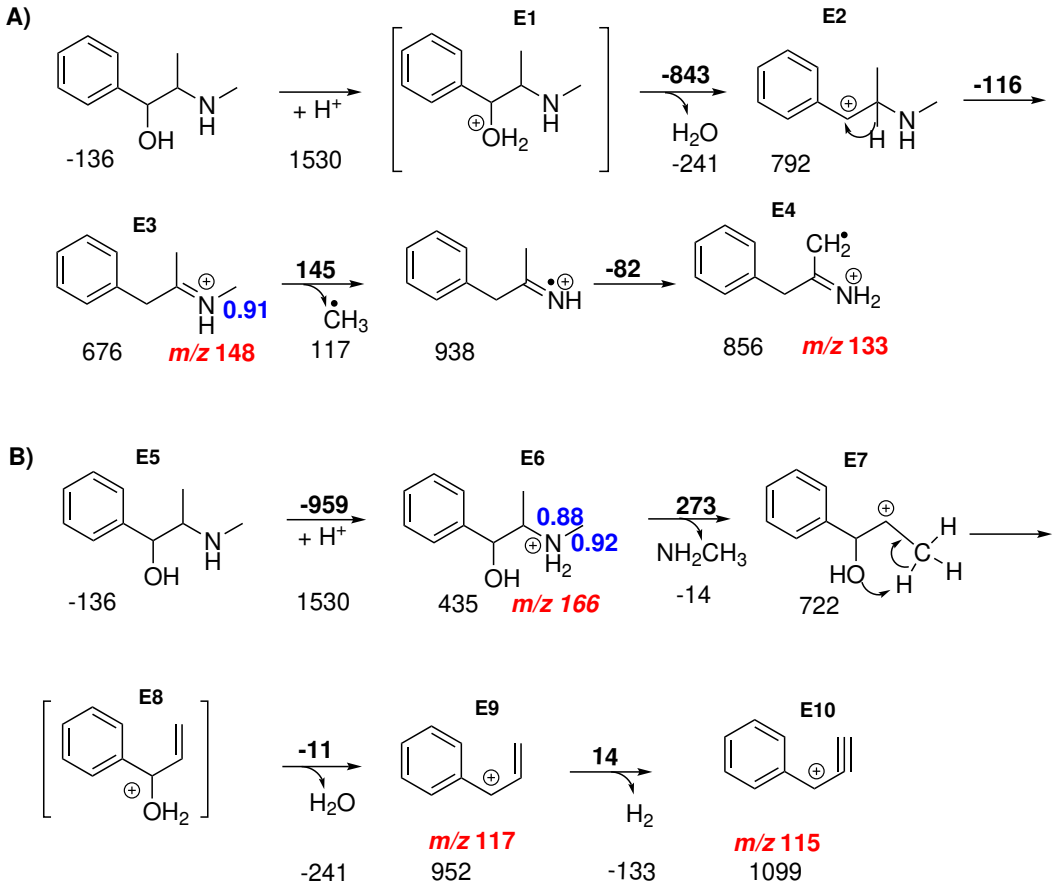


Abbildung 16: Fragmentierungprozess von Ephedrin, durch `ChemFrag` erstellt: Die Fragmentierung startet an zwei verschiedenen Protonierungspositionen. Die Energie der Fragment-Ionen, die unter den Strukturen stehen und die Reaktionsenergien, die auf den Pfeilen stehen, sind in kJ/mol angegeben. Die Schwellwerte entsprechen der später erläuterten Parameterkombination 1. Die Laufzeit beträgt 58 Sekunden auf einem 8-Kern-System (entnommen aus [66]).

Im ersten Schritt analysieren wir den Fragmentierungsweg A) aus Abbildung 16. Ausgehend von dem neutralen Molekül bis zu dem Fragment E3 mit einem Masse-zu-Ladungs-Verhältnis von 148 simuliert `ChemFrag` diesen Vorgang in einer Reaktion, ohne Zwischenrechnung von `MOPAC`. In dieser Reaktion erkennt `ChemFrag` zunächst, dass das protonierte Molekül (E1) durch das positiv geladene Wasser instabil ist.
Um ein stabiles Molekül zu generieren, detektiert `ChemFrag` das protonierte Wasser und überführt das Proton zu dem benachbarten sekundären Kohlenstoffatom (E2). Um ein energetisch instabiles Fragment zu verhindern, spaltet `ChemFrag` Wasser ab (Reaktion von E1 auf E2) und überführt im nächsten Schritt E2 zu E3. Für diese Überführung wendet `ChemFrag` einen Protonen-Shift (Abbildung 10(4)) an, da ein sekundäres Kohlenstoff-Kation energetisch relativ schwach im Vergleich zu einem

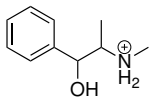
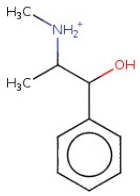
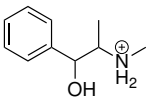
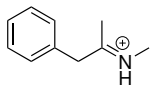
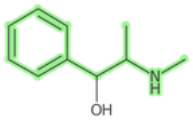
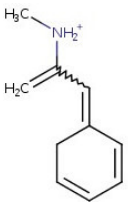
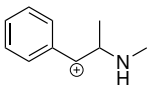
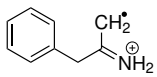
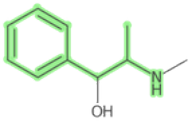
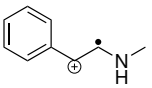
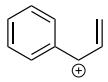
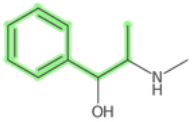
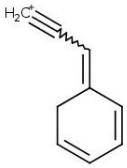
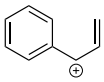
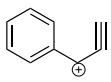
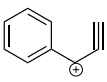
tertiären Kohlenstoff-Kation ist. Diesen Fakt sehen wir auch an den Bindungsenthalpien, wo E2 eine um 116 kJ/mol höhere Bindungsenthalpie hat als E3. `MOPAC` bewertet die Kohlenstoff-Stickstoff-Bindung des Fragment E3 als schwächste Bindung mit einer Bindungsordnung von 0,91. Daher führt `ChemFrag` für diese Bindung eine homolytische Spaltung zwischen dem positiv geladenen Stickstoffatom und der Methylgruppe durch. Die bereits erläuterten Regelanwendungen ermöglichen, dass `ChemFrag` die beiden Fragment-Ionen mit m/z 148 und 133 annotiert. Diese Reaktionen zeigen bereits, dass das Erkennen instabiler Fragment-Ionen und die Anwendung entsprechender Umlagerungen ein wichtiger Bestandteil von `ChemFrag` ist, um chemisch plausible Fragment-Ionen zu erzeugen.

Als zweite alternative Protonierungsposition erkennt `ChemFrag` das Stickstoffatom. Der zugehörige Fragmentierungsweg ist in B) abgebildet. Das protonierte Molekül-Ion E6 hat einen m/z-Wert von 166 und annotiert damit diesen Peak. Es zeichnet sich mit der Reaktionsenergie von -959 kJ/mol als bevorzugtes und energetisch stabiles Molekül-Ion aus. `ChemFrag` favorisiert für das protonierte Molekül-Ion E6 aufgrund der Bindungsordnung von 0,88 die heterolytische Abspaltung von Methylamin. In Folge der Abspaltung formt `ChemFrag` das Intermediat E8 aus dem Fragment-Ion E7. Diese Umformung ermöglicht `ChemFrag` durch einen Protonen-Shift vom neuen positiv geladenen Kohlenstoffatom zu der Hydroxylgruppe. Aufgrund der Instabilität von E8 entfernt `ChemFrag` das Wassermolekül und das Kation wird zum gebundenen Kohlenstoffatom überführt. Das entstehende Fragment-Ion E9 erläutert den Peak mit m/z 117. Nach Anwendung der Regel zur Entfernung von Wasserstoff generiert ChemFrag auch das letzte Fragment-Ion E10 mit m/z 115. Mit dieser Abfolge von chemischen Regeln können wir demonstrieren, dass `ChemFrag` chemisch realistische Umlagerungen simuliert und Regeln auf Basis der Bindungsordnungen für homolytische und heterolytische Spaltungen besitzt. Nachfolgend stellen wir das Ergebnis von `ChemFrag` den Ergebnissen aus der Literatur und denen der Programme `MetFrag`, `CFM-ID` und `SIRIUS` gegenüber.

Vergleich des Reaktionsweges von Ephedrin mit etablierten Methoden
Um zu bewerten, wie chemisch sinnvoll die Vorhersagen für die Fragment-Ionen der einzelnen Peaks durch `ChemFrag` sind, vergleichen wir das Ergebnis mit dem publiziertem Fragmentierungsweg von Thevis [8]. Zusätzlich werden die Vorhersagen der Fragment-Ionen den Annotationen aus `MetFrag`, `CFM-ID`, und `SIRIUS` gegenübergestellt. Die einzelnen Fragment-Ionen führt die Tabelle 2 auf.

Vergleich mit der Literatur:
Betrachten wir die Tabelle 2 so erkennen wir, dass für die Peaks mit m/z 166,

Tabelle 2: Visualisierung der Ergebnisse aus `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den aus der Literatur bekannten Strukturen zu den zugehörigen Peaks für Ephedrin. Die Energieberechnung der Strukturen aus `MetFrag` ist nicht möglich, da die Position der positiven Ladung nicht bekannt ist (angelehnt an [66]).
* Orginialbilder der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	`ChemFrag`	`MetFrag` *	`CFM-ID` *	Literatur
166	 435 kJ/mol		 435 kJ/mol	 435 kJ/mol
148	 676 kJ/mol	 $[C_{10}H_{14}N]^+$	 812 kJ/mol	 868 kJ/mol
133	 856 kJ/mol	 $[C_9H_{11}N]^+$		 925 kJ/mol
117	 952 kJ/mol	 $[C_9H_{10}\text{-}H]^+$	 **	 952 kJ/mol
115	 1099 kJ/mol			 1099 kJ/mol

117 und 115 `ChemFrag` die selben Fragment-Ionen vorhersagt, wie sie in der Literatur von Thevis gezeichnet sind. Für den Peak mit m/z 148 (Molekül E3) simuliert

ChemFrag einen Hydrid-Shift um ein stabileres Fragment-Ion zu generieren. Diese Umlagerung ist in der angegebenen Literatur nicht dargestellt. Aufgrund dieser unterschiedlichen Generierung der Fragment-Ionen sind ebenfalls die Fragment-Ionen des Peaks 133 verschieden. Sie unterscheiden sich an verschiedenen Positionen der Methylgruppen-Abspaltung. Um zu bestimmen, welche Struktur stabiler ist, betrachten wir die Energie der Strukturen. Am Beispiel von m/z 133 sehen wir, dass das Fragment-Ion aus ChemFrag eine niedrigere Energie besitzt und damit energetisch zu bevorzugen ist. Die energetische Analyse zeigt uns damit, dass ChemFrag tendenziell stabilere Fragment-Ionen produziert als die in der Literatur diskutierten Ergebnisse.

Vergleich mit MetFrag *und* CFM-ID:
Bei dem Vergleich der Ergebnisse für MetFrag und CFM-ID besteht das Problem, dass beide Programme keine 3D-Strukturen als Ausgabe generieren. Im Fall von CFM-ID wurden deshalb auf Basis der Ausgabebilder die zugehörigen Strukturen generiert. Anschließend erfolgte für diese Strukturen die Energieberechnung. Leider ist ein direkter Vergleich der Ausgaben von MetFrag nicht so simpel, da MetFrag nur die Substrukturen des Moleküls hervorhebt, die an dem Fragment-Ion beteiligt sind. Dadurch sind beispielsweise die Positionen der positiv geladenen Atome oder Radikale nicht ersichtlich. Das hat zur Folge, dass keine korrekte Energieberechnung erfolgen kann.
Tabelle 2 beinhaltet auch den Vergleich der zwei Programme mit ChemFrag. Da sich das neutrale geladene Ausgangsmolekül und das protonierte Molekül nur durch das positiv geladene Atom unterscheiden, gibt MetFrag keine Struktur für das protonierte Molekül an. Beim Vergleich der vorhergesagten Protonierungspositionen von ChemFrag und CFM-ID sehen wir, dass beide die gleiche Struktur generieren. Im Gegensatz dazu steht die Vorhersage des Fragment-Ions mit m/z 148. Hier bestimmt CFM-ID eine chemische Struktur, die energetisch um 136 kJ/mol schlechter ist als die von ChemFrag. Der positive Effekt des Proton-Shift, als Umlagerung in ChemFrag, ist an diesem Fragment-Ion sehr gut erkennbar. Die niedrigere Energie deutet auf ein deutlich stabileres Fragment-Ion hin, welches chemisch plausibler sein wird. Wie bereits erwähnt, hebt MetFrag die Substrukturen nur hervor. Dadurch sind die Umlagerungen für das Fragment-Ion mit m/z 148 und das Radikal am Fragment-Ion mit m/z 133 im Ergebnis von MetFrag nicht ersichtlich. Bemerken müssen wir an dieser Stelle, dass CFM-ID chemisch ungültige Strukturen hervor sagt, wie die Struktur für m/z 117, für die eine Energieberechnung nicht möglich ist. Weder CFM-ID noch MetFrag können ein Fragment mit m/z 115 hervorsagen. Zusammenfassend konnten wir evaluieren, dass ChemFrag chemisch realistischere Fragment-Ionen erstellt als die Ansätze von CFM-ID und MetFrag.

Vergleich mit ***SIRIUS*** *und* ***CSI:FingerID***:
Neben der Ausgabe des vollständigen Reaktionsweges ist es möglich, Fragmentierungsbäume aus den Aufgaben von `ChemFrag` abzuleiten. Aus der Einleitung wissen wir bereits, dass auch das Programm `SIRIUS` [54] Fragmentierungsbäume vorhersagt. Um abschließend die Leistungsfähigkeit von `ChemFrag` für Ephedrin zu evaluieren, vergleichen wir die erstellen Fragmentierungsbäume der beiden Programme. In Abbildung 17 sehen wir den nachgebildeten Fragmentierungsbaum aus `SIRIUS`.
Der von `SIRIUS` generierte Fragmentierungsbaum enthält ausschließlich die Summenformeln der Fragment-Ionen. Die Erweiterung um `CSI:FingerID` [57] ermöglicht im nächsten Schritt eine Vorhersage der Strukturen für die angegebenen Summenformeln.

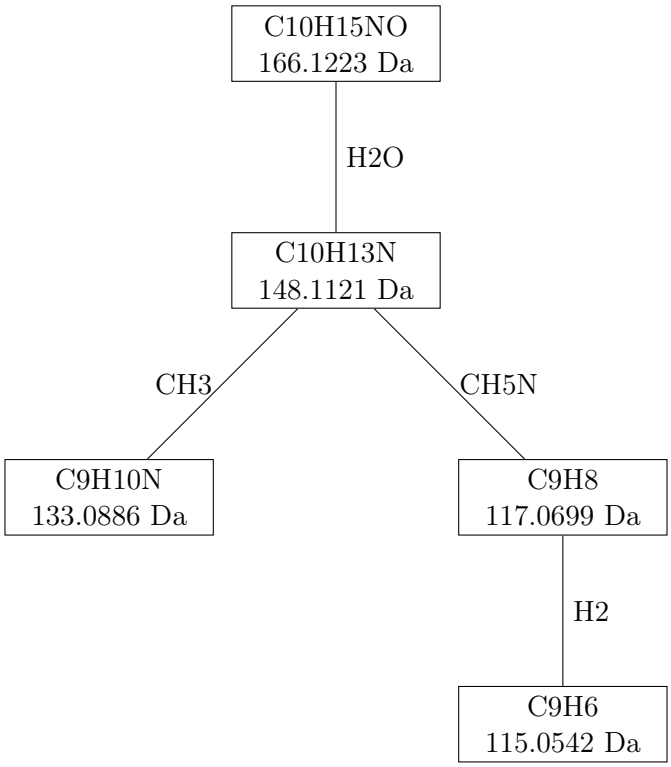
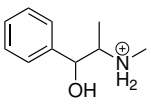
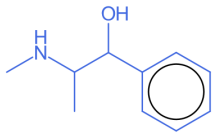
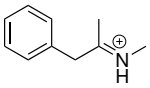
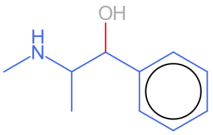
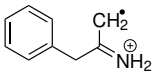
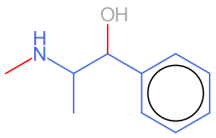
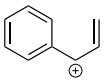
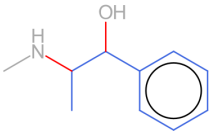
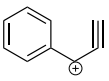
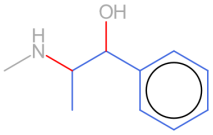


Abbildung 17: Nachbildung des von `SIRIUS` generierten Fragmentierungsbaums für die Eingaben von Ephedrin (nachgebildet von [54]).

Daher bilden wir in Abbildung 21 zunächst den Fragmentierungsbaum ab und fügen in der Tabelle 3 die Summenformeln mit den chemischen Strukturen zusammen.
Basierend auf der Tabelle 3 und den Fragmentierungsbäumen von `SIRIUS` (Abbildung 17) und `ChemFrag` (Abbildung 18) vergleichen wir im kommenden Schritt die Ergebnisse der drei Programme. Der von `ChemFrag` generierte Fragmentierungsbaum in Abbildung 18 zeigt die beiden Reaktionswege, die durch die zwei unterschiedlichen Protonierungspositionen entstehen.

Tabelle 3: Zuordnung der chemischen Strukturen zu den aus SIRIUS ermittelten Summenformeln sowie der Vergleich zu den aus ChemFrag generierten Strukturen für Ephedrin.

m/z	ChemFrag	SIRIUS	CSI:FingerID
166		C10H15NO	
148		C10H13N	
133		C9H10N	
117		C9H8	
115		C9H6	

Diese Unterscheidung ist im von SIRIUS produzierten Fragmentierungsbaum nicht erkennbar. Dort werden sowohl die Fragment-Ionen mit m/z 133 und m/z 117 aus dem Fragment-Ion m/z 148 gebildet. ChemFrag jedoch sagte vorher, dass das Fragment-Ion mit m/z 117 durch die Protonierung des Stickstoffsatoms und einer sich anschließenden Wasserabspaltung entsteht. Gleichermaßen sagen beide Programme vorher, dass das Fragment-Ion m/z 155 durch eine Wasserstoffabspaltung aus dem Fragment-Ion m/z 117 gebildet wird. Vergleichen wir im zweiten Schritt noch die Strukturen der Fragment-Ionen erkennen wir für CSI:FingerID, dass die am Fragment-Ion beteiligten Atome und Bindungen farblich hervorgehoben sind. Die gespaltene Bindung ist rot dargestellt und die abgespaltenen Teilstrukturen grau. Wie bei MetFrag sind auch hier keine Umlagerungen, Radikale oder die Positionen der Ladung ersichtlich. Positiv für den Vergleich ist hervorzuheben, dass ChemFrag und CSI:FingerID die gleichen Atome und Bindungen für die Fragment-Ionen kennzeichnen. Wie bereits angedeutet, lassen sich für die Fragement-Ionen m/z 133, m/z 117 und m/z 115 die

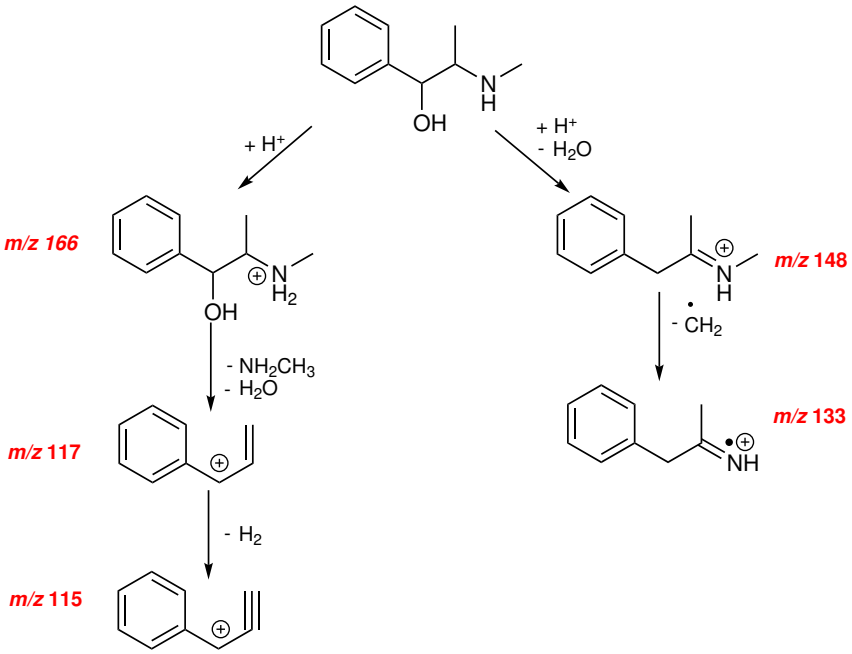


Abbildung 18: Darstellung des von `ChemFrag` generierten Fragmentierungsbaums für die Eingaben von Ephedrin.

Umlagerungen zur Erzeugung energetisch stabiler Fragment-Ionen nicht erkennen. Wir ziehen damit als abschließendes Fazit dieses Vergleichs, dass sich die Fragmentierungsbäume beider Programme stark ähneln. Beide Programme können für alle Peaks Strukturen und Summenformeln vorhersagen. Die vorhergesagten Atome und Bindungen der Strukturen der Fragment-Ionen stimmen überein und deuten auf die Richtigkeit der Programme `ChemFrag` und `CSI:FingerID` hin.

Beispiel 2: Kokain

In unserem zweiten Beispiel betrachten wir die Fragmentierungs-Ergebnisse für Kokain. Kokain gehört zu der Gruppe der Tropan-Alkaloide [75]. Es wirkt sowohl als starkes Stimulans und als Betäubungsmittel. Das Massenspektrum und die chemische Struktur von Kokain sind in Abbildung 19 ersichtlich.

Die Analyse des Fragmentierungsweges von Kokain ist komplexer als die von Ephedrin. Er enthält verschiedene Umlagerungen und spezielle Spaltungsregeln. Betrachten wir zunächst die reine Anzahl an generierten und selektierten Fragment-Ionen. Die Tabelle 4 enthält dazu die Übersicht. Positiv nehmen wir zur Kenntnis, dass `ChemFrag` nur vier Fragmentierungsebenen benötigt, um die Fragmentierung zu simulieren. Aufgrund der sehr guten Auswahl des Energieintervalls (siehe Parameterkombination1 in Kapitel 4.4) mussten in jeden Schritt nur wenige Fragment-Ionen für den nächsten Schritt selektiert werden. Besonders ersichtlich ist dies bei den Umlagerun-

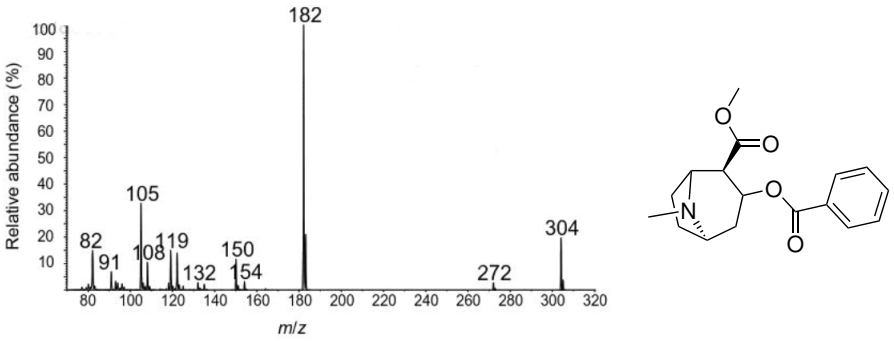


Abbildung 19: Positiv ionisiertes ESI CID Massenspektrum von Kokain aus Thevis [8] und die zugehörige Molekülstruktur (entnommen aus [66]).

Tabelle 4: Übersicht zur Anzahl an generierten und selektierten Fragment-Ionen von Kokain für die vier Fragmentierungsebenen (angelehnt an [66]).

Schritt	Fragmente		Umlagerungen		Umlagerungen mit Abspaltungen	
	generiert	selektiert	generiert	selektiert	generiert	selektiert
0	11	1	-	-	35	16
1	28	11	164	11	183	47
2	164	26	472	13	200	19
3	215	15	379	1	50	5
4	99	0	220	0	11	0

gen, wo beispielsweise im dritten Schritt 379 neue Fragment-Ionen generiert wurden, `ChemFrag` aber nur eine als chemisch relevant einstufte. Damit reduziert `ChemFrag` den Fragmentierungsbaum deutlich und verhindert frühzeitig chemisch nicht sinnvolle Fragmentierungen. Nach den Erkenntnissen zur Anzahl der Fragment-Ionen werden wir im Folgenden den von `ChemFrag` vorhergesagten Fragmentierungsweg und die Annotation der Fragment-Ionen aus Abbildung 19 nachvollziehen.

Wie aus der Literatur bekannt (Vergleich Abbildung S-5) und von `ChemFrag` vorhergesagt, hat Kokain mehrere mögliche Protonierungspositionen. Die erste mögliche Protonierung am Stickstoffatom sehen wir in Abbildung 20 A).

Beginnend bei dem Molekül C1 mit der am stärksten favorisierten Protonierungsposition, dem Stickstoffatom, erfolgt eine erste Bindungsspaltung auf Basis der niedrigsten Bindungsordnung. Die schwächste Bindung mit 0,86 befindet sich zwischen dem Stickstoffatom und dem Kohlenstoffatom im Ring. Diese Spaltung führt zu einem instabilen Fragment C2. Nach Anwendung des Protonen-Shifts entsteht das deutlich

stabilere Fragment C3. Die Energiedifferenz der Reaktionsenergie von-204 kJ/mol, zeigt die energetische Bevorzugung dieser Reaktion. Die schwache Bindung (BO = 0,88) zwischen dem Stickstoffatom und dem Kohlenstoffatom führt zur Spaltung dieser Bindung. Dabei spaltet `ChemFrag` Methylamin als Neutralverlust ab und bildet das instabile Fragment C4. Um ein stabileres Fragment zu erhalten, wendet `ChemFrag` einen weiteren Protonen-Shift an. Dieser überträgt das Proton vom Kohlenstoffatom zum Sauerstoffatom des Esters (C5). Das positiv geladene Sauerstoffatom im Ester führt zu einem instabilen Fragment. Die Folge wäre, dass eine energetische Berechnung nicht möglich wäre. Um das zu verhindern, führt `ChemFrag` eine Regel zur Aufspaltung des Esters durch (Regel S-2(k)). Dabei bildet sich die Benzoesäure als Neutralverlust. Ein weiterer Protonen-Shift vom gebildeten Fragment-Ion C6 zu C7 führt zu einer erneuten Esterspaltung und dem experimentell ermittelten Fragment-Ion C8 mit m/z 119. Gleichzeitig spaltet `ChemFrag` in diesem Schritt Methanol ab. Nach der Spaltung der schwächsten Bindung (BO = 0,90) zwischen dem Kation und dem Kohlenstoffatom des 7-Rings wendet `ChemFrag` einen Allyl-Shift an, um das energetisch favorisierte Fragment C10 mit m/z 91 zu generieren.

Der zweite Fragmentierungsweg (B) startet mit der am zweitstärksten energetisch favorisierten Protonierungsposition. Diese befindet sich am Sauerstoff der Estergruppe (C11). Die beiden Bindungen des protonierten Sauerstoffatoms sind infolge der Protonierung instabil, sodass `ChemFrag` sie spaltet. `ChemFrag` erstellt damit die zwei neuen Fragment-Ionen C12 und C13, wobei C12 das Fragment-Ion mit m/z 105 annotiert. Auf das Fragment-Ion C13 wendet `ChemFrag` zwei verschiedene Protonen-Shifts an, da sowohl das Stickstoffatom als auch das einfach gebundene Sauerstoffatom ein Proton binden können. Durch den Proton-Shift zum Stickstoffatom entsteht das stabile Fragment-Ion C14 mit m/z 182. Im Gegensatz dazu bildet der Shift auf das Sauerstofftatom ein instabiles Zwischenprodukt (C15), welches durch Abspaltung von Methanol zu dem stabilen Fragment-Ion mit m/z 150 (C16) wird. Im letzten Fragmentierungsschritt spaltet `ChemFrag` Kohlenstoffmonoxid ab und ermöglicht damit die Annotierung des Peaks m/z 122 mit dem Fragment-Ion C18. Um den von `ChemFrag` simulierten Fragmentierungsweg zu evaluieren, vergleichen wir ihn wieder mit jedem Fragment-Ionen aus den Vorhersagen von `MetFrag`, `CFM-ID`, `SIRIUS` und dem publiziertem Reaktionswege aus der Literatur [8].

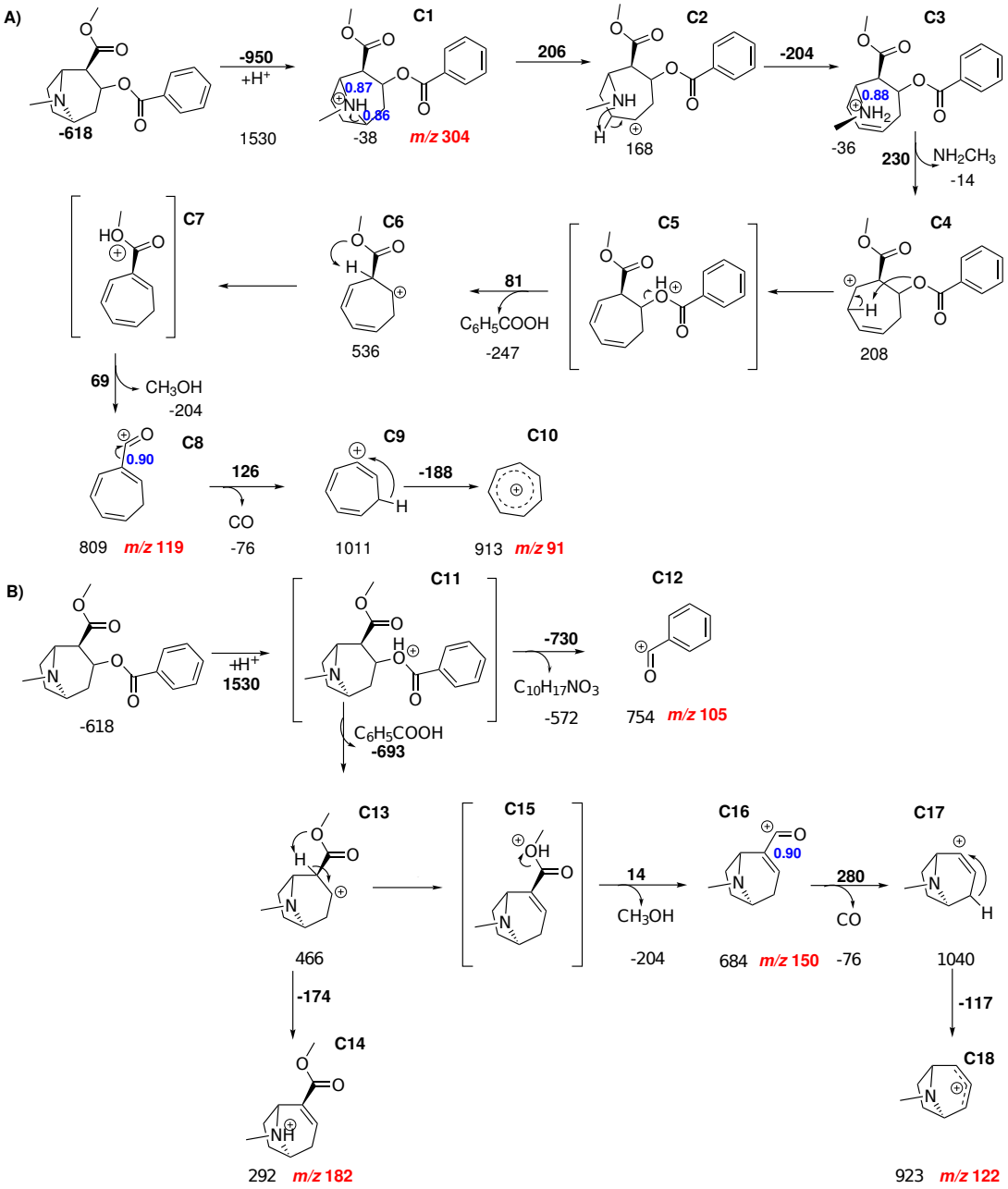


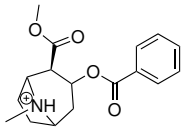
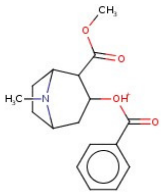
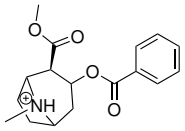
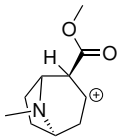
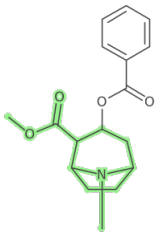
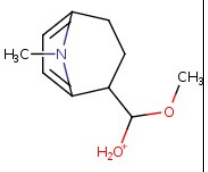
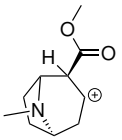
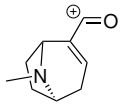
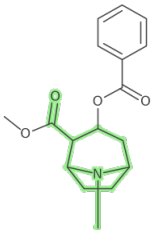
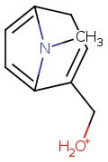
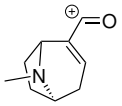
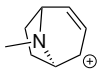
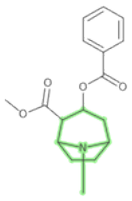
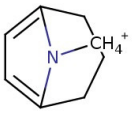
Abbildung 20: Fragmentierungsprozess von Kokain durch `ChemFrag` erstellt: Die Fragmentierung startet an zwei verschiedenen Protonierungspositionen. Die Energie der Fragment-Ionen und die Reaktionsenergien sind in kJ/mol angegeben. Die Schwellwerte entsprechen der später erläuterten Parameterkombination 1. Die Laufzeit beträgt 796 Sekunden auf einem 8-Kern-System (entnommen aus [66]).

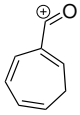
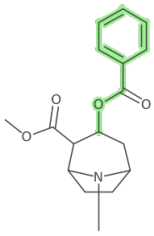
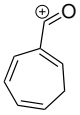
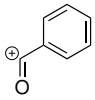
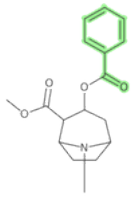
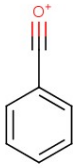
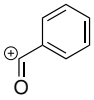
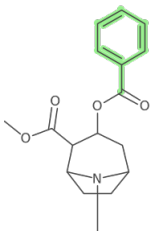
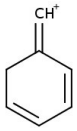
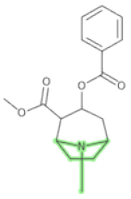
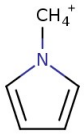
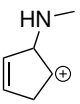
Vergleich des Reaktionsweges von Kokain mit etablierten Methoden Wie bereits bei der Evaluierung der Ergebnisse von Ephedrin, stellen wir die vorhergesagten Fragment-Ionen aus `ChemFrag` den publizierten Wegen von Thevis und den Strukturen aus `MetFrag`, `CFM-ID` und `SIRIUS` gegenüber.

Vergleich mit der Literatur:
Die Ergebnisse von `ChemFrag` sind fast identisch mit dem publizierten Fragmentierungsweg von Thevis [8] (Abbildung S-5). Ein Unterschied besteht nur bei dem Fragment-Ion mit m/z 82, da dieses von `ChemFrag` nicht generiert werden kann. Der Grund könnte in einer fehlenden Regel für komplexe Zyklisierungen liegen, da dafür komplexe Umlagerungsregeln notwendig sind.

Vergleich mit `MetFrag` und `CFM-ID`:
Tabelle 5 zeigt uns, dass alle drei Programme fast alle Peaks erläutern können. Während `CFM-ID` das protonierte Molekül C11 für den Peak m/z 304 annimmt, berechnet `ChemFrag` C1 dafür, da das Fragment-Ion C1 eine geringe Energie als C11 hat und damit energetisch stabiler ist. `MetFrag` annotiert nur Fragment-Ionen, die in der Masse kleiner sind als der Peak des Molekül-Ions. Für die Ionen m/z 182, 150, 122 und 119 hebt `MetFrag` die gleichen Bindungen hervor, die auch in `ChemFrag` für diese Fragment-Ionen existieren. Wie bereits erwähnt, lässt sich nicht erkennen an welchen Positionen die Ladung oder Atomumlagerungen in den von `MetFrag` generierten Strukturen vorliegen. Im Vergleich mit `CFM-ID` zeigt sich, dass die Fragment-Ionen mit m/z 182, 150, 122 und die 105 die selben Atome wie die Strukturen von `ChemFrag` enthalten. Jedoch unterscheiden sie sich in der Position ihrer Doppelbindungen und den Positionen der positiven Ladung. Die Folge dieser Unterschiede ist eine höhere Energie der von `CFM-ID` generierten Strukturen im Vergleich zu den `ChemFrag` ermittelten. Beispielsweise liegt für den Peak m/z 150 eine Energiedifferenz von 224 kJ/mol vor. Dieser signifikante Energieunterschied verdeutlicht, dass `ChemFrag` chemisch realistischere Strukturen vorhersagt als `CFM-ID`. Da `CFM-ID` für den Peak m/z 122 eine Struktur mit einem fünf-fach gebundenen Kohlenstoffatom annotiert, ist dafür keine Energieberechnung möglich. Beim Vergleich der Strukturen für den Peak m/z 91 ist festzustellen, dass `ChemFrag` einen stabilen 7-Ring vorhersagt, wohingegen `CFM-ID` einen 6-Ring bildet. Dieser ist energetisch um 236 kJ/mol schlechter als der 7-Ring. Anhand dieser Auswertungen schlussfolgern wir, dass `CFM-ID`, mit einem Maschine-Learning-Ansatz, noch mehr chemisches Wissen benötigt, um die Vorhersage chemisch nicht plausibler Fragment-Ionen zu verhindern. Dieses Wissen ist in `ChemFrag` durch den regelbasierten und den quantenchemischen Ansatz bereits abgebildet.

Tabelle 5: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von Kokain. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind. (angelehnt an [66])
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	`ChemFrag`	`MetFrag` *	`CFM-ID` *	Literatur[8]
304	 -38 *kJ/mol*		 62 *kJ/mol*	 -38 *kJ/mol*
182	 466 *kJ/mol*	 $[C_{10}H_{16}NO_2]^+$	 571 *kJ/mol*	 466 *kJ/mol*
150	 684 *kJ/mol*	 $[C_9H_{13}NO\text{-}H]^+$	 908 *kJ/mol*	 684 *kJ/mol*
122	 1010 *kJ/mol*	 $[C_8H_{13}N\text{-}H]^+$	 **	 1010 *kJ/mol*

m/z	ChemFrag	MetFrag *	CFM-ID *	Literatur[8]
119	 809 *kJ/mol*	 $[C_8H_6O]+H^+$		 809 *kJ/mol*
105	 754 *kJ/mol*	 $[C_7H_5O]^+$	 754 *kJ/mol*	 754 *kJ/mol*
91	 913 *kJ/mol*	 $[C_7H_5+H]+H^+$	 1149 *kJ/mol*	 913 *kJ/mol*
82		 $[C_5H_9N\text{-}H]^+$	 **	 733 *kJ/mol*

Vergleich mit `SIRIUS` *und* `CSI:FingerID`

Aus der Ausgabe von `ChemFrag` können wir wieder den Fragmentierungsbaum generieren, sodass dieser abschließend mit den Ergebnissen aus `SIRIUS` und `CSI:FingerID` verglichen werden kann. Damit können wir auch die Leistungsfähigkeit von `ChemFrag` für Kokain evaluieren. Abbildung 21 zeigt den nachgebildeten Fragmentierungsbaum aus `SIRIUS`.

Der von `SIRIUS` generierte Fragmentierungsbaum enthält ausschließlich die Summen-

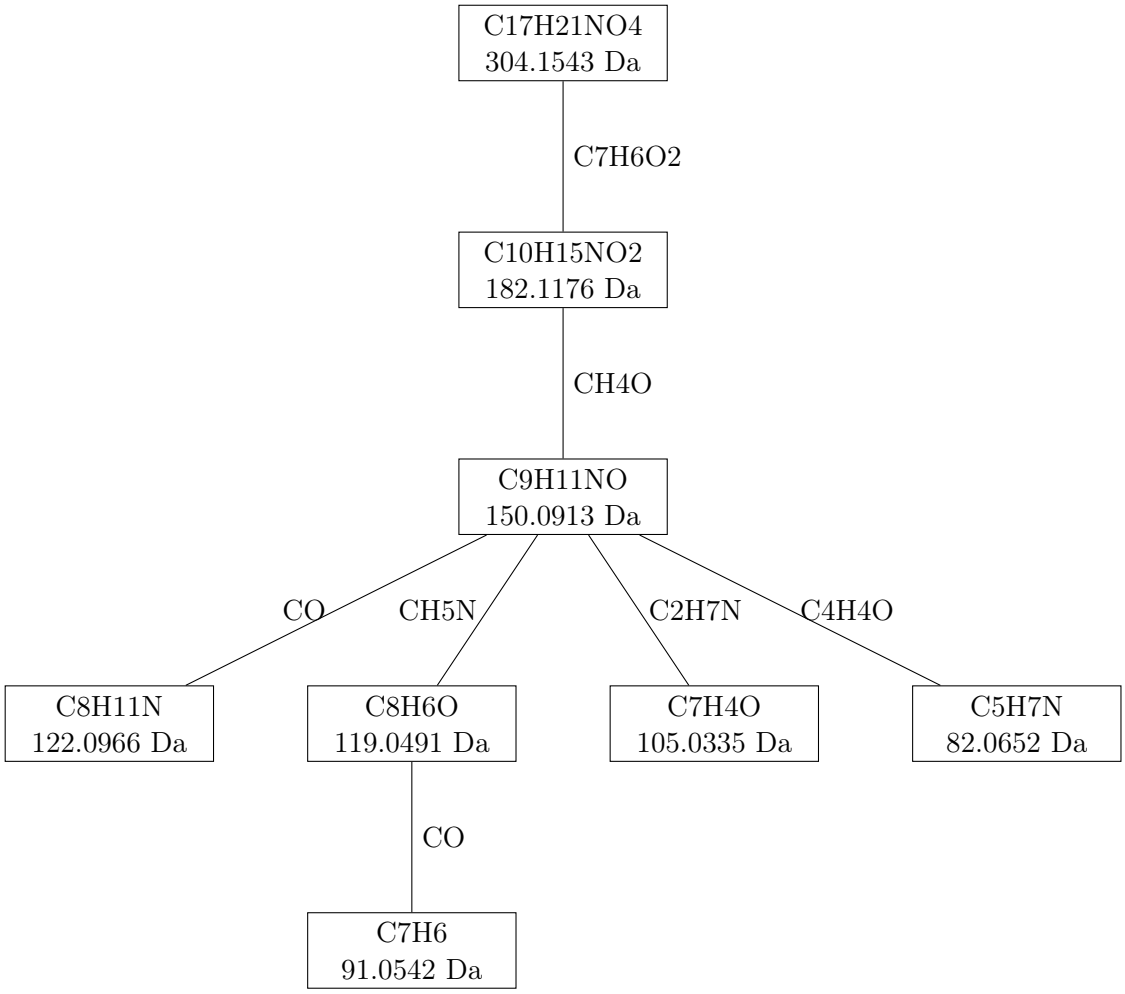
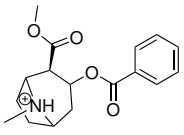
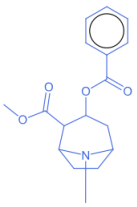
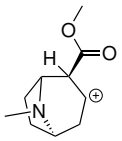
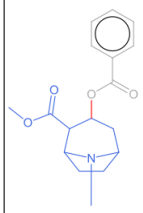
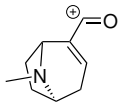
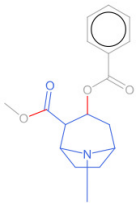
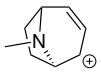
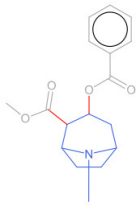
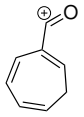
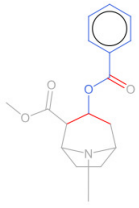
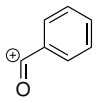
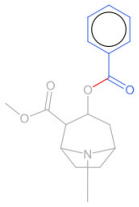


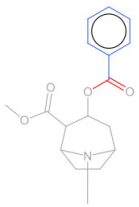
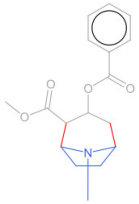
Abbildung 21: Nachbildung des von `SIRIUS` generierten Fragmentierungsbaums für die Eingaben von Kokain (nachgebildet von [54]).

formeln der Fragment-Ionen. Die Erweiterung um `CSI:FingerID` [57] ermöglicht im nächsten Schritt eine Vorhersage der Strukturen für die angegebenen Summenformeln. Daher bilden wir in Abbildung 21 zunächst den Fragmentierungsbaum ab und fügen in der Tabelle 6 die Summenformeln mit den chemischen Strukturen zusammen.

Für unseren Vergleich betrachten wir die Fragmentierungswege und die Vorhersagen der Fragment-Ionen genauer. Beginnen werden wir mit den Ergebnissen aus `SIRIUS` und `CSI-FingerID`. Aus der Abbildung 21 erkennen wir, dass für alle Fragment-Ionen Summenformeln bestimmt werden können. In der zusätzlichen Ausgabe von `CSI:FingerID` sind die am Fragment-Ion beteiligten Atome und Bindungen farblich hervorgehoben.

Tabelle 6: Zuordnung der chemischen Strukturen zu den aus `SIRIUS` ermittelten Summenformeln sowie der Vergleich zu den aus `ChemFrag` generierten Strukturen für Kokain.

m/z	`ChemFrag`	`SIRIUS`	`CSI:FingerID`
304		C17H21NO4	
182		C10H15NO2	
150		C9H11NO	
122		C8H11N	
119		C8H6O	
105		C7H4O	

m/z	ChemFrag	SIRIUS	CSI:FingerID
91		C7H6	
82		C5H7N	

Die abgespaltenen Neutralverluste sind dagegen grau unterlegt. Wie bei `MetFrag` ist es auch bei den aus `CSI:FingerID` generierten Strukturen nicht möglich, die Postion des positiv geladenen Atoms sowie Umlagerungen der Bindungen zu erkennen. Vergleicht man die Kanten und somit die Edukt-Produkt-Beziehung im Fragmentierungsbaum aus Abbildung 21 mit den farblich hervorgehobenen Atomen und Bindungen in der Tabelle 6 erkennen wir zwei mögliche Widersprüche zwischen den Summenformeln und den Strukturen. Es handelt sich dabei um die Kanten von m/z 150 zu m/z 119 sowie von m/z 150 zu m/z 105. In der Struktur für m/z 150 ist der 7-Ring mit dem Stickstoffatom und der Methylgruppe sowie die Aldehyd-Gruppe hervorgehoben. Der Bereich des Benzen-Rings mit der Aldehyd-Gruppe ist als abgespalten markiert. Aus der Struktur mit m/z 150 wird laut dem Fragmentierungsbaum zum einen CH_5N abgespalten, sodass das Fragment-Ion mit m/z 119 gebildet wird, und zum anderen C_2H_7N mit der Bildung des Fragment-Ions für m/z 105. Diese beiden Strukturen weisen jedoch den Benzen-Ring mit der Aldehyd-Gruppe als Teil des Fragments auf. An dieser Stelle erkennen wir den Widerspruch, da in der Vorgängerstruktur mit m/z 150 diese Teilstruktur abgespalten war. Hier stellt sich nun die Frage, ob der vorhergesagte Weg im Fragmentierungsbaum oder die Darstellung der chemischen Struktur korrekt sind.

Vergleichen wir dazu den Fragmentierungsbaum auf Basis der Ausgabe von `ChemFrag` aus Abbildung 22 erkennen wir die zwei Fragmentierungswege aus Abbildung 20 wieder. Im Gegensatz zu dem Fragmentierungsbaum aus `SIRIUS` entwickelt sich nur das Fragment-Ion mit m/z119 aus der am Stickstoffatom protonierten Ausgangsstruktur (m/z 304). Gleich ist zwischen beiden Programmen, dass sich das Fragment-Ion mit m/z 91 aus dem Fragment-Ion mit m/z 119 und das Fragment-Ion mit m/z 122 sich aus dem Fragment-Ion mit m/z 150 entwickelt. Deutlich verschieden ist jedoch, dass das Fragment-Ion m/z 150 nicht aus der Protonierung des Stickstoffatoms hervor-

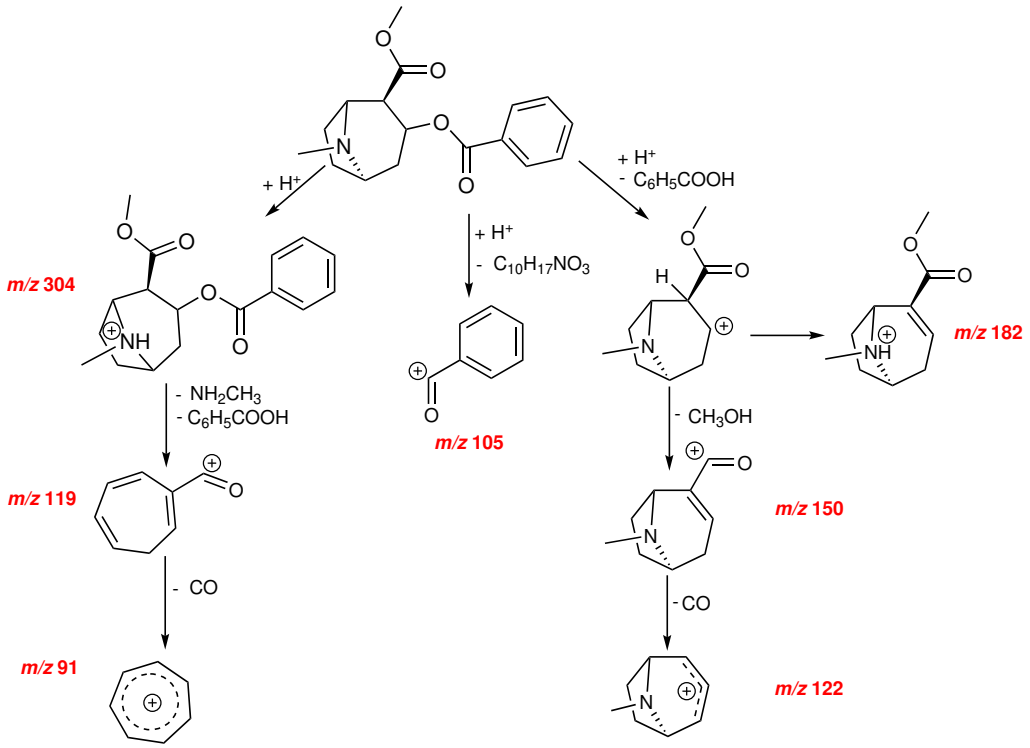


Abbildung 22: Darstellung des von `ChemFrag` generierten Fragmentierungsbaums für die Eingaben von Kokain.

kommt, sondern aus der des Esters. Aufgrunddessen unterscheiden sich die Fragmentierungsbäume der beiden Programme deutlich, da nur das Fragment-Ion mit m/z 122 eine Substruktur des Fragments m/z 150 ist. Vergleichen wir zusätzlich noch die hervorgehobenen Atome und Bindungen aus der Vorhersage von `CSI:FingerID` sind diese bis auf die Fragment-Ionen m/z 119 und m/z 91 zu den Ausgaben von `ChemFrag` identisch. Erkennbar ist nicht, wie bereits erwähnt, die Position der Ladung.

Aus unserem Vergleich können wir nun schlussfolgern, dass `SIRIUS` eine andere Reihenfolge in der Abspaltung der Neutralverluste vorhersagt als `ChemFrag`. Zusätzlich existieren Widersprüche in der Darstellung der Strukturen mit `CSI:FingerID` und den Wegen im Fragmentierungsbaum aus `SIRIUS`. Die vorgesagten Strukturen mit `CSI:FingerID` weisen eine große Ähnlichkeit mit denen aus `ChemFrag` auf, sodass wir davon ausgehen können, dass beide Programme chemisch plausible Strukturen für die Annotation bestimmen können.

4.4.3. Bewertung und Minimierung der quantenchemischen Berechnungen

Nachdem wir zuvor die chemische Leistungsfähigkeit für die Annotation der Fragment-Ionen diskutiert haben, nehmen wir noch Bezug zu dem Beginn dieses Kapitels. Darin haben wir den quantenchemischen Ansatz evaluiert. Nachteilig für diesen Ansatz ist die hohe Laufzeit durch die Integration von `RDKit` und `MOPAC`. Daher wird dieses Kapitel zum einen errechnen, wie groß der Anteil dieser Programme an der Gesamtlaufzeit von `ChemFrag` ist und wie sich die Möglichkeiten der Laufzeitoptimierung aus Kapitel 4.2.2 auswirken.

Experiment 4.3. Das Ziel des folgenden Experiments ist zu bestimmen, wie der Anteil der Laufzeit zwischen der Generierung neuer Fragment-Ionen mit deren Neutralverlusten und der Generierung der 3D-Koordinaten und der Bestimmung der Bindungsenthalpien und Bindungsordnungen ist. Der Prozess der Koordinatenerzeugung enthält, wie bereits im Kapitel 4.1.3 beschrieben, die Erzeugung von 2D-Koordinaten mittels `CDK` und die anschließende Berechnung von 3D-Koordinaten durch `RDKit` auf Basis der 2D-Koordinaten. Für die Bestimmung der Energie und der Bindungsordnungen betrachten wir die Ausführung von `MOPAC` und das anschließende Einlesen der Ergebnisse aus der `MOPAC` Ausgabe. Im Experiment messen wir die Zwischenlaufzeiten dieser drei Abschnitte während des Fragmentierungsprozesses von Kokain. Der Anteil von `RDKit` an der Gesamtlaufzeit beträgt lediglich 3,4 %, was 27 Sekunden im Fragmentierungsprozess entspricht. Für die quantenchemischen Berechnungen errechnen wir 684 Sekunden. Das entspricht 86 % der Gesamtlaufzeit.

Betrachten wir den gesamten Fragmentierungsprozess, so nehmen die Koordinaten-Generierung und die quantenchemischen Rechnungen in der dritten Fragmentierungebene mit 42 % an der Gesamtlaufzeit der Fragmentierung den größten Anteil an. Den geringsten Anteil stellt erwartungsgemäß die Protonierung dar, da in dieser Ebene nur die Protonierung und die Umlagerung, zur Vermeidung instabiler Moleküle, existieren. Wir können daher schlussfolgern, dass die Ausführung von `MOPAC` den größten Anteil an der Laufzeit von `ChemFrag` besitzt.

Experiment 4.4. Das vorherige Experiment unterstreicht die Annahmen aus Kapitel 4.2.2, dass die Anzahl quantenchemischer Berechnungen reduziert werden muss. Als eine Möglichkeit listete das Kapitel 4.2.2 die Minimierung redundanter Berechnungen während des gesamten Annotationsprozesses des Eingabe-Moleküls auf. Das betrifft hauptsächlich die Reduktion quantenchemischer Rechnungen gleicher Fragment-Ionen oder Neutralverluste. Um die Effektivität dieses Ansatzes zu bewerten, zeigen wir für die ersten Fragmentierungsschritte von Kokain die Reduzierung

der Molekülberechnungen.
Die Verwendung des Programms `MET` ermöglicht die Erkennung von äquivalenten Molekülen und den Ausschluss möglicher Berechnungen gleicher Moleküle. Statt redundanter Berechnungen, bilden `ChemFrag` und `MET` die Ergebnisse der Bildungsenthalpie und der Bindungsordnungen des bereits berechneten Moleküls auf das äquivalente Molekül ab. Dadurch verhindern sie die erneute Berechnung der 3D-Koordinaten durch `RDKit` und die Ausführung von `MOPAC`. Abbildung 23 zeigt den Anteil der Molekülberechnungen für die ersten Fragmentierungsschritte von Kokain. Dazu wurden fünf Gruppen für die Iterationen definiert:

- Fragment-Ionen
- Neutralverluste
- Fragment-Ionen aus Umlagerungen
- Fragment-Ionen aus Umlagerungen mit anschließenden Spaltungen
- Neutralverluste von Fragment-Ionen aus Umlagerungen mit anschließenden Spaltungen

Aus der Abbildung 23 erkennen wir, dass im Protonierungsschritt die Fragment-Ionen alle unterschiedlich sind, da 100 % der protonierten Moleküle berechnet werden müssen. Das entspricht auch den Erwartungen, da das Molekül an unterschiedlichen Positionen protoniert wurde. Durch die Umlagerungen können dann gleiche Fragment-Ionen und mit einer größeren Anzahl auch gleiche Neutralverluste entstehen. In der ersten Fragmentierungsebene zeichnen sich noch viele unterschiedliche Fragment-Ionen ab, wobei jedoch deren Neutralverluste in weniger als 50 % neu berechnet werden müssen. Die Anzahl an neu berechneten Fragment-Ionen, die aus Umlagerungen entstanden sind, liegt ebenfalls bei rund 50 %. Für den zweiten Fragmentierungsschritt bemerken wir dann noch einmal bei den Neutralverlusten besonders den sinnvollen Einsatz der Hash Map, da dort nur noch 20 % der Neutralverluste berechnet werden müssen. Dass die Fragment-Ionen fast vollständig neu berechnet werden müssen, ist zu erwarten, da `ChemFrag` durch die potenzielle Anwendung der 30 Spaltungsregeln auf die einzelnen Fragment-Ionen des vorherigen Schrittes, eine Vielzahl diverser Fragment-Ionen generiert.

4.4.4. Zusammenfassung der chemischen Plausibilität von `ChemFrag`

Die Verwendbarkeit von `MOPAC` konnten wir erfolgreich am Vergleich der Protonenaffinitäten mit Hunter *et al.* [70] zeigen. Weiterhin ist unsere Annahme, dass Bindungsordnungen die Stabilität einer Bindung bewerten durch den Vergleich der Bindungslängen mit Alex *et al.* [7] unterstrichen. Auch die chemische Plausibilität der

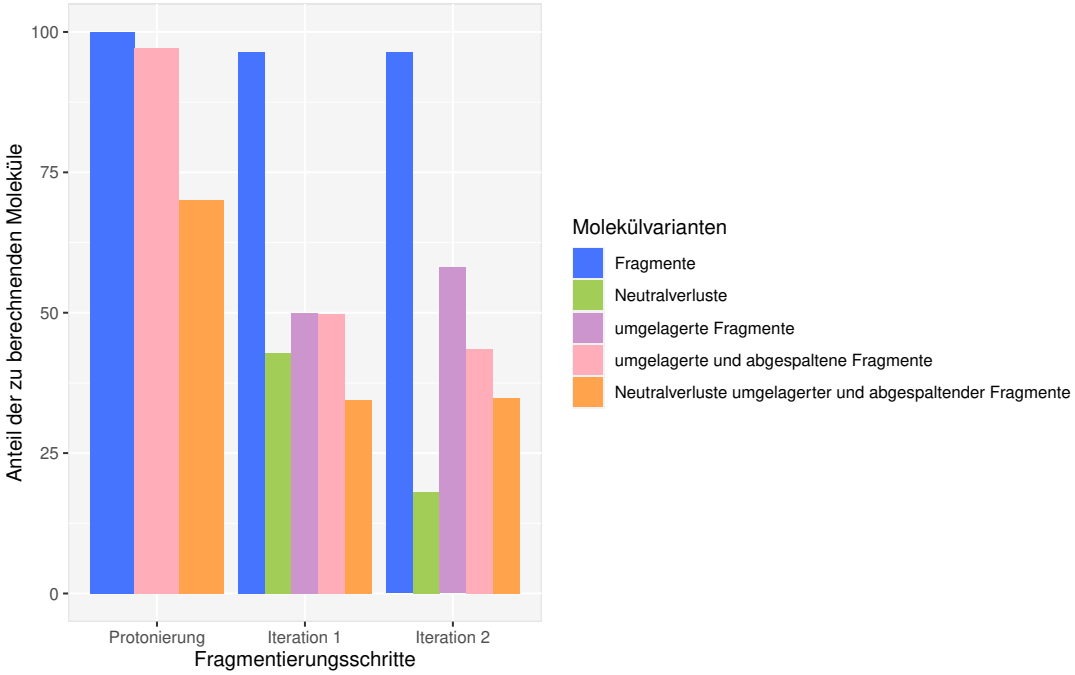


Abbildung 23: Berechnung der ersten Fragmentierungsebenen von Kokain. Die x-Achse zeigt die Ergebnisse für die Protonierung und die der ersten und zweiten Fragmentierungsebene für die verschiedenen Molekültypen. Die y-Achse zeigt den Anteil der Moleküle, die in den verschiedenen Fragmentierungsebenen berechnet werden müssen. Die Laufzeit beträgt 1465 Sekunden auf einem 8-Kern-System unter Verwendung der Hash-Map von `MET`. Im Vergleich dazu beträgt die Laufzeit 2125 Sekunden auf dem selben System ohne die Einbindung von `MET`. (angelehnt an [66])

Vorhersagen der Fragmentierungswege und der Annotation der Fragment-Ionen zeigten wir anhand ausgewählter Beispiele von Doping-Substanzen. Speziell für Ephedrin und Kokain sehen wir, dass `ChemFrag` chemisch sinnvolle Fragmentierungswege und damit Annotationen der Fragment-Ionen vorhersagen kann. Diese sind, wie die Gegenüberstellung zu `MetFrag`, `CFM-ID` und `SIRIUS` zeigt, oftmals chemisch plausibler als bisher etablierte Methoden. Besonders erkennbar ist dies an geringeren Energien, wie bei dem Fragment-Ion 148 von Ephedrin oder an der Vermeidung chemisch nicht sinnvoller Strukturen, wie beim Peak 122 von Kokain. Weitere Analysen von Doping-Substanzen im Kapitel C des Anhang listen auf, dass `ChemFrag` 19 von 21 Fragment-Ionen annotieren kann. Bei den zwei nicht annotierten Fragment-Ionen handelt es sich um das Fragment-Ion 132 von Clenbuterol, welches eine Intensität von unter 10 % hat sowie um das Fragment-Ion 131 mit einer Intensität von 40 % bei Pethidin. Das nicht aufgeklärte Fragment-Ion von Clenbuterol könnte aufgrund seiner geringen Intensität in der Annotation weggelassen werden. Anders sieht es bei der Annotation

des Fragment-Ions von Pethidin aus. Für dieses müsste in `ChemFrag` noch eine Regel zur Aufspaltung der Ringstruktur implementiert werden. Betrachten wir noch die Ergebnisse zu JTV-519 in Tabelle S-6 und LDG-2226 in Tabelle S-3 so sehen wir, dass `ChemFrag` alle Fragment-Ionen annotieren kann und `CFM-ID` keine oder nur die protonierte Struktur. Aus diesen Analysen können wir den Rückschluss ziehen, dass `ChemFrag` insbesondere `CFM-ID` in der chemisch plausiblen Aufklärung von MS/MS-Spektren überlegen ist. Aus dem Vergleich zu `SIRIUS` und `CSI:FingerID` kommen wir zu dem Schluss, dass die Programme ähnliche Fragmentierungswege vorhersagen und die am Fragment-Ion beteiligten Atome und Bindungen oftmals identisch sind. Auch hier hat `ChemFrag` den Vorteil, dass die Position der Ladung sowie der Radikale und Umlagerungen von Bindungen dargestellt werden können. Der Vergleich zu `SIRIUS` zeigte auf, dass `SIRIUS` und `CSI:FingerID` in sich Widersprüche in der Reihenfolge abgespaltener Neutralverluste aufweisen können.

4.5. Ergebnisse zur Parameterermittlung und Anwendbarkeit des regelbasierten Ansatzes

Nachdem das vorherige Experiment die Leistungsfähigkeit von `ChemFrag` an verschiedenen Doping-Substanzen zeigte, werden wir nun `ChemFrag` auf einen größeren Datensatz anwenden. Die MS/MS-Daten dazu stammen aus dem 2016 durchgeführten Wettbewerb „Critical Assessment of Small Molecule Identification“ (CASMI) [76]. Aus dem Wettbewerb wurde der Trainingsdatensatz der Kategorie zwei und drei verwendet. Dieser enthält 234 Listen von Fragment-Ionen im positiven Modus, aus 285 unterschiedlichen Substanzen. Für jeden Kandidaten werden die Strukturen im SDF oder CSV Format sowie als SMILES, InChI und InChIKey bereit gestellt.

In diesem Kapitel werden mehrere Aspekte von `ChemFrag` untersucht. Ein wichtiger Punkt wird die Beurteilung der Wichtigkeit der einzelnen Regeln sein. Zusätzlich führen wir auf Basis dieses Datensatz eine Parameteroptimierung der Reaktionsenergie- und Bindungsordnungswerte durch und bestimmen eine optimale Fragmentierungstiefe. Abschließend werden wir Bewertungsmethoden aus Kapitel 4.1.4 auf ihre Eignung evaluiert.

Bevor wir die Experimente durchführen, listen wir nachfolgend die Parameter auf, die für jedes Experiment von `ChemFrag` gleich waren.

Tabelle 7: Übersicht der Parameter (ϵ und κ) zur Berechnung des Intervalls der Reaktionsenergie der protonierten, der fragmentierten und der umgelagerten Moleküle sowie der Bindungsordnung zur Spaltung der Bindungen.

Kombination	Tprot	Tfrag	Trearr	TBO
1	50	150	100	0.08
2	50	150	100	1.00
3	50	200	150	0.08

- absolute Massenabweichung (-d): 0.01
- relative Massenabweichung (-ppm): 10
- Ionisierungsmethode (-pa): prot (Protonierung)
- semi-emprische Methode (-m): PM7 (PM7 aus MOPAC)
- Fragmentierungstiefe (-TD): 200

Zusätzliche Parameter für die Größe der Reaktions- und Bindungsordnungsintervalle sind die Werte von Tprot, Tfrag, Trearr und TBO. Der Parameter Tprot hat den Wert von ϵ, um das Reaktionsintervall im Protonierungsschritt zu bestimmen. Um das Reaktionsintervall nach der Anwendung der homolytischen oder heterolytischen Spaltung oder den Spaltungregeln zu errechnen, enthält der Parameter Tfrag den Wert für ϵ. In Trearr ist dann der Wert für ϵ enthalten, der für die Berechnung des Reaktionsintervalls nach der Anwendung der Umlagerungsregeln notwendig ist. In dem Parameter TBO ist der Wert für κ enthalten, der die Größe des Bindungsordnungsintervalle beeinflusst. Für diese Parameter haben wir drei Parameterkombinationen zusammengestellt, die in den Experimenten getestet werden. Tabelle 7 zeigt die Parameterkombinationen.

Beginnen werden wir die Experimente mit der Ermittlung der optimalen Fragmentierungstiefe.

4.5.1. Bestimmung der optimalen Fragmentierungstiefe

Die Fragmentierungstiefe bestimmt, wieviele Fragmentierungsebenen ausgeführt werden sollen. Eine Fragmentierungsebene besteht dabei aus den Bindungsspaltungen auf Grundlage der Bindungsordnungen und Spaltungsregeln sowie sich anschließenden Umlagerungen. Damit hat die Fragmentierungstiefe einen erheblichen Einfluss auf die Anzahl generierter Fragment-Ionen und auf die Laufzeit von `ChemFrag` sowie

Tabelle 8: Darstellung der vorkommenden Fragmentierungstiefen mit deren Häufigkeiten für die Parameterkombination 1.

Baumtiefe	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18
Häufigkeit	5	4	17	36	36	50	28	29	20	4	7	7	4	1	1	1

auf eine umfangreiche Annotation der Fragment-Ionen. Ist die Fragmentierungstiefe zu klein, werden oftmals nur Fragment-Ionen mit einem großen m/z-Verhältnis annotiert, da zu dem Zeitpunkt nur wenige Spaltungen stattgefunden haben. Auf der anderen Seite führt eine hohe Fragmentierungstiefe zu einer hohen Anzahl an Fragment-Ionen, die keine Annotationen bewirken, da ihre Masse zu gering ist. Weiterhin bewirkt jede weitere Fragmentierungstiefe eine Erhöhung der Laufzeit aufgrund neuer Berechnungen in `RDKit` und `MOPAC` für jedes neue Fragment-Ion. Es ist daher das Ziel, ein ausgewogenes Verhältnis zwischen der Anzahl an annotierten Fragment-Ionen in einer akzeptablen Laufzeit zu erreichen. Dieses Verhältnis möchten wir im folgenden Experiment ermitteln.

Experiment 4.5: `ChemFrag` hat zwei Möglichkeiten, um den Fragmentierungsprozess zu beenden. Eine Möglichkeit stellt das Erreichen der maximalen Fragmentierungstiefe dar und `ChemFrag` stoppt außerdem, sobald keine neuen Fragment-Ionen für den nächsten Schritt selektiert werden können. Das tritt genau dann ein, wenn alle Fragment-Ionen eine kleinere Masse als der kleinste m/z-Wert aufweisen. Für das Experiment legen wir einen hohen Schwellwert (TD = 200) für die Fragmentierungstiefe fest. Ziel ist es damit herauszufinden, ab welcher Tiefe keine selektierten Fragment-Ionen mehr vorliegen.

`ChemFrag` ruft dieses Experiment mit der Parameterkombination 1 auf. Es berechnet für alle Moleküle des CASMI-Datensatzes die vollständige Fragmentierung, bis keine selektierten Fragment-Ionen mehr für die weitere Fragmentierung vorhanden sind. Die Tabelle 8 zeigt uns, ab welchen Fragmentierungstiefen dies geschieht.
Aus der Tabelle 8 erkennen wir, dass eine geeignete Fragmentierungstiefe zwischen fünf und zehn liegt. Dabei beendete `ChemFrag` die meisten Fragmentierungen bei der Fragmentierungstiefe sieben. Auch der Median von sieben und der Mittelwert mit 7,38 weisen auf eine Wahl sieben als geeignete Fragmentierungstiefe hin. Fragmentierungstiefen, die kleiner als fünf oder größer als zehn sind, sind gleichermaßen wenig vertreten. Möchte man die Laufzeit von `ChemFrag` reduzieren, kann das durch die Fragmentierungstiefe erfolgen. Für diesen Fall kann der Parameter TD auf die Fragmentierungstiefe sieben festgelegt werden.

Führen wir das gleiche Experiment mit den Parameterkombinationen 2 und 3 aus, bilden die Tabellen S-7 und S-8 die Ergebnisse ab. Vergleichen wir diese Auflistungen mit den Ergebnissen aus Tabelle 8 stellen wir fest, dass die Fragmentierungstiefen im selben Intervall liegen. Lediglich in Tabelle S-7 ist die Fragmentierungstiefe bis 19 erreicht. Dieser Wert liegt wiederum nur um eins höher als bei den beiden anderen Experimenten. Wir schließen daher darauf, dass das Intervall der Fragmentierungstiefen sich bei wechselnden Parametern kaum verändert. Ebenfalls können wir unsere Annahme bestätigen, dass zwischen fünf und zehn die meisten Moleküle ihre maximale Fragmentierungstiefe erreichen. Auch in den Tabellen S-7 und S-8 weißt die Fragmentierungstiefe sieben die größte Anzahl auf. Die Fragmentierungstiefe sieben ist damit ein angemessener Kompromiss zwischen der Laufzeit und der Möglichkeit weitere kleine Fragment-Ionen zu bestimmen.

4.5.2. Anwendbarkeit der Regeln

Nachdem wir zuvor auf Basis des CASMI-Datensatzes die optimale Fragmentierungstiefe bestimmt haben, legen wir nun den Fokus auf die Anwendbarkeit der implementierten Regeln. Wie aus dem Kapitel 4.1.2 bekannt, enthält `ChemFrag` 51 Umlagerungs- und Spaltungsregeln.

Experiment 4.6: Das Experiment hat das Ziel zu bewerten, welche Regeln besonders häufig zur Anwendung kommen und welche zur Erläuterung der Fragment-Ionen verwendet werden. Dazu analysieren wir zunächst die Anwendung auf dem gesamten CASMI-Datensatz, indem `ChemFrag` für jedes Molekül des Datensatzes zählt, wie häufig eine Regel während der Fragmentierung ausgeführt wird.
Um die Häufigkeit der Regeln im ersten Experiment zu betrachten, unterscheiden wir zwischen den Umlagerungsregeln und den verschiedenen Spaltungsregeln. Bei den Umlagerungsregeln (Tabelle 10) wurden am meisten der Hydrid-Shift und der Protonen-Shift mit 29.309 bzw. 19.349 Aufrufen angewendet. Mit 9.903 Aufrufen hat `ChemFrag` die OCC-Regel am dritthäufigsten aufgerufen. Die wenigsten Aufrufe (5) verzeichnet die Sulfide Zyklisierung, wohingegen die Regeln COCreation, CN-Rearr PericyclicShift und ThionoThioloRearr keine Anwendung fanden. Betrachten wir nun im nächsten Schritt die Spaltungsregeln (Tabelle 9). Am häufigsten wendet `ChemFrag` die homolytische Spaltung an (24.076). Mit 13.603 Matchings wurde die Abspaltung von Wasserstoff (H_2) am zweithäufigsten angewendet. Die beiden Spezialumlagerungen, zur Vermeidung instabiler Moleküle, mit Abspaltung von Wasser und einer Hydroxylgruppe kamen fast gleich häufig mit 9.771 bzw. 7.152 vor. Am seltensten rief `ChemFrag` die Regel zur SO2 Entfernung (5 mal) auf. Kein Matching erfolgte bei den Regeln ReduC2H2Aro, QinoneCleavage, AmmoniumRemoval, OniumRule, EsterRule, RetroeneReaction, SOCleavQinone, BenzylRule.

Tabelle 9: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Spaltungsregeln für Parametersatz 1. Rot markiert die drei häufigsten Spaltungsregeln (ausgenommen der Spezialregeln).

Regel	Matchings	Vorkommen in Fragment-Ionen
homolytic cleavage:	24076	25534
RemH2General:	13603	6632
heterolytic cleavage:	3955	3973
COReduction:	2459	8393
ChlorCleavage:	1921	1139
LCleavage:	1275	249
RemovalHrad:	537	151
AllylRule:	343	13
EthenRemoval:	279	3764
InductiveCleavage:	277	14
McLaffertyRearrangement:	248	264
CarboxylCleavage:	171	550
AromaticElimination:	154	24
OrthoRemoval:	137	384
RDAReaction:	130	70
RemH2_C2C4:	111	0
OriginH20:	101	0
NHTransition:	67	9
PhosphoRemoval:	47	0
H2OSulfonCleavage:	9	33
CONCleavage:	8	0
SOCleavage:	8	0
SO2Removal:	5	11
ReduC2H2Aro:	0	0
QinoneCleavage:	0	
AmmoniumRemoval:	0	0
OniumRule:	0	0
EsterRule:	0	0
RetroeneReaction:	0	0
SOCleavQinone:	0	0
BenzylRule:	0	0
RemOH:	9771	12155
RemoveH2O:	7152	11466

Tabelle 10: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Umlagerungsregeln für Parametersatz 1. Rot markiert die drei häufigsten Regeln.

Regel	Matchings	Vorkommen in Fragment-Ionen
Hydrid-Shift	29309	9852
ProtonShift	19349	26449
OCCRearr:	9903	2833
HShiftRad:	2434	668
NCCProtShift:	2278	0
CCFeElRearr:	1294	1414
CORearr:	1103	0
NposRad:	789	0
AllylShift:	775	131
CNFeElRearr:	717	70
Cyclisation:	105	0
COFeElRearr:	14	0
SulfideCyclisation:	5	0
ThionoThioloRearr:	0	0
CNRearr:	0	0
AnionBackw:	0	0
PericyclicShift:	0	0
COCreation:	0	0
AnionRearr:	0	0

An dieser Stelle ist es notwendig darauf hinzuweisen, dass ein Aufruf einer Regel nicht gleichbedeutend mit der Beteiligung zur Annotation eines Fragment-Ions ist. Beispielsweise können auf dem Fragmentierungsweg Fragment-Ionen entstehen, die keinem Peak im Massenspektrum entsprechen.

Nachfolgend untersuchen wir deshalb, welche Regeln bei der Erläuterung der Fragment-Ionen beteiligt sind. Da `ChemFrag` für einen Peak verschiedene energetisch stabile Fragment-Ionen, beispielsweise durch Umlagerungen, vorhersagen kann, kann die Anzahl der verwendeten Regeln in den Fragment-Ionen zur Annotation höher sein als die Anzahl der Matchings. Zum Beispiel kommt es nach der Abspaltung von Kohlenstoffmonoxid (COReduction) meist zu einem Protonen- oder Hydrid-Shift. Damit

wird die Regel zur COReduction zwar nur ein Mal angewendet, jedoch entstehen durch die Umlagerungen weitere Fragment-Ionen mit der gleichen Masse.

Bei der Auswertung der angewendeten Regeln für die Annotation der Fragment-Ionen unterscheiden wir wieder zwischen den Umlagerungs- und den Spaltungsregeln. Bei den Umlagerungsregeln sind die Protonen-Shifts am meisten (26.449) bei der Erläuterung der Fragment-Ionen vertreten. Am zweithäufigsten sind die Hydrid-Shifts (9.852) bei den Annotationen vermerkt. Auffallend ist hierbei, dass zwar 29.309 Matchings für den Hydrid-Shift auftraten, jedoch nur 9.852 bei den Aufklärungen von Fragment-Ionen beteiligt sind. Mit 2.833 Vorkommen ist die Umlagerung OCC am dritthäufigsten bei Fragment-Ionen vorkommend. Wir erkennen daran, dass die drei Regeln, die am häufigsten gematcht wurden, auch bei der Aufklärung der Fragment-Ionen am häufigsten beteiligt sind. Weiterhin ist zu benennen, dass bei den NC-CProtShift, CORearr, NPosRad, Cyclisation, COFeElRearr und SulfideCyclisation zwar Matchings vorlagen, diese Regeln aber nicht in der Erläuterung von Fragment-Ionen vorkamen. Besonders auffallend ist auch, dass Umlagerungsregeln wie HShift-Rad 2.434 Matchings hatten, jedoch nur 668 in den Annotationen auftraten.

Ähnliches erkennen wir auch bei den Spaltungsregeln. Die homolytische Spaltung ist am häufigsten mit 25.534 Vorkommen bei den Fragment-Ionen vertreten. Am zweithäufigsten (8.393) kommt die COReduction vor. Diese lag bei den Matchings (2.459) noch auf Platz vier. Mit 6.632 Vorkommen ist die RemH2General Regel bei der Aufklärung der Fragment-Ionen beteiligt. Wie bei den Umlagerungen existieren auch hier Regeln, bei denen zwar ein Matching vorkam, die jedoch nicht bei der Annotation der Fragment-Ionen beteiligt sind. Dazu zählen RemH2_C2C4, OriginH20, PhosphoRemoval, CONCleavage, SOCleavage. Die beiden Spezialregeln, zur Vermeidung instabiler Fragment-Ionen waren mit 12.155 (RemOH) und 11.466 (RemoveH2O) bei der Erklärung der Fragment-Ionen beteiligt. Damit zeigt sich besonders bei den Spezialregeln deren hohe Wichtigkeit.

Vergleichen wir diese Ergebnisse mit den Tabellen S-9 bis S-12, in denen `ChemFrag` mit Parameterkombinationen 2 und 3 die Fragmentierung bestimmt hat, erkennen wir, dass es sich stets um die selben drei häufigsten Regeln handelt. Lediglich bei den Spaltungsregeln mit dem Parametersatz 3 (Tabelle S-12) tritt RemH2General häufiger in den Fragment-Ionen auf als die heterolytische Spaltung. Der Unterschied beträgt lediglich 16. Daraus ziehen wir die Erkenntnis, dass bei verschiedenen Parametereinstellungen für den CASMI-Datensatz die Regeln der homolytischen und heterolytischen Spaltung, die Abspaltung von Wasserstoff sowie Kohlenmonoxid zu den wichtigsten Spaltungsregeln gehören. Ebenso zählen der Hydrid-Shift, der Protonen-

Shift und OCC Umlagerung zu den wichtigsten Umlagerungsregeln.

Dieses Experiment basierte bisher auf dem gesamten Datensatz. Jedoch kann es vorkommen, dass einzelne Stoffgruppen nur wenig im Datensatz vertreten sind. Das führt dazu, dass auch die Spezialregeln dieser Gruppen wenig Anwendung finden. Daher unterteilen wir für die nächste Auswertung den Datensatz in strukturelle chemische Gruppen. Der Datensatz gliedert sich in folgende 11 Gruppen:

1) Strukturen mit einfach gebundenem Schwefel
2) Strukturen mit zweifach gebundenem Schwefel
3) Strukturen, mit sowohl einfach, als auch zweifach gebundenem Schwefel oder Phosphor
4) Carbonsäuren
5) Strukturen mit Ester-Gruppen
6) Strukturen mit 6-Ring-System
7) Strukturen aus langkettigen Kohlenstoffen (optional einzelne Stickstoffatome)
8) Strukturen mit Aldehyd/Keton-Gruppen
9) Strukturen mit Alkohol-Gruppen
10) Strukturen mit Ether-Gruppen
11) Strukturen bestehend aus Ether-Gruppen und Alkohol- oder Aldehyd-Gruppen oder Strukturen bestehend aus Alkohol- und Aldehyd-Gruppen

Wir vergleichen nun für alle Gruppen die Matchings der Regeln mit denen des gesamten Datensatzes. Hierbei stellen wir fest, dass bei allen Umlagerungsregeln die Regeln Hydrid-Shift, Proton-Shift, OCCRearr am häufigsten auftreten. Dabei tritt der Hydrid-Shift bis auf bei Gruppe 1 am häufigsten auf. In Gruppe 1 matchte `ChemFrag` die OCCRearr Umlagerung am häufigsten (OCCRearr 795, Hydrid-Shift 756). Bei den Gruppen 2, 4-7, 9-11 ist der Protonen-Shift die zweithäufigste Umlagerung. Für die Gruppe 3 und 8 ist die OCCRearr Umlagerung am zweithäufigsten. Die häufigere Anwendung der OCCRearr Umlagerung liegt an der Struktur der Gruppen, wie wir beispielhaft an der Struktur in Abbildung 24 sehen.

Beispielsweise kommt es in der Gruppe der Aldehyde/Ketone sehr häufig zu einer Protonierung des Sauerstoffatoms. Ist dieses dann mit einem Kohlenstoffatom verbunden, welches wiederum eine Doppelbindung zu einem weiteren Kohlenstoff besitzt, bildet `ChemFrag` eine Doppelbildung zwischen dem Sauerstoffatom und dem

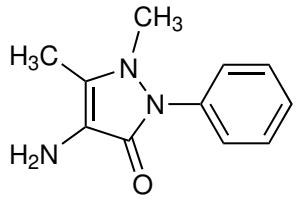


Abbildung 24: Darstellung einer Keton-Struktur aus der Gruppe 8 des CASMI-Datensatz.

Kohlenstoffatom. Dadurch entsteht ein energetisch stabiles Fragment. Die Doppelbindung zwischen den Kohlenstoffatomen wandelt `ChemFrag` in eine Einfachbindung um. Dadurch wandert die positive Ladung vom Sauerstoffatom zu dem einfach gebundenen Kohlenstoffatom.

Für die Spaltungsregeln ist zu beobachten, dass für alle Gruppen die homolytische Spaltung am häufigsten und die Abspaltung von Wasserstoff (RemH2General) am zweithäufigsten angewendet wurde. Abhängig von der Gruppe matchte `ChemFrag` die Regeln der heterolytischen Spaltung (Gruppe 1-3, 6-11) und die COReduction (Gruppe 4-5) am dritthäufigsten. Zu vermerken ist ebenfalls, dass je nach Gruppe unterschiedliche Regeln häufiger benutzt werden. So werden in den Gruppen 1-3 spezielle Regeln zur Spaltung und Umlagerung von Schwefelverbindungen angewendet. Dazu zählen H2OSulfonCleavage (9), SOCleavage (8), SO2Removal (5), SOCleavQinone (4) und SulfideCyclisation (5). Weiterhin kommt die Carboxylcleavage (141 in Gruppe 4) und EthenRemoval (169 in Gruppe 5) häufig zum Einsatz, in den Strukturen, wo diese Substrukturmerkmale vorhanden sind. In den Gruppen 8 und 11 matcht `ChemFrag` die Regel zur COReduction verhältnismäßig häufig. Das bedeutet in der Gruppe 8 306 mal und in Gruppe 11 315 mal.

Anhand dieser Analyse ist es `ChemFrag` nun möglich, im Vorfeld Regeln zu selektieren, die zur Aufklärung der Fragment-Ionen notwendig sind. Dafür wurde in `ChemFrag` nach diesem Experiment eine Gruppierung der Regeln eingeführt. So sind beispielsweise Regeln, zum Protonen-Shift, Hydrid-Shift, Zyklisierung oder auch Wasserstoffabspaltung in einer allgemeingültigen Gruppe zusammengefasst. Spezifische Gruppen mit Regelsätzen gibt es für Phosphor, Schwefel oder auch Anionen-Regeln. Nach dem Einlesen des SMILES überprüft `ChemFrag`, welche Elemente das Molekül enthält und wählt anhand dessen die Gruppen zur Regelanwendung aus. Dadurch verhindert `ChemFrag` das mehrmalige Überprüfen chemischer Regeln, die aufgrund der Elementzusammensetzung keine Anwendung finden können.

4.5.3. Scoring und Laufzeit am CASMI-Datensatz

Nach der vorherigen Analyse der Regelanwendung, werden wir uns in diesem Unterkapitel mit der Bewertung der Annotation der Fragment-Ionen durch `ChemFrag` auseinander. Ziel ist das Ergebnis der Fragmentierung in einem geeigneten Zahlenwert widerzuspiegeln. Zusätzlich möchten wir in die Auswertung noch die Laufzeit mit einfließen lassen, um zu bestimmen, welche Parameterkombination ein Optimum an aufgeklärten Fragment-Ionen in akzeptabler Laufzeit ermöglicht. Beginnen werden wir mit der Bewertung der Scoringfunktionen aus Kapitel 4.1.4.

Experiment 4.7: Die Gleichungen 2 und 3 aus Kapitel 4.1.4 zeigten verschiedene Berechnungsmöglichkeiten des Scores eines Annotationsprozesses. Mit der Gleichung 2 wird die absolute Anzahl erklärter Fragment-Ionen berechnet. Im Gegensatz dazu inkludiert die Gleichung 3 Informationen über die Intensität der Fragment-Ionen aus dem Massenspektrum. Für dieses Experiment berechnen wir den absoluten und den gewichteten Score für die 11 Gruppen des CASMI-Datensatzes, ohne die Fragmentierungstiefe zu beschränken. Dabei lief das Experiment jeweils mit den Parameterkombinationen 1 bis 3. Tabelle 11 stellt die beiden berechneten Scores gegenüber.
Aus der Tabelle erkennen wir, dass `ChemFrag` mit der Parameterkombination 1 die wenigsten Fragment-Ionen und mit der Parameterkombination 3 die meisten Fragment-Ionen annotieren kann. Vergleichen wir die Zahlen des absoluten Scores können durch die Parameterkombination 1 54 %, mit der Parameterkombination 2 62 % und mit der Kombination 3 68 % aufgeklärt werden. Das scheint im ersten Eindruck ein deutlicher Unterschied zu sein. Wie wir aber bereits diskutiert haben, ist es für die Aufklärung besonders sinnvoll die Fragment-Ionen mit hoher Intensität aufzuklären, da sie während der Fragmentierung häufig gebildet werden. Daher betrachten wir nun den gewichteten Score. Auch hier annotiert `ChemFrag` mit der Parameterkombination 1 die wenigsten Fragment-Ionen. Betrachten wir jedoch den prozentualen Anteil, ist der Unterschied nicht mehr so groß, wie bei dem absoluten Score. Für die Parameterkombination 1 klärt `ChemFrag` 68 %, für die Parameterkombination 2 73 % und für die Parameterkombination 3 77 % auf. Der prozentuale Unterschied von 68 % zu 73 % oder zu 73 % zu 77 % ist weniger stark als zwischen 54 % und 62 % oder 62 % und 68 %. Auch erkennen wir durch die Scores, dass `ChemFrag` mehr als die Hälfte der Fragment-Ionen annotiert und, dass die hohen Fragment-Ionen mit Intensitäten höhere Bedeutung haben. Daher können wir aus diesem Experiment als eine Schlussfolgerung ziehen, dass die gewichteten Score die tatsächliche Annotation besser bewerten als die absoluten Scores und zum anderen, dass `ChemFrag` eine gute Annotation für diesen strukturell diversen Datensatz liefert.

Tabelle 11: Vergleich des absoluten und des gewichteten Score für den CASMI-Datensatz mit Unterteilung in die 11 Strukturgruppen. Die Scores wurden für die Fragmentierungen auf Basis der drei Parameterkombinationen aus Tabelle 7 berechnet. Die absoluten Scores sind in Klammern geschrieben.

Gruppe	Moleküle je Gruppe	Kombination 1	Kombination 2	Kombination 3
1	29	3184/4742 (50/96)	3237/4742 (61/96)	3577/4742 (66/96)
2	15	1463/3663 (29/96)	1791/3663 (43/96)	1716/3663 (42/96)
3	13	1257/2820 (25/84)	1235/2820 (28/84)	1433/2820 (34/84)
4	27	3829/4984 (81/118)	4112/4984 (86/118)	4213/4984 (90/118)
5	23	2814/5005 (62/137)	3311/5005 (83/137)	3778/5005 (96/137)
6	49	5983/7664 (98/155)	6486/7664 (116/155)	6657/7664 (125/155)
7	18	2193/2925 (37/67)	2231/2925 (36/67)	2274/2925 (40/67)
8	32	4012/5548 (71/127)	4321/5548 (83/127)	4591/5548 (92/127)
9	10	1168/1768 (20/37)	1394/1798 (26/37)	1425/1798 (29/37)
10	8	923/1355 (15/30)	1046/1355 (18/30)	1089/1355 (21/30)
11	26	5198/6507 (140/202)	5254/6507 (143/202)	5541/6507 (148/202)
Gesamt	**250**	**32054/47011 (626/1150)**	**34418/47011 (723/1150)**	**36294/47011 (783/1150)**

Experiment 4.8: Nach den Erörterungen zum Scoring legen wir nun das Augenmerk auf die Laufzeit von `ChemFrag`. Dafür messen wir die Laufzeit für die 11 Gruppen des CASMI-Datensatzes, ohne Beschränkung der Fragmentierungstiefe, insgesamt drei mal, wobei stets eine andere Parameterkombinationen aus Tabelle 7 verwendet wird. Der Vergleich der Laufzeit mit dem Score soll eine Abschätzung ermöglichen, inwieweit sich eine veränderte Parameterkombination positiv auf die Anzahl annotierter Fragment-Ionen auswirkt und ob die dazugehörige Laufzeit in

Tabelle 12: Vergleich der Laufzeit mit den absoluten Scores auf dem CASMI-Datensatz unter Berücksichtigung der drei unterschiedlichen Parameterkombinationen aus Tabelle 7. Die Ergebnisse sind in die strukturbezogenen Gruppen unterteilt. Die Zeiten sind in Stunden angegeben und in Klammern stehen die absoluten Scores.

Gruppe	Moleküle je Gruppe	Kombination 1	Kombination 2	Kombination 3
1	29	1,5 (50/96)	3,4 (61/96)	4,2 (66/96)
2	15	0,8 (29/96)	3,0 (43/96)	4,0 (42/96)
3	13	0,6 (25/84)	1,7 (28/84)	2,4 (34/84)
4	27	2,9 (81/118)	7,6 (86/118)	11,8 (90/118)
5	23	2,7 (62/137)	6,9 (83/137)	9,7 (96/137)
6	49	3,2 (98/155)	7,3 (116/155)	10,1 (125/155)
7	18	0,7 (37/67)	1,8 (36/67)	2,1 (40/67)
8	32	1,5 (71/127)	3,3 (83/127)	5,0 (92/127)
9	10	2,4 (20/37)	2,6 (26/37)	2,5 (29/37)
10	8	0,6 (15/30)	2,1 (18/30)	3,7 (21/30)
11	26	3,6 (140/202)	8,1 (143/202)	11,1 (148/202)
Gesamt	**250**	**20,5h (626/1150)**	**47,8h (723/1150)**	**66,6h (783/1150)**

einem vertretbaren Bereich liegt. Tabelle 12 zeigt die Laufzeiten aus den drei Parameterkombination und zusätzlich die Summe der absoluten Scores für jede Gruppe.

Wir erkennen aus der Tabelle 12, dass ein direkter Zusammenhang zwischen einer höheren Laufzeit und der Anzahl erklärter Scores besteht. Betrachten wir den Unterschied zwischen Parameterkombination 1 und Parameterkombination 2, handelt es sich dort nur um die Änderung des Parameters κ für die Bindungsordnung. Diese wurde von 0.08 auf 1.00 erhöht. Das führt dazu, dass mehr Bindungen durch homo -oder heterolytische Spaltung gespalten werden. Die Folge sind mehr gebildete Fragment-Ionen, die sowohl im Bereich der Fragment-Ionen als auch dann bei den umgelagerten Fragment-Ionen entstehen. Diese höhere Anzahl an zu berechnenden Fragment-Ionen führt zu mehr als einer Verdopplung der Laufzeit und nur zur Erklärung von 97 zusätzlichen Fragment-Ionen. Vergleichen wir nun dazu die Ergebnisse aus Parameterkombiantion 1 und Parameterkombination 3 sehen wir, dass es zu einem dreifachen Anstieg der Laufzeit kam und 157 mehr erklärten Fragment-Ionen. Hier wurde der Parameter ϵ für das Reaktionsintervall verändert. Eine Erhöhung um

jeweils 50 kJ/mol führt zu einer höheren Anzahl annotierter Fragment-Ionen (14 % im absoluten Score), die im nächsten Fragmentierungsschritt zulässig sind. An dieser Stelle muss diskutiert werden, ob die Steigerung der erklärten Fragment-Ionen der deutlichen Erhöhung der Laufzeit im Nutzen steht. Stellen wir die Erkenntnisse aus Parameterskombination 2 und 3 gegenüber, ist es lohnenswerter das Reaktionsintervall zu erhöhen, da das Verhältnis zwischen erhöhter Laufzeit und der Anzahl annotierter Fragment-Ionen ein besseres ist als bei der Erhöhung der Bindungsordnung. Wir schlussfolgern daher für die Einstellung von `ChemFrag`, dass 0.08 für den Parameter κ ein geeigneter Wert ist. Um mehr Fragment-Ionen zu annotieren, ist die Veränderung des Parameters ϵ am sinnvollsten. Für den Nutzer empfiehlt es sich zunächst mit der Parameterkombination 1 zu starten und wenn eine Verbesserung der erklärten Fragment-Ionen notwendig ist auf die Parameterkombination 3 zu wechseln, da dort eine deutliche Steigerung der Anzahl annotierter Fragment-Ionen möglich ist.

4.5.4. Zusammenfassung auf Basis des CASMI-Datensatzes

Im ersten Schritt bestimmten wir für die optimale Fragmentierungstiefe den Wert sieben. Weiterhin zeigten die Experimente, dass der gewichtete Score eine bessere Bewertung eines Massenspektrums ermöglicht. Die Einbeziehung der Intensität ist von großem Vorteil, da besonders die Fragment-Ionen mit hoher Intensität aufgeklärt sein sollten. Abhängig von der Gerätekonfiguration ist bei einer minimalen Intensität unklar, ob es sich um ein Fragment-Ion oder um Rauschen handelt. Weiterhin konnten wir schlussfolgern, dass die Parameterkombination 1 aus Tabelle 7 eine sinnvolle Kombination ist, um eine Anfangsfragmentierung zu generieren, auf der weitere Optimierungen möglich sind. Durch diese Parameter entsteht eine hohe Aufklärungsrate in einer angemessenen Laufzeit. Auf Basis dieser Parameterkombination zeigten sich die Regeln der homo- und heterolytischen Spaltung, die Abspaltung von Kohlenstoffmonoxid und Wasserstoff sowie die Spezialregeln von hoher Anwendung. Bei den Umlagerungsregeln weisen der Hydrid-Shift und Proton-Shift sowie die Umlagerung einer Sauerstoff-Kohlenstoff-Kohlenstoff-Kette hohe Aufrufe und Beteiligungen an den annotierten Fragment-Ionen auf.

4.6. Vorhersage neuer Fragmentierungswege für Naturstoffe

In diesem abschließenden Experimentekapitel möchten wir die erlernten Parameter auf weitere Stoffgruppen anwenden. Dabei beziehen wir uns auf ESI-MS/MS-Spektren unter anderem für verschiedene Naturstoffe, die aus der Arbeitsgruppe von Prof. Csuk am Institut für Chemie der Martin-Luther-Universität stammen. Es handelt sich zum einen um die Stoffgruppe der Steroide, die sich durch anel-

lierte Ringsysteme strukturell auszeichnen. Zum anderen wenden wir `ChemFrag` auf Kombinationen von Kohlenwasserstoffen an. Diese kennzeichnen sich durch längere Kohlenwasserstoffketten mit vereinzelten Stickstoff- oder Sauerstoffatomen sowie maximal einem Benzenring aus. Für diese verschiedenen Substanzklassen werden wir einige ausgewählte Analysen zur Annotation der Massenspektren durchführen. Die Grundlage der Ergebnisse lieferten die Bachelorarbeit von Pauline Walesch [77] und die Masterarbeit von Antonia Schmidt [78].

4.6.1. Annotation von Fragment-Ionen der Steroide

Für die Substanzklasse der Steroide stehen uns folgende zehn Substanzen zur Verfügung: α-Hydroxyestrondiacetat, Equilin-3-acetat, 9(11)-Dehydro-Estron, Estriol-3-methylether, Estriol-3-acetat, Estriol-3-methylether, Dehydrocyanomethyl-Estradiol, Cyananomethyl-estradiol, Estriol-16-acetat und 17-β-Estradiol. Die genannten Steroide zeichnen sich durch eine Kombination von 5- und 6-Ringen aus. Aufgrund der Verknüpfung der Ring-Strukturen ist eine chemisch sinnvolle Aufklärung der Fragment-Ionen anspruchsvoll [79]. `ChemFrag` besitzt dazu im regelbasierten Ansatz Implementierungen zur Spaltung und Umlagerung der Ringstrukturen. Diese wären durch einen quantenchemischen Ansatz kaum zu simulieren. Das erste Ziel ist daher nach einer umfangreichen Implementierung neuer Regeln die Fragmentierung eines Steroids mit dem zugehörigen publizierten Fragmentierungsweg zu vergleichen. Dafür wenden wir `ChemFrag` auf 17-β-Estradiol an. Im zweiten Schritt bewerten wir die Ergebnisse von `ChemFrag` an den übrigen Steroiden.

Beispiel 3: 17-β-Estradiol

Die chemische Substanz 17-β-Estradiol, auch als Estradiol oder Östradiol bekannt, ist ein körpereigenes Hormon. Bei Frauen wird diese Form des Estradiols in großer Konzentration in den Eierstöcken produziert. Im Vergleich dazu bilden die Nebennierenriede und die Hoden beim Mann geringe Konzentrationen aus. 17-β-Estradiol besitzt die stärkste Wirkung der Estradiole im Körper und ist an der Auslösung des Eisprungs und am Aufbau der Gebärmutterschleimhaut beteiligt [80]. Aus der Publikation von Ma *et al.* im Jahr 2018 ist die Fragmentierung von 17-β-Estradiol aufgeklärt [81] (Abbildung S-14). Sie lehnt sich an die Erkenntnisse von Bourcier *et al.* aus dem Jahr 2010 an [82].

Die Fragmentierung, mit Parameterkombination 1, von 17-β-Estradiol durch `ChemFrag` basiert auf dem SMILES `CC12CCC3C(C1CCC2O)CCC4=C3C=CC(=C4)O` und der Liste der Fragment-Ionen aus der MassBank mit dem Index AU279702. Die Abbildung 25 zeigt den ermittelten Fragmentierungsweg durch `ChemFrag`.

Bei der Fragmentierung fällt im ersten Schritt die Protonierung der Hydrodoxylgruppe (E1) auf. Diese führt, wie in den früheren Kapiteln beschrieben, zur Abspaltung

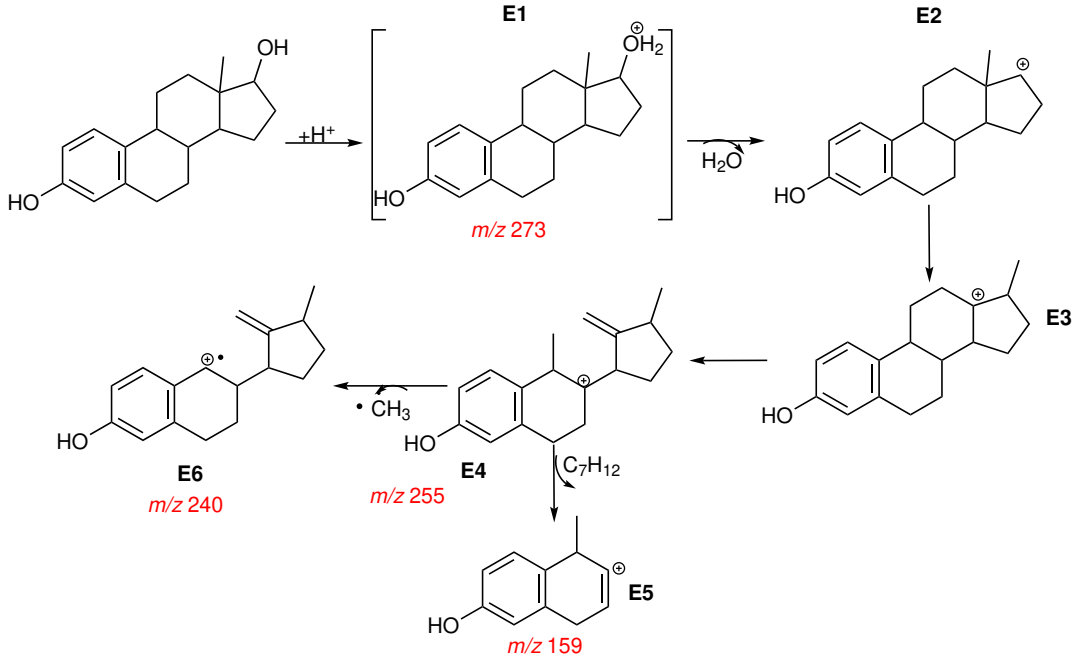


Abbildung 25: Darstellung des Fragmentierungsweges von 17-β-Estradiol [78].

von Wasser (E2). Aufgrund dieser Abspaltung kann es vorkommen, dass das protonierte Molekül nur mit einer geringen Intensität im Massenspektrum erfasst wurde. Besonders charakteristisch für die Fragmentierung von Estradiol ist die Verschiebung der Methylgruppe von dem verbindenden Kohlenstoffatom zwischen dem 5-Ring und 6-Ring zum Kohlenstoffatom des 5-Rings, zu sehen von E2 zu E3. `ChemFrag` simuliert diese Verschiebung durch eine Regel im regelbasierten Ansatz, die nach einer Literaturrecherche eingefügt wurde. Es handelt sich dabei um eine Umlagerung von einem sekundären Carbenium-Ion (E2) zu einem teritären Carbenium-Ion (E3). Diese charakteristische Umlagerung unterscheidet die Ergebnisse der Fragmentierung von `ChemFrag` mit denen aus `MetFrag` und `CFM-ID`. Nach Anwendung der Regel ist das verbindende Kohlenstoffatom positiv geladen und erläutert das Fragment-Ion E3 mit einer Masse von 255 Da. Die Position der positiven Ladung führt zu einer heterolytischen Spaltung, die eine Ringöffnung bewirkt und das Fragment-Ion E4 gebildet wird. Aufgrund der Masse des Fragment-Ions E4 kann `ChemFrag` so, den Peak m/z 255 erklären. Davon ausgehend können nun zwei Spaltungen möglich sein. Entweder es kommt zur Abspaltung eines Methyl-Radikals (E6), welches das Fragment-Ion m/z 240 annotiert oder zur Abspaltung des 5-Rings. Nach dieser Abspaltung bildet sich das Naphthalenol-Derivat (E5), welches mit m/z 159 das Fragment-Ion aufklären kann.

Die Annotation durch `ChemFrag` ähnelt stark den Ergebnissen aus Ma *et al.* (Ab-

Tabelle 13: Auflistung der gewichteten und in Klammern gesetzt die absoluten Scores für die neun Steroide im Vergleich von `ChemFrag`, `MetFrag` und `CFM-ID` [78].

Molekül	`ChemFrag`	`MetFrag`	`CFM-ID`
α-Hydroxyestrondiacetat	205/219 (5/7)	158/219 (5/7)	55/219 (1/7)
Equilin-3-acetat	180/188 (4/5)	166/188 (4/5)	14/188 (1/5)
Estriol-3-methylether	302/384 (10/15)	202/384 (9/15)	100/384 (1/15)
9(11)-Dehydro-Estron	134/234 (4/5)	140/234 (4/5)	125/234(3/5)
Estriol-3-acetat	119/227 (4/6)	181/227 (4/6)	38/227 (1/6)
Estriol-3-methylether	123/225 (4/7)	34/225 (4/7)	100/225 (1/7)
Dehydrocyanomethyl-Estradiol	326/346 (5/7)	226/346 (4/7)	100/346 (1/7)
Cyananomethyl-estradiol	285/325 (5/9)	213/325 (6/9)	100/325 (1/9)
Estriol-16-acetat	164/164 (4/4)	152/164 (3/4)	12/164 (1/4)
Gesamt-Score	1838/2312 (45/65)	1472/2312 (43/65)	644/2312 (11/65)

bildung S-14). Besonders die implementierten Regeln zur Umlagerung der Methylgruppe sowie die Öffnung des 6-Rings zeigen hier eine große Ähnlichkeit zwischen `ChemFrag` und der Literatur. Wir konnten damit an einem Beispiel nachweisen, dass `ChemFrag` chemisch sinnvolle Annotationen vorhersagen kann und für die Vorhersage von Fragmentierungswegen von Steroiden geeignet sein könnte.

4.6.1.1. Erläuterungen zu den weiteren Steroid Varianten Auf Basis der Erkenntnisse von 17-β-Estradiol ist das sich anschließende Ziel `ChemFrag`‘s Eignung auf einer größeren Datengrundlage zu evaluieren. Daher wenden wir `ChemFrag` auf die verbliebenen neun Steroide an. `ChemFrag` bestimmt die Fragmentierung mit der Parameterkombination 1 und einer Fragmentierungstiefe von sieben. Den Score für das Experiment berechnen wir mit den Gleichungen 2 und 3. Die Tabelle 13 listet sowohl die absoluten als auch die gewichteten Scores der neun Steroid-Varianten auf, die durch die Annotation mittels `ChemFrag`, `MetFrag` und `CFM-ID` erzielt wurden. In der Analyse zeigt sich, dass `ChemFrag` 45 der 65 gemessenen Fragment-Ionen für den gesamten Datensatz annotiert. Vergleichend dazu erklärte `MetFrag` 43 und CFM-ID 11 der Fragment-Ionen.

Anhand der Scores der drei Programme erkennen wir, dass `ChemFrag` die meisten Fragment-Ionen unter Einbeziehung der Intensität aufklärt. Für `MetFrag` sehen wir,

dass der Score um etwa 400 niedriger ausfällt, was der fehlenden Protonierung des Moleküls geschuldet sein könnte. Wenn das protonierte Molekül eine hohe Intensität hat, schlägt sich diese Nicht-Erläuterung im Score deutlich nieder. Vergleichend dazu ist der Score von `ChemFrag` annähernd dreifach so hoch, wie der von `CFM-ID`. Hier sehen wir auch in den absoluten Scores, dass meist nur ein, maximal drei, Fragment-Ionen annotiert werden konnten. Wir schlussfolgern daher aus diesem Experiment, dass `ChemFrag` die Fragment-Ionen in der Gesamtanzahl gleichwertig annotieren kann wie `MetFrag` und besser als `CFM-ID`. Vergleichen wir zusätzlich die generierten Strukturen zeichnet sich auch hier ab, dass `ChemFrag` chemisch sinnvollere Ergebnisse erzielt als beispielsweise `CFM-ID`. Beispielsweise bei 9(11)-Dehydroestron generiert `CFM-ID` zwei Fragment-Ionen mit einem positiv geladenen Wasser [78].

Aus der Tabelle 13 sehen wir weiterhin, dass `ChemFrag` für die Moleküle Estradiol-3-methylether und Dehydrocyanomethylestradiol sehr gute Ergebnisse erzielt. Eine Recherche nach publizierten Fragmentierungswegen ergab keine Ergebnisse. Daher ermöglichen wir es erstmals auf Basis von `ChemFrag` einen Fragmentierungsweg für diese Strukturen vorherzusagen. Die Abbildungen S-11 und S-12 im Anhang visualisieren die vorhergesagten Fragmentierungswege. Wir können damit hervorheben, dass `ChemFrag` für die Vorhersage neuer Fragmentierungswege eine Verbesserung ist und die Basis für eine chemisch plausible Annotation liefern kann.

4.6.2. Annotation von Fragment-Ionen langkettiger Kohlenwasserstoffe

Nach der Anwendung auf die Steroide betrachten wir abschließend eine weitere Substanzklasse der Naturstoffe. Es handelt sich um langkettige Kohlenwasserstoffe, die mit mindestens einem Benzenring verknüpft sind. Für die folgenden acht Substanzen berechnete `ChemFrag` die Fragment-Ionen: *N*-Ethylnicotinamid, 2-Cyano-2-phenylbutansäureetylester, 2-Cyano-3-methylhexansäureethylester, Nikotinamid, 2,4-Diamino-6-hydroxylmethylpteridine, Gluconsäurephenylhydrazid, Moxonidin, Hippursäuremethylester. Auf Basis der Parameterkombination 1 berechnete `ChemFrag` die Fragmentierung.

Tabelle 14 vergleicht auch hier wieder die Anzahl annotierter Fragment-Ionen zwischen `ChemFrag` und `MetFrag`. Ein Vergleich mit CFM-ID 3.0 war nicht möglich, da der Webserver keine Annotation der Fragment-Ionen ermöglichte.
Rechnerisch erkennen wir, dass `ChemFrag` vier Fragment-Ionen mehr annotierten kann als `MetFrag`. Die Verbesserung von `ChemFrag` in der Aufklärung der Fragment-Ionen bemerken wir besonders im gewichteten Score, wo `ChemFrag` einen Score von 1476/1558 im Vergleich zu `MetFrag` mit 1061/1558 hat. Dieser höhere Score lässt sich wieder durch die Annotation von Fragment-Ionen mit hoher Intensität sowie

Tabelle 14: Auflistung der gewichteten und in Klammern gesetzt die absoluten Scores für die acht Strukturen im Vergleich von `ChemFrag` und `MetFrag`.

Molekül	`ChemFrag`	`MetFrag`
N-Ethylnicotinamid	165/165 (4/4)	113/165 (3/4)
2-Cyano-2-phenylbutansäureetylester	153/153 (3/3)	105/153 (2/3)
2-Cyano-3-methylhexansäureethylester	194/214 (4/5)	30/214 (2/5)
Nikotinamid	195/195 (4/4)	155/195 (3/4)
2,4-Diamino-6-hydroxylmethylpteridine	192/216 (3/4)	185/216 (3/4)
Gluconsäurephenylhydrazid	253/260 (8/9)	233/260 (8/9)
Moxonidin	136/152 (3/5)	121/152 (4/5)
Hippursäuremethylester	203/203 (4/4)	119/203 (3/4)
Gesamt-Score	1476/1558 (33/38)	1061/1558 (29/38)

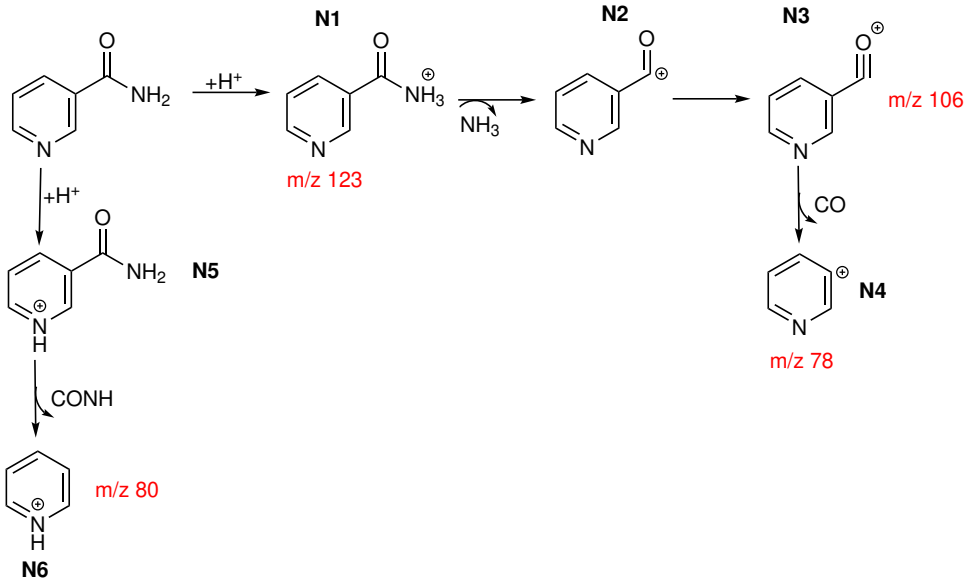


Abbildung 26: Darstellung des von `ChemFrag` vorhergesagten Fragmentierungsweges von Nikotinamid.

dem annotierten $[M+H]^+$-Ion begründen, auf die `ChemFrag` besonderen Wert legt. Daher können wir auch für diese Stoffgruppe schlussfolgern, dass `ChemFrag` in der Anzahl der Annotation gleichwertig mit `MetFrag` ist. Erfolgt dann noch ein Vergleich mit der Literatur, so zeigen sich deutlich Übereinstimmungen in den vorhergesagten

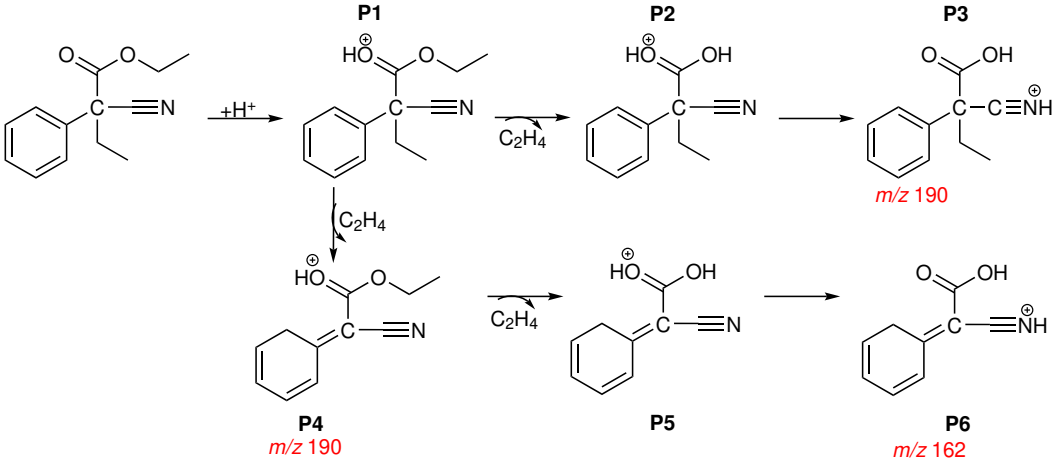


Abbildung 27: Darstellung des von `ChemFrag` vorhergesagten Fragmentierungsweges von 2-Cyano-2-phenylbutansäureetylester [77].

Fragment-Ionen. Besonders hervorzuheben ist dabei der Reaktionsweg von Nikotinamid, der starke Ähnlichkeit zum Fragmentierungsweg von Hau *et al.* [83] aufweist. Abbildung 26 visualisiert die Fragmentierung und der Vergleich zu Hau *et al.* ist im Anhang (Abbildung S-13) abgebildet.

Auch für diese Stoffklasse ist wiederum nicht für jedes Molekül ein Fragmentierungsweg veröffentlicht. Insgesamt existieren für vier Moleküle keine Wege, unter anderem für 2-Cyano-2-phenylbutansäureetylester. Daher nutzen wir `ChemFrag` um einen Fragmentierungsweg für 2-Cyano-2-phenylbutansäureetylester vorherzusagen. Diesen betrachten wir nun in Abbildung 27 detaillierter.

Beispiel 4: 2-Cyano-2-phenylbutansäureetylester
2-Cyano-2-phenylbutansäureetylester ist ein Zwischenprodukt bei der Synthese der Medikamente Hexamidin und Phenobarbital [84]. Bei Hexamidin handelt es sich um ein Antibiotikum zur Behandlung von Infektionen im Hals- und Rachenbereich oder der Augen [85]. Im Gegensatz dazu wird Phenobarbital als Antiepileptikum eingesetzt [85].

Die Fragmentierung zeichnet sich durch eine hohe Anwendung des regelbasierten Ansatzes aus. Als energetisch bevorzugte Protonierung gilt das Sauerstoffatom der Phenylbutansäure im Ester. Nach der Protonierung erfolgen zwei Reaktionswege. Zum einen kann durch die Abspaltung von Ethen aus dem Alkohol-Teil des Esters und eines anschließendem Proton-Shifts das Fragment-Ion P3 erzeugt werden. Dieses besitzt eine Masse von 190,09 und annotiert damit den Peak mit m/z 190,09. Die Umlagerung von P2 auf P3 begünstigt die Stabilität des Fragments. Ausgehend von

P1 kann es durch eine McLafferty-Umlagerung auch zur Abspaltung von Ethen kommen. Dieses Mal spaltet `ChemFrag` das Ethen aus der Phenylbutansäure ab. Dieses Fragment-Ion (P4) hat ebenfalls eine Masse von 190,09, ist jedoch energetisch um 153 *kJ/mol* höher als P3. Nach der Abspaltung von Ethen aus dem Alkohol-Teil des Esters bildet sich das Fragment-Ion P5, welches durch einen Proton-Shift das stabile Fragment-Ion P6 bildet. Dieses annotiert das Fragment-Ion mit einer Masse von 162.06.

Mit der Herleitung des neuen Fragmentierungsweges können wir die Einsatzfähigkeit von `ChemFrag` hervorheben und das positive Zusammenspiel aus dem regelbasierten und dem quantenchemischen Ansatz unterstreichen. Diese Kombination ermöglicht eine Anwendung von `ChemFrag`, um Fragmentierungswege und chemisch plausible Annotationen zu gewährleisten.

4.6.3. Zusammenfassung zur Fragmentierung von verschiedenen Substanzklassen

Ziel dieses Kapitels war es, mit der ausgewählten Parameterkombination 1 und der ermittelten Fragmentierungstiefe sieben durch `ChemFrag` Annotationen von MS/MS-Spektren aus verschiedenen Substanzklassen, unter anderem Naturstoffgruppen, zu bestimmen. Positiv hervorzuheben ist, dass für beide Gruppen Fragmentierungswege aus der Literatur bestätigt werden konnten. Weiterhin ist die Anzahl annotierter Fragment-Ionen mindestens gleichwertig mit `MetFrag` und überlegen zu `CFM-ID`. Durch den regelbasierten Ansatz ist es `ChemFrag` möglich, neue chemisch sinnvolle Fragmentierungswege dieser Naturstoffe vorherzusagen. Diese wurden an drei Beispiel bildlich dargestellt. Wir schlussfolgern daher, dass `ChemFrag` eine sehr gute Grundlage für die Annotation von Fragment-Ionen liefert und auch für die chemisch sinnvolle Vorhersage unbekannter Fragmentierungswege verwendet werden kann. Diese Anwendung ist ein Vorteil von `ChemFrag` im Vergleich zu existierenden Methoden.

4.7. Schnittstellen für `ChemFrag`

In dem vorherigen Kapitel stellten wir die Optimierung und Leistungsfähigkeit von `ChemFrag` vor. Für dieses Kapitel legen wir unser Augenmerk auf die Schnittstellen von `ChemFrag` für dessen Aufruf sowie dessen Ausgabe und die Visualisierung der Ergebnisse.

4.7.1. Aufruf und Visualisierung

Der Aufruf von `ChemFrag` kann über eine Konsolenanweisung oder über eine Webseite [86] gesteuert werden. Das in `ChemFrag` notwendige Programm `MOPAC` benötigt

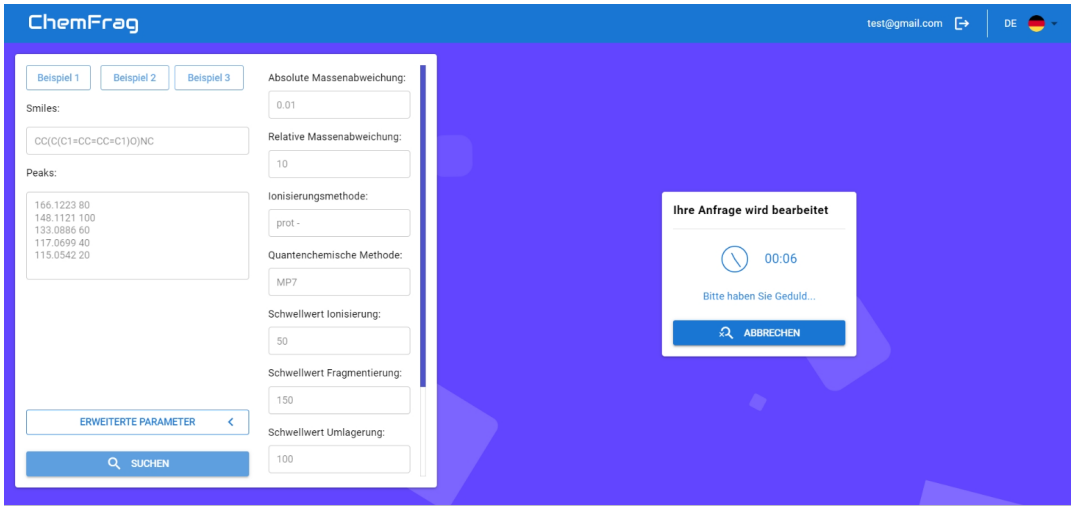


Abbildung 28: Darstellung der webseiten-basierten Eingabe der Parameter für Ephedrin [86].

einen Lizenzerwerb. Für akademische Einrichtungen ist es frei, muss jedoch beantragt werden. Im Vergleich dazu muss es für nicht-akademische Einrichtungen kostenpflichtig erworben werden. Aufgrund der Beschränkung der Nutzung von `MOPAC` ist der webseiten-gesteuerte Zugriff mit einem Login versehen. Nach einem erfolgreichen Login können die Parameter (Abbildung 28) für `ChemFrag` eingetragen werden. Bevor `ChemFrag` mit den eingegebenen Daten startet, erfolgt ein Abgleich mit einer Datenbank, ob bereits eine Berechnung für das gegebene Molekül mit der Parameterkombination durchgeführt wurde. Fällt der Vergleich positiv aus, wird das Ergebnis direkt ausgegeben. Andernfalls startet `ChemFrag` die Fragmentierung.

Die Ansicht der Ergebnisse ist über verschiedene Methoden möglich. Zum einen existiert eine Textdatei in der für jeden Peak die Namen erklärender Fragment-Ionen aufgeführt sind. Zusätzlich sind in einem Ordnersystem für jeden m/z-Wert die erklärten Fragment-Ionen als Bild im png-Format gespeichert. Das Bild bildet neben der Struktur auch den m/z-Wert, die Bildungsenthalpien und die Reaktionsenergie ab. Zusätzlich sind die Bindungsordnungen an alle Bindungen geschrieben, die im Intervall der Bindungsordnung liegen. Die Abbildung 29 zeigt ein Beispiel aus dem Ephedrin Reaktionsweg.
Neben der Ordner basierten Darstellung generiert `ChemFrag` webbasierte Ausgaben, die im Browser angezeigt werden können. Dazu gehören die Ausgabe des Reaktionsweges und die Auflistung der Peaks. Die Webseite (Abbildung 30) zur Fragment-Ionen Darstellung enthält Informationen zum m/z-Verhältnis, zur Bildungsenthalpie (Energie) und Reaktionsenergie. Zusätzlich ist die Struktur visualisiert und kann als mol-Datei und als Bild heruntergeladen werden. Weiterhin ermöglicht die Webseite

die Anzeige der eingegebenen Parameter und die Darstellung des Fragmentierungsweges des Fragment-Ionen. Für Anmerkungen ist ein Kommentar-Feld vorhanden.

Um den Reaktionsweg nachvollziehen zu können, existiert eine Visualisierung des Fragmentierungsbaumes (Abbildung Fig:ChemFrag-BeispielTree-Ephedrin), die schrittweise angezeigt wird. Im ersten Schritt erscheinen alle Molekül-Ionen aus dem Protonierungsschritt. Dafür wird eine Kurzbezeichnung als Namen verwendet. Der ausführliche Name kann in einer Textdatei im Ausgabeordner eingesehen werden. Bei einem Klick auf den Kurznamen erscheint die Struktur, wie in Abbildung 29. Der schwarze Pfeil öffnet die nächste Fragmentierungebene und zeigt die Strukturen, die aus diesem Edukt-Fragment entstanden sind. Durch die Pfeile ist es möglich zu erkennen, welche Strukturen für den nächsten Schritt ausgewählt wurden.

4.8. Limitierungen von `ChemFrag`

Die vorherigen Kapitel betrachteten den Aufbau und die Ergebnisse von `ChemFrag`. Nun möchten wir kurz noch darauf eingehen, welche Limitierungen für `ChemFrag` existieren.

Ein Nachteil von `ChemFrag` bildet die lange Laufzeit. Diese ist jedoch nicht der Architektur von `ChemFrag` geschuldet, sondern der Einbindung von `RDKit` und `MOPAC`. Abhängig von der Größe der Moleküle verlängert sich dadurch auch die Laufzeit. Zu einem weiteren Problem zählt die Reproduzierbarkeit der Ergebnisse. Die Reproduzierbarkeit hängt stark von der Koordinaten-Generierung und der anschließenden Optimierung durch `MOPAC` ab. Leider ist es bereits in `CDK` nicht möglich für gleiche Fragment-Ionen aus unterschiedlichen Annotationsdurchläufen durch `ChemFrag` identische 2D-Koordinaten für die Eingabe von `RDKit` zu generieren. Zusätzlich

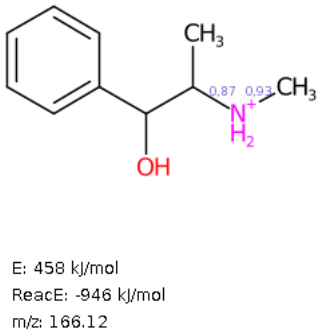


Abbildung 29: Ausgabe der png Datei am Beispiel von Ephedrin. Dieses Bild ist im Unterornder des zugehörigen m/z-Wertes im Ausgabepfad gespeichert.

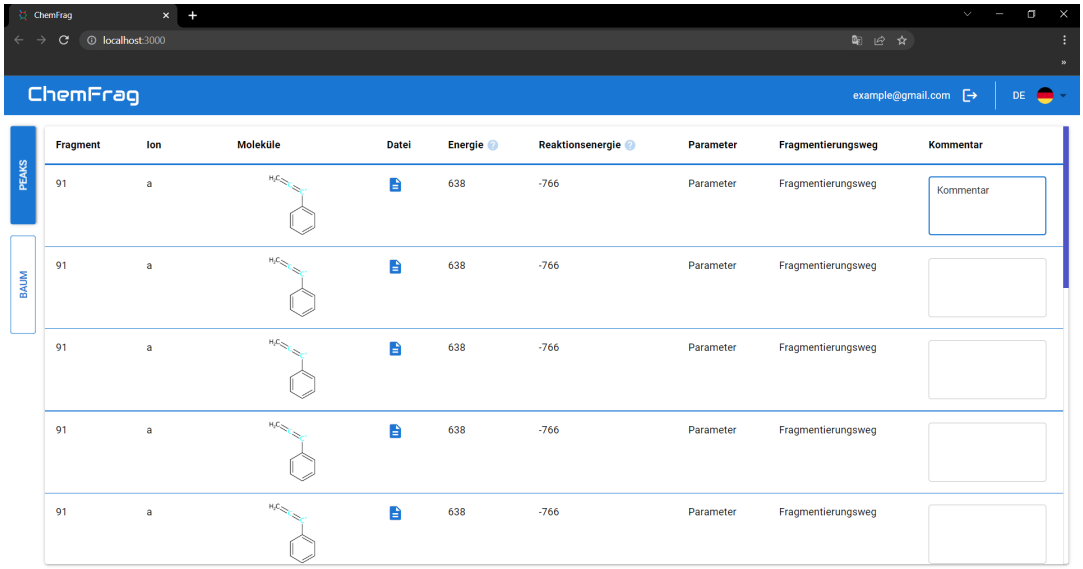


Abbildung 30: Ausschnitt der Ausgabe der Fragment-Ionen von Ephedrin [86].

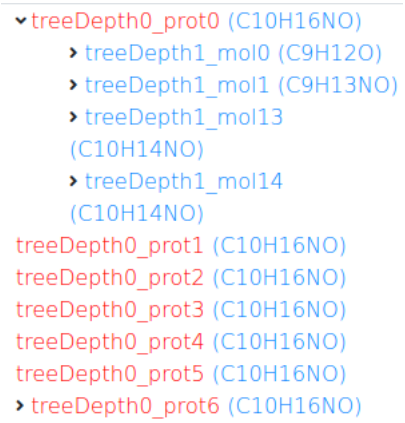


Abbildung 31: Ausgabe des Reaktionsweges im html Format für Ephedrin.

verwendet `RDKit` ein randomisiertes Optimierungsverfahren zur Generierung der 3D-Koordinaten, sodass selbst bei gleicher Eingabe der 2D-Koordinaten die 3D-Koordinaten in verschiedenen Durchläufen leicht unterschiedlich sein können. Das hat zur Folge, dass auch `MOPAC` seine Optimierung mit unterschiedlichen Startkoordinaten startet. Die Folge sind kleinere Unterschiede in den Bindungsordnungen und in den Bildungsenthalpien. Wie wir wissen, errechnet `ChemFrag` aus den Bildungsenthalpien die Reaktionsenergien jedes Fragments. Liegt ein Fragment-Ion durch eine veränderte Bildungsenthalpie nicht mehr im Reaktionsintervall führt `ChemFrag` davon keine weitere Fragmentierung durch. Das hat zur Folge, dass mögliche ganze Fragmentierungswege im Ergebnis nicht mehr abgebildet sind. Aus diesem Grund ist der Einsatz der Datenbank aus dem vorherigem Kapitel von besonderer Bedeu-

tung, da dort die Ergebnisse der Fragmentierung mit den Parameterkombinationen gespeichert sind. Dadurch lassen sich die Unterschiede zwischen mehreren Durchläufen exakt bestimmen und errechnete Fragmentierungswege gehen nicht verloren. Als ein weiterer Nachteil ist zu benennen, dass momentan alle Spaltungs- und Umlagerungsregeln in `ChemFrag` in Java implementiert sein müssen. Das bedeutet, dass neue Regeln nur eingefügt werden können, wenn der Nutzer Implementierungen in der Programmiersprache Java durchführen kann. Das stellt ein Hindernis in der Erweiterbarkeit von `ChemFrag` dar.

4.9. Zusammenfassung und Ausblick

Nach den genannten Limitierungen möchten wir nun die positiven Ergebnisse aus `ChemFrag` nochmal kurz zusammenfassen sowie darauf eingehen, wie die genannten Limitierungen behoben werden könnten.
Die chemische Plausibilität und Leistungsfähigkeit von `ChemFrag` mit dem regelbasierten und quantenchemischen Ansatz zeigten wir an mehreren Beispielen. Dazu zählen zum einen Moleküle aus dem Bereich der Doping Substanzen und zum anderen aus dem Bereich der Naturstoffe. Beide Klassen unterscheiden sich in ihrer Molekülstruktur grundlegend. Die Doping Substanzen enthalten nur wenige verbundene Ringsysteme und dafür vermehrt Seitenketten auf Kohlenstoff Basis, wohingegen die Naturstoffe eine Reihe von zusammenhängenden Ringsystemen aufweisen. Für beide Stoffklassen konnten wir zeigen, dass `ChemFrag` die Fragmentierungswege mit einer hohen Genauigkeit und chemisch sinnvollen Strukturen aufklären konnte. Besonders im Vergleich zu existierenden Programmen, wie `MetFrag` oder `CFM-ID` konnten wir erkennen, dass `ChemFrag` mehr Fragment-Ionen annotiert und diese gleichzeitig chemisch plausibel sind. Es kam beispielsweise nicht dazu, dass fünfach-gebundene Kohlenstoffatome, wie im Fall von `CFM-ID`, generiert werden. Viele der aufgeklärten Fragment-Ionen spiegeln die Ergebnisse aus publizierten Fragmentierungswegen, wie am Beispiel von Kokain oder Ephedrin, wider. Weiterhin konnten wir zeigen, dass die Reihenfolge der Fragmentierungen im Vergleich zu denen aus `SIRIUS` teilweise widerspruchsfreier und chemisch plausibler sind. Gleichzeitig konnten wir im Bereich der Naturstoffe Fragmentierungswege herleiten, die noch nicht publiziert wurden. Dazu zählt beispielsweise der Fragmentierungsweg von Estradiol-3-methylether, Dehydrocyanomethylestradiol und 2-Cyano-2-phenylbutansäureetylester.

Zusätzlich zur Auswertung der Leistungsfähigkeit in der Annotation durch `ChemFrag` führten wir Experimente zur Optimierung durch. Wir erkannten, dass eine Fragmentierungstiefe von sieben vielversprechende Ergebnisse liefert, die im zeitlich akzeptablen Rahmen liegen. Weiterhin haben wir default Parameter für die Intervalle zur Reaktionsenergie und für die Bindungsordnungen ermittelt. Diese sind 50 für die Pro-

tonierung (Tprot), 150 für die Strukturen aus der Fragmentierung (Tfrag) sowie 100 für die umgelagerten Strukturen (Trearr). Als Parameter für die Bindungsordnung (TBO) stellte sich ein Wert von 0,08 als optimal heraus. Weiterhin haben wir eine Analyse für die Verwendung der Regeln und deren Erfolg für die Annotierung von Fragment-Ionen durchgeführt. Als besonders wichtig haben sich der Hydrid-Shift- und Protonen-Shift, die homolytische Spaltung sowie die Abspaltung von Wasserstoff herausgestellt. Die weiteren Regeln werden je nach chemischer Gruppe unterschiedlich häufig eingesetzt. Abschließend haben wir weiter hervorgehoben, dass es in `ChemFrag` notwendig ist, die Berechnung und Fragmentierung gleicher Fragment-Ionen zu vermeiden. Abschließend können wir zusammenfassen, dass die aktuelle Entwicklung von `ChemFrag` bereits ein gutes Programm zur chemisch plausiblen Annotation von MS/MS-Spektren bereitstellt.

Für eine stärkere Verwendung von `ChemFrag` wären einige Anpassungen sinnvoll. Das betrifft zum einen die Implementation der Regeln. Um diese flexibler erstellen und von externen AnwenderInnen einpflegen zu können, soll die feste Implementierung der Regeln aus `ChemFrag` herausgenommen werden. Dafür ist die Idee eine eigene leicht verständliche domain-spezifische Sprache zu entwickeln, in der AnwenderInnen neue Regeln implementieren können. Diese implementierten Regeln müssten über einen eigenen Compiler aufgerufen und in Java-Quellcode umgewandelt werden, der anschließend von `ChemFrag` angewendet wird. Da für diese Anwendung ein flexibler Ansatz zur Substruktursuche notwendig ist, ist das Ziel diesen zu implementieren und den `SMARTS`-basierten Ansatz dadurch zu ersetzen. Weiterhin wäre es sinnvoll, `CDK` und `RDKit` so zu modifizieren, dass stets die gleichen Koordinaten generiert werden und damit eine höhere Reproduzierbarkeit erreicht werden kann. Sollte das nicht möglich sein, besteht die Idee eine Datenbank von Fragment-Ionen und Neutralverlusten aufzubauen, in der häufig generiert Fragment-Ionen/Neutralverluste mit ihrer Bindungsenthalpie und Bindungsordnungen gespeichert sind. Diese Werte könnten dann direkt aus der Datenbank gelesen und verwendet werden, sobald `ChemFrag` das Fragment-Ion bildet. Damit könnten Aufrufe von `RDKit` und `MOPAC` verhindert werden.

5. Molecule Equivalence Tester - MET

Die Analyse von Molekülen ist ein wichtiger Bestandteil der Chemoinformatik [87, 88, 89, 90, 91, 92]. Aufgrund der vielfältigen Atom- und Bindungskombinationen entsteht eine Vielzahl von Molekülen, die beispielsweise in den Datenbanken PubChem [93], KEGG [94] oder ChEBI [95] gespeichert sind. Um zu testen, ob eine Molekülstruktur bereits in einer der Datenbanken gespeichert ist, überprüft man, ob die gesuchte Molekülstruktur *äquivalent* zu einer gespeicherten Molekülstruktur ist. Molekülstrukturen werden als *äquivalent* bezeichnet, wenn ihre Atome so aufeinander abgebildet werden können, dass ihre Struktur in zweidimensionaler Darstellung und deren chemische Eigenschaften identisch sind. Um das Vorhandensein in einer Datenbank zu überprüfen, müssen daher die strukturellen und chemischen Eigenschaften beider Moleküle verglichen werden. Oftmals ist das optische Erkennen der Äquivalenz schwer möglich, wie wir an Abbildung 32 erkennen, weshalb dafür chemoinformatische Programme zum Einsatz kommen.

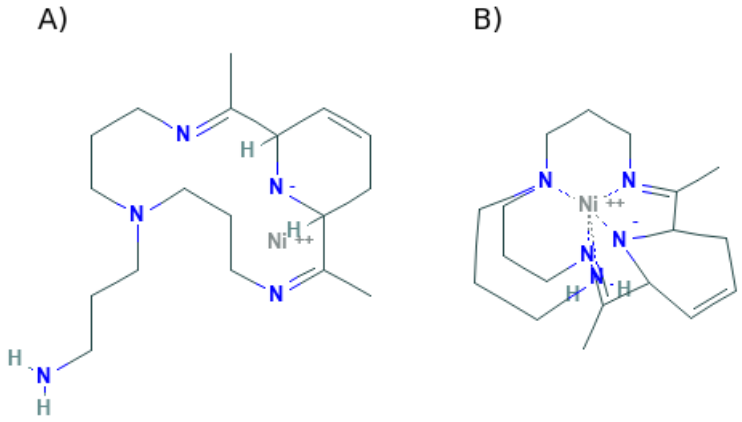


Abbildung 32: Molekülstruktur A) ist ein Bild aus der PubChem Datenbank mit der CID 50934715.
Molekülstruktur B) ist ein Bild aus der PubChem Datenbank mit der CID 6397461. Beide Moleküle sind in ihrer zweidimensionalen Darstellung äquivalent, was jedoch optisch schwer zu erkennen ist (übernommen aus [96]).

In den folgenden Kapiteln möchten wir auf die Herausforderung der Isomorphie und im Speziellen des Molekülvergleichs genauer eingehen. Die Erläuterungen zur Entwicklung von `MET` basieren auf der Veröffentlichung:

> Jördis-Ann Schüler, Steffen Rechner, Matthias Müller-Hannemann: *MET: a Java package for fast molecule equivalence testing.* Journal of Cheminformatics **12** (2020), 73-85

Wir vergleichen in diesem Kapitel Methoden des Äquivalenzvergleichs durch SMILES oder InChI, als String, Anwendungen aus chemoinformatischer Software sowie verschiedene Varianten der Graphisormphie. Anschließend werden wir für die Graphisomorphie existierende Methoden und Programme genauer betrachten und deren Verwendbarkeit prüfen. Im Detail werden wir auf das entwickelte Programm MET [96] (Molecule Equivalence Tester) sowie dessen Optimierungen eingehen.

5.1. Existierende Chemoinformatik-Software zur Moleküläquivalenz

Wie in Kapitel 2 dargestellt, existieren verschiedene Algorithmenbibliotheken in der Chemoinformatik. Besonders viel Anwendung finden dabei `CDK` [16] und `RDKit` [17]. Mit diesen Bibliotheken sind unterschiedliche Ansätze zum Molekülvergleich möglich, die wir nachfolgend betrachten möchten.

Äquivalenzvergleich auf Basis der String-Notation Das Kapitel 2 erklärt bereits, dass Moleküle in String-Formaten wie SMILES [12] und InChI [19] dargestellt werden können. Dies legt nahe, dass nach einer erfolgreichen Umwandlung einer Molekülstruktur in einen SMILES oder InChI der Äquivalenztest über einen reinen Stringvergleich erfolgen kann. Um zu festzustellen, ob zwei Molekülstrukturen äquivalent sind, würde es ausreichen, zu überprüfen, ob die zugehörigen Strings identisch sind. Dafür müsste jedoch gelten, dass die Stringdarstellung eindeutig wäre.
SMILES oder InChI können beispielsweise durch Chemoinformatik-Programme wie `CDK` und `RDKit` generiert werden. Ein wesentlicher Nachteil der SMILES ist jedoch, dass die SMILES Darstellung nicht eindeutig definiert ist. So können verschiedene Implementationen unterschiedliche SMILES für ein und dasselbe Molekül generieren. Betrachten wir dazu das Molekül 2-Chloroprop-1-en-amid aus Abbildung 36. Die generierten SMILES können `C=C(Cl)C(=O)N` oder `ClC(C(=O)N)=C` sein. Im Gegensatz dazu wäre der InChI eindeutig. Nachteil beider String-Darstellungen ist, dass der Äquivalenztest via String-Vergleich keine Zuordnung der äquivalenten Atome liefert. Dazu würde ein weiterer Algorithmus notwendig sein.

Aufgrund der zwei genannten Nachteile werden wir unser Augenmerk in dem kommenden Abschnitt auf die Bestimmungsmöglichkeiten zur Moleküläquivalenz aus verschiedenen Chemoinformatik-Bibliotheken legen.

Äquivalenzvergleich durch Chemoinformatik-Bibliotheken Die Bibliotheken `CDK`, Small Molecule Subgraph Detector (`SMSD`) [97] und `RDKit` zählen zu den bekanntesten Möglichkeiten aus der Chemoinformatik, um die Äquivalenz zweier

Molekülstrukturen zu überprüfen. `CDK` und `RDKit` können wie bereits erläutert, Moleküldarstellungen einlesen und durch implementierte Algorithmen Äquivalenztest darauf durchführen. `SMSD` ist eine Java-Bibliothek, die in `CDK` für den Äquivalenzvergleich eingebunden werden kann. Diese Programme arbeiten für eine Vielzahl an Molekülen korrekt. Jedoch beachten sie nicht immer alle chemischen Eigenschaften, wie Isotope oder Radikale, wie die Abbildungen 33 und 34 zeigen.

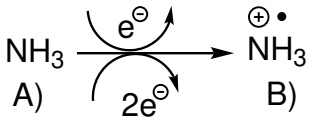


Abbildung 33: Beispiel: Ionisierung des Stickstoffatoms von Ammoniak (A) durch Elektronenstoßionisation in ein einfach positiv geladenes Ammoniak-Radial (B). Diese zwei Moleküle werden durch den Isomorphie-Test von `CDK` als äquivalent angesehen (übernommen aus [96]).

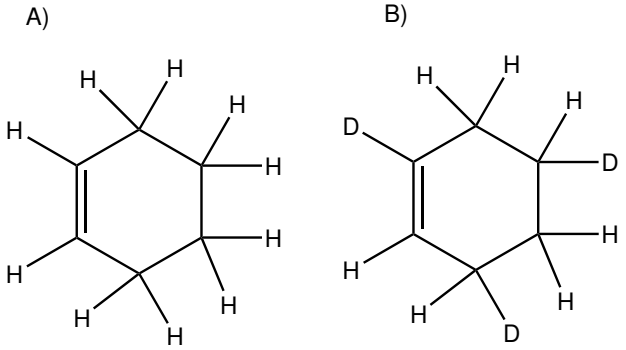


Abbildung 34: Struktur (A): Cyclohex-1-en, Struktur (B) Cyclohex-1-en, in dem drei Wasserstoffe durch Deuterium ersetzt sind. Diese zwei Moleküle werden durch den Isomorphie-Test von `CDK` als äquivalent angesehen (übernommen aus [96]).

Anderseits können weitere Algorithmen aus `CDK` und `SMSD` diese Eigenschaften aus Abbildung 33 und 34 erkennen. Jedoch verzeichnen sie dann eine hohe Laufzeit. Die Korrektheit und die Laufzeit dieser Programme werden wir im Kapitel 5.6. in den Ergebnissen analysieren.
Aus den Abbildungen 33 und 34 sehen wir, dass die existierenden Implementierungen aus den Chemoinformatik-Programmen die Moleküläquivalenz nicht vollständig korrekt bewerten. Zusätzlich ist es in den existierenden Programmen schwer zu erkennen, welche Eigenschaften eines Moleküls für den Äquivalenzvergleich verwendet werden sollen. Aus diesem Grund möchten wir im Rahmen dieser Arbeit ein neues Programm zur Erkennung der Moleküläquivalenz entwickeln. Aus Kapitel 2 wissen wir bereits, dass Molekülstrukturen auch als Graphen modelliert werden können. Die

Äquivalenz zweier Graphen kann durch einen Test auf Graphisomorphie nachgewiesen werden, wofür bereits viele Algorithmen existieren [98]. Daher möchten wir auch in unserer Entwicklung Molekülstrukturen als Graph repräsentieren. Dazu werden wir nachfolgend die Grundlagen der Graphisomorphie betrachten.

5.2. Äquivalenzvergleich auf Basis von Graphen

Das Kapitel 2 führte die verschiedenen Varianten der Überführung von Molekülstrukturen in Graphen ein. Das waren zum einen die einfachen Graphen sowie die gelabelten Graphen. Für beide Repräsentationsarten existieren verschiedene Arten zur Erkennung der Graphisomorphie. Bevor wir diese detaillierter vergleichen, möchten wir uns zunächst mit dem Begriff Graphisomorphie auseinandersetzen.

5.2.1. Grundlagen der Graphisomorphie

Garey und Johnson [99] definieren zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ genau dann als isomorph, falls eine Abbildung $f : V_1 \to V_2$ existiert mit den Eigenschaften:

1. $f : V_1 \to V_2$ ist bijektiv und
2. für alle $x, y \in V_1$ gilt: $\{x, y\} \in E_1$ genau dann, wenn $\{(f(x), f(y))\} \in E_2$.

Für gelabelte Graphen gilt zusätzlich die Forderung $l_1(v) = l_2(v)$ für jeden Knoten $v \in V_1$.
Eine derartige Abbildung $f : V_1 \to V_2$ heißt Isomorphismus von G_1 nach G_2. Die Abbildung 35 zeigt ein Beispiel.

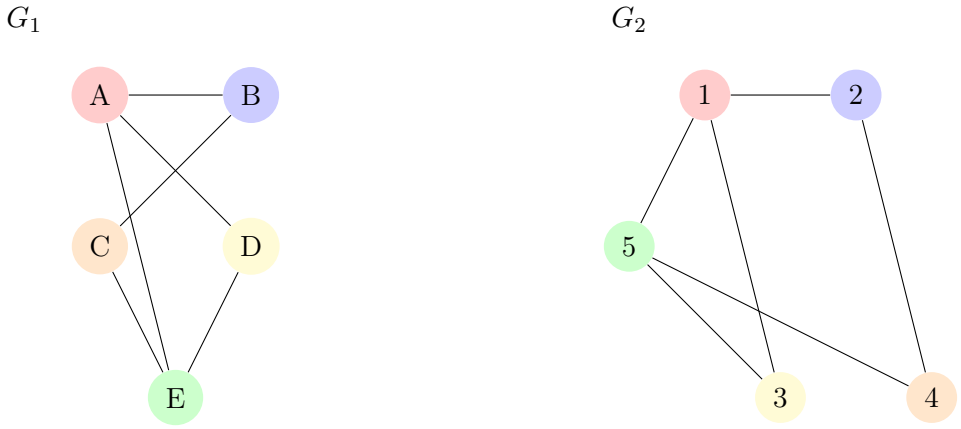


Abbildung 35: Isomorphismus zwischen den Graphen G_1 und G_2. Die Abbildung f bildet die Knoten zwischen G_1 und G_2 wie folgt ab: $A \mapsto 1$, $B \mapsto 2$, $D \mapsto 3$, $C \mapsto 4$, $E \mapsto 5$.

Komplexität Für das Graphisomorphie-Problem ist im Allgemeinen nicht bekannt, ob es in der Klasse P liegt oder NP-vollständig ist [22, 99]. Im Allgemeinen sind daher keine Polynomialzeit-Algorithmen bekannt. Einen Algorithmus in quasipolynomialer Zeit zur Lösung des Graphisomorphie-Problems auf allgemeinen Graphen präsentierte Babai im Jahr 2016 [100]. Die Laufzeit wird mit $O(n^{\text{polylog}(n)})$ abgeschätzt, wobei n die Knotenanzahl und das polylog(n) ein Polynom in $\log(n)$ ist. Für bestimmte Spezialfälle sind schärfere Abschätzungen bekannt, sodass eine Lösung in polynomialer Zeit bestimmbar ist. Einige dieser Fälle schauen wir uns im Folgenden genauer an.
Für planare Graphen kann die Isomorphie in polyonomieller Zeit überprüft werden. Bekannte Algorithmen dafür sind der Linearzeit-Algorithmus von Mehlhorn und Mutzel [101] und der Hopcroft-Wong Algorithmus [102].
Für gradbeschränkte Graphen zeigte Luks, dass diese ebenfalls in polynomieller Zeit auf Isomorphie überprüfbar sind [103]. Jedoch ist dieser Algorithmus in der Praxis ungeeignet, weil die Laufzeit sehr hoch ist [22].

Ein weit verbreiteter Algorithmus zur Lösung des Graphisomorphie-Problems ist der `nauty` Algorithmus von McKay [104]. Dieser wandelt gelabelte Graphen in eine eindeutige kanonische Form um. Der Isomorphietest kann dann durch simples Vergleichen der kanonischen Darstellung erfolgen. `nauty` hat im Allgemeinen eine exponentielle Laufzeit [104]. Für viele Instanzen ist er jedoch sehr schnell.
Einer der ersten in der Praxis häufig angewendeten Algorithmen ist der 1971 publizierte Isomorphie-Algorithmus von Ullmann [105]. Er basiert auf einem rekursiven Backtracking-Ansatz, dessen Grundidee wir in Kapitel 5.4.2.4. genauer beschreiben werden. Ullmann veröffentlichte 2011 eine substantielle Verbesserung des Algorithmus [9].
Cordella *et al.* publizierten 1998 den VF-Algorithmus [106]. Auf ihn folgten eine Reihe von Entwicklungen zu `VF2`, `VF2Plus` bis hin zu `VF2++` aus dem Jahr 2018 [10]. Die Grundidee dieser verwandten Algorithmen ist die iterative Erweiterung von Teillösungen unter Nutzung bestimmter Lösbarkeitskriterien. Da `VF2++` die aktuellste Erweiterung der VF-Algorithmen ist und als vergleichender Algorithmus zu `MET` dienen wird, stellen wir diesen später detaillierter vor.

Anwendbarkeit auf Moleküle Bei der Umwandlung einer Molekülstruktur in einen Graphen repräsentieren die Knoten die Atome und Kanten die Bindungen der Molekülstruktur. Chemische Eigenschaften können als ganze oder reelle Zahlen in einem gelabelten Graphen oder als Dummy-Atome in einem einfachen Graphen modelliert werden. Zwei Molekülstrukturen sind genau dann äquivalent, wenn die

zwei zugehörigen Graphen isomorph zu einander sind.
Eine Möglichkeit zur Äquivalenzbestimmung stellte 2009 Chowdary *et al.* [21] vor. Dieser Ansatz wandelt Molekülstrukturen zunächst in einfache Graphen um und testet diese anschließend auf Planarität. Viele Moleküle, wie DNA und RNA oder Fullerene können als planare Graphen dargestellt werden. In diesem Fall wird einer der oben genannten Polymonialzeit-Algorithmen für planare Graphen verwendet. Sollte keine Planarität vorliegen, wird auf den Algorithmus von Luks [103] zurückgegriffen. Beispiele für Moleküle, die nicht planar dargestellt werden können, sind anorganische oder verknüpfte Polymere [20, 22]. Werden Molekülstrukturen in gelabelte Graphen umgewandelt, können sie beispielsweise wie in `CDK` der Ullmann-Algorithmus oder auch von `VF2++`, wie Jüttner 2018 [10] zeigte, für den Äquivalenzvergleich angewendet werden.

Die Nachteile der Algorithmen aus `CDK` haben wir in Kapitel 5.1 bereits benannt. Für die Verwendbarkeit der Algorithmen aus Kapitel 5.2 müssen die Molekülstrukturen zunächst in Graphen überführt werden.
Oftmals gestaltet sich die Integration vorhandener Bibliotheken in chemoinformatische Software schwierig. Um diese Herausforderungen zu überwinden, fand im Rahmen dieser Arbeit die Entwicklung des Programms `MET` (Molecule Equivalence Tester) statt.

`MET`‘s Ziel ist die Überprüfung der exakten Äquivalenz zweier Molekülstrukturen, die in 2D repräsentiert sind. Die Beachtung aller chemischer Eigenschaften, im Speziellen der Isotope und der Radikale, ist darin als Ziel verankert. Unter Beachtung der Eigenschaften ist das Ziel von `MET` eine bessere Laufzeit zu erzielen als die vorhandenen Implementierungen der Isomorphie-Algorithmen in `CDK`, `SMSD` und `RDKit`. Zusätzlich soll eine Einbindung des Algorithmus in `CDK` möglich sein.

In den folgenden Kapiteln sehen wir uns den von `MET` verwendeten Äquivalenzbegriff genauer an. Ebenfalls erläutern wir detailliert die Architektur von `MET`.

5.3. Moleküläquivalenz in MET

`MET` modelliert eine Molekülstruktur als Graph $G = (V, E)$, mit einer Knotenmenge V und einer Kantenmenge E. Jeder Knoten hat ein zugehöriges Knotenlabel $l(v) \in \mathbb{R}^s$ aus s ganzen Zahlen, welche jeweils Atomeigenschaften repräsentieren. Eine Zahl könnte beispielsweise die Ordnungszahl des zugehörigen Atoms sein. Weitere mögliche Eigenschaften erläutert das Kapitel 5.4.1.1.

Für die Bestimmung der Äquivalenz definieren wir, wie zuvor, zwei Molekülstruktu-

ren $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ mit den dazugehörigen Knotenlabeln l_1 und l_2 als äquivalent genau dann, wenn eine bijektive Funktion $f : V_1 \to V_2$ existiert, sodass

- $\{f(u), f(v)\} \in E_2$ genau dann wenn $\{u, v\} \in E_1$ und
- $l_1(v) = l_2(v)$ für jeden Knoten $v \in V_1$.

Durch die erste Bedingung ist gesichert, dass beide Graphen die selbe Struktur haben. Mit der zweiten Bedingung ist die Kompatibilität der Knotenlabel garantiert.

5.4. Methodik von `MET`

Der in `MET` implementierte Algorithmus benötigt als Eingabe zwei Molekülstrukturen. Die Repräsentation der Molekülstruktur erfolgt entweder über eine Darstellung aus `CDK` oder als SDF-Dateiformat.
Der Ansatz von `MET` besteht aus zwei Phasen. Im ersten Teil erfolgt die Umwandlung der Molekülstruktur in einen gelabelten Graphen. Der zweite Teil enthält den Isomorphietest. Betrachten wir das kurz genauer:

1. In der ersten Phase konvertiert der Algorithmus die beiden Molekülstrukturen in gelabelte Graphen. Dabei wird jede Molekülstruktur in einen Graphen $G = (V, E)$ mit der Knotenmenge V und Kantenmenge E transformiert, wodurch die zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ generiert werden. Anschließend erstellt `MET` die Knotenlabel $l_1 : V_1 \to \mathbb{R}^s$ und $l_2 : V_2 \to \mathbb{R}^s$. Diesen Prozess beschreiben wir gleich genauer. Dabei ist wichtig, dass zwei Knoten nur dann die identischen Labels erhalten, wenn ihre zugehörigen Atome identische Eigenschaften haben.

2. In der zweiten Phase läuft zuerst ein schneller Vortest. Dieser prüft, ob G_1 und G_2 offensichtlich nicht isomorph sind. Wenn der Vortest eine potentielle Isomorphie feststellt, führt `MET` den eigentlichen Isomorphie-Algorithmus aus, der entscheidet, ob die gelabelten Graphen isomorph sind.

In den folgenden Unterkapiteln erklären wir die zwei Phasen detaillierter.

5.4.1. Phase 1

Die erste Phase überführt die 2D-Repräsentation zweier Molekülstrukturen in gelabelte Graphen. Für die Überführung ersetzt `MET` jedes Atom durch einen Knoten und jede Bindung durch eine Kante. Um die chemischen Eigenschaften beizubehalten, werden die Atomeigenschaften als Knotenlabel und die Bindungstypen als Kantenlabel gespeichert. Nachfolgend betrachten wir die verwendeten chemischen Eigenschaften genauer.

5.4.1.1. Atomeigenschaften

`MET` ermöglicht es dem Anwender, die zu verwendenden chemischen Eigenschaften auszuwählen, die für den Äquivalenztest berücksichtigt werden sollen. Es handelt sich um die Folgenden:

- Ordnungszahl (z.B. 1 für Wasserstoff, 6 für Kohlenstoff)
- Anzahl an gebundenen Wasserstoffatomen und gebundenen Deuterium-Atomen oder Tritium-Atomen
- Formalladung (z.B. 0 für neutral, +1 für einfach positiv, -1 für einfach negativ)
- Anzahl an freien Elektronen (Radikalen)
- Anzahl an gebundenen Einfach-, Zweifach- und Dreifachbindungen

Tabelle 15: Liste von klassischen chemischen Eigenschaften für die Molekülstruktur in Abbildung 36 (übernommen aus [96]).

Eigenschaft	Atom 1	Atom 2	Atom 3	Atom 4	Atom 5	Atom 6
Ordnungszahl	6	6	6	17	8	7
Wasserstoffanzahl	2	0	0	0	0	2
Deuteriumanzahl	0	0	0	0	0	0
Formalladung	0	0	0	0	0	0
Radikalanzahl	0	0	0	0	0	0
Anzahl Einfachbindungen	0	2	2	1	0	1
Anzahl Doppelbindungen	1	1	1	0	1	0
Anzahl Dreifachindungen	0	0	0	0	0	0

Zusätzliche Eigenschaften können einfach in `MET` integriert werden, solange sie als Zahlen repräsentiert werden können. Tabelle 15 zeigt die zugehörigen Atomeigenschaften der Molekülstruktur aus Abbildung 36.

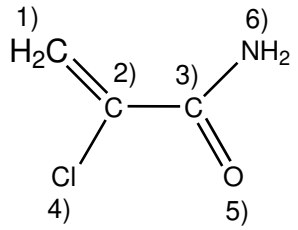


Abbildung 36: Molekülstruktur von 2-Chloroprop-1-en-amid zur Erklärung der klassischen Atomeigenschaften (übernommen aus [96]).

Basierend auf den ausgewählten Eigenschaften, erhält jeder Knoten v ein Label $l(v) = (p_1, p_2, ...p_s)$, wobei jedes p_i eine bestimmte Atomeigenschaft repräsentiert. Dabei erhalten zwei Knoten nur dann identische Labels, wenn ihre zugehörigen Atome identische Eigenschaften haben.

5.4.1.2. Nachbarschaftsdeskriptor

Neben den chemischen Eigenschaften, erhält jeder Knoten v eine zusätzliche Struktureigenschaft $d(v)$. Diese nennt sich Nachbarschaftsdeskriptor. Dabei handelt es sich um eine ganze Zahl, die die Informationen zur lokalen Nachbarschaft des Knotens kodiert. `MET` sieht vor, dass zwei Knoten den identischen Nachbarschaftsdeskriptor haben, wenn:

a) sie die gleichen chemischen Eigenschaften haben und

b) ihre lokale Nachbarschaft strukturell identisch ist.

Um den Nachbarschaftsdeskriptor zu berechnen, definiert der Algorithmus iterativ für jeden Knoten v ein Integer $d_i(v)$, der seine lokale Nachbarschaft bis zu einer Tiefe von i beschreibt. Für die Initialisierung gilt:

$$d_0(v) := hash(l(v)).$$

Dabei bezeichnet $hash$ eine Hashfunktion, die Tupel von ganzen oder rationalen Zahlen auf Integer abbildet. Damit enthält der initiale Deskriptor $d_0(v)$ nur Informationen von dem Knoten v selbst. Für $1 \leq i \leq k$, gilt:

$$d_i(v) := \sum_{(v,w) \in E} d_{i-1}(w).$$

Der Wert k entspricht dabei der maximalen Tiefe, die der nächste Abschnitt genauer betrachtet. Nach der Berechnung von $d_i(v)$ für $0 \leq i \leq k$, verwendet `MET`

$$d(v) := hash(d_0(v), d_1(v), ..., d_k(v))$$

als Nachbarschaftsdeskriptor für den Knoten v.
Tabelle 16 demonstriert die Berechnung des Nachbarschaftsdeskriptors am Beispiel aus Abbildung 37. Darin können wir mehrere Beobachtungen erkennen:

- In diesem Beispiel sollen die Knoten C und D identische chemische Eigenschaften haben. Da sie zusätzlich eine symmetrische Nachbarschaftsstruktur haben, erhalten sie identische Werte für d_i für alle $i \geq 0$. Somit erhalten sie den gleichen Nachbarschaftsdeskriptor d.

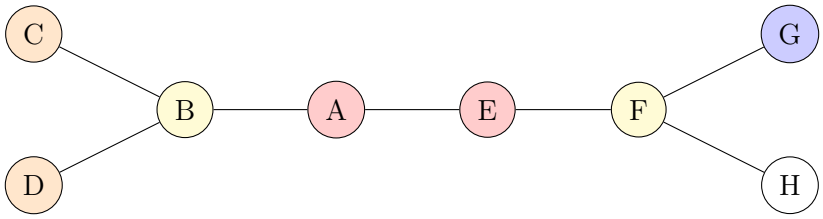


Abbildung 37: Beispiel-Graph. Die Knoten A und E, C und D, sowie B und F haben vermeintlich die selben chemischen Eigenschaften und sind jeweils mit gleichen Farben markiert (übernommen aus [96]).

Tabelle 16: Beispiel zur Berechnung des Nachbarschaftsdeskriptors basierend auf Abbildung 37. Die Werte für d_0 sind beispielhaft und folgen keiner speziellen Semantik (übernommen aus [96]).

	d_0	d_1	d_2	d_3	d
A	5	8	27	39	78
B	3	19	14	65	30
C	7	3	19	14	90
D	7	3	19	14	90
E	5	8	25	41	60
F	3	17	14	59	75
G	10	3	17	14	88
H	2	3	17	14	48

- Die Knoten G und H zeichnen sich durch verschiedene chemische Eigenschaften ($d_0(G) \neq d_0(H)$) aus. Ihre Nachbarschaftsstruktur ist jedoch exakt symmetrisch ($d_i(G) = d_i(H)$ für alle $i \geq 1$). Dennoch erhalten sie unterschiedliche Nachbarschaftsdeskriptoren.

- Das Beispiel legt für doe Knoten A und E identische chemische Eigenschaften ($d_0(A) = d_0(E)$) fest. Zusätzlich haben deren unmittelbar benachbarten Atome die selben chemischen Eigenschaften ($d_1(A) = d_1(E)$). Jedoch unterscheidet sich die lokale Nachbarschaft ab einer Tiefe von zwei und mehr ($d_i(A) \neq d_i(E)$ für alle $i \in \{2, 3\}$) . Dadurch erhalten A und E unterschiedliche Nachbarschaftsdeskriptoren.

5.4.1.3. Maximale Tiefe

Ein wichtiger Aspekt bei der Berechnung des Nachbarschaftsdeskriptors ist die maximale Tiefe k der Nachbarschaft. Auf der einen Seite enthält ein großer Wert von k viele Informationen zur lokalen Nachbarschaft jedes Knotens. Das hat eine mögliche

Reduzierung der Laufzeit beim Isomorphietest zur Folge. Auf der anderen Seite benötigt die Berechnung des Nachbarschaftsdeskriptors des Molekülgraphens $G = (V, E)$ eine Laufzeit von $O(k \cdot |E|)$. Das spricht dafür, k als möglichst kleine Konstante zu wählen. Das Kapitel 5.6., zur Parameterermittlung, enthält eine Studie zum Einfluss von k auf die Laufzeit der Benchmark-Experimente.

5.4.1.4. Fingerprint

Neben der Umwandlung der Molekülstruktur in einen Graphen repräsentiert `MET` intern das Molekül zusätzlich als *Fingerprint*. Die Grundlage des Fingerprints bilden die Atomeigenschaften. `MET` errechnet für jede Atomeigenschaft die Summe über alle Atome. Diese berechneten Werte der Atomeigenschaften verbindet `MET` mit Unterstrichen zu einem String. Betrachten wir die Molekülstruktur aus Abbildung 36 so generiert `MET` den folgenden String:

$$fingerprint(2-Chloroprop-1-enamid) : 50\_4\_0\_0\_0\_6\_4\_0\_-1117291562$$

Die Reihenfolge der Atomeigenschaft im Fingerprint entspricht der Reihenfolge aus Tabelle 15. Der letzte Wert bildet den Nachbarschaftsdeskriptor ab. Im Kapitel 5.7. werden wir die Anwendung der Fingerprints und deren Eindeutigkeit genauer untersuchen. Nach der Umwandlung und Kodierung der chemischen und strukturellen Eigenschaften aus den Molekülstrukturen in Graphen folgt die Phase 2.

5.4.2. Phase 2

In der zweiten Phase testet `MET`, ob die zwei gelabelten Graphen G_1 und G_2 mit den Knotenlabeln l_1 und l_2 isomorph zueinander sind. Im ersten Schritt läuft dazu ein Vortest. Daran schließt sich der eigentliche Isomorphietest an.

5.4.2.1. Vortest

Bevor der richtige Isomorphie-Test startet, führt `MET` einen Vortest aus. Dieser soll schnell entdecken, ob die Graphen offensichtlich *nicht* isomorph sind. Dazu werden die charakteristischen Eigenschaften der Graphen G_1 und G_2 bestimmt und verglichen. Diese Eigenschaften sind:

- Gesamtanzahl der Atome in der Molekülstruktur
- Gesamtanzahl der Bindungen in der Molekülstruktur
- Gesamtanzahl der Radikale in der Molekülstruktur
- Gesamtanzahl der Wasserstoffe und der Deuterium/Tritium-Atome in der Molekülstruktur

- sowie die Formalladung des Moleküls

Unterscheiden sich die Graphen in mindestens einer dieser Eigenschaften, können sie nicht isomorph sein. Zusätzlich erstellt `MET` die Label-Sets $\{l_1(v) : v \in V_1\}$ und $\{l_2(v) : v \in V_2\}$ und prüft diese auf Gleichheit. Sind die Sets nicht identisch, kann mit Sicherheit darauf geschlossen werden, dass die zwei Molekülstrukturen in ihrer 2D-Repräsentation nicht äquivalent sind. Sind jedoch beide Bedingungen erfüllt, führt `MET` im nächsten Schritt den Äquivalenztest aus.

5.4.2.2. Äquivalenztest

Nach einem erfolgreichen Vortest versucht `MET` einen Isomorphismus zwischen G_1 und G_2 zu finden. Dafür stehen zwei Algorithmen zur Verfügung. Zum einen kann der in der Lemon Bibliothek [107, 108] implementierte `VF2++` Algorithmus verwendet werden. Dieser ist besonders auf großen Instanzen schnell [10]. Für unsere Zwecke ist nachteilig, dass `VF2++` in C++ implementiert ist und sich daher nur auf Umwegen in eine Java Anwendung integrieren lässt. Im Gegensatz dazu ist in `MET` ein eigener einfacher Backtracking-Algorithmus eingebunden, den wir im Folgenden kurz skizzieren. Dieser hat den Vorteil, dass er direkt in Java implementiert ist.
Als Eingabe erhält der Algorithmus zwei Graphen G_1 und G_2 mit den zugehörigen Knoten-Labeln l_1 und l_2.

5.4.2.3. Kandidatenmenge

Zunächst konstruiert der Algorithmus 1 für alle Knoten aus G_1 und G_2 sogenannte Kandidatenmengen $can_1(v)$ und $can_2(v)$.

Algorithmus 1: ErstelleKandidatenMenge$(G_1, G_2, \ell_1, \ell_2)$

Eingabe : Graph $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$, Knotenlabel ℓ_1 ,ℓ_2.
Ausgabe: Kandidatenmengen can_1 und can_2

```
1  for v ∈ V1 do
2      can1(v) ← ∅;
3  for w ∈ V2 do
4      can2(w) ← ∅;
5  for v ∈ V1 do
6      for w ∈ V2 do
7          if ℓ1(v) = ℓ2(w) then
8              can1(v) ← can1(v) ∪ {w};
9              can2(w) ← can2(w) ∪ {v};
10 return can1, can2
```

Für jeden Knoten $v \in V_1$ ist die entsprechende Kandidatenmenge definiert als die

Menge

$$can_1(v) := \{w \in V_2 : l_1(v) = l_2(w)\}$$

von Knoten aus G_2, die dem Knoten v zugeordnet werden könnten, da sie das gleiche Label besitzen. Symmetrisch bestimmt der Algorithmus für jeden Knoten $w \in V_2$ aus G_2 eine Kandidatenmenge

$$can_2(w) := \{v \in V_1 : l_1(v) = l_2(w)\}$$

von Knoten aus G_1, die w zugewiesen werden können.

5.4.2.4. Backtracking

Das Herzstück des Isomorphie-Algorithmus ist der Algorithmus 2 mit einem Backtracking-Ansatz.

Algorithmus 2: KNOTENZUWEISUNG(can_1, can_2, f)

Data: Graph $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$.
Eingabe : Kandidatenmenge can_1 und can_2, Funktion $f\colon V_1 \to V_2 \cup \{\text{NIL}\}$
Ausgabe: Isomorphiefunktion $f\colon V_1 \to V_2$, oder NIL wenn kein Isomorphismus existiert.

```
1  if ∀v ∈ V1: f(v) ≠ NIL then
2      return f;     // alle Knoten zugewiesen, f ist ein Isomorphismus
3  Wähle ein v ∈ V1 mit f(v) = NIL und minimalem |can1(v)|
4  for w ∈ can1(v) do
5      f(v) ← w;                                          // weise v w zu
6      can'1, can'2 ← KANDIDATENREDUZIERUNG(can1, can2, v, w);
7      f' ← KNOTENZUWEISUNG(can'1, can'2, f);          // rekursiver Aufruf
8      if f' ≠ NIL then
9          return f';                                  // f' ist ein Isomorphismus
10 return NIL;                                         // kein Isomorphismus möglich
```

Dieser rekursive Algorithmus selektiert zunächst einen Knoten $v \in V_1$ mit der kleinsten Kandidatenmenge. Der Algorithmus versucht dann dem Knoten v einen Knoten w aus seiner Kandidatenmenge $can_1(v)$ zuzuweisen. Nach Konstruktion der Kandidatenmengen ist sichergestellt, dass v und w identische Knotenlabels besitzen.
Nach der Zuweisung der zwei Knoten v und w zueinander, reduziert der Algorithmus die Kandidatenmengen aller Knoten (was wir gleich im Detail beschreiben) und versucht die übrigen Knoten einander zu zuweisen. Dies kann, muss aber nicht erfolgreich sein:

- Ist die Zuweisung jedes Knoten aus G_1 zu G_2 erfolgreich, hat der Algorithmus die Isomorphie der beiden Graphen bestimmt. Der Algorithmus antwortet mit einer positiven Antwort und der Isomorphie-Funktion f.

- Ist das rekursive Zuweisen der übrigen Knoten nicht erfolgreich, wird die Zuweisung $f(v) = w$ rückgängig gemacht. Anschließend führt `MET` erneut den Backtracking-Schritt mit dem nächsten Knoten der Kandidatenmenge aus. Nachdem alle Kandidaten auf diese Weise ohne positive Antwort getestet wurden, weiß der Algorithmus, dass G_1 und G_2 nicht isomorph sein können.

5.4.2.5. Kandidatenreduktion

Durch das Zuweisen von w zu v können mehrere Kandidatenmengen, wie im Algorithmus 3 gezeigt, reduziert werden:

1. Offensichtlich können v und w aus jeder Kandidatenmenge entfernt werden, in der sie aktuell enthalten sind.
2. Für jede Kante $\{u, v\} \in E_1$ existiert eine Kante $\{z, w\} \in E_2$. Somit kann für jede Kante $\{u, v\} \in E_1$ jeder Knoten aus $can_1(u)$ entfernt werden, der nicht adjazent zu dem Knoten w ist.
3. Symmetrisch kann aus $can_2(z)$ jeder Knoten entfernt werden, der nicht adjazent zu v ist.
4. Weiterhin können die Kandidatenmengen von v und w gelöscht werden.

Beispiel Zur Veranschaulichung der Kandidatenreduktionsregeln betrachten wir das Beispiel aus Abbildung 38.

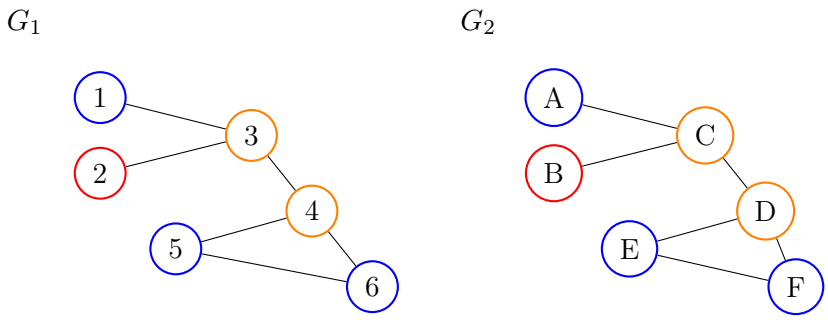


Abbildung 38: Zwei isomorphe Beispielgraphen zur Demonstration der Kandidatenreduktion. Knoten mit gleicher Farbe haben das gleiche Label.

Algorithmus 3: KANDIDATENREDUZIERUNG(can_1, can_2, v, w)

Data: Graph $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$.
Eingabe : Kandidatenmenge can_1 und can_2, Knoten $v \in V_1, w \in V_2$
Ausgabe: Aktualisierte Kandidatenmenge $can'_1 \colon V_1 \to 2^{V_2}$ und
$can'_2 \colon V_2 \to 2^{V_1}$ kleinerer Größe.

```
1  for u ∈ can_1(v) do
2      can_2(u) ← can_2(u) \ {v}
3  for u ∈ can_2(w) do
4      can_1(u) ← can_1(u) \ {w}
5  for {u, v} ∈ E_1 do
6      for z ∈ can_1(u) do
7          if {w, z} ∉ E_2 then
8              can_1(u) ← can_1(u) \ {z};
9              can_2(z) ← can_2(z) \ {u};
10 for {u, w} ∈ E_2 do
11     for z ∈ can_2(u) do
12         if {v, z} ∉ E_1 then
13             can_2(u) ← can_2(u) \ {z};
14             can_1(z) ← can_1(z) \ {u};
15 for {u, w} ∈ E_2 do
16     can_2(u) ← can_2(u) \ {z ∈ V_1 : {v, z} ∉ E_1}
17 can_1(v) ← ∅;
18 can_2(w) ← ∅;
19 return can_1, can_2
```

Der Entscheidungsbaum in Abbildung 39 demonstriert die Veränderung der Kandidatenmengen im Laufe des Algorithmus. Jeder Knoten dieses Entscheidungsbaumes ist mit den dazugehörigen Kandidatenlisten can_1 beschriftet. (Die Kandidatenlisten can_2 sind nicht zu sehen.) Der Pfad von der Wurzel bis zum linken Blatt oder auch zum rechten Blatt wären Pfade im Rekursionsbaum, die die Äquivalenz nachweisen. Aus der Struktur des Entscheidungsbaums in Abbildung 39 lassen sich bereits allgemeine Aussagen über die Effizienz des Backtracking-Verfahrens schließen.

- Jeder Pfad von der Wurzel zu einem Blatt entspricht einer partiellen Zuordnung der Knoten aus G_1 und G_2. Ein Pfad der Länge $|V|$ entspricht einer gefundenen Isomorphie-Funktion.
- Die maximale Höhe des Entscheidungsbaums ist die Anzahl $|V|$ der Knoten. Allerdings haben im Allgemeinen nicht alle Pfade diese Länge, da die Rekursion durch eine leere Kandidatenmenge früher abbrechen kann.
- Der Entscheidungsbaum ist umso breiter, je größer die Kandidatenmengen sind.

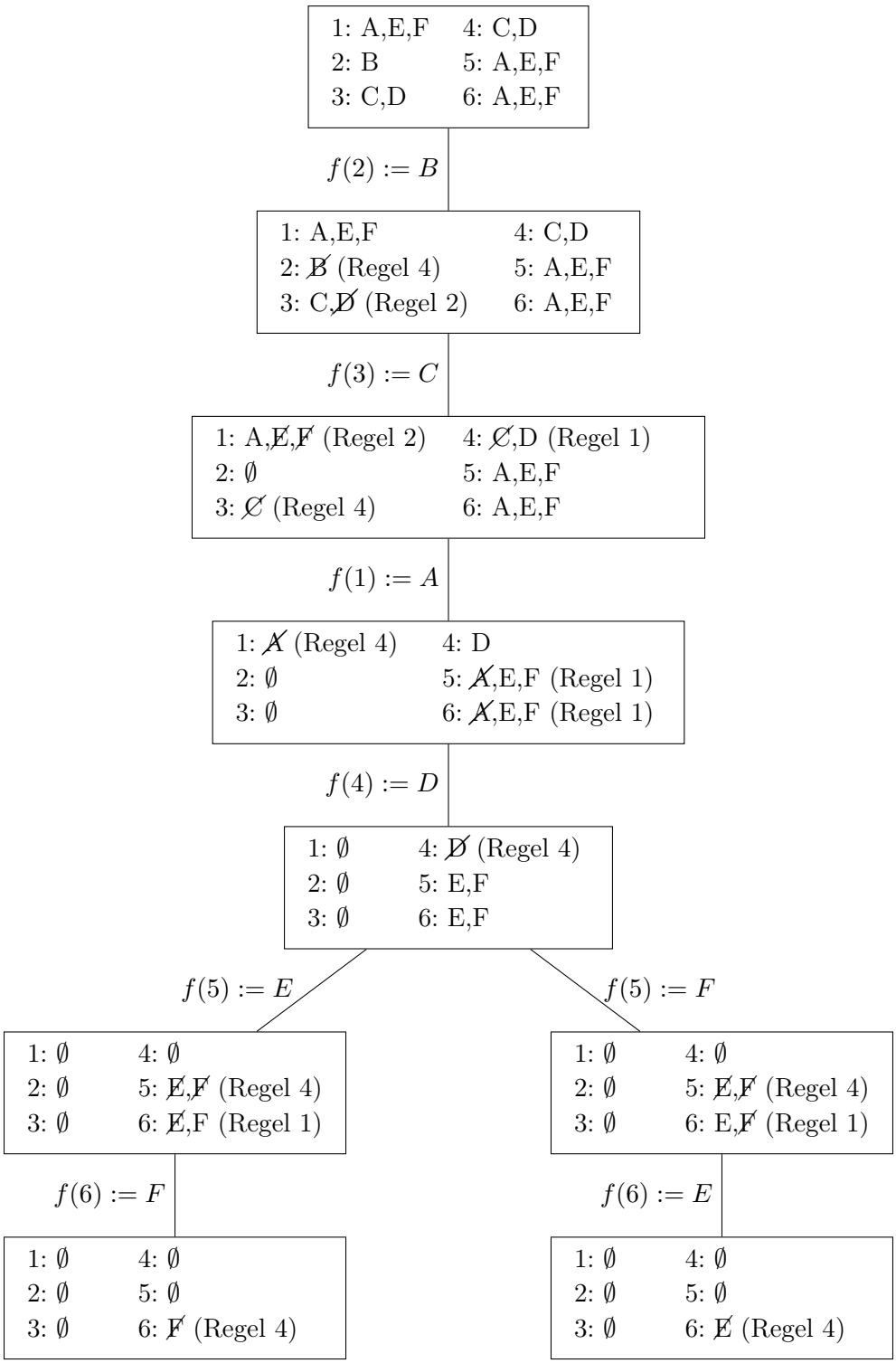


Abbildung 39: Vollständiger Entscheidungsbaum für G_1 und G_2. Jeder Knoten des Entscheidungsbaumes enthält die Kandidatenlisten can_1.

Möglichst kleine Kandidatenmengen führen daher unmittelbar zu einer geringen Anzahl rekursiver Aufrufe und damit einer geringen Laufzeit.

5.4.2.6. Beschreibung der Architektur von VF2++

Wie bereits am Anfang des Kapitels erwähnt, wird VF2++ als Vergleichsalgorithmus zu MET dienen. Daher werden wir nun kurz die Architektur von VF2++ vorstellen und diese mit der von MET vergleichen.
Der grundlegende Ablauf von VF2++ ist ähnlich zu MET. VF2++ enthält ebenfalls eine Isomorphiefunktion f, die rekursiv aufgebaut wird. Dazu wird in jedem Rekursionsschritt ein Knoten v aus V_1 gewählt und einem Knoten w aus V_2 zugewiesen. Ein Forward-Checking prüft dann, ob nach dieser Zuweisung noch eine vollständige Isomorphiefunktion f erzeugt werden kann. Falls nein, wird die Zuweisung v zu w rückgängig gemacht und ein anderes Paar (v, w) versucht. Fällt jedoch das Forward-Checking positiv aus, führt VF2++ die Rekursion aus.
Im Unterschied zu MET enthält VF2++ für das Forward-Checking komplexere Forward-Checking-Regeln, die dort Cutting-Regeln genannt werden. Außerdem vollzieht VF2++ eine komplexere Auswahl des Knotens v im Rekursionsschritt. Während MET v so wählt, dass $|can1(v)|$ minimal ist und dann die Knoten w aus $can1(v)$ in beliebiger Reihenfolge durchläuft, bestimmt VF2++ eine fest Reihenfolge der Knoten V_1 initial. Dies geschieht über eine viel komplexere Strategie, die verschiedene Breitensuchen im Molekülgraph umfassen und die Labels sowie den Grad der Knoten berücksichtigt. Zudem gibt es weitere sekundäre und tertiäre Auswahlkriterien. Weiterhin unterscheidet sich VF2++ zu MET darin, dass es keine expliziten Kandidatenmengen und -reduktionsregeln verwendet, sondern die Kandidaten ad hoc neu bestimmt werden.

5.4.3. Zusammenfassung der Architektur von MET

Nachdem wir im Detail den Algorithmus von MET betrachtet haben, möchten wir die wichtigsten Eigenschaften dazu noch einmal zusammenfassen. MET besteht aus zwei Phasen. Diese umfassen den Aufbau des Molekülgraphen mit geeigneten Knoten- und Kantenlabeln, die die Atom- und Struktureigenschaften widerspiegeln. Ein Vortest überprüft, ob zwei Molekülstrukturen in der Summe ihrer Moleküleigenschaften übereinstimmen. Ein sich anschließender Äquivalenztest versucht die Knoten einander zu zuordnen. Das Augenmerk dieses Algorithmus liegt dabei in der Bestimmung der Nachbarschaftsdeskriptoren und der ständigen Reduzierung der Kandidatenmenge in jedem Schritt. Damit unterscheidet sich MET klar von der Kandidatenauswahl aus VF2++. Die folgenden Kapitel untersuchen die Einflüsse des Backtracking-Algorithmus sowie der Kandidatenreduktion in MET. Weiterhin führen wir Experimente zur Bestimmung der optimalen Nachbarschaftstiefe durch und vergleichen die Ergebnisse von MET mit existierenden Methoden.

5.5. Experimente zur Analyse der Kandidatenreduktion

Bevor wir `MET` mit anderen Methoden vergleichen, untersuchen wir in diesem Kapitel den Effekt des Algorithmus zur Kandidatenreduktion auf die Größe der entstehenden Entscheidungsbäume. Damit einhergehend werden wir die Auswirkungen auf die Laufzeit von `MET` untersuchen.
Um den Effekt der Kandidatenreduktion zu bewerten, zählen wir die Anzahl der Knoten des Entscheidungsbaum einmal mit allen beschriebenen Regeln aus Kapitel 5.5.2.5 zur Kandidatenreduktion, und einmal ohne die Regeln zwei und vier aus Kapitel 5.5.2.5. Die Gesamtanzahl der Knoten in diesem Baum entspricht einer unteren Schranke für die Gesamtlaufzeit des Backtracking-Algorithmus im schlechtesten Fall.
In den folgenden drei Experimenten verwenden wir dafür 228.812 Molekülstrukturen aus der PubChem Datenbank. Die Größe dieser Molekülstrukturen reicht für diese Experimente von drei Atomen bis über 399 Atomen. Für die folgenden Experimenten vergleichen wir jeweils die Rekursionsaufrufe nur zwischen den Molekülstrukturen, von denen wir wissen, dass sie zueinander äquivalent sind.

Experiment 5.1: In diesem ersten Experiment möchten wir den Einfluss der Reduzierung der Kandidaten quantifizieren. Dazu bestimmen wir experimentell die Gesamtanzahl von Knoten des Entscheidungsbaum mit und ohne die Regeln zwei und vier zur Kandidatenreduktion. Das Experiment baut somit vollständige Entscheidungbäume auf, wo mehrere äquivalente Zuordnungen enthalten sind (Vergleich Abbildung 39). Der Quotient aus beiden Größen beschreibt uns den Effekt der Kandidatenreduktion. Abbildung 40 zeigt das Ergebnis dieses Experiments.

Die Abbildung 40 visualisiert uns, dass die Kandidatenreduktion mit den Regeln zwei und vier die Anzahl der Knoten des Entscheidungsbaumes um einen exponentiellen Faktor reduziert. Die implementierte Kandidatenreduktion ist damit ein unverzichtbares Mittel zur Reduktion der Laufzeit von `MET`.

Experiment 5.2: Im vorangegangenen Experiment sind wir vereinfachend davon ausgegangen, dass der Algorithmus stets den gesamten Entscheidungsbaum aufbaut. Im Falle der Isomorphie bricht die Rekursion jedoch häufig deutlich früher ab, sobald er nämlich eine Isomorphie feststellt, da eine äquivalente Zuordnung der Knoten genügt.
Wir untersuchen daher im nächsten Schritt experimentell, welcher Anteil des Entscheidungsbaums typischerweise durchsucht werden muss, bis die Rekursion abbricht. Dazu bestimmen wir einerseits die Knotenzahl des vollständigen Entscheidungsbaums sowie die Anzahl der besuchten Knoten bis zum Abbruch der Rekursion (laut

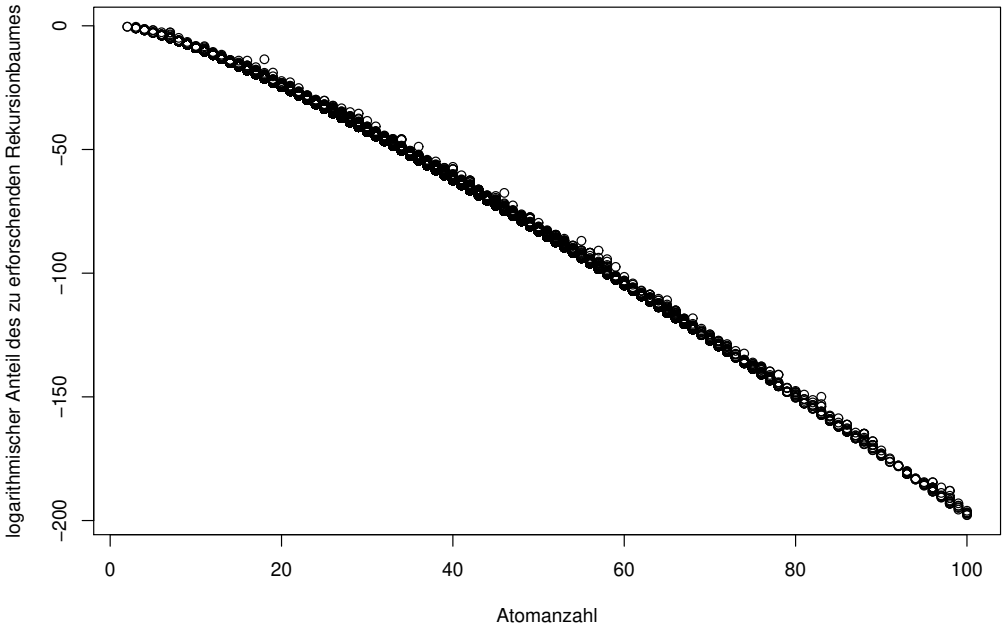


Abbildung 40: Darstellung des Anteils des Entscheidungsbaum mit und ohne die Regeln zwei und vier aus dem Algorithmus 3 in Abhängigkeit zur Molekülgröße. Der Anteil berechnet sich aus dem Quotienten zwischen der Anzahl der Rekurisonsaufrufe aus Algorithmus 3 mit den Regeln zwei und vier und der Anzahl der Rekursionsaufrufe ohne die Regeln zwei und vier.

Algorithmus 2) und bilden wieder den Quotienten beider Größen. Dieser Quotient liegt zwischen null und eins. Werte nahe eins bedeuten, dass der Entscheidungsbaums fast vollständig durchsucht werden muss. Je kleiner der Quotient ist, desto größer ist der Effekt des frühzeitigen Abbruchs.

In Abbildung 41 sehen wir das Ergebnis dieses Experiments. Wir erkennen, dass für Molekülstrukturen bis 40 Atome oftmals der vollständige Entscheidungsbaum durchsucht wird, um die Äquivalenz zu bestimmen. Das liegt daran, dass durch die kleine Atomanzahl nur geringe Kandidatenmengen vorliegen und diese bei der Zuordnung vollständig betrachtet werden. Daher entspricht bei kleinen Molekülstrukturen der durchsuchte Teil des Entscheidungsbaum fast dem vollständigen Entscheidungsbaum.
Für die Molekülstrukturen mit mindestens 41 bis 300 Atomen muss oftmals weniger als 75 % des Entscheidungsbaums durchlaufen werden. Noch stärker sehen wir diese Tendenz bei Molekülstrukturen mit mehr als 300 Atomen. Hier werden nur 2 % des Entscheidungsbaums durchsucht. Das lässt die Schlussfolgerung zu, dass für eine Vielzahl der Fälle nicht der vollständige Entscheidungsbaum durchsucht werden

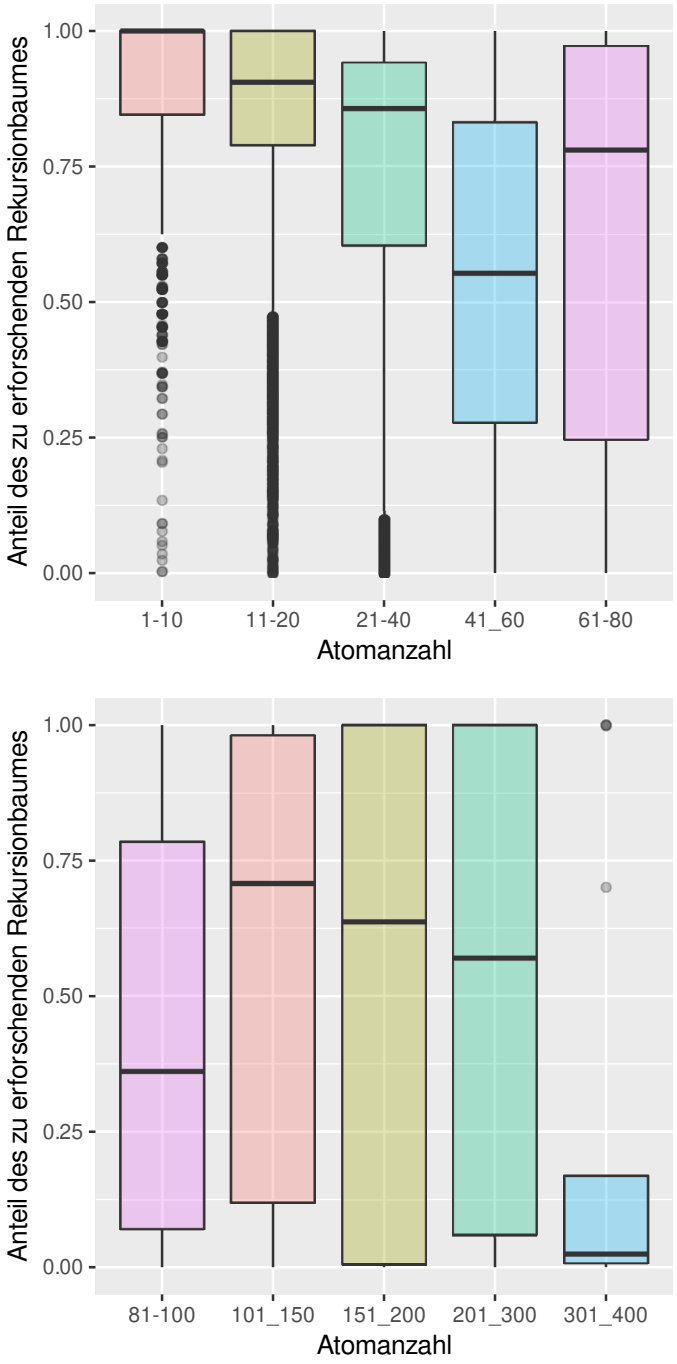


Abbildung 41: Darstellung des Anteils des zu erforschenden Entscheidungsbaums mit frühzeitigen Abbruch der Rekursion im Vergleich zum vollständigen Testen der Kandidatenmenge in Abhängigkeit zur Molekülgröße. Der Anteil berechnet sich aus dem Quotienten zwischen der Anzahl der Rekurisonsaufrufe mit frühzeitigen Abbruch und aus der Anzahl im vollständigen Entscheidungsbaums.

muss, um die Isomorphie festzustellen und ein vorzeitiger Abbruch gerechtfertigt ist.

Experiment 5.3: Nachdem wir in den vorherigen zwei Experimenten erkannt haben, dass die Anwendung der Kandidatenreduktion nach Algorithmus 2 sowie der Rekursionsabbruch in Algorithmus 3 eine positive Auswirkung auf die Laufzeit haben, betrachten wir nun die absolute Anzahl der Rekursionsaufrufe in Abhängigkeit von der Molekülgröße (siehe Abbildung 42).

Abbildung 42 lässt uns schlussfolgern, dass häufig eine lineare Anzahl von Rekursionsaufrufen ausreicht, um die Isomorphie zu finden. Bei dem zu sehenden Ausreißer handelt es sich um die Struktur aus Abbildung 43. Diese Struktur besteht fast vollständig aus kombinierten Benzen-Ringen. Das führt innerhalb von `MET` dazu, dass die Knoten nicht ausreichend divers durch die Atomeigenschaften und den Nachbarschaftsdeskriptor unterschieden werden können. Die Folge sind große Kandidatenmengen und damit Entscheidungsbäume mit vielen Knoten in einer Ebene. Dadurch steigt die Anzahl der Backtracking-Schritte stark an.

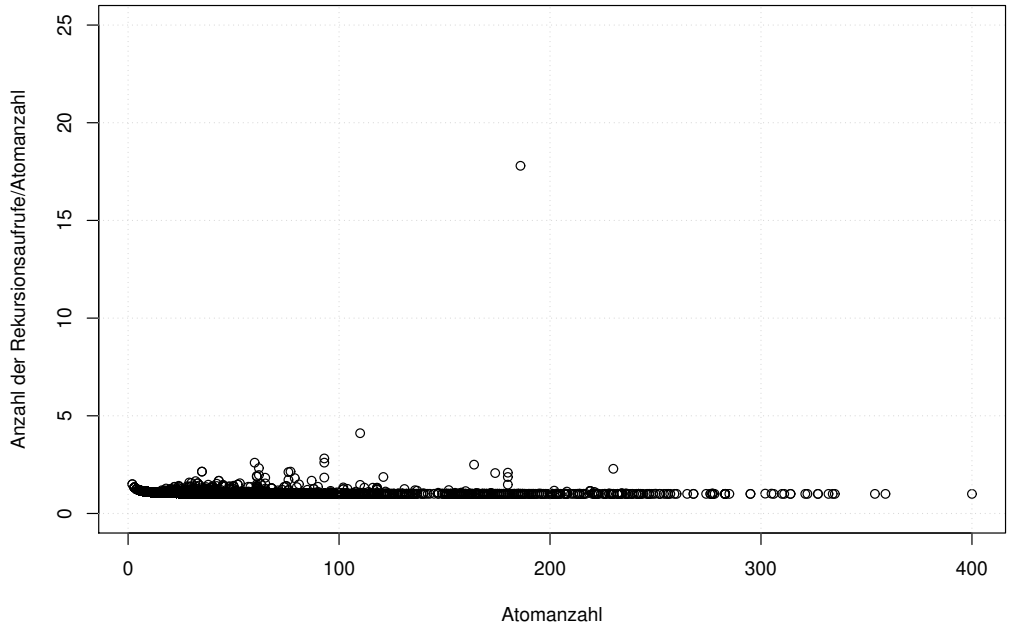


Abbildung 42: Skalierung der Rekursionsaufrufe in Abhängigkeit zur Atomananzahl.

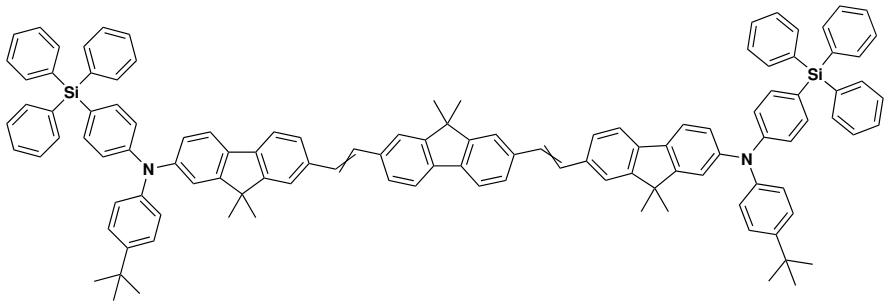


Abbildung 43: Die Molekülstruktur (CID: 72242453) weist 226 Rekursionsaufrufe bei einer Atomanzahl von 121 auf.

Zusammenfassung der Analyse zur Kandidatenreduktion In diesem Kapitel führten wir eine Analyse zur Auswirkung der Kandidatenreduktion für `MET` durch. Ohne die eingeführte Reduktion der Kandidatenmengen (Algorithmus 3) hätte `MET` einen exponentiellen Anstieg der Laufzeit in Abhängigkeit zur Molekülgröße. Damit wäre eine effiziente Überprüfung der Äquivalenz zweier Molekülstrukturen nicht umsetzbar. Weiterhin können wir aus dem Experiment 5.2 schlussfolgern, dass `MET` im Großteil aller Fälle nicht den vollständigen Entscheidungsbaum durchlaufen muss, um die Isomorphie zweier Molekülstrukturen festzustellen. Das verdeutlicht die Effizienz des Algorithmus 2. Im letzten Experiment evaluierten wir sogar, dass die Anzahl der Rekursionsaufrufe linear zur Atomanzahl ist. Wir sehen hier damit, dass wir mit dem Ansatz der Kandidatenbestimmung zu Beginn von `MET` und der integrierten Kandidatenreduktion einen leistungsfähigen Algorithmus entwickeln konnten. Nach diesen Erkenntnissen werden wir jetzt dazu übergehen, `MET` mit existierenden Methoden zu vergleichen und dabei eine optimale Nachbarschaftstiefe zu bestimmen.

5.6. Experimente zur Laufzeit und Korrektheit von `MET`

Dieses Kapitel enthält im ersten Abschnitt ein Experiment zur Bestimmung der optimalen Nachbarschaftstiefe k. Anschließend führen wir mehrere Experimente durch, um die Laufzeit und Effizienz von `MET` mit `VF2++` und `SMSD` zu vergleichen. Zusätzlich überprüfen wir die Korrektheit von `MET` durch den Vergleich der Ergebnisse mit InChI‘s und SMILES.

Um die Performance der Implementierung von `MET` mit anderen existierenden Methoden zu quantifizieren, wurde ein Benchmark Datensatz erstellt, dessen Aufbau wir nun betrachten.

Benchmark Datensatz Als Benchmark Datensatz verwenden wir die Molekülstrukturen der PubChem Datenbank. Der Datensatz der PubChem enthält eine

Vielzahl von SDF-Dateien, in denen jeweils 25.000 Molekülstrukturen gespeichert sind. Für die Vorbereitung der folgenden Experimente werden die Molekülstrukturen aus den SDF-Dateien in mehrere Gruppen unterteilt. Um diese Gruppen genauer beschreiben zu können, führen wir den Begriff der *Äquivalenzfamilien* und *Äquivalenzklassen* ein.

Eine Äquivalenzklasse enthält nur äquivalente Molekülstrukturen. Daher sind die Fingerprints der Moleküle identisch und der Isomorphietest gibt eine äquivalente Zuordnung der Atome der Moleküle zurück. Jede Äquivalenzklasse wird durch den Fingerprint eines Moleküls repräsentiert, den wir daher *Repräsentanten* nennen. Eine Äquivalenzfamilie besteht aus mehreren (mindestens einer) Äquivalenzklassen, die sich durch den gleichen Fingerprint des Repräsentanten auszeichnen. Deshalb charakterisiert sich eine Äquivalenzfamilie durch genau diesen einen Fingerprint.

Bei den Gruppen meinen wir daher die Äquivalenzklassen. Der allgemeine Ablauf für die Erstellung der Äquivalenzklasse ist wie folgt:

1. Die Moleküldarstellungen werden eins nach dem anderen aus der Eingabedatei gelesen und in einen Graphen $G = (V, E)$ mit den Knotenlabeln l umgewandelt.

2. Um G in seine Äquivalenzklasse einzuordnen, wird die Menge an Eigenschaften $\{l(v) : v \in V\}$ verwendet und der Fingerprint gebildet. Darauf basierend wird eine Liste mit Repräsentanten gesammelt, die die gleichen Eigenschaftsmengen haben und dadurch möglicherweise äquivalent zu G sind. Diese Liste wird L_G genannt und enthält die Repräsentanten der Äquivalenzklassen der Äquivalenzfamilie.

3. Für jeden Repräsentanten $R \in L_G$, wird getestet, ob G äquivalent zu R ist. Dazu sind in `MET` die folgenden Algorithmen/Ansätze implementiert:
   - `MET`: Verwendung des Isomorphie-Algorithmus aus Kapitel 5.4.2.
   - `VF2++`: Verwendung des `VF2++`-Algorithmus aus der Lemon-Bibliothek [10]
   - `CDKMCS, MCSPlus, Vlib, Default` aus dem `SMSD` Paket [97]
   - `CDK:SMILES`: Verwendung von `CDK` zur Erstellung kanonischer SMILES [16]
   - `RDKit:SMILES`: Verwendung von `RDKit` zur Erstellung kanonischer SMILES [109]
   - `CDK:InChI`: Verwendung von `CDK` zur Erstellung der InChI-Repräsentation [16]

4. Ist G zu keinem Repräsentanten aus L_G äquivalent, wird eine neue Äquivalenzklasse mit G als Repräsentanten gebildet.

5. Ist der Fingerprint von G zu keinem Fingerprint einer Äquivalenzfamilie gleich, wird eine neue Äquivalenzfamilie mit einer Äquivalenzklasse und G als Repräsentanten gebildet.

Alle Experimente werden auf einem Debian/Linux 10 System mit 38 CPU-Kernen und 250 GB Hauptspeicher ausgeführt. Die Implementierungen verwenden Java 12, `CDK` 2.3, `RDKit` 2018.09.1 (Ubuntu Paket) und `SMSD` 2.2.0.

5.6.1. Parameterermittlung

In den ersten Experimenten haben wir das Ziel den optimalen Wert für den Parameter der Nachbarschaftstiefe k zu finden. Wir erinnern uns, der Nachbarschaftsdeskriptor spiegelt die Atomeigenschaften von Atomen aus einem festgelegten Radius um ein Atom wider. Dadurch besteht in `MET` die Möglichkeit, Atome mit gleichen Atomeigenschaften durch ihre Nachbarschaft zu unterscheiden. Weiterhin werden wir evaluieren, welcher Graphisomorphie-Algorithmus in der zweiten Phase von `MET` die schnellste Laufzeit liefert.

Experiment 5.4: Dieses Experiment untersucht den Einfluss der Nachbarschaftstiefe k auf die Laufzeit von `MET` und `VF2++`. Dazu messen wir die Laufzeit des Benchmark-Experiments mit mehreren Werten für $2 \leq k \leq 20$. Das Experiment erhält als Eingabe 95.945.527 Molekülstrukturen aus der PubChem Datenbank (Stand 12.04.2022).

Abbildung 44 zeigt das Ergebnis dieses Experiments. Wir erkennen, dass kleine Werte mit $k < 6$ zu einer großen Laufzeit führen. Das ist nicht überraschend, da mit einem kleinen k die Knoten-Deskriptoren nur wenig Informationen zur lokalen Nachbarschaft jedes Atoms enthalten. Im Gegenzug sehen wir, dass ab einer Tiefe von $k > 6$ es zu keinem signifikanten Anstieg oder Fall der Laufzeit kommt. Die Laufzeit bleibt ansatzweise konstant für $k \geq 6$. Basierend auf diesen Beobachtungen, schlussfolgern wir, dass $k = 6$ ein angemessener Wert für diese Anwendung ist. Diese Schlussfolgerung gleicht ähnlich publizierten Experimenten von Boyle [110] und Probst [111], die herausfanden, dass eine Beachtung von vier bis sechs Bindungen um ein Atom die beste Performance zur Darstellung kleiner organischer Moleküle als Molekül-Fingerprints liefert.
Eine weitere Beobachtung des Experiments 5.4 ist, dass der Algorithmus von `MET` den von `VF2++` für alle k in der Performance übertrifft. Für $k = 6$ ist `MET` drei Stunden schneller als `VF2++`. Ein Grund könnte sein, dass `VF2++` in C++ implementiert ist und zusätzliche Aufrufe des Java Native Inferfaces notwendig sind. Wir können daher schlussfolgern, dass `MET` gegenüber `VF2++` leicht überlegen ist.

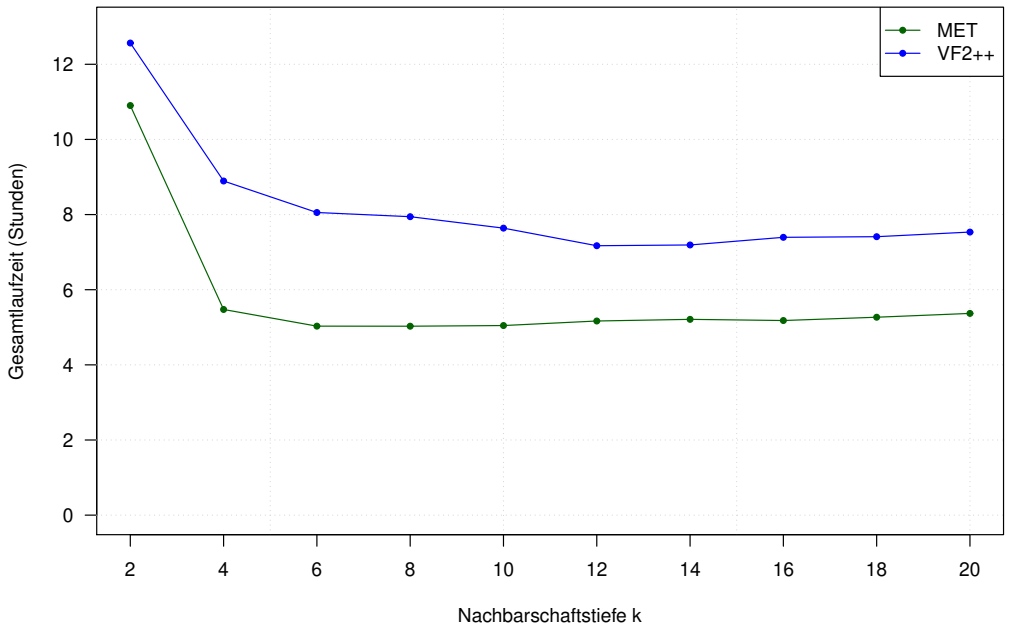


Abbildung 44: Einfluss des Parameters k der Nachbarschaftstiefe auf die Laufzeit der Algorithmen `MET` und `VF2++` (angelehnt an [96]).

Experiment 5.5: Das zweite Experiment studiert im Detail ob `MET` oder `VF2++` in der zweiten Phase des Äquivalenztest am besten arbeitet. Dazu wurden die Molekülstrukturen des Benchmark Datensatzes in sieben Gruppen entsprechend ihrer Molekülgröße unterteilt.

- Gruppe 1: 56.410.359 Molekülstrukturen mit bis zu 25 Atomen,
- Gruppe 2: 35.528.631 Molekülstrukturen mit 26 bis 45 Atomen,
- Gruppe 3: 3.773.370 Molekülstrukturen mit 46 bis 100 Atomen,
- Gruppe 4: 168.714 Molekülstrukturen mit 101 bis 150 Atomen,
- Gruppe 5: 40.251 Molekülstrukturen mit 151 bis 200 Atomen,
- Gruppe 6: 22.745 Molekülstrukturen mit 201 bis 400 Atomen, und
- Gruppe 7: 1.457 Molekülstrukturen mit mehr als 400 Atomen.

Ziel des Experiments ist die Messung der Laufzeiten von `MET` und `VF2++` für jede einzelne Gruppe. Für jede Gruppe lief dazu das Experiment individuell ab. Anschließend wurde für jede Molekülgruppe ein Quotient aus beiden Laufzeiten gebildet ($\frac{\texttt{MET}}{\texttt{VF2++}}$), um den Speed-Up zu bestimmen. Ein Quotient von eins besagt, dass beide Algorithmen etwa gleich schnell sind. Ein Quotient kleiner als eins besagt, dass `MET` schneller ist als `VF2++`.

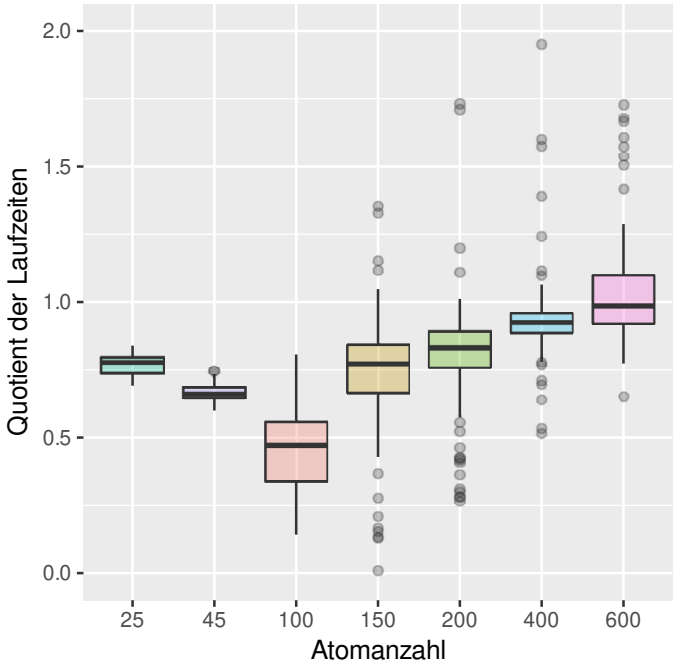


Abbildung 45: Laufzeit von `MET` geteilt durch die Laufzeit von `VF2++` (mit $k = 6$). Die Mediane sind mit dicken horizontalen Linien markiert (angelehnt an [96]).

Abbildung 45 zeigt das Ergebnis des Experiments. Wir erkennen, dass `MET` im Mittel für jede Gruppe `VF2++` überlegen ist, da die Linien zum Median unterhalb von eins liegen. Es sind jedoch auch Ausreißer in beiden Richtungen zu erkennen. Zusätzlich stellen wir fest, dass `MET` besonders für einen Großteil der kleinen Molekülstrukturen schneller ist als `VF2++`. Hingegen sinkt die Effizienz bei steigender Atomanzahl (über 400 Atome). Ob diese Entwicklung abhängig vom Nachbarschaftsparameter k ist, möchten wir direkt im Anschluss analysieren.

Dafür werden wir anhand der Gruppierung die Bestimmung der Nachbarschaftstiefe k für die sieben verschiedenen Molekülgrößen bewerten. Das Experiment misst die Laufzeit von `MET` und `VF2++` mit $2 \leq k \leq 20$ individuell für die Gruppen eins, drei und sieben. Dadurch werden kleine, mittlere und große Molekülstrukturen im Experiment widergespiegelt. Auszugsweise stellen die Abbildungen 46 bis 48 die Laufzeitanalyse in Abhängigkeit von k für die Gruppen eins, drei und sieben dar. Eine Übersicht der weiteren vier Molekülgrößengruppen befindet sich im Anhang in den Abbildungen S-15 bis S-18.
Für die kleinen Molekülstrukturen sehen wir nur geringe Unterschiede in der Laufzeit in Abhängigkeit von k. Den deutlichsten Unterschied erkennen wir in der Gruppe drei. Hier sinkt die Laufzeit ab einem $k = 6$ deutlich und verändert sich nicht

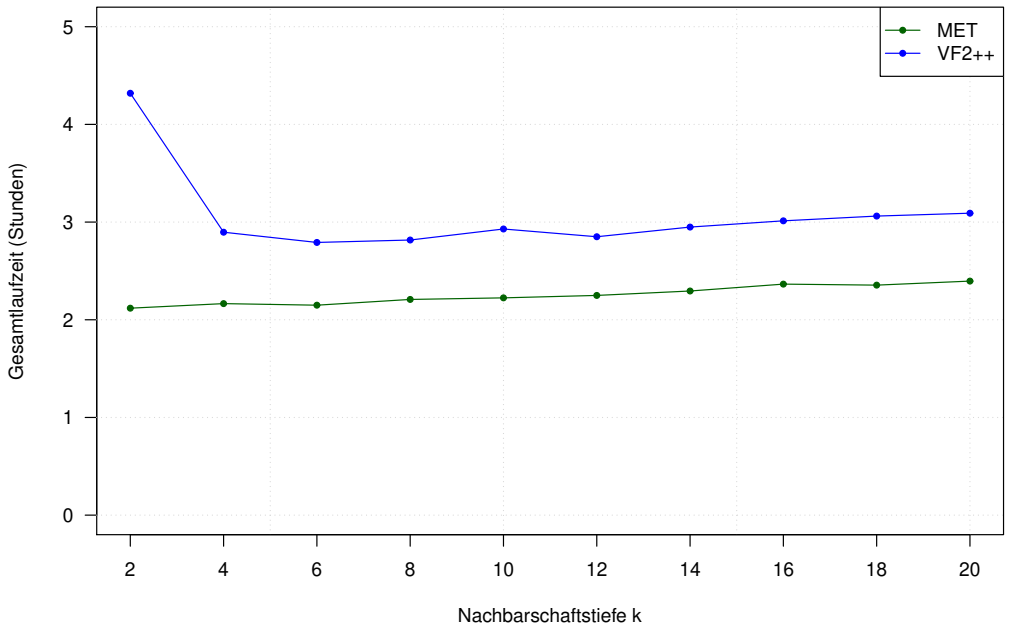


Abbildung 46: Laufzeitanalyse der Gruppe eins für $1 \leq k \leq 20$.

signifikant bei ansteigendem k. Dieser Kurvenverlauf gleicht sich mit dem aus Abbildung 44. Betrachten wir noch die Gruppe sieben, ist bei den Werten für $k > 6$ ebenfalls keine signifikante Laufzeitveränderung erkennbar.

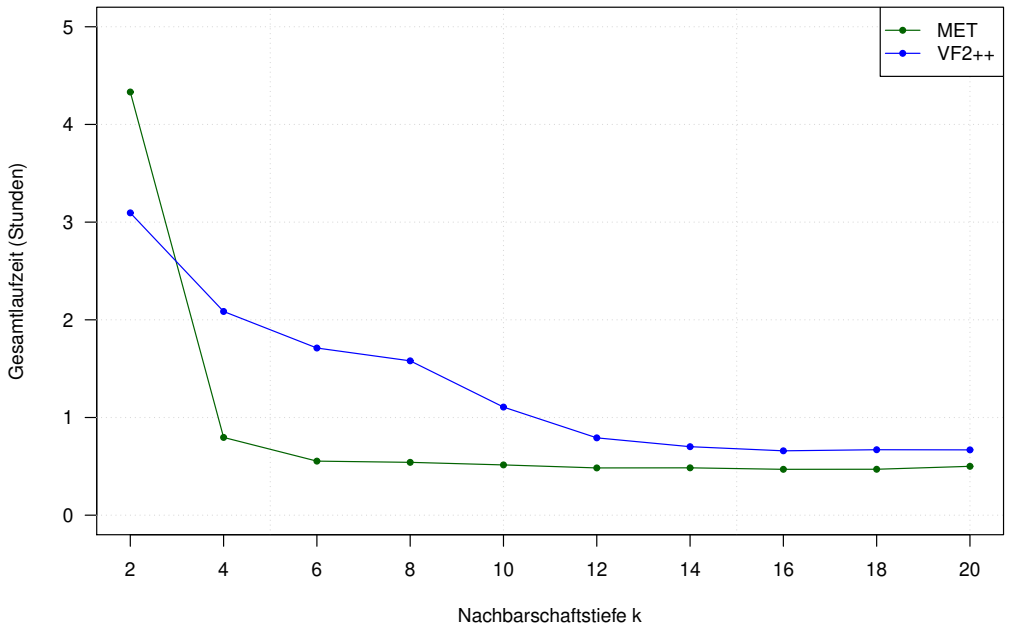


Abbildung 47: Laufzeitanalyse der Gruppe drei für $1 \leq k \leq 20$.

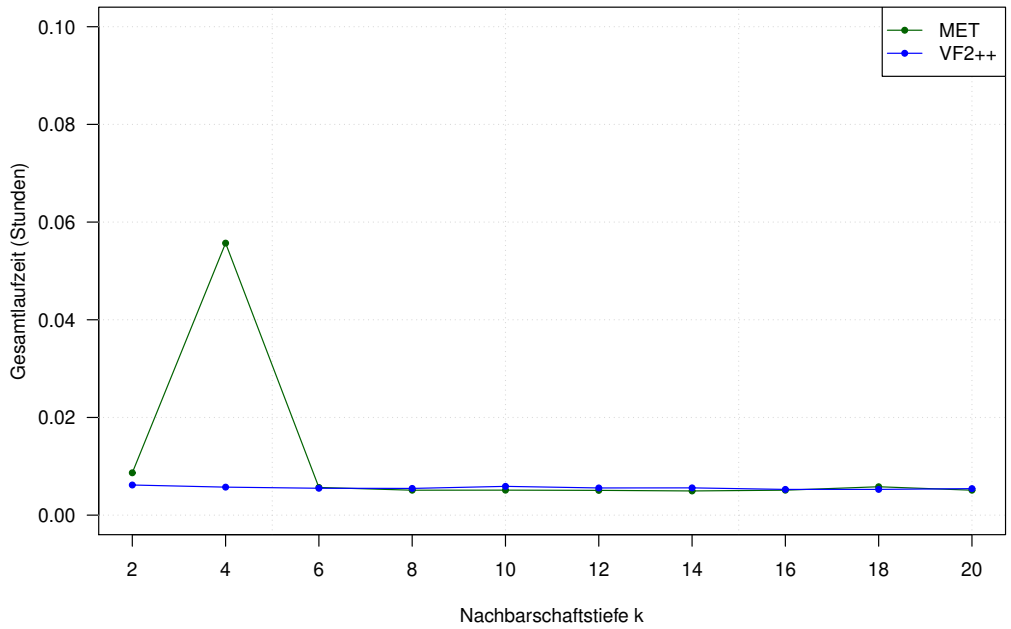


Abbildung 48: Laufzeitanalyse der Gruppe sieben für $1 \leq k \leq 20$.

Tabelle 17: Auflistungen der Laufzeit für die Gruppe 7 (Atomanzahl größer 400) für $50 \leq k \leq 400$. Die Laufzeit ist in Sekunden angegeben.

k	6	50	100	150	200	400	600
Laufzeit	31,9	33,0	32,0	34,9	34,5	39,0	42,9

Daher möchten wir für diese Gruppe zusätzlich quantifizieren, ob ein signifikant größeres k die Laufzeit reduzieren könnte. Dazu betrachten wir die Laufzeiten in Tabelle 17, die mit $6 \leq k \leq 600$ berechnet wurden.
Betrachten wir zunächst den Bereich $6 \leq k \leq 200$ sehen wir, dass hier keine deutliche Veränderung der Laufzeit vorliegt. Von $k = 6$ zu $k = 200$ erhöht sich die Laufzeit nur um 2,4 Sekunden. Im Gegensatz dazu liegt der Unterschied bei $k = 600$ bei 11,0 Sekunden. Wir können damit quantifizieren, dass eine größere Betrachtung der Nachbarschaftsbeziehung durch eine größere Nachbarschaftstiefe nicht zu einer Reduzierung der Laufzeit führt, sondern, dass die Wahl von $k = 6$ auch für größere Molekülstrukturen zu bevorzugen ist.

Wir schlussfolgern daher, dass für alle Molekülgrößen die Nachbarschaftstiefe mit $k = 6$ ein optimaler Wert ist. Zusätzlich ziehen wir die Schlussfolgerung, dass `MET` besonders für Molekülstrukturen bis zu einer Größe von 400 Atomen eine schnellere Laufzeit aufweist als `VF2++`. Daraus resultiert als offene Frage, welchen Anteil an der Laufzeit die Isomorphielalgorithmen der beiden Programme an der Phase zwei

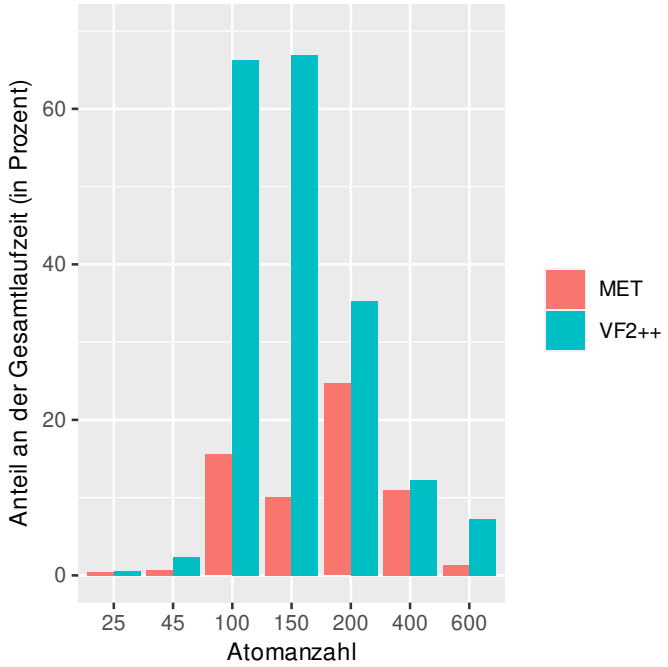


Abbildung 49: Anteil der Isomorphie-Algorithmen `MET` und `VF2++` (Phase 2) an der Gesamtlaufzeit in Prozent für jede Molekülgruppe, mit $k = 6$ (angelehnt an [96]).

haben.

Experiment 5.6: Basierend auf Experiment 5.5 möchten wir daher im nächsten Schritt überprüfen, wie groß der Anteil der Laufzeit des Äquivalenztestes an der Gesamtlaufzeit ist. Dazu verwenden wir wieder den Datensatz aus Experiment 5.5 und messen separat die Laufzeit der zweiten Phase. Diese Laufzeit geteilt durch die Gesamtlaufzeit ergibt den `Anteil des Isomorphie-Algorithmus` (inklusive des Vortests) an der Gesamtlaufzeit. Abbildung 49 stellt das Ergebnis des Experiments dar.

Es ist klar ersichtlich, dass der Anteil des Isomorphie-Algorithmus von `MET` für alle Molekülgruppen kleiner ist als der von `VF2++`. Im Besonderen für die Molekülstrukturen mit 50 bis 200 Atomen ist der Algorithmus von `MET` signifikant schneller. Dieses Ergebnis unterstreicht damit das Resultat von Experiment 5.5 und damit den Einsatz von `MET` für Molekülstrukturen bis 400 Atomen. Betrachten wir die Molekülgruppen drei und vier nochmals genauer, sind die Laufzeitunterschiede zwischen den Algorithmen `MET` und `VF2++` dort am ausgeprägtesten. Dieser Unterschied spiegelt sich ebenfalls in dem Ergebnis von Experiment 5.5 wider. Ein Grund für diesen Unterschied könnte die optimale Berechnung des Nachbarschaftsdeskriptors ($k = 6$) sein.

Dadurch können die Atome und Molekülstrukturen relativ eindeutig durch einen Repräsentanten, beispielsweise einen Fingerprint, abgebildet werden und eine schnelle Entscheidung zur Äquivalenz getroffen werden.

Wir schlussfolgern aus diesem Experiment, dass der Isomorphiealgorithmus aus `VF2++` der laufzeitbestimmende Faktor für mittelgroße Molekülstrukturen ist. Hingegen ist bei `MET` die Vorverarbeitung und insbesondere die Bestimmung der Nachbarschaftsumgebung laufzeitbestimmend. Um einen noch genaueren Vergleich über die Performance der Isomorphiealgorithmen zu erhalten, müssten isomorphe Molekülstrukturen verglichen werden.

Experiment 5.7 Wir wollen die abschließende Idee aus dem vorherigen Experiment aufgreifen und die Äquivalenz für isomorphe Molekülstrukturen untersuchen. `MET` konnte in den vorherigen Experimenten viele Nicht-Äquivalenzen bereits durch ein Fehlschlagen des Vortest oder des Fingerprint-Vergleichs feststellen. Deshalb bewerten wir in diesem Experiment die Effizienz der Algorithmen von `MET` und `VF2++` an einem Datensatz bei dem von vornherein bekannt ist, dass nur äquivalente Molekülstrukturen vorhanden sind.
Dazu wurde aus jeder Molekülgrößen-Gruppe eine Teilmenge an Äquivalenzklassen als Eingabe für das Benchmark Datensatz selektiert. Dadurch ist sicher gestellt, dass alle Paare an zu testenden Molekülstrukturen äquivalent sind.

Das Experiment misst nun die Laufzeit für diesen neuen Datensatz und zeigt das Ergebnis in Abbildung 50. Der Quotient bildet sich wieder aus der Laufzeit von `MET` und `VF2++`. Wir erkennen auch wieder ein ähnliches Verhalten wie im Experiment 5.5. Das bedeutet, dass beide Algorithmen in etwa gleich gut für sehr kleine als auch sehr große Molekülstrukturen arbeiten. Der Unterschied zwischen `MET` und `VF2++` liegt wieder in der Überlegenheit von `MET` für die mittelgroßen Molekülstrukturen. Damit ist die offene Frage aus dem Experiment 5.6 beantwortet, ob die Überlegenheit von `MET` nur auf das schnelle Erkennen einer nicht vorhandenen Isomorphie begrenzt ist oder wie hier dargestellt sich auch auf äquivalente Molekülstrukturen bezieht.

Zusammenfassung der Parameterbestimmung Die Experimente 5.4 bis 5.7 zeigen, dass der Isomorphiealgorithmus aus `MET` dem von `VF2++` für kleine bis mittelgroße Molekülstrukturen überlegen ist. Das spricht dafür, dass für den Bereich bis 400 Atome im Molekül (mit impliziten Wasserstoffen) `MET` als Isomorphiealgorithmus eingebunden wird. Für diesen Algorithmus stellte sich heraus, dass eine Bestimmung der Nachbarschaftstiefe von sechs ein guter Kompromiss zwischen der Laufzeit und dem Informationsgewinn darstellt. Damit gehen wir einher mit Erkenntnissen von

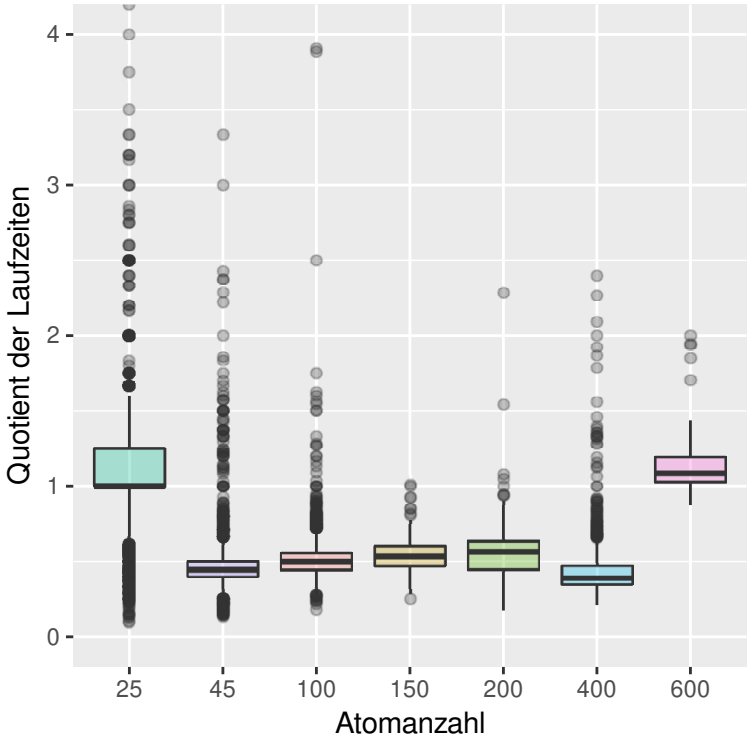


Abbildung 50: Laufzeit von `MET` geteilt durch die Laufzeit von `VF2++` (mit $k = 6$) auf einem Datensatz, der nur Isomere enthält. Die Mediane sind mit dicken horizontalen Linien markiert. (angelehnt an [96])

Boyle [110] und Probst [111] und zeigten, dass die Einbeziehung der erweiterten Nachbarschaft ebenfalls für den Äquivalenzvergleich von Molekülstrukturen sinnvoll ist. Nachdem wir die beiden Algorithmen `MET` und `VF2++` genauer in ihrer Laufzeit verglichen haben, möchten wir anschließend noch die korrekte Bestimmung der Äquivalenz durch `MET` untersuchen.

5.6.2. Vergleich mit etablierten chemoinformatischen Ansätzen

In den beiden abschließenden Experimenten werden wir evaluieren, wie sich der Algorithmus von `MET` im Vergleich zu den etablierten Äquivalenztests aus `SMSD` und den Methoden auf Basis der kanonischen SMILES und InChI aus `CDK` und `RDKit` verhält.

Experiment 5.8: Dieses Experiment wird zeigen, dass die Laufzeit einiger Äquivalenzalgorithmen sehr groß ist. Aus diesem Grund werden die Tests in diesem Experiment nicht mit dem kompletten Datensatz, sondern mit einem verkleinerten durchgeführt. Dieser Datensatz enthält daher eine kleine Teilmenge (23.254 Molekülstrukturen) der PubChem Datenbank, die in ihre Äquivalenzfamilien unterteilt ist. Das Experiment misst für die verschiedenen Methoden der Moleküläquivalenz, die in der

Tabelle 18: Laufzeit und Anzahl abweichender Ergebnisse für einen kleinen Datensatz an Molekülstrukturen ($k = 6$) (angelehnt an [96]).

Algorithmus	Millisekunden	# abweichende Ergebnisse
`MET`	7.693	0/692
`VF2++`	6.488	0/692
`SMSD: CDKMCS`	18.677	9/692
`SMSD: MCSPlus`	2.740.136	9/692
`SMSD: Vlib`	4.940.974	0/692
`SMSD: Default`	24.902.968	0/692
`RDKit: SMILES`	54.872	0/692
`CDK: SMILES`	7.399	0/692
`CDK: InChI`	6.563	0/692

Phase zwei ausgeführt werden können, die Laufzeit. Anschließend überprüft es, wo bei den erkannten Äquivalenzen falsch positive oder falsch negative Ergebnisse vorliegen. Tabelle 18 zeigt das Ergebnis dieses Experiments.

Anhand der Spalte der abweichenden Ergebnisse beobachten wir als erstes, dass die zwei Algorithmen (`CDKMCS`, `MCSPlus`) aus `SMSD` nicht alle äquivalenten Molekülpaare korrekt identifizieren. Beispielsweise werden die 2D-Strukturen (Abbildung 51) von Methadon-Hydrochlorid (PubChem CID: 14184) und Levomethadon hydrochlorid (PubChem CID: 22266) als unterschiedlich klassifiziert. Interessanterweise sind dies Strukturen ohne Radikale und Isotope.
Betrachten wir von den korrekten Algorithmen aus `SMSD` deren Laufzeit, stellen wir fest, dass die Laufzeit von `MET` um ein vielfaches kleiner ist als die aller Algorithmen

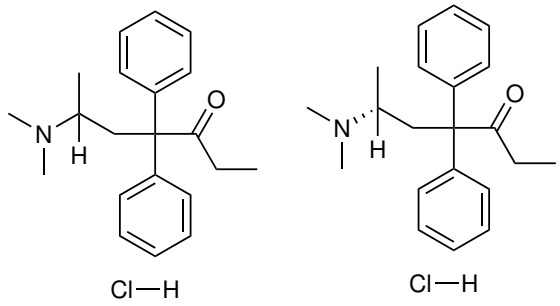


Abbildung 51: Darstellung der 2D-Strukturen von Methadon-Hydrochlorid (links, PubChem CID: 14184) und Levomethadon hydrochlorid (rechts, PubChem CID: 22266).

Tabelle 19: Laufzeit und Anzahl abweichender Ergebnisse für den PubChem Datensatz ($k = 6$) (angelehnt an [96]).

Algorithmus	Millisekunden	# abweichende Ergebnisse
`MET`	18.206.705	0/18.554.268
`VF2++`	29.093.114	0/18.554.268
`CDK: SMILES`	63.182.381	1.190/18.554.268
`CDK: InChI`	38.050.677	3.245/185.54.268

aus `SMSD`. Aus diesem Grund werden diese Algorithmen in weiteren Experimenten nicht mehr betrachtet. Das Experiment zeigt uns zusätzlich, dass die Äquivalenztests auf Basis der kanonischen SMILES und InChI korrekt sind. Zusätzlich beobachten wir, dass die Laufzeit von `RDKit` auffällig größer ist als die von `CDK`.

Aus diesem Experiment schlussfolgern wir somit, dass nur die SMILES und InChI basierenden Algorithmen aus `CDK` vergleichbar mit `MET` und `VF2++` sind. Deshalb untersuchen wir diese Algorithmen weiter auf dem großen Datensatz.

Experiment 5.9: Im letzten Experiment vergleichen wir die Laufzeit und die Korrektheit der SMILES und InChI Methoden aus `CDK` mit denen von `MET` und `VF2++`. Das Vorgehen ist das selbe wie aus Experiment 5.8. Jedoch werden diese Algorithmen auf dem PubChem Datensatz aus Experiment 5.5 ausgeführt. Die Tabelle 19 sowie die Abbildungen 52 und 54 zeigen das Ergebnis.

Betrachten wir die Laufzeit der InChI Methode aus `CDK` ist zu beobachten, dass `MET` um den Faktor zwei schneller ist als `CDK`.
Zusätzlich stellen wir fest, dass `CDK` abweichende Ergebnisse für 3.245 Molekülpaare produziert. Darin sind drei Paare enthalten, bei denen für mindestens eine Molekülstrukturen kein InChI durch `CDK` konstruiert wird. Bei allen verbliebenen Paaren handelt es sich um falsch positive Ergebnisse von `CDK`. Zum Beispiel erhalten die Molekülstrukturen mit der CID 18671247 und 60160843 (Abbildung 53) den selben InChI, obwohl sie sich in der Position der Doppelbindungen unterscheiden. Ein weiteres Beispiel sind die Molekülstrukturen mit CID 5151983 und 3799953, die sich in der Position des Protons unterscheiden, aber den selben InChI durch `CDK` erhalten.

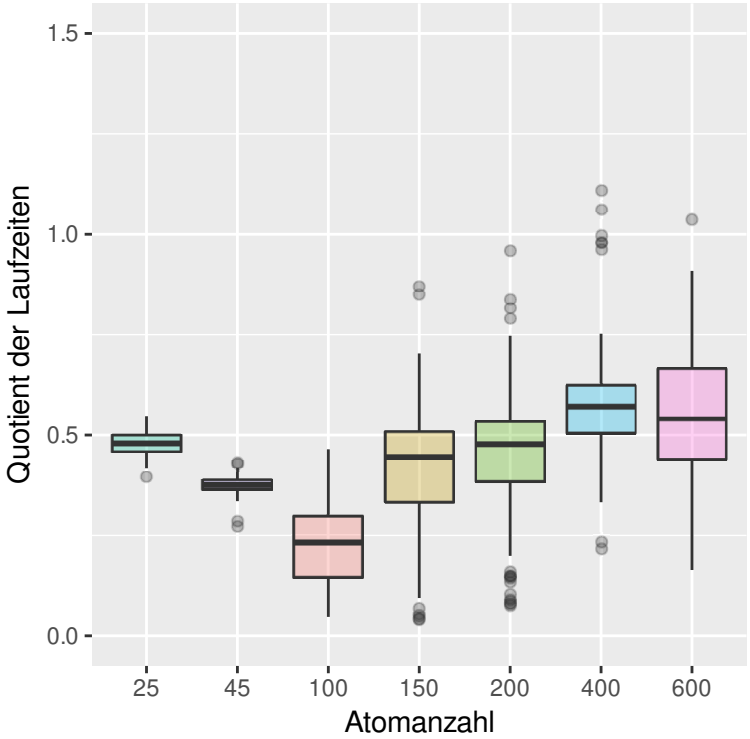


Abbildung 52: Laufzeit von `MET` (mit $k = 6$) geteilt durch die Laufzeit von InChI (`CDK`). Die Mediane sind mit dicken horizontalen Linien markiert. (angelehnt an [96])

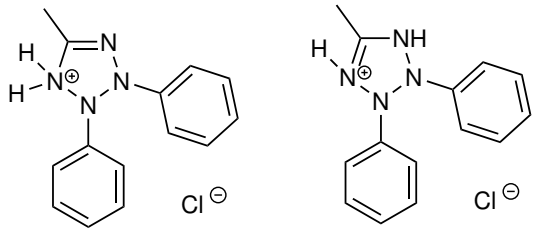


Abbildung 53: Die linke Abbildung zeigt die Struktur des Moleküls mit der CID 18671247 und die rechte Abbildung die des Moleküls mit CID 60160843. Der generierte InChI aus `CDK` ist `/InChI=1S/C14H14N4.ClH/c1-12-15-17(13-8-4-2-5-9-13)18(16-12)14-10-6-3-7-11-14;/h2-11H,1H3,(H,15,16);1H`

Bei der Auswertung der SMILES Methode erkennen wir, dass die Laufzeit von `MET` nur ein drittel der von `CDK` beträgt. Es gibt 1.190 der 18.554.268 Molekülpaare, für die die SMILES Methode unterschiedlichere Ergebnisse als `MET` liefert. Diese abweichenden Ergebnisse sind falsch negative von `CDK`. Zum Beispiel erstellt `CDK` für die Molekülstrukturen mit CID 414487 und 49791694 (Abbildung 55) verschiedene SMILES. Betrachten wir im Vergleich dazu die SMILES aus der PubChem oder die

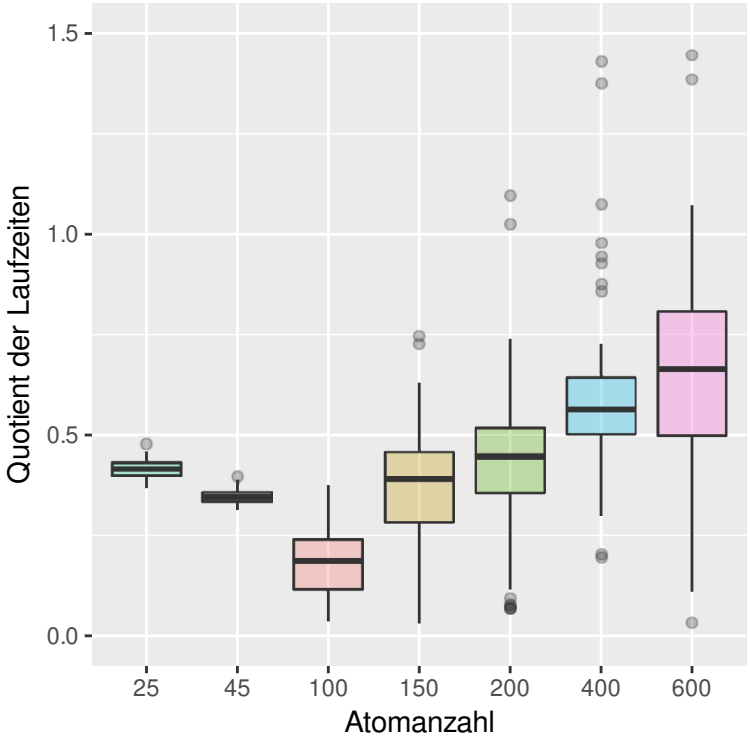


Abbildung 54: Laufzeit von `MET` (mit $k = 6$) geteilt durch die Laufzeit von SMILES (`CDK`). Die Mediane sind mit dicken horizontalen Linien markiert (angelehnt an [96]).

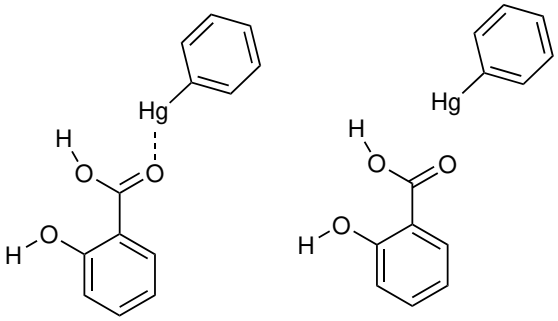


Abbildung 55: Die linke Abbildung zeigt die Struktur des Moleküles mit der CID 414487 und die rechte Abbildung die des Moleküls mit CID 49791694. Die generierten SMILES aus `CDK` sind `C=1C=CC(=C(C1)C(=O)O)O.C=1C=CC(=CC1)[Hg]` und `C=1C=CC(=C(C1)C(O)=O)O.C=1C=CC(=CC1)[Hg]`.

kanonischen SMILES aus `RDKit` sind diese jeweils gleich. Diese 1190 Paare enthalten 93 Paare, für die `CDK` keine zugehörigen SMILES erstellen konnte.

Um die abweichenden Ergebnisse der InChI und SMILES zu quantifizieren, erfolgte

eine automatisierte und eine manuelle Prüfung. Für die automatisierte Prüfung wurde jede äquivalente Zuordnung zweier Molekülstrukturen, die sich aus dem String Vergleich ergab, mit `MET` verifiziert. Erkannte `MET` keine Äquivalenz zwischen den Molekülstrukturen, wurden die 2D-Strukturen visuell verglichen. Damit konnten die falsch positiven Ergebnisse selektiert werden. Das selbe Prinzip wurde auch umgekehrt angewendet, indem `MET` die Nicht-Äquivalenz zwischen Molekülstrukturen überprüft, die aus einem String-Vergleich hervorging. Erkannte `MET`, dass die Molekülstrukturen äquivalent sind, hat es falsch negative Ergebnisse bestimmt. Auch hier erfolgte wieder die visuelle Überprüfung der 2D-Strukturen, um die Korrektheit von `MET` sicher zu stellen.

Das Experiment unterstreicht, dass `MET` sowohl in der Laufzeit als auch in der Korrektheit einem Äquivalenzvergleich auf Basis von SMILES oder InChI überlegen ist. Besonders die vollständige Korrektheit des Äquivalenzvergleichs in `MET` ist dabei herauszustellen. Durch die gelabelten Knoten können verschiedene chemische Eigenschaften eines Atom im Knoten korrekt abgebildet werden. Wichtige Informationen wie Isotope oder auch Radikale können bei der Umwandlung in die String-basierte Darstellung von SMILES und InChI‘s verloren gehen.

Zusammenfassung des Vergleichs Aus den Experimenten 5.8 und 5.9 schlussfolgern wir, dass `MET` eine signifikante Verbesserung zu den existierenden Programmen bietet, da es nicht auf die fehlerbehaftete Generierung der SMILES und InChI angewiesen ist. Bei den SMILES entstanden falsch negative Ergebnisse, wohingegen InChI‘s falsch positive Ergebnisse lieferten. In beiden Methoden gab es Molekülstrukturen, für die keine SMILES oder InChI Darstellung generiert werden konnte. `MET` hingegen überzeugte durch eine korrekte Zuordnung der Molekülstrukturen und deren äquivalenter Atome. Zusätzlich ist die Implementierung von `MET` signifikant schneller als die Isomorphiealgorithmen aus `RDKit` und `CDK` ist.

5.7. Experimente zur Fingerprint Darstellung in `MET`

Nachdem wir in dem vorherigen Kapitel die Laufzeit und die korrekte Zuordnung äquivalenter Molekülstrukturen überprüft haben, werden wir den Fokus noch auf die Fingerprints in `MET` legen. Sie sorgen neben dem Vortest für die frühzeitige Erkennung einer Nicht-Äquivalenz und dienen damit einer signifikanten Reduzierung der Laufzeit. Wir werden uns in den Experimenten mit dem Aufbau der Fingerprints und den Größen der Äquivalenzfamilien in `MET` auseinandersetzen. Dazu werden wir zuerst überprüfen, wie eindeutig der Fingerprint eines Moleküls ist, da das ein entscheidender Faktor zur Feststellung der Nicht-Äquivalenz zweier Molekülstrukturen ist. Anschließend untersuchen wir die Größe der Äquivalenzfamilien und wie diese

minimiert werden können.

5.7.1. Eindeutigkeit der Fingerprint Darstellung

In diesem ersten Experiment werden wir die Eindeutigkeit des Fingerprints quantifizieren. Zur Erinnerung: Der Fingerprint dient einer kurzen Repräsentation des Moleküls und wird vor dem Äquivalenztest überprüft. In diesem Experiment inbegriffen, ist ein Vergleich der Atomeigenschaften. Wir möchten herausfinden, welche Atomeigenschaften am unterschiedlichsten unter den Molekülen sind.

Dazu erinnern wir uns zunächst wie `MET` äquivalente Molekülstrukturen verwaltet. `MET` fügt äquivalente Molekülstrukturen in einer Äquivalenzklasse zusammen. Verschiedene Äquivalenzklassen fasst `MET` in sogenannten Äquivalenzfamilien zusammen. Zur Erinnerung wiederholen wir kurz die Begrifflichkeiten der Äquivalenz- und Äquivalenzfamilie. Jede Äquivalenzklasse wird durch einen Molekül-Repräsentanten repräsentiert. Dessen Eigenschaften bildet sein Fingerprint ab. Das bedeutet, dass die unterschiedlichen Repräsentanten der Äquivalenzklassen innerhalb einer Äquivalenzfamilie den selben Fingerprint als Repräsentanten besitzen. Die Größe der Äquivalenzfamilie hat direkten Einfluss auf die Laufzeit von `MET`. Es gilt, je größer die Äquivalenzfamilien sind, desto mehr Äquivalenztests mit den Repräsentanten der Äquivalenzklassen müssen durchgeführt werden, um eine Molekülstruktur der korrekten Äquivalenzklasse zuordnen zu können oder eine neue Äquivalenzklasse zu erstellen. Daraus resultiert ein Anstieg der Laufzeit. Das Ziel ist daher, die Repräsentanten möglichst eindeutig im Fingerprint zu charakterisieren. Dafür sehen wir uns in diesem Abschnitt eine bestimmte Größe von Äquivalenzfamilien an. Wir werden dort zum einen analysieren, wie stark sich die einzelnen Eigenschaftswerte im Fingerprint unterscheiden und zum anderen eine Äquivalenzfamilie im Detail betrachten.

5.7.1.1. Vergleich der Repräsentanten innerhalb einer Äquivalenzfamilie

Ziel dieser Analyse ist es herausfinden, wie sich die Eigenschaftswerte im Fingerprint zwischen den Repräsentanten der Äquivalenzfamilien unterscheiden. Dieses Experiment soll damit Aufschluss geben, welche chemische oder strukturelle Eigenschaft eine Molekülstruktur am charakteristischsten beschreiben kann und daher zu einer Unterscheidbarkeit zu anderen Molekülstrukturen führt. Für unseren Vergleich betrachten wir die Gruppe eins der Molekülstrukturen (Atomanzahl maximal 25) aus dem Benchmark Datensatz. In dieser Gruppe existieren 495.824.23 Äquivalenzfamilien. Darin enthalten sind 45 Äquivalenzfamilien der Größe sechs. Das bedeutet, dass es 45 Äquivalenzfamilien gibt, die jeweils sechs Äquivalenzklassen existieren, die sich durch den gleichen Fingerprint ihres Repräsentanten auszeichnen. Im ersten Schritt untersuchen wir für diese 45 Äquivalenzfamilien die Fingerprints der Repräsentanten.

Hierfür stellen wir uns die Frage, wie viele unterschiedliche Werte jeweils für die acht Eigenschaftswerte des Fingerprints existieren. Erinnern wir uns, dass der Fingerprint aus folgenden Werten besteht:

- Summe der Ordnungszahlen der Atome in der Molekülstruktur
- Anzahl der Wasserstoffe in der Molekülstruktur
- Anzahl der Einfachbindungen in der Molekülstruktur
- Anzahl der Doppelbindungen in der Molekülstruktur
- Anzahl der Dreifachbindungen in der Molekülstruktur
- Anzahl des Deuterium-Wasserstoffs in der Molekülstruktur
- Summe der Formalladungen der Atome in der Molekülstruktur
- Anzahl der Radikale in der Molekülstruktur
- Summe der Nachbarschaftsdeskriptoren aller Atome in der Molekülstruktur

Unser Experiment zeigt für die 45 Repräsentanten der Äquivalenzfamilien, dass

- 33 unterschiedliche Werte für die Summe der Ordnungszahlen
- 12 unterschiedliche Werte für die Summe der Wasserstoffe
- 6 unterschiedliche Werte für die Summe der Einfachbindungen
- 2 unterschiedliche Werte für die Summe der Doppelbindungen
- 17 unterschiedliche Werte für die Summe der Dreifachbindungen
- 2 unterschiedliche Werte für die Summe der Deuterium-Wasserstoffe
- 4 unterschiedliche Werte für die Summe der Formalladungen
- 2 unterschiedliche Werte für die Summe der Radikale
- 33 unterschiedliche Werte der Nachbarschaftsdeskriptoren

existieren.

Wir erkennen anhand der Auflistung, dass die Summe der Ordnungszahlen sowie die Summe der Nachbarschaftsdeskriptoren am stärksten die Molekülstrukturen voneinander unterscheiden. Wir leiten außerdem aus der Auflistung ab, dass die Summe der Bindungstypen nicht stark zur Charakterisierung beitragen. Da die Summe der Ordnungszahlen nicht veränderbar ist, können weitere Optimierungen nur an der

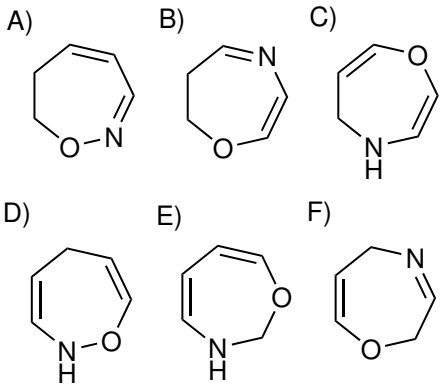


Abbildung 56: Darstellungen der Repräsentanten Moleküle der sechs Äquivalenzklassen einer Äquivalenzfamilie. Es handelt sich um folgende Moleküle:
A) 6,7-Dihydrooxazepine (CID: 18955278)
B) 6,7-Dihydro-1,4-oxazepine (CID: 70377288)
C) 4,5-Dihydro-1,4-oxazepine (CID: 19370107)
D) 2,5-Dihydrooxazepine (CID: 91414074)
E) 2,3-Dihydro-1,3-oxazepine (CID: 58550601)
F) 2,5-Dihydro-1,4-oxazepine (CID: 91534915)
Der Fingerprint aller Äquivalenzklassen ist 45_10_4_0_7_0_0_0_-54801393.

Zusammensetzung der Nachbarschaftsdeskriptoren vorgenommen werden. Ziel sollte es somit sein, diese noch eindeutiger zu machen. Darauf aufbauend vermuten wir ebenfalls, dass die Differenzierungsmöglichkeiten durch einen verbesserten Nachbarschaftsdeskriptor nicht nur Einfluss auf die Anzahl der Äquivalenzfamilien haben, sondern damit auch einhergehend auf die Größe der Äquivalenzfamilien. Denn je unterschiedlicher die Fingerprints der Repräsentanten sind, auf desto mehr einzelne Äquivalenzfamilien werden sich diese verteilen.

Im nächsten Schritt betrachten und analysieren wir detaillierter, ob die Fingerprints die Repräsentanten ausreichend unterscheidbar charakterisieren. Wir untersuchen dazu eine Äquivalenzfamilie der Größe sechs. Dafür sehen wir uns die Fingerprints der sechs Repräsentanten sowie deren zwei dimensionale Struktur an. Die Abbildung 56 zeigt die sechs Repräsentanten der Äquivalenzklassen einer Äquivalenzfamilie. Wir erkennen, dass die Molekülstrukturen sich strukturell ähneln. Sie besitzen alle einen 7-Ring, der aus zwei Doppelbindungen mit einem Stickstoffatom und einem Sauerstoffatom besteht. Daraus schlussfolgern wir, dass die ersten acht Zahlen des Fingerprints des Repräsentanten identisch sein müssten. Einen Unterschied zwischen den Molekülstrukturen sollten wir nur in ihren Nachbarschaftsbeziehungen erkennen. Die ersten genannten Schlussfolgerungen spiegeln sich fast vollständig in den Fingerprints wider. Auffällig ist jedoch, dass der Nachbarschaftsdeskriptor gleich ist. Hier

gleichen sich unsere optische Erkenntnis und die Berechnung nicht. Daraus resultiert die Frage, wodurch dieser Unterschied entsteht. Um diesen zu beantworten, berechnen wir zunächst die Nachbarschaftsdeskriptoren am Beispiel der Moleküle A) und B) händisch.

A) 1820509383

B) 746767526

Wir sehen, dass sich die händisch errechneten Nachbarschaftsdeskriptoren von den berechneten unterscheiden. Die durch `MET` errechneten Nachbarschaftsdeskriptoren weisen auf einen Überlauf, während der Summation der Nachbarschaftsdeskriptoren aller Atome, durch deren Integer Darstellung hin. Da wir zuvor gesehen haben, dass der Nachbarschaftsdeskriptor mit am charakteristischsten für den Fingerprint ist, werden wir im folgenden Abschnitt erörtern, wie sich der Überlauf auswirkt und wie er behoben werden kann.

5.7.1.2. Kodierung des Nachbarschaftsdeskriptors

Der vorhergehende Abschnitt hatte mit der Frage geendet, wie sich der Überlauf während der Summation der Nachbarschaftsdeskriptoren auf die Größe der Äquivalenzklassen auswirkt und welche Möglichkeiten existieren, die fehlerhafte Berechnung zu vermeiden. In den folgenden Experimenten untersuchen wir daher, wie die Berechnung der Nachbarschaftsdeskriptoren korrekt erfolgen kann. Dafür werden wir berechnen, wie groß die Äquivalenzfamilien im PubChem-Datensatz sind und wie häufig diese Größe vorkommt. Weiterhin werden wir analysieren, welche der sieben Gruppen, aus der Unterteilung der Molekülgrößen, die größten Äquivalenzfamilien aufweisen.

Experiment 5.10: Das erste Experiment dient der Bestimmung der Größe der Äquivalenzfamilien. Dazu betrachten wir die Berechnung des Nachbarschaftsdeskriptors und vergleichen Möglichkeiten zur Minimierung der Größe der Äquivalenzfamilien. Für das Experiment verwendet `MET` die Molekülstrukturen des Benchmark Datensatzes, die aus der PubChem Datenbank stammen. Aus diesem Datensatz ordnet `MET` äquivalente Molekülstrukturen in Äquivalenzklassen ein und fügt diese in Äquivalenzfamilien zusammen.
In Tabelle 20 sehen wir, wie groß die Äquivalenzfamilien aus dem PubChem-Datensatz sind und wie häufig jede Größe vorkommt. Ziel sollte es sein, möglichst viele Äquivalenzfamilien kleiner Größe zu erhalten, da dann beim Einfügen einer neuen Molekülstruktur in eine Äquivalenzklasse innerhalb einer Äquivalenzfamilie wenige Repräsentanten-Moleküle überprüft werden müssen. Die Tabelle zeigt uns, dass insgesamt 40 Äquivalenzfamilien existieren, die mehr als 150 Äquivalenzklassen haben. Um eine

Tabelle 20: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors erfolgte eine Integer-Darstellung.

Größe	1	2	3	4	5	6-10
Häufigkeit	80967900	194181	8188	3628	1535	2768

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	712	303	117	62	28	40

neue Molekülstruktur einer dieser Äquivalenzklassen zu zuordnen, müssten daher im schlechtesten Fall 150 Äquivalenztests zwischen der neuen Molekülstruktur und den Molekül-Repräsentanten der Äquivalenzklassen durchgeführt werden.
Ein Grund für den gleichen Wert des Nachbarschaftsdeskriptors ist, wie im vorherigen Kapitel beschrieben, die Kodierung als Integer-Wert. Durch die Summe der Nachbarschaftsdeskriptoren aller Atome entsteht ein Überlauf. Die Berechnung der Summe erfolgt nach Algorithmus 4.

Algorithmus 4: Berechnung des Nachbarschaftsdeskriptors als Integer

```
1 int oldValue = neighborhoodDescriptor.get(i);
2 int newValue = (31 * oldValue + sum_of_neighbourhooddescriptor);
```

Um diese Fehlerquelle zu beheben, wenden wir mehrere Ansätze zur Darstellung der Summe an. Als erste Möglichkeit speichert `MET` die Summe statt des Zahlentyp Integer als Long. Die dadurch entstehende Verbesserung der Äquivalenzfamiliengrößen zeigt die Tabelle S-14 im Anhang. In ihr erkennen wir beispielsweise, dass sich die Anzahl der Äquivalenzfamilien mit einer Größe von mehr als 150 Äquivalenzklassen um 10 reduziert hat. Als zweite Möglichkeit bildet `MET` die errechnete Summe im Zahlentyp Long zusätzlich in einer Hashtabelle ab. Die Verteilung der Hashtabelle erfolgt über eine große Primzahl (größer 2^{30}), wodurch die Äquivalenzfamilien nochmals in ihrer Größe reduziert wurden. Die Abwandlung des Algorithmus 4 zeigt der Algorithmus 5. Dafür wurden vier Primzahlen, die größer als 2^{30} miteinander verglichen. Alle Primzahlen (Tabelle 21, S-15, S-16, S-17) weisen auf eine deutliche Reduzierung der Äquivalenzklassen pro Äquivalenzfamilie im Vergleich zur Long Darstellung. Der direkte Vergleich der Ergebnisse zeigt, dass die Berechnung mit der Primzahl 1073741857 (Tabelle 21) das beste Ergebnis erzielt.
Das können wir beispielsweise daran erkennen, dass keine Äquivalenzfamilien mit mehr als 150 Äquivalenzklassen mehr existieren. Weiterhin hat sich die Zahl der Äquivalenzfamilien, die nur eine Äquivalenzklasse enthalten, um 147.997 Äquivalenzfamilien erhöht. Das unterstreicht positiv, dass die neue Berechnung die Repräsentanten-

Algorithmus 5: Berechnung des Nachbarschaftsdeskriptors als Long unter Einbindung einer Primzahl

```
1 long oldValue = neighborhoodDescriptor.get(i);
2 long newValue =
    (31 * oldValue + sum_of_neighbourhooddescriptor)%1073741857;
```

Tabelle 21: Übersicht zu den Größen der Äquivalenzfamilien (erste Zeile) und deren Häufigkeit (zweite Zeile). Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors wurde die Primzahl 1073741857 verwendet.

Größe	1	2	3	4	5	6-10
Häufigkeit	81.115.897	125.384	7.331	3.436	1.476	2.561

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	857	384	120	70	19	0

Moleküle besser in ihrem Fingerprint charakterisiert und weniger Äquivalenzvergleiche notwendig sein werden. Auf Basis dieser Erkenntnisse wurden die Berechnungen der Experimente 5.4. bis 5.10. durchgeführt.

Experiment 5.11: In diesem sich anschließenden Experiment untersuchen wir nun, bei welcher Molekülgröße die größten Äquivalenzfamilien auftreten. Betrachten wir für dieses Experiment daher wieder die Unterteilung des PubChem-Datensatzes in seine sieben Gruppen in Abhängigkeit der Molekülgröße. Ziel dieses Experiments ist es, herauszufinden, wie die veränderte Kodierung sich in der Laufzeit der Gruppen widerspiegelt. Im Experiment vergleichen wir dazu die berechnete Laufzeit unter Verwendung des Algorithmus 4 und Algorithmus 5 für die sieben Gruppen.

Abbildung 57 zeigt die Quotienten aus den Laufzeiten. Der Quotient setzt sich aus der Laufzeit von `MET` mittels der Berechnung nach Algorithmus 4 geteilt durch die Laufzeit von `MET` mittels der Berechnung nach Algorithmus 5 zusammen. Ein Wert von unter eins bedeutet, dass die Ausgangsvariante, mit dem Zahlentyp Integer, schneller ist. Wir erkennen, dass für die Gruppe eins und fünf die Integer-Darstellung minimal besser ist als die eben besprochene zusätzliche Abbildung in einer Hashtabelle. Im Gegensatz dazu zeigt der Quotient für die Gruppen zwei bis vier einen Wert von über eins. Das bedeutet, dass sich hier die Kodierung des Nachbarschaftsdeskriptors als Long mit der abgewandelten Hashfunktion positiv auf eine Laufzeitreduzierung auswirken. Beispielsweise kommt es für die Gruppe drei (46-100 Atome) zur einer Laufzeitreduktion von rund 6 Minuten.
Wir können daher abschließend daraus schlussfolgern, dass die veränderte Abbildung

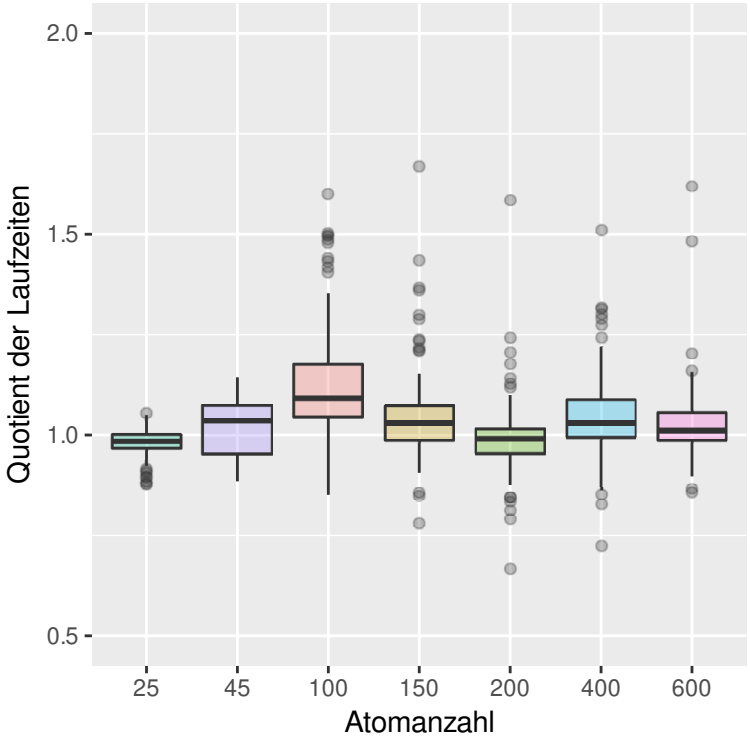


Abbildung 57: Laufzeit von `MET` (mit $k = 6$) mit der Darstellung des Nachbarschaftsdeskriptors als Integer geteilt durch die Laufzeit von `MET` mit der Darstellung des Nachbarschaftsdeskriptors als Long (modulo 1073741857). Die Mediane sind mit dicken horizontalen Linien markiert.

der Summe sich durch die Reduktion der Äquivalenzfamilien positiv auf die Laufzeit auswirkt. Wie sich die Größe der Äquivalenzfamilien über diese sieben Gruppen des Datensatz verteilt, betrachten wir im folgenden Experiment.

Experiment 5.12: Dieses Experiment untersucht, welche Gruppe des PubChem Datensatzes die größten Äquivalenzfamilien besitzt. Dazu berechnen wir für jede der sieben Gruppen, wie häufig die unterschiedlichen Größen an Äquivalenzfamilien vorkommen. Aus dem Experiment 5.11. wissen wir bereits, dass die Gruppen zwei bis vier in der Laufzeit eine Verbesserung zeigen.

Wir erkennen aus Tabelle 22, dass die größten Äquivalenzfamilien in den Gruppen drei (Atomamzahl 100-149) und vier (Atomanzahl 150-199) vorkommen. Die größte Äquivalenzfamilie aus diesen zwei Gruppen enthält 142 Äquivalenzklassen.

Basierend auf der Erkenntnis, dass in den Gruppen drei und vier die größten und häufigsten Äquivalenzfamilien vorkommen, können wir Rückschlüsse auf die vorher-

Tabelle 22: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit für die sieben Gruppen von Molekülgrößen. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors wurde die Primzahl 1073741857 verwendet.

Größe	1	2	3	4	5	6-10
Häufigkeit Gruppe 1	49582662	58296	1053	226	54	25
Häufigkeit Gruppe 2	28333145	39730	2366	1091	407	418
Häufigkeit Gruppe 3	2995261	29273	3597	1907	958	1930
Häufigkeit Gruppe 4	140269	1240	259	167	46	163
Häufigkeit Gruppe 5	35149	410	35	27	8	23
Häufigkeit Gruppe 6	20409	242	24	17	3	1
Häufigkeit Gruppe 7	1268	42	9	1	0	1

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit Gruppe 1	0	0	0	0	0	0
Häufigkeit Gruppe 2	77	21	0	0	0	0
Häufigkeit Gruppe 3	711	337	101	61	19	0
Häufigkeit Gruppe 4	47	24	14	8	0	0
Häufigkeit Gruppe 5	22	2	5	1	0	0
Häufigkeit Gruppe 6	0	0	0	0	0	0
Häufigkeit Gruppe 7	0	0	0	0	0	0

gehenden Experimente ziehen. Wir sahen, dass die stärksten Unterschiede von `MET` zu den Algorithmen in den Gruppen drei und vier lagen. Das lässt sich darauf zurückführen, dass dort die meisten Äquivalenzvergleiche aufgrund der Größe der Äqui-

valenzfamilien vorkamen.

An dieser Stelle erkennen wir noch ein Mal die Wichtigkeit der Fingerprints. Die Tabelle 22 bildet die Partitionierung der Molekülstrukturen der PubChem Datenbank ab. Sollte nun eine neue Molekülstruktur eingefügt werden, genügt es den Fingerprint der Molekülstruktur zu bestimmen und in die Äquivalenzklasse mit dem selben Fingerprint des Repräsentanten einzufügen. Dafür sind im schlechtesten Fall 150 Vergleiche für eine Molekülstruktur aus der Gruppe drei nötig. Ohne die Partitionierung hätten 3.034.155 Äquivalenzvergleiche erfolgen müssen. Mit großer Wahrscheinlichkeit sind nicht die kompletten 3.034.155 Äquivalenzvergleiche notwendig, da bereits der Vortest zum Ausschluss einiger Molekülstrukturen führen wird. Der Vergleich der 150 Äquivalenzvergleiche zu den 3.034.155 für die Gruppe drei hebt den positiven Effekt der Partitionierung auf Basis der Fingerprints hervor. Zusätzlich ist die Berechnung des Fingerprints kostengünstig, da es sich nur um die Konkatenation bereits berechneter Werte handelt.

Nachdem wir die Berechnung der Nachbarschaftsdeskriptoren verbessert und deren Laufzeit analysiert haben, möchten wir abschließend Bezug auf den Ausgangspunkt des Kapitels 5.7.1. nehmen.

5.7.1.3. Optimierung der Charakterisierung der Repräsentanten

Um die verbesserte Darstellung des Nachbarschaftsdeskriptors zu demonstrieren, schauen wir uns nochmal die Anzahl der Äquivalenzfamilien für die Gruppe eins der Molekülgröße an. Wir vergleichen dazu die Gesamtanzahl an Äquivalenzfamilien sowie die Anzahl an Äquivalenzfamilien, die sechs Äquivalenzklassen beinhaltet.

Insgesamt existieren nun 496.452.33 Äquivalenzfamilien, dies entspricht einem Zuwachs von 62.810 Äquivalenzfamilien. Wir schließen daraus, dass sich die Äquivalenzklassen auf mehr Äquivalenzfamilien verteilt haben, da die Fingerprints stärker divers sind und bei der Zuordnung einer Molekülstruktur in eine Äquivalenzklasse weniger Äquivalenztests durchgeführt werden müssen. Betrachten wir nun noch die Anzahl von Äquivalenzfamilien, die genau sechs Äquivalenzklassen enthalten, sehen wir, dass für diese Gruppe nur noch 20 statt 45 Äquivalenzfamilien existieren. Die von uns zuvor betrachtete Äquivalenzfamilie mit den sechs Äquivalenzklassen (Abbildung 56) hat sich auf vier Äquivalenzfamilien aufgeteilt. Zwei davon haben die Größe zwei und die weiteren zwei enthalten nur eine Äquivalenzklasse.

Im Detail bedeutet das, dass die Moleküle C) und E) in einer Äquivalenzfamilie sind und durch den Fingerprint 45_10_4_0_7_0_0_0_2071692738 repräsentiert wer-

den. Die Moleküle A) und F) bilden eine weitere Äquivalenzfamilie mit dem Fingerprint 5_10_4_0_7_0_0_0_1820509383. Das Molekül B) bildet seine eigene Äquivalenzfamilie mit der Repräsentation durch 45_10_4_0_7_0_0_0_746767526. Der Fingerprint 45_10_4_0_7_0_0_0_3145434595 repräsentiert die Äquivalenzklasse mit dem Molekül D).

Zusammenfassung zur Eindeutigkeit der Fingerprints Zusammenfassend können wir feststellen, dass die Summe der Nachbarschaftsdeskriptoren als Zahlentyp Long sowie die zusätzliche Abbildung der Werte in einer großen Hashtabelle einen positiven Effekt zeigen. Diesen erkannten wir an der Präzisierung der Repräsentanten und dass in deren Folge eine deutliche Reduktion der Äquivalenzfamilien, die aus mehreren Äquivalenzklassen bestehen, erfolgte. Damit einhergehend verzeichnen wir eine Laufzeitreduktion von 384 Sekunden für die Gruppe eins und über alle Gruppen von 6 Minuten. Ebenfalls stellte sich die Wichtigkeit der Fingerprints heraus. Die Fingerprints dienen nach dem Vortest einer zweiten zusätzlichen Reduktion von Isomorphiebestimmungen zwischen beiden Molekülengraphen. Durch die bereits berechneten Eigenschaften der Molekülstruktur, die dann konkatiniert werden, kann effizient ermittelt werden, ob insbesondere die Nachbarschaftsbeziehungen zwischen zwei Molekülstrukturen übereinstimmen. Weiterhin dient der Fingerprint dazu, die Anzahl an Äquivalenzvergleichen beim Einfügen einer neuen Molekülstruktur in die Grupperierung zu minimieren, da lediglich die Repräsentanten einer Äquivalenzfamilie mit der einzufügenden Molekülstruktur verglichen werden müssen.

5.8. Einbindung von `MET` in `ChemFrag`

Das Ziel dieses Kapitels ist es den Nutzen von `MET` zur Erkennung der Moleküläquivalenz in `ChemFrag` hervorzuheben. Dazu verwenden wir die Version von `MET`, die die Optimierungen aus dem vorhergehenden Kapitel enthält. `ChemFrag` integriert `MET` an zwei Positionen. Die eine Position ist die Abfrage, ob ein Fragment-Ion/Neutralverlust bereits berechnet wurde. Im Falle einer Äquivalenz gibt `MET` dafür die Zuordnung der äquivalenten Atome zurück. Dadurch müssen in `ChemFrag` keine redundanten quantenchemischen Rechnungen durchgeführt werden. Weiterhin verwendet `ChemFrag` `MET`, um Fragment-Ionen zu erkennen, die bereits weiter fragmentiert wurden. Damit verhindert `ChemFrag`, dass gleiche Fragment-Ionen mehrfach fragmentiert werden. Das Speichern der Fragment-Ionen erfolgt in `ChemFrag` in Hashtabellen. Ziel ist es nun, zu überprüfen, ob die verschiedenen Molekülvergleiche in `ChemFrag` durch die Anwendung von `MET` verbessert werden können. Dabei werden wir zum einen die Laufzeit vergleichen und zum anderen überprüfen, ob die Erkennung der Moleküläquivalenz durch `MET` exakter ist als bei dem bisher integrierten SMILES Vergleich. Diese zwei Experimente führen wir an der Fragmentierung von

Kokain durch.

Bevor wir den Vorteil von MET in ChemFrag erkennen können, müssen wir zunächst die Methode zur Moleküläquivalenz in ChemFrag beschreiben, die vor dem Einbau von MET enthalten war. ChemFrag verwendete in der Ausgangsvariante für die Hashtabellen Strings als Schlüssel und speicherte in der Hashtabelle, als Hash-Map die Fragment-Ionen mit dem Typ IAtomContainer aus CDK. Um die Fragment-Ionen in Strings abzubilden, nutzte ChemFrag die SMILES Notation. Wenn ChemFrag ein neues Fragment-Ion überprüfte, erstellte es im ersten Schritt seinen SMILES und glich diesen mit den Schlüsseln in der Hashtabelle ab. Sollte der SMILES bereits vorhanden sein und ChemFrag benötigte Informationen, wie die Bindungsordnungen aus dem gespeicherten Fragment-Ion, um diese auf das neue Fragment-Ion abbilden, musste ChemFrag zunächst die Matching-Funktion aus CDK anwenden. Diese bestimmte die äquivalenten Atome zwischen den zwei Fragment-Ionen, sodass im nächsten Schritt von den Atomen auf die Bindungen geschlossen werden konnte. Nachteilig war an dieser Vorgehensweise, dass nach dem String-Vergleich nochmals ein Äquivalenztest durchgeführt werden musste, um die äquivalenten Atome zu bestimmen und anschließend über die Atome auf die Bindungen zugreifen zu können. Dieses Vorgehensweise ist nötig, da der eigentliche Äquivalenzvergleich in ChemFrag sehr zeitintensiv ist (siehe Tabelle 18) und ein Vortest mittels SMILES zu einer Reduktion der Laufzeit führte. ChemFrag benötigte für die Annotation des MS/MS-Spektrums von Kokain mit dieser Methode 2100 Sekunden.

Im Vergleich dazu ist MET anders in ChemFrag integriert. Alle bereits berechneten oder fragmentierten Moleküle sind in Hash-Sets enthalten. Im Hash-Set sind die Fragment-Ionen vom Typ Molecule aus MET gespeichert. Führt ChemFrag nun den Äquivalenzvergleich durch, prüft es, ob das Fragment-Ion oder der Neutralverlust bereits gespeichert ist. Im Falle der Äquivalenz gibt MET die Zuordnung der äquivalenten Atome und Bindungen mit deren Labels für die Bindungsordnungen zurück. Damit ist die Abbildung der Bindungsordnungen effizient möglich. Mit dieser Methodik benötigt ChemFrag nur 796 Sekunden und damit nur ein drittel der Laufzeit des SMILES basierten Ansatzes. Wir stellen damit fest, dass die Einbindung von MET eine deutliche Verbesserung der Laufzeit von ChemFrag bewirkt.

Für den Vergleich der Äquivalenz haben wir uns ebenfalls die Anzahl äquivalenter Molekülstrukturpaare ausgeben lassen. Diese umfassen sowohl die Fragment-Ionen als auch die Neutralverluste. Mit der SMILES basierten Methode generiert ChemFrag 6.652 äquivalente Molekülstrukturpaare. Im Vergleich dazu konnte MET 15.744 äquivalente Molekülstrukturpaare detektieren. Auch hier spiegelt sich die Erkenntnis aus

dem Experiment 5.9 wider, dass der SMILES basierte Äquivalenzvergleich falsch negative Ergebnisse liefert. Aufgrund der 9.092 weniger erkannten äquivalenten Molekülstrukturen musste `ChemFrag` in SMILES basierten Variante 9.092 mehr quantenchemische Rechnungen mittels `MOPAC` durchführen. Daher lässt sich der deutliche Unterschied in der Laufzeit von rund 23 Minuten mit der Vielzahl zusätzlicher quantenchemischer Rechnungen erklären.

Mit diesem abschließenden Experiment konnten wir den Nutzen und die Vorteile des neu entwickelten Moleküläquivalenztesters nochmals besonders hervorheben. Die Entwicklung von `MET` ermöglicht nun für `ChemFrag` Laufzeit reduzierte Annotation und die Beachtung, dass äquivalente Molekülstrukturen erkannt und zusätzliche quantenchemische Berechnungen verhindert werden.

5.9. Zusammenfassung

Das Kapitel „Molecule Equivalence Tester - MET“ stellte den neuen Algorithmus zur Bestimmung von Moleküläquivalenzen vor. `MET` nutzt chemische und strukturelle Eigenschaften der Moleküle, um deren Strukturen in gelabelte Graphen zu überführen. Um die Laufzeit des Isomorphie-Algorithmus zu reduzieren, basiert der Algorithmus auf der Konstruktion von möglichst eindeutigen Knoten-Labeln. Dafür essentiell ist die Einführung des Knoten-Labels, das die Nachbarschaft eines Atoms widerspiegelt. Die Experimente dazu wiesen darauf hin, dass eine lokale Nachbarschaft von sechs für eine gute Charakterisierung ausreicht. Weiterhin zeigten die Experimente, dass die Kodierung des Nachbarschaftsdeskriptors einen großen Einfluss auf die korrekte Widerspiegelung der lokalen Nachbarschaft hat. Durch die chemischen und strukturellen Eigenschaften erstellt `MET` Kandidatenmengen und enthält entscheidende Regeln zu deren Reduzierung. Wir konnten zeigen, dass die Verwendung der Kandidatenmengen und deren Reduktionsregeln zu einer Minderung der Rekursionsaufrufe führt. Des Weiteren beobachten wir, dass sich die Anzahl der Rekursionsaufrufe linear zur Atomanzahl verhält.

In der zweiten Phase von `MET` wird ein generischer Isomorphiealgorithmus aus gelabelten Graphen genutzt. Hierfür zeigten die Experimente, dass dieser implementierte Backtracking-Algorithmus vergleichbar mit dem aktuell schnellsten Algorithmus aus `VF2++` ist. Weitere Experimente demonstrierten, dass `MET` schneller ist als die Algorithmen aus `SMSD`, `CDK` und `RDKit`. Zusätzlich brachten die Experimente hervor, dass `MET` robuster ist als die Konstruktion von SMILES und InChI aus `CDK`. Zusammenfassend können wir feststellen, dass `MET` für Molekülstrukturen bis 400 Atome (Wasserstoff nicht mit eingerechnet) schneller ist als der aktuell schnellste Algorithmus `VF2++`. Zusätzlich weist `MET` Korrektheit in der Zuordnung der äquivalenten Atome

auf und ist damit korrekter als der schnelle String-Vergleich mittels SMILES und InChI, die falsch positive oder falsch negative Ergebnisse erzielen. Hervorzuheben ist ebenfalls, dass `MET` Radikale und Isotope im Äquivalenzvergleich korrekt mitberücksichtigen kann.

Die Implementation von `MET` in Java ermöglicht zusätzlich die Kompatibilität zu `CDK` und kann damit dort existierende fehlerhafte Isomorphie-Algorithmen ersetzen. Diese Ersetzung wurde in `ChemFrag` bereits umgesetzt, sodass dort die Korrektheit zur Überprüfung der Moleküläquivalenz von Fragment-Ionen und Neutralverlusten deutlich verbessert werden konnte.

5.10. Ausblick

In zukünftigen Projekten soll der Molekülvergleich von der 2D-Ebene auf die 3D-Ebene erweitert werden. Dazu muss sowohl die R/S-Isomerie der Atome als auch die cis/trans-Isomerie der Bindungen eingebunden werden. Diese Isomerien sollen über das Labeling der Knoten und Kanten ermöglicht werden. Weiterhin ist das Ziel, die Knotenauswahl beispielsweise über einen Fibonacci-Heap statt eines Binären Heaps in der Prioritätswarteschlange zu realisieren. Um die Knoten effizienter zu zuweisen, sollten weitere Ansätze des „constraint satisfaction programming“ umgesetzt werden [112, 113]. Dafür eignet sich beispielsweise die Verwendung von „constraint Graphen“ vor der Ausführung des Backtrackings. Um die Anwendbarkeit von `MET` zu erweitern, könnten ebenfalls Proteine betrachtet werden. Um deren Äquivalenz zu bestimmen, stellen die Knoten dann Aminosäuren dar. An dieser Stelle müssten die Knoten-Label verändert werden, indem statt der Ordnungszahl der 1-Buchstaben- oder 3-Buchstaben-Code einer Aminosäure verwendet wird. Eigenschaften, wie die Anzahl der Radikale oder der Formalladung, könnten beibehalten werden. Ebenfalls wird der Ansatz der Nachbarschaftsbeziehung weiter integriert sein. Neben der Anwendung der exakten Äquivalenz, ist ein zukünftiges Projekt die Anwendung auf Teiläquivalenzen, im Sinne der Subgraphisomorphie oder auch der Ähnlichkeitssuche. Die Anwendung der Ähnlichkeitssuche ist besonders in der Wirkstoffsuche von großem Interesse.

6. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit entstanden zwei neue Programme, die im Bereich Chemoinformatik Einsatz finden können. Dabei handelt es sich als erstes um das Programm `ChemFrag` zur Annotation von Fragment-Ionen-Spektren. Erste Ergebnisse dazu wurden 2018 [66] publiziert. Ziel von `ChemFrag` ist eine chemisch plausible Annotation von Fragment-Ionen-Spektren. Dafür vereint `ChemFrag` einen regelbasierten Ansatz und einen quantenchemischen Ansatz. Der quantenchemische Ansatz bestimmt auf Basis der semi-empirischen Methode PM7 aus `MOPAC` die schwächsten Bindungen in einem Fragment-Ion. Diese können dann durch Implementierungen der homolytischen- und heterolytischen Spaltung getrennt werden. Zusätzlich dient `MOPAC` der Berechnung der Bildungsenthalpien, die `ChemFrag` für die Bestimmung der Reaktionsenergien benötigt. Die Reaktionsenergie bilden die Basis für die Entscheidung, ob ein Fragment-Ion im nächsten Schritt weiter fragmentiert werden soll. Um ein größeres Spektrum an Fragment-Ionen zu erzeugen, enthält `ChemFrag` den regelbasierten Ansatz. Hierfür beinhaltet `ChemFrag` 31 Spaltungsregeln und 20 Umlagerungsregeln. Die Spaltungsregeln enthalten beispielsweise Implementationen zur Abspaltung von Wasserstoff, Wasser oder auch die α-Spaltung. Die Umlagerungsregeln dienen dazu, chemisch plausible und energetisch stabile Fragment-Ionen zu erzeugen. Hier sind insbesondere der Proton-Shift und der Hydrid-Shift zu nennen.

Experimente zeigten, dass `ChemFrag` für verschiedene Stoffgruppen angewendet werden kann. Ausführlich haben wir das Verhalten von `ChemFrag` an den Doping-Substanzen Ephedrin und Kokain beschrieben. Ebenso konnten wir aufführen, dass `ChemFrag` für weitere Doping-Substanzen sowie für langkettige Kohlenwasserstoffe und Steroide chemisch plausible Annotationen vornehmen kann. Besonders die Vorhersage noch nicht veröffentlichter Fragmentierungswege für Estradiol-3-methylether, Dehydrocyanomethylestradiol und 2-Cyano-2-phenylbutansäureetylester ist hervorzuheben. Weiterhin stellten wir fest, dass `ChemFrag` große Übereinstimmungen in der Annotation zu den etablierten Methoden, wie `MetFrag`, `CFM-ID` und `SIRIUS` und insbesondere zu publizierten Fragmentierungswegen aufweist. Anderseits konnten wir auch aufzeigen, dass es bei den etablierten Methoden zur Vorhersage chemisch nicht korrekter Fragment-Ionen, wie beispielsweise ein fünffach gebundenes Kohlenstoffatom, kommen kann. Im Vergleich zu `MetFrag` und `SIRIUS` mit `CSI:FingerID` kann `ChemFrag` die Position der Ladung und Radikalen sowie die Umlagerung von Bindungen und chemischen Gruppen exakt darstellen. `ChemFrag` ermöglicht ebenfalls die Vorhersage der Fragmentierungswege und die Zusammenfassung in einem Fragmentierungsbaum, der die Fragment-Ionen zur Annotation der Fragment-Ionen enthält. Experimente zur Parameteroptimierung von `ChemFrag` bewerteten eine Fragmentie-

rungstiefe von sieben als geschickten Tradeoff zwischen Laufzeit und Güte. Weiterhin bestimmten die Experimente die Parameterkombination 1 (Tprot = 50, Tfrag = 150, Trearr = 100, TBO = 0.08, TD = 7) als Ausgangskonfiguration für den Annotationsprozess.

`ChemFrag's` Rechenzeit wird von `MOPAC` und `RDKit` dominiert wird, die für den quantenchemischen Ansatz notwendig sind. Um deren Rechenzeit zu minimieren, ist es sinnvoll, gleiche Fragment-Ionen und Neutralverluste zu erkennen und die äquivalenten Atome und Bindungen zwischen äquivalenten Fragment-Ionen korrekt einander zu zuordnen, um redundante Berechnungen zu vermeiden. Auf Grundlage dieser Anforderung entstand das Programm `MET`. Dessen Ziel ist das schnelle und korrekte Ermitteln äquivalenter Molekülstrukturen sowie die Zuordnung der äquivalenten Atome. Dafür bildet `MET` Molekülstrukturen in gelabelte Graphen ab. Die Knoten der Graphen enthalten als Label die chemischen Eigenschaften der Atome, wie Ordnungszahl, Anzahl implizierter Wasserstoffe oder auch Radikale, sowie einen strukturellen Eigenschaftswert, den Nachbarschaftsdeskriptor. Der Nachbarschaftsdeskriptor ist ein essentieller Bestandteil von `MET`, der dessen Erfolg ausmacht, da er in einem Wert die Eigenschaften der Atome zusammenfasst, die sich in einem festgelegten Radius um das betrachtete Atom befinden. Experimente ergaben den Wert sechs als optimalen Radius. Weiterhin enthält `MET` einen Backtracking-Algorithmus sowie einen intelligenten Algorithmus zur Kandidatenreduktion. Insbesondere die Kandidatenreduktion erwies sich in den Experimenten als ein wichtiger Faktor zur Reduktion der Laufzeit. Dies sahen wir insbesondere daran, dass sich die Anzahl der Rekursionsaufrufe oft linear zur Atomanzahl verhält. Einen Anteil zur Laufzeitreduktion bringt ebenfalls die Fingerprint Repräsentation der Moleküle. Dieser ist ein String und enthält die Konkatenation der berechneten Eigenschaften und wird vor der Ausführung des Isomorphie-Algorithmus verglichen. Nur wenn beide Fingerprints identisch sind, wird der Isomorphie-Algorithmus ausgeführt.

In unseren Experimenten verglichen wir die Äquivalenzerkennung von `MET` mit den Ergebnissen aus `CDK`, `SMSD` und `VF2++` sowie den String-basierten Vergleichen auf Basis von SMILES und InChI's. Hier konnten wir feststellen, dass `MET` den etablierten Bibliotheken `CDK` und `SMSD` sowohl in der korrekten Erkennung der Äquivalenz als auch in der Laufzeit deutlich überlegen ist. Weiterhin zeigten die Experimente, dass die SMILES Vergleiche falsch negative und die InChI Vergleiche falsch positive Ergebnisse erzeugen. Im direkten Vergleich der Laufzeit war auch hier `MET` den String-basierten Vergleichen überlegen. Ebenfalls ist zu nennen, dass `MET` für alle Molekülstrukturen eine Umwandlung in die Graph Repräsentation durchführen konnte und somit für alle Molekülstrukturen der PubChem Datenbank eine Äquivalenz-

ordnung möglich war. Im Gegensatz dazu konnten für 93 Molekülstrukturen keine SMILES und für drei Molekülstrukturen keine InChI‘s erzeugt werden. Im direkten Vergleich zu `VF2++` sahen wir, dass `MET` für die Einordnung aller Molekülstrukturen der PubChem in Äquivalenzfamilien und Äquivalenzklassen signifikant schneller war als `VF2++`. Die Zuordnung äquivalenter Molekülstrukturen stimmten zwischen `MET` und `VF2++` überein. Damit unterstrichen wir, dass die Verwendung von `MET` statt `VF2++` für die Erkennung und Zuordnung äquivalenter Molekülstrukturen gerechtfertigt ist.

Die Zusammenführung von `MET` in `ChemFrag` erzielte bereits erste positive Ergebnisse. Am Beispiel von Kokain sahen wir eine Reduzierung der Laufzeit um 38 %, da mehr äquivalente Fragment-Ionen und Neutralverluste erkannte wurden. Die Folge sind weniger Aufrufe des Programms `MOPAC`. In einem nächsten Schritt soll `MET` um einen Algorithmus zur Erkennung von Teilstrukturen erweitert werden, damit die Erkennung der Teilstrukturen im regelbasierten Ansatz aus Basis der SMARTS durch ein erweitertes `MET` ersetzt werden kann. Weiterhin besteht das Ziel `MET` für die Bewertung von Ähnlichkeiten von Molekülen zu auszubauen. Für den Äquivalenzvergleich der Molekülstrukturen soll zukünftig nicht nur die 2D-Ebene mitbetrachtet werden, sondern auch die 3D-Ebene. Des Weiteren ist eine Idee die Anwendbarkeit von `MET` auf den Äquivalenzvergleich von Proteinen zu erweitern. Für `ChemFrag` wird das Augenmerk auf der Erweiterung des Anion-Modus liegen. Weiterhin ist eine Kombination mit `MetFrag` wünschenswert, mit dem Ziel das Ranking der Kandidaten in `MetFrag` genauer zu bestimmen. Um `ChemFrag` flexibler erweitern zu können, soll die feste Implementierung der Regeln aus `ChemFrag` herausgenommen werden und stattdessen eine eigene leicht verständliche domain-spezifische Sprache entwickelt werden. Mit ihr sollen die Spaltungs- und Umlagerungsregeln auch von externen AnwenderInnen, ohne Kenntnissen in Java, implementiert werden.

7. Literaturverzeichnis

[1] B. Z. Manfred Hesse Herbert Meier. *Spektroskopische Methoden in der organischen Chemie*. Thieme Taschenlehrbuch (2005). DOI: https://doi.org/10.1002/prac.19813230126.

[2] M. M. Rinschen, J. Ivanisevic, M. Giera und G. Siuzdak. *Identification of bioactive metabolites using activity metabolomics*. Nature Reviews Molecular Cell Biology **20** (Juni 2019), 353–367. DOI: 10.1038/s41580-019-0108-4.

[3] F. Hufsky, K. Scheubert und S. Böcker. *Computational mass spectrometry for small-molecule fragmentation*. Trends in Analytical Chemistry **53** (2014), 41–48. DOI: http://dx.doi.org/10.1016/j.trac.2013.09.008.

[4] S. Grimme. *Towards First Principles Calculation of Electron Impact Mass Spectra of Molecules*. Angew. Chem. Int. Ed. **52** (2013), 6306–6312. DOI: 10.1002/anie.201300158.

[5] P. Cayley. *LVII. On the mathematical theory of isomers*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **47** (1874), 444–447. DOI: 10.1080/14786447408641058.

[6] I. Mayer und Á. Gömöry. *Use of energy partitioning for predicting primary mass spectrometric fragmentation steps: A preliminary account*. International Journal of Quantum Chemistry **48** (1993), 599–605. DOI: 10.1002/qua.560480854.

[7] A. Alex, S. Harvey, T. Parsons, F. S. Pullen, P. Wright und J.-A. Riley. *Can density functional theory (DFT) be used as an aid to a deeper understanding of tandem mass spectrometric fragmentation pathways?* Rapid Communications in Mass Spectrometry **23** (2009), 2619–2627. DOI: 10.1002/rcm.4163.

[8] M. Thevis. *Mass Spectrometry in Sports Drug Testing*. Bd. 1. Wiley & Sons (2010).

[9] J. R. Ullmann. *Bit-vector Algorithms for Binary Constraint Satisfaction and Subgraph Isomorphism*. Journal of Experimental Algorithmics **15** (2010), 1.6:1–1.6:64. DOI: 10.1145/1671970.1921702.

[10] A. Jüttner und P. Madarasi. *VF2++—An improved subgraph isomorphism algorithm*. Discrete Applied Mathematics **242** (2018). Computational Advances in Combinatorial Optimization, 69–81. DOI: https://doi.org/10.1016/j.dam.2018.02.018.

[11] M. Karthikeyan und R. Vyas. *Practical Chemoinformatics*. Springer India (2014). DOI: 10.1007/978-81-322-1780-0.

[12] D. Weininger. *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.* Journal of Chemical Information and Computer Sciences **28** (1988), 31–36. DOI: 10.1021/ci00057a005.

[13] D. Weininger, A. Weininger und J. L. Weininger. *SMILES. 2. Algorithm for generation of unique SMILES notation.* Journal of Chemical Information and Computer Sciences **29** (1989), 97–101. DOI: 10.1021/ci00062a008.

[14] N. M. O'Boyle. *Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI.* Journal of Cheminformatics **4** (2012), 22. DOI: 10.1186/1758-2946-4-22.

[15] Daylight. URL: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

[16] E. L. Willighagen, J. W. Mayfield, J. Alvarsson, A. Berg, L. Carlsson, N. Jeliazkova, S. Kuhn, T. Pluskal, M. Rojas-Chertó, O. Spjuth, G. Torrance, C. T. Evelo, R. Guha und C. Steinbeck. *The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching.* Journal of Cheminformatics **9** (2017), 33. DOI: 10.1186/s13321-017-0220-4.

[17] *The RDKit Documentation (https://www.rdkit.org/docs/).* 2020.

[18] S. R. Heller, A. McNaught, I. Pletnev, S. Stein und D. Tchekhovskoi. *InChI, the IUPAC International Chemical Identifier.* Journal of Cheminformatics **7** (2015), 23. DOI: 10.1186/s13321-015-0068-4.

[19] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi und I. Pletnev. *InChI - the worldwide chemical structure identifier standard.* Journal of Cheminformatics **5** (2013), 7. DOI: 10.1186/1758-2946-5-7.

[20] J.-L. Faulon und A. Bender. *Handbook of Chemoinformatics Algorithms.* Taylor und Francis Group (2010).

[21] C. S. Chowdary und P. Mitra. *Novel Method for Improving the Exact Matching of the Molecular Graphs.* International Journal of Recent Trends in Engineering **1** (2009), 254–259.

[22] J.-L. Faulon. *Isomorphism, Automorphism Partitioning, and Canonical Labeling Can Be Solved in Polynomial-Time for Molecular Graphs.* Journal of Chemical Information and Computer Sciences **38** (1998), 432–444. DOI: 10.1021/ci9702914.

[23] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann und E. Willighagen. *The Chemistr Deve Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics.* J. Chem. Inf. Comput. Sci. **43** (2003), 493–500. DOI: 10.1021/ci025584y.

[24] M. D. Lechner. *Einführung in die Quantenchemie. Aufbau der Atome und Moleküle, Spektroskopie*. Springer Spektrum (2017). DOI: https://doi.org/10.1007/978-3-662-49883-5.

[25] W. Thiel. "Chapter 21 - Semiempirical quantum-chemical methods in computational chemistry". *Theory and Applications of Computational Chemistry*. Hrsg. von C. E. Dykstra, G. Frenking, K. S. Kim und G. E. Scuseria. Elsevier 2005, 559–580. DOI: https://doi.org/10.1016/B978-044451719-7/50064-0.

[26] J. J. Stewart. *MOPAC: a semiempirical molecular orbital program*. Journal of computer-aided molecular design **4** (1990), 1–103. DOI: 10.1007/BF00128336.

[27] J. J. P. Stewart. *PM7 accuracy*. 2017. URL: http://openmopac.net/PM7_accuracy/PM7_accuracy.html.

[28] F. Rasche, A. Svatoš, R. K. Maddula, C. Böttcher und S. Böcker. *Computing Fragmentation Trees from Tandem Mass Spectrometry Data*. Analytical Chemistry **83** (2011), 1243–1251. DOI: 10.1021/ac101825k.

[29] Ö. C. Zeki, C. C. Eylem, T. Reçber, S. Kır und E. Nemutlu. *Integration of GC–MS and LC–MS for untargeted metabolomics profiling*. Journal of Pharmaceutical and Biomedical Analysis **190** (2020), 113509–113520. DOI: https://doi.org/10.1016/j.jpba.2020.113509.

[30] C. Ruttkies, E. L. Schymanski, S. Wolf, J. Hollender und S. Neumann. *MetFrag relaunched: incorporating strategies beyond in silico fragmentation*. Journal of Cheminformatics **8** (2016), 1–16. DOI: 10.1186/s13321-016-0115-9.

[31] D. A. Dias, O. A. Jones, D. J. Beale, B. A. Boughton, D. Benheim, K. A. Kouremenos, J.-L. Wolfender und D. S. Wishart. *Current and Future Perspectives on the Structural Identification of Small Molecules in Biological Systems*. Metabolites **6** (2016), 1–29. DOI: 10.3390/metabo6040046.

[32] G. Münzenberg. *Development of mass spectrometers from Thomson and Aston to present*. International Journal of Mass Spectrometry **349-350** (2013). 100 years of Mass Spectrometry, 9–18. DOI: https://doi.org/10.1016/j.ijms.2013.03.009.

[33] J. H. Gross. *Massenspektrometrie - Ein Lehrbuch*. Springer Spektrum (2013).

[34] S. Bienz, L. Bigler, T. Fox und H. Meier. *Spektroskopische Methoden in der organischen Chemie*. 9. Auflage. Georg Thieme Verlag KG (2016).

[35] NIST Mass Spectrometry Data Center. URL: https://chemdata.nist.gov/.

[36] S. Banerjee und S. Mazumdar. *Electrospray Ionization Mass Spectrometry: A Technique to Access the Information beyond the Molecular Weight of the Analyte*. International journal of analytical chemistry (2012), 1–41. DOI: 10.1155/2012/282574.

[37] L. Gruber und A. Gruner. *Grundlagen und Verfahren der Massenspektrometrie*. Springer-Verlag Berlin Heidelberg (2015). DOI: 10.1007/978-3-662-45538-8_36-1.

[38] Wiley Science Solutions. URL: https://sciencesolutions.wiley.com/mass-spectral-databases/.

[39] M. Vinaixa, E. L. Schymanski, S. Neumann, M. Navarro, R. M. Salek und O. Yanes. *Mass spectral databases for LC/MS- and GC/MS-based metabolomics: State of the field and future prospects*. Trends Anal. Chem. **78** (2016), 23–35. DOI: http://dx.doi.org/10.1016/j.trac.2015.09.005.

[40] K. Scheubert, F. Hufsky, F. Rasche und S. Böcker. *Computing Fragmentation Trees from Metabolite Multiple Mass Spectrometry Data*. J. Comput. Biol. **18** (2011), 1383–1397. DOI: 10.1089/cmb.2011.0168.

[41] METLIN. URL: https://www.agilent.com/en/product/liquid-chromatography-mass-spectrometry-lc-ms/lc-ms-application-solutions/life-sciences-solutions/metlin-metabolomics-database-library.

[42] GNPS. URL: https://gnps.ucsd.edu/ProteoSAFe/static/gnps-splash.jsp.

[43] MassBank. URL: https://massbank.eu/MassBank/.

[44] D. H. Nguyen, C. H. Nguyen und H. Mamitsuka. *Recent advances and prospects of computational methods for metabolite identification: a review with emphasis on machine learning approaches*. Briefings in Bioinformatics **20** (Aug. 2019), 2028–2043. DOI: 10.1093/bib/bby066.

[45] HighChem,Ltd. Bratislava, Slovakia, versions after 5.0 available from Thermo Scientific, Waltham.

[46] MS Manager, version 11.01, Advanced Chemistry Development, Toronto, Ontario, Canada, 2077.

[47] A. Kerber, R. Laue, M. Meringer und K. Varmuza. *MOLGEN-MS: Evaluation of low resolution electron impact mass spectra with MS classification and exhaustive structure generation*. Adv Mass Spectrom **15** (2001), 939–40.

[48] E. L. Schymanski, M. Meringer und W. Brack. *Matching structures to mass spectra using fragmentation patterns: are the results as good as they look?* Analytical chemistry **81** (2009), 3608–3617. DOI: 10.1021/ac802715e.

[49] A. W. Hill und R. J. Mortishire-Smith. *Automated assignment of high-resolution collisionally activated dissociation mass spectra using a systematic bond disconnection approach*. Rapid Commun. Mass Spectrom. **19** (2005), 3111–3118. DOI: 10.1002/rcm.2177.

[50] M. Heinonen, A. Rantanen, T. Mielikäinen, J. Kokkonen, J. Kiuru, R. A. Ketola und J. Rousu. *FiD: a software for ab initio structural identification of product ions from tandem mass spectrometric data*. Rapid Communications in Mass Spectrometry **22** (2008), 3043–3052. DOI: 10.1002/rcm.3701.

[51] S. Wolf, S. Schmidt, M. Müller-Hannemann und S. Neumann. *In silico fragmentation for computer assisted identification of metabolite mass spectra*. BMC Bioinformatics **11** (2010), 1–12. DOI: 10.1186/1471-2105-11-148.

[52] C. Ruttkies, N. Strehmel, D. Scheel und S. Neumann. *Annotation of metabolites from gas chromatography/atmospheric pressure chemical ionization tandem mass spectrometry data using an in silico generated compound database and MetFrag*. Rapid Communications in Mass Spectrometry **29** (2015), 1521–1529. DOI: 10.1002/rcm.7244.

[53] F. Rasche, K. Scheubert, F. Hufsky, T. Zichner, M. Kai, A. Svatoš und S. Böcker. *Identifying the Unknowns by Aligning Fragmentation Trees*. Analytical Chemistry **84** (2012), 3417–3426. DOI: 10.1021/ac300304u.

[54] K. Dührkop, M. Fleischauer, M. Ludwig, A. A. Aksenov, A. V. Melnik, M. Meusel, P. C. Dorrestein, J. Rousu und S. Böcker. *SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information*. Nature Methods **16** (Apr. 2019), 299–302. DOI: 10.1038/s41592-019-0344-8.

[55] S. Böcker, M. C. Letzel, Z. Lipták und A. Pervukhin. *SIRIUS: decomposing isotope patterns for metabolite identification†*. Bioinformatics **25** (Nov. 2008), 218–224. DOI: 10.1093/bioinformatics/btn603.

[56] M. Heinonen, H. Shen, N. Zamboni und J. Rousu. *Metabolite identification and molecular fingerprint prediction through machine learning*. Bioinformatics **28** (2012), 2333–2341. DOI: 10.1093/bioinformatics/bts437.

[57] K. Dührkop, H. Shen, M. Meusel, J. Rousu und S. Böcker. *Searching molecular structure databases with tandem mass spectra using CSI:FingerID*. Proceedings of the National Academy of Sciences **112** (2015), 12580–12585. DOI: 10.1073/pnas.1509788112.

[58] F. Allen, A. Pon, M. Wilson, R. Greiner und D. Wishart. *CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra*. Nucleic Acids Research **42** (2014), 94–99. DOI: 10.1093/nar/gku436.

[59] F. Wang, D. Allen, S. Tian, E. Oler, V. Gautam, R. Greiner, T. O. Metz und D. S. Wishart. *CFM-ID 4.0 - a web server for accurate MS-based metabolite identification*. Nucleic Acids Research **50** (Mai 2022), W165–W174. DOI: 10.1093/nar/gkac383.

[60] Y. Djoumbou-Feunang, A. Pon, N. Karu, J. Zheng, C. Li, D. Arndt, M. Gautam, F. Allen und D. S. Wishart. *CFM-ID 3.0: significantly improved ESI-MS/MS prediction and compound identification*. Metabolites **9** (2019), 72. DOI: 10.3390/metabo9040072.

[61] F. Allen, R. Greiner und D. Wishart. *Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification*. Metabolomics **11** (2014), 98–110. DOI: 10.1007/s11306-014-0676-4.

[62] C. A. Bauer und S. Grimme. *Elucidation of Electron Ionization Induced Fragmentations of Adenine by Semiempirical and Density Functional Molecular Dynamics*. J. Phys. Chem. A **118** (2014), 11479–11484. DOI: 10.1021/jp5096618.

[63] J. Cautereels, M. Claeys, D. Geldof und F. Blockhuys. *Quantum chemical mass spectrometry: ab initio prediction of electron ionization mass spectra and identification of new fragmentation pathways*. Journal of Mass Spectrometry **51** (2016), 602–614. DOI: 10.1002/jms.3791.

[64] J. Cautereels und F. Blockhuys. *Quantum Chemical Mass Spectrometry: Verification and Extension of the Mobile Proton Model for Histidine*. Journal of The American Society for Mass Spectrometry **28** (Juni 2018), 1227–1235. DOI: 10.1007/s13361-017-1636-9.

[65] B. G. Janesko, L. Li und R. Mensing. *Quantum Chemical Fragment Precursor Tests: Accelerating de novo annotation of tandem mass spectra*. Analytica Chimica Acta **995** (2017), 52–64. DOI: 10.1016/j.aca.2017.09.034.

[66] J.-A. Schüler, S. Neumann, M. Müller-Hannemann und W. Brandt. *ChemFrag: Chemically meaningful annotation of fragment ion mass spectra*. Journal of Mass Spectrometry **53** (2018), 1104–1115. DOI: 10.1002/jms.4278.

[67] F. L. Pilar. *Bond-order/bond-length and bond-energy/bond-length relations for carbon-oxygen bonds*. Journal of Molecular Spectroscopy **5** (1961), 72–77. DOI: 10.1016/0022-2852(61)90068-6.

[68] A. Weissberg und S. Dagan. *Interpretation of ESI(+)-MS-MS spectra—Towards the identification of "unknowns"*. International Journal of Mass Spectrometry **299** (2011), 158–168. DOI: http://dx.doi.org/10.1016/j.ijms.2010.10.024.

[69] S. Riniker und G. A. Landrum. *Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation*. Journal of Chemical Information and Modeling **55** (2015), 2562–2574. DOI: 10.1021/acs.jcim.5b00654.

[70] E. P. L. Hunter und S. G. Lias. *Evaluated Gas Phase Basicities and Proton Affinities of Molecules: An Update*. Journal of Physical and Chemical Reference Data **27** (1998), 413–656. DOI: 10.1063/1.556018.

[71] I. Mayer und Á. Gömöry. *Semiempirical quantum chemical method for predicting mass spectrometric fragmentations*. Journal of Molecular Structure: THEOCHEM **311** (1994), 331–341. DOI: 10.1016/S0166-1280(09)80070-5.

[72] K. Wiberg. *Application of the pople-santry-segal CNDO method to the cyclopropylcarbinyl and cyclobutyl cation and to bicyclobutane*. Tetrahedron **24** (1968), 1083–1096. DOI: 10.1016/0040-4020(68)88057-3.

[73] R. Improta, G. Scalmani und V. Barone. *Radical cations of DNA bases: some insights on structure and fragmentation patterns by density functional methods*. International Journal of Mass Spectrometry **201** (2000), 321–336. DOI: 10.1016/S1387-3806(00)00225-6.

[74] A. Gossauer. *Struktur und Reaktivität der Biomoleküle*. Helvetica Chimica Acta, Zürich (2006).

[75] E. Schrott und H. P. T. Ammon. *Inhaltsstoffe von Arzneipflanzen*. Springer (2012), 99–113. DOI: 10.1007/978-3-642-13125-7_6.

[76] C. A. of Small Molecule Identification. URL: http://casmi-contest.org/2017/index.shtml.

[77] P. D. Walesch. *Verfikation und Optimierung von ChemFrag für verschiedene Naturstoffklassen*. Bachelorarbeit. Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, (2022).

[78] A. Schmidt. *Analysen an Steroiden zur Verfikation von ChemFrag sowie zur Inhibierung von Acetylcholin- und Buterylcholoninesterase*. Masterarbeit. Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, (2022).

[79] F. Guan, L. R. Soma, Y. Luo, C. E. Uboh und S. Peterman. *Collision-induced dissociation pathways of anabolic steroids by electrospray ionization tandem mass spectrometry*. Journal of the American Society for Mass Spectrometry **17** (2006), 477–489. DOI: 10.1016/j.jasms.2005.11.021.

[80] K. M. Durante und N. P. Li. *Oestradiol level and opportunistic mating in women*. Biology Letters **5** (2009), 179–182. DOI: 10.1098/rsbl.2008.0709.

[81] L. Ma und S. R. Yates. *A review on structural elucidation of metabolites of environmental steroid hormones via liquid chromatography–mass spectrometry.* TrAC Trends in Analytical Chemistry **109** (2018), 142–153. DOI: 10.1016/j.trac.2018.10.007.

[82] S. Bourcier, C. Poisson, Y. Souissi, S. Kinani, S. Bouchonnet und M. Sablier. *Elucidation of the decomposition pathways of protonated and deprotonated estrone ions: application to the identification of photolysis products.* Rapid Communications in Mass Spectrometry **24** (2010), 2999–3010. DOI: 10.1002/rcm.4722.

[83] J. Hau, R. Stadler, T. A. Jenny und L. B. Fay. *Tandem mass spectrometric accurate mass performance of time-of-flight and Fourier transform ion cyclotron resonance mass spectrometry: a case study with pyridine derivatives.* Rapid Communications in Mass Spectrometry **15** (2001), 1840–1848. DOI: 10.1002/rcm.444.

[84] A. Vitvitskaya, F. Naidis, E. Katsnelson und I. Karpinskaya. *Synthesis of Etyhl Ether or Phenylethylcyanoetic acid.* Khimiko-Farmatsevticheskii Zhurnal **15** (1981), 85–88.

[85] Gelbe Liste. URL: https://www.gelbe-list.de/wirkstoffe/.

[86] A. Chepurniak. *Design und Implementation eines Web-Services für die Annotation von Massenspektren mit ChemFrag.* Bachelorarbeit. Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, (2022).

[87] A. G. Maldonado, J. P. Doucet, M. Petitjean und B.-T. Fan. *Molecular similarity and diversity in chemoinformatics: From theory to applications.* Molecular Diversity **10** (Feb. 2006), 39–79. DOI: 10.1007/s11030-006-8697-1.

[88] N. Singh, R. Guha, M. A. Giulianotti, C. Pinilla, R. A. Houghten und J. L. Medina-Franco. *Chemoinformatic Analysis of Combinatorial Libraries, Drugs, Natural Products, and Molecular Libraries Small Molecule Repository.* Journal of Chemical Information and Modeling **49** (Apr. 2009), 1010–1024. DOI: 10.1021/ci800426u.

[89] A. Varnek und I. I. Baskin. *Chemoinformatics as a Theoretical Chemistry Discipline.* Molecular Informatics **30** (2011), 20–32. DOI: 10.1002/minf.201000100.

[90] P. Willett. *Chemoinformatics – similarity and diversity in chemical libraries.* Current Opinion in Biotechnology **11** (2000), 85–88. DOI: 10.1016/S0958-1669(99)00059-2.

[91] C. N. Parker und S. K. Schreyer. "Application of Chemoinformatics to High-Throughput Screening". *Chemoinformatics: Concepts, Methods, and Tools for Drug Discovery*. Hrsg. von J. Bajorath. Humana Press 2004, 85–110. DOI: 10.1385/1-59259-802-1:085.

[92] A. Kerber, R. Laue, M. Meringer, C. Rücker und E. Schymanski. *Mathematical Chemistry and Chemoinformatics: Structure Generation, Elucidation and Quantitative Structure-Property Relationships*. De Gruyter (2013). DOI: doi:10.1515/9783110254075.

[93] PubChem. URL: https://pubchem.ncbi.nlm.nih.gov/.

[94] KEGG. URL: https://www.genome.jp/kegg/pathway.html.

[95] ChEBI. URL: https://www.ebi.ac.uk/chebi/.

[96] J.-A. Schüler, S. Rechner und M. Müller-Hannemann. *MET: a Java package for fast molecule equivalence testing*. Journal of Cheminformatics **12** (2020), 73. DOI: 10.1186/s13321-020-00480-1.

[97] S. A. Rahman, M. Bashton, G. L. Holliday, R. Schrader und J. M. Thornton. *Small Molecule Subgraph Detector (SMSD) toolkit*. Journal of Cheminformatics **1** (2009), 12. DOI: 10.1186/1758-2946-1-12.

[98] T. Akutsu und H. Nagamochi. *Comparison and Enumeration of Chemical Graphs*. Computational and Structural Biotechnology Journal **5** (2013), e201302004. DOI: 10.5936/csbj.201302004.

[99] M. R. Garey und D. S. Johnson. *Computers and Intractability: A Guide to the Theorey of NP-Completness*. W.H. Freeman & CoNew York (1979).

[100] L. Babai. *Graph Isomorphism in Quasipolynomial Time [Extended Abstract]. Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Association for Computing Machinery (2016), 684–697. DOI: 10.1145/2897518.2897542.

[101] K. Mehlhorn und P. Mutzel. *On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm*. Algorithmica **16** (Aug. 1996), 233–242. DOI: 10.1007/BF01940648.

[102] J. E. Hopcroft und J. K. Wong. *Linear Time Algorithm for Isomorphism of Planar Graphs (Preliminary Report). Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*. STOC '74. Association for Computing Machinery (1974), 172–184. DOI: 10.1145/800119.803896.

[103] E. M. Luks. *Isomorphism of graphs of bounded valence can be tested in polynomial time*. Journal of Computer and System Sciences **25** (1982), 42–65. DOI: 10.1016/0022-0000(82)90009-5.

[104] B. D. McKay. *Practical Graph Isomorphism*. Congressus Numerantium **30** (1981), 45–87.

[105] J. R. Ullmann. *An Algorithm for Subgraph Isomorphism*. Journal of the Assocmtlon for Computing Machinery **23** (Jan. 1976), 31–42. DOI: 10.1145/321921.321925.

[106] L. P. Cordella, P. Foggia, C. Sansone und M. Vento. *Subgraph Transformations for the Inexact Matching of Attributed Relational Graphs. Graph Based Representations in Pattern Recognition*. Hrsg. von J.-M. Jolion und W. G. Kropatsch. Springer Vienna (1998), 43–52.

[107] V. Carletti, P. Foggia und M. Vento. *VF2 Plus: An Improved version of VF2 for Biological Graphs. Graph-Based Representations in Pattern Recognition*. Springer International Publishing (2015), 168–177. DOI: 10.1007/978-3-319-18224-7_17.

[108] B. Dezső, A. Jüttner und P. Kovács. *LEMON – an Open Source C++ Graph Template Library*. Electronic Notes in Theoretical Computer Science **264** (2011). DOI: 10.1016/j.entcs.2011.06.003.

[109] N. Schneider, R. A. Sayle und G. A. Landrum. *Get Your Atoms in Order—An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm*. Journal of Chemical Information and Modeling **55** (2015), 2111–2120. DOI: 10.1021/acs.jcim.5b00543.

[110] N. M. O'Boyle und R. A. Sayle. *Comparing structural fingerprints using a literature-based similarity benchmark*. Journal of Cheminformatics **8** (2016), 36. DOI: 10.1186/s13321-016-0148-0.

[111] D. Probst und J.-L. Reymond. *A probabilistic molecular fingerprint for big data settings*. Journal of Cheminformatics **10** (2018), 66. DOI: 10.1186/s13321-018-0321-8.

[112] V. Kumar. *Algorithms for Constraint-Satisfaction Problems: A Survey*. Bd. 13(1). AI Magazine (1992). DOI: 10.1609/aimag.v13i1.976.

[113] P. Beek. *Principles and Practice of Constraint Programming - CP 2005*. Springer Berlin, Heidelberg (2005). DOI: 10.1007/11564751.

[114] S. Böcker und F. Rasche. *Towards de novo identification of metabolites by analyzing tandem mass spectra*. Bioinformatics **24** (2008), i49–i55. DOI: 10.1093/bioinformatics/btn270.

[115] K. Dührkop und S. Böcker. *Research in Computational Molecular Biology (Part: Fragmentation Trees Reloaded)*. Hrsg. von T. M. Przytycka. Bd. 9029. Springer International Publishing (2015), 65–79.

[116] M. Holčapek, R. Jirásko und M. Lísa. *Basic rules for the interpretation of atmospheric pressure ionization mass spectra of small molecules*. Journal of Chromatography A **1217** (2010). Mass Spectrometry: Innovation and Application. Part VI, 3908–3921. DOI: https://doi.org/10.1016/j.chroma.2010.02.049.

[117] J. Gasteiger. *Chemoinformatics: a new field with a long tradition*. Analytical and Bioanalytical Chemistry **384** (Jan. 2006), 57–64. DOI: 10.1007/s00216-005-0065-y.

[118] P. Willett. *Chemoinformatics: a history*. Wiley Interdisciplinary Reviews: Computational Molecular Science **1** (2011), 46–56. DOI: 10.1002/wcms.1.

[119] N. Brown. *Chemoinformatics—an Introduction for Computer Scientists*. ACM Comput. Surv. **41** (Feb. 2009). DOI: 10.1145/1459352.1459353.

[120] F. K. Brown u. a. *Chemoinformatics: what is it and how does it impact drug discovery*. Annual reports in medicinal chemistry **33** (1998), 375–384.

[121] M. A. Stravs, K. Dührkop, S. Böcker und N. Zamboni. *MSNovelist: de novo structure generation from mass spectra*. Nature Methods (Mai 2022). DOI: 10.1038/s41592-022-01486-3.

Anhang

A. Spaltungs- und Umlagerungsregeln

A.1. Umlagerungsregeln

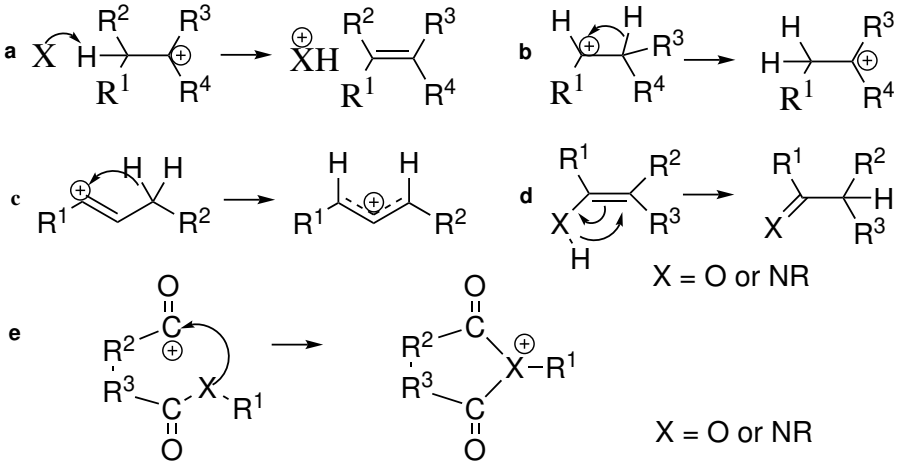
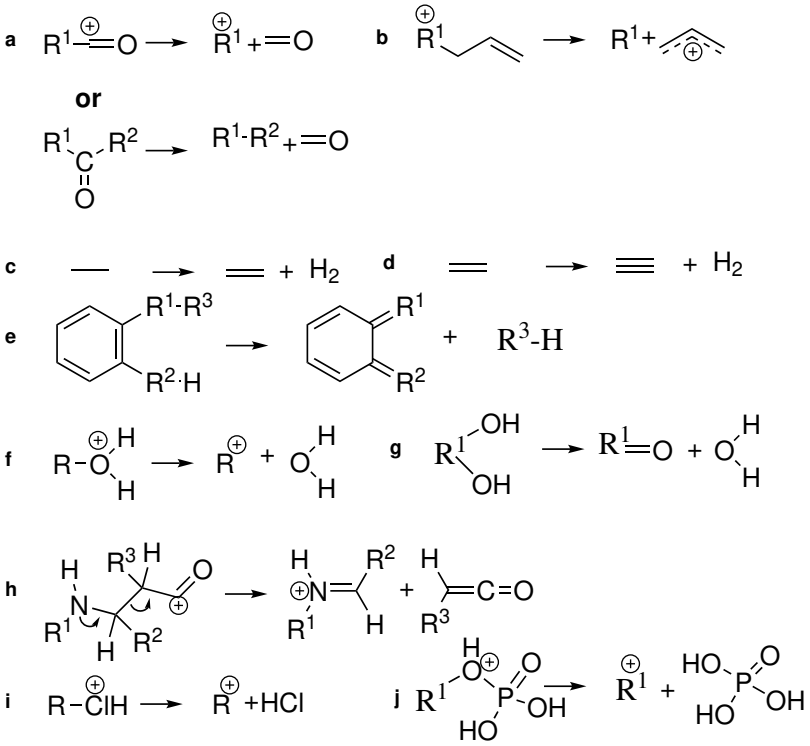


Abbildung S-1: Visualisierung der wichtigsten Umlagerungsreaktionen (angelehnt an [66]).
SMARTS von a) und b) `[C+,c+]-[C,c]`
SMARTS von c) `[C+,c+]~[C,c]~[CH2,ch2]`
SMARTS von d) `[O,N][C,c]=[C,c]`
SMARTS von e) `O=[C+][C,c,N,n][C,c,N,n]C(=O)[O,N][C,c,N,n]`

A.2. Spaltungsregeln



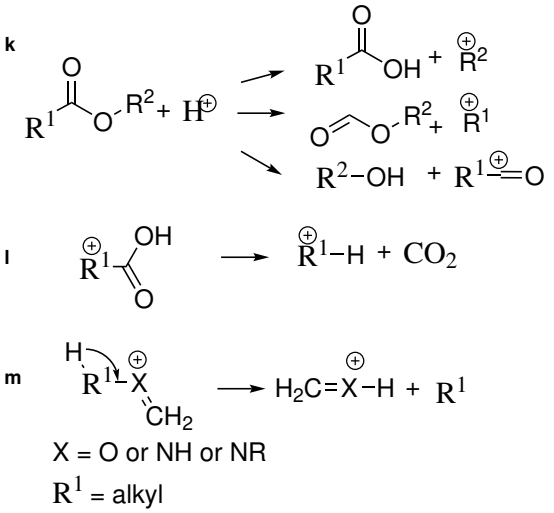


Abbildung S-2: Visualisierung der wichtigsten Spaltungsregeln (angelehnt an [66]).
SMARTS von a) `[C,c]=[O,o]` or `O=[C,c]([C,c])[C,c]`
SMARTS von b) `[C]CC=C`
SMARTS von c) `[C][C]`
SMARTS von d) `[C]=[C]`
SMARTS von e) `[C,N,O]c1[c][c][c][c]c1[C,O,N][O,N,C]`
SMARTS von f) `[OH+]`
SMARTS von g) `OCO`
SMARTS von h) `O=[C+][C,c][C,c]N`
SMARTS von i) `[Cl]`
SMARTS von j) `[C,N,O,S]OP(=O)(O)O`
SMARTS von k) `[C,c][O+][C,c]([C,c])=O`
SMARTS von l `[C,c][C,c](=O)[OH]`
SMARTS von m) `[C]=[O+,N+][C]`

B. Bewertung der Protonenaffinitäten mit PM7 aus MOPAC

Tabelle S-1: Auflistung der experimentell bestimmten Protonenaffinitäten von Hunter [70] sowie der durch PM7 bestimmten Protonenaffinitäten für 172 Reaktionen. Alle Energien sind in *kJ/mol* (angelehnt an [66]).

Edukt	Produkt	exp.	PM7
c1ccc2c(c1)nc1c(c2)cccc1	c1ccc2c(c1)[nH+]c1c(c2)cccc1	-972.6	-945
C(NC)C	C([NH2+]C)C	-942.2	-925
NN	N[NH3+]	-853.2	-829
N=N	N=[NH2+]	-803.0	-766
NH2	[NH3 +]	-773.4	-866
N	[NH4+]	-853.6	-866
CN	C[NH3+]	-899.0	-890
CCN	CC[NH3+]	-912.0	-899
C(CCCCC)CCCCN	C(CCCCC)CCCC[NH3+]	-930.4	-905
N(C)C	[NH2+](C)C	-929.5	-908
CCNCC	CC[NH2+]CC	-952.4	-940
C(C)(C)N	C(C)(C)[NH3+]	-923.8	-917
C(CN)(C)C	C(C[NH3+])(C)C	-924.776	-907
[C@H](C)(CC)N	[C@@H](C)(CC)[NH3+]	-929.679	-922
C(CNCCCC)CC	C(C[NH2+]CCCC)CC	-968.478	-946
C(CNCC(C)C)(C)C	C(C[NH2+]CC(C)C)(C)C	-958.078	-958
[C@@H](C)(CC)N[C@H](C)CC	[C@@H](C)(CC)[NH2+][C@H](C)CC	-980.674	-963
C(CC)NCCC	C(CC)[NH2+]CCC	-962.3	-938
C1CNCCN1	C1CNCC[NH2+]1	-943.7	-927
C=C(CN)C	C=C(C[NH3+])C	-917.5	-910
C=CCN	C=CC[NH3+]	-909.5	-896
C(=N)(C)C	C(=[NH2+])(C)C	-932.2	-927
[C-]#[N+]CC	C=[NH+]CC	-851.3	-1760
C(C#N)C#N	C(C#[NH+])C#N	-723.0	-770
CC#N	CC#[NH+]	-779.2	-813
C[N+]#[C-]	C[N+]#C	-839.1	-938

N(=N)C	N(=[NH2+])C	-845.0	-820
[NH+]#[C-]	[NH+]#C	-772.3	-955
N#C	[NH+]#C	-712.9	-771
CC=N	CC=[NH2+]	-855.1	-906
C=NC	C=[NH+]C	-884.6	-877
N(=N\C)/C	N(=[NH+]\C)/C	-865.1	-833
C(C)(C)(CN)C	C(C)(C)(C[NH3+])C	-928.3	-914
C(N)(C)(CC)C	C([NH3+])(C)(CC)C	-937.8	-936
C(C)(NC)C	C(C)([NH2+]C)C	-952.4	-940
C1CCCN1	C1CCC[NH2+]1	-948.3	-919
CCC#N	CCC#[NH+]	-794.1	-821
CCCC#N	CCCC#[NH+]	-798.4	-824
C(=N\CC)/C	C(=[NH+]\CC)\C	-941.9	-926
C1CNCCC1	C1C[NH2+]CCC1	-954.0	-937
C1CN(CC1)C	C1C[NH+](CC1)C	-965.6	-940
C1CN(N(C1)C)C	C1C[N@H+](N(C1)C)C	-959.3	-917
c1(ccccc1)N	c1(ccccc1)[NH3+]	-882.5	-872
N=C	[NH2+]=C	-852.9	-868
CN	C[NH3+]	-832.8	-890
c1cnccc1	c1c[nH+]ccc1	-930.0	-888
c1ncncn1	c1ncnc[nH+]1	-848.8	-807
c1c2c(ccc1)cccn2	c1c2c(ccc1)ccc[nH+]2	-953.175	-914
c1c2c(ccc1)ccnc2	c1c2c(ccc1)cc[nH+]c2	-951.676	-910
c1ccccc1N	c1c(cccc1)[NH3+]	-882.477	-872
c1cnccn1	c1cncc[nH+]1	-877.1	-840
c1cccnn1	c1ccc[nH+]n1	-907.2	-846
c1ccncn1	c1ccnc[nH+]1	-885.8	-848
C=CC#N	C=CC#[NH+]	-784.7	-818
C=C(C)N	C=C(C)[NH3+]	-941.8	-870
NC=C	[NH3+]C=C	-898.9	-856
CN(C=C)C	C[NH+](C=C)C	-956.8	-898
c1cnccc1CC	c1c[nH+]ccc1CC	-951.1	-907
c1cccc(c1)CC#N	c1cccc(c1)CC#[NH+]	-805.5	-833

O	[OH3+]	-691.0	-678
CO	C[OH2+]	-754.3	-723
C(C)O	C(C)[OH2+]	-776.4	-767
CCCO	CCC[OH2+]	-786.5	-761
OCCCC	[OH2+]CCCC	-789.2	-764
C(CCCO)C	C(CCC[OH2+])C	-795.0	-766
C(CCCO)CC	C(CCC[OH2+])CC	-799.0	-767
C(CCCO)CCC	C(CCC[OH2+])CCC	-799.0	-770
C(CCCO)CCCC	C(CCC[OH2+])CCCC	-799.0	-771
C(O)(C)(C)C	C([OH2+])(C)(C)C	-802.6	-835
O[C@H](C)CC	[OH2+][C@@H](C)CC	-815.0	-796
CC(C)CO	CC(C)C[OH2+]	-793.7	-768
CC(C)O	CC(C)[OH2+]	-793.0	-790
CC(C)(CO)C	CC(C)(C[OH2+])C	-795.5	-773
C=O	C=[OH+]	-712.9	-723
CC=O	CC=[OH+]	-768.5	-773
C(C=O)C	[OH+]=CCC	-786.0	-782
C(CC=O)C	C(CC=[OH+])C	-792.7	-773
C(=O)CCCC	C(=[OH+])CCCC	-796.6	-788
CCCC(=O)CCC	CCCC(=[OH+])CCC	-845.0	-823
CCCCC(=O)CCCC	CCCCC(=[OH+])CCCC	-853.7	-827
O=CC=C	[OH+]=CC=C	-797.0	-795
C/C=C/C=O	C/C=C/C=[OH+]	-830.8	-817
C(C(=O)C)C	C(C(=[OH+])C)C	-827.3	-816
O=C/C(=C\C)/C	[OH+]=C/C(=C\C)/C	-843.9	-834
CC(=CC=O)C	CC(=CC=[OH+])C	-856.9	-845
O=C(C)C	[OH+]=C(C)C	-812.0	-803
CCC(=O)CC	CCC(=[OH+])CC	-836.8	-822
CC(=O)CCC	CC(=[OH+])CCC	-832.7	-812
C(C(=O)C)(C)(C)C	C(C(=[OH+])C)(C)(C)C	-840.1	-828
CC(C)C(=O)C(C)C	CC(C)C(=[OH+])C(C)C	-850.3	-832
C/C=C/C(=O)C	C/C=C/C(=[OH+])C	-864.3	-842

CC(C)C(=O)C	CC(C)C(=[OH+])C	-836.3	-820
CC(=O)/C=C/CC	CC(=[OH+])/C=C/CC	-865.6	-841
CCC(=O)CCC	CCC(=[OH+])CCC	-843.2	-821
C/C=C(/C)\C(=O)C	C/C=C(\C)/C(=[OH+])C	-866.4	-880
COC	C[OH+]C	-792.0	-743
C=CCOCC=C	C=CC[OH+]CC=C	-827.4	-807
CCO/C=C/C	CC[OH+]/C=C/C	-867.6	-785
CCOCC=C	CC[OH+]CC=C	-833.7	-808
O=CO	[OH+]=CO	-742.0	-726
CC(=O)O	CC(=[OH+])O	-783.7	-765
OC(=O)CC	OC(=[OH+])CC	-797.2	-741
OC(=O)C(=C)C	OC(=[OH+])C(=C)C	-816.7	-780
C1CCCCC1=O	C1CCCCC1=[OH+]	-841.0	-819
c1ccccc1O	c1ccccc1[OH2+]	-817.3	-724
O1[C@@H](C1)C	[OH+]1[C@@H](C1)C	-803.3	-774
COC=C	C[OH+]C=C	-895.2	-737
CCOC	CC[OH+]C	-808.6	-777
C=COCC	C=C[OH+]CC	-870.1	-769
C(C=O)(C)C	C(C=[OH+])(C)C	-797.3	-777
C1COCCO1	C1C[OH+]CCO1	-797.4	-753
C1COCOC1	C1COC[OH+]C1	-825.4	-783
C(OC)(C)C	C([OH+]C)(C)C	-826.3	-800
C(OC)CC	C([OH+]C)CC	-814.9	-776
O(CC)CC	[OH+](CC)CC	-828.4	-806
C1COC=CC1	C1C[OH+]C=CC1	-865.8	-750
O=C/C=C/CC	[OH+]=C/C=C/CC	-839.0	-835
CCO/C=C/C	CC[OH+]/C=C/C	-876.9	-785
CC(C)OCC	CC(C)[OH+]CC	-842.7	-826
CC(C)(OC)C	CC(C)([OH+]C)C	-841.6	-830
CCCCOC	CCCC[OH+]C	-820.3	-779
C#CCOCC#C	C#CC[OH+]CC#C	-783.9	-802

C1CC(=O)CCC1=O	C1CC(=O)CCC1=[OH+]	-812.5	-779
CC(=CC(=O)C)C	CC(=CC(=[OH+])C)C	-878.7	-857
C1COCCCC1	C1C[OH+]CCCC1	-834.2	-795
O(C(C)C)C(C)C	[OH+](C(C)C)C(C)C	-855.5	-843
CCCOCCC	CCC[OH+]CCC	-837.9	-805
COCC(C)(C)C	C[OH+]CC(C)(C)C	-825.8	-796
O=COC	[OH+]=COC	-782.5	-789
CC(=O)OC	CC(=[OH+])OC	-821.6	-790
O=COCC	[OH+]=COCC	-799.4	-779
C=CC(=O)OC	C=CC(=[OH+])OC	-825.8	-799
CCC(=O)OC	CCC(=[OH+])OC	-830.2	-803
COC(=O)CCC	COC(=[OH+])CCC	-836.4	-805
CC(=O)OCCC	CC(=[OH+])OCCC	-836.6	-834
C(C)(C)(C)OCC	C(C)(C)(C)[OH+]CC	-856.0	-845
C1CCCCC1CO	C1CCCCC1C[OH2+]	-802.1	-759
C1CCCCC1OC	C1CCCCC1[OH+]C	-840.5	-807
c1cccc(c1)C(=O)O	c1cccc(c1)C(=[OH+])O	-821.1	-795
O=C=O	O=C=[OH+]	-540.5	-587
C(=O)(C)C(=O)C	C(=[OH+])(C)C(=O)C	-801.9	-755
C=CC=O	C=CC=[OH+]	-797.0	-784
O=C=CC	[OH+]=C=CC	-834.1	-679
C=C(C=O)C	C=C(C=[OH+])C	-808.7	-801
C=CC(=O)C	C=CC(=[OH+])C	-834.7	-817
C(=CC(=O)C)(C)C	C(=CC(=[OH+])C)(C)C	-878.7	-871
C(=O)N	C(=[OH+])N	-822.2	-809
[N+](=O)([O-])C	[N+](=O)(O)C	-754.6	-676
CC(=O)N	CC(=[OH+])N	-863.6	-836
C(=O)NC	C(=[OH+])NC	-851.3	-839
C=CC(=O)N	C=CC(=[OH+])N	-870.7	-840
CN(C)C=O	CN(C)C=[OH+]	-887.5	-863
C(C(=O)N)C	C(C(=[OH+])N)C	-876.2	-844
CC(=O)NC	CC(=[OH+])NC	-888.5	-863

C(C(=O)NC)C	C(C(=[OH+])NC)C	-920.4	-866
C(C(=O)N)(C)C	C(C(=[OH+])N)(C)C	-878.6	-851
CCNC(=O)C	CCNC(=[OH+])C	-898.0	-869
CC(=O)N(C)C	CC(=[OH+])N(C)C	-908.0	-884
CCCNC=O	CCCNC=[OH+]	-878.4	-847
C=CC(=O)N	C=CC(=[OH+])N	-904.3	-843
C(C(=O)N)(C)(C)C	C(C(=[OH+])N)(C)(C)C	-889.0	-863
CC(=C)C(=O)N(C)C	CC(=C)C(=[OH+])N(C)C	-911.5	-897
CN(C)C(=O)/C=C/C	CN(C)C(=[OH+])/C=C/C	-930.3	-908
CN(C)C(=O)CCC	CN(C)C(=[OH+])CCC	-921.7	-885
N(C(=O)C)(CC)CC	N(C(=[OH+])C)(CC)CC	-925.4	-895
N(C(=O)C(C)C)(C)C	N(C(=[OH+])C(C)C)(C)C	-923.7	-892
C(C(=O)N(C)C)(C)(C)C	C(C(=[OH+])N(C)C)(C)(C)C	-927.1	-910
c1cc(ccc1C(=O)N)C	c1cc(ccc1C(=[OH+])N)C	-900.9	-868
c1c(cccc1C(=O)N)C	c1c(cccc1C(=[OH+])N)C	-900.9	-861
c1cc(ccc1C(=O)C)N	c1cc(ccc1C(=[OH+])C)N	-908.8	-894
c1ccccc1[N+](=O)[O-]	c1ccccc1[N+](=O)O	-800.3	-632
OCCN	OCC[NH3+]	-930.3	-876

C. Studie zur Annotation von Massenspektren aus Dopingsubstanzen

C.1. Vergleich der Ergebnisse der Annotation des Massenspektrums für Ephedrin

Publizierter Fragmentierungsweg für Ephedrin durch Thevis

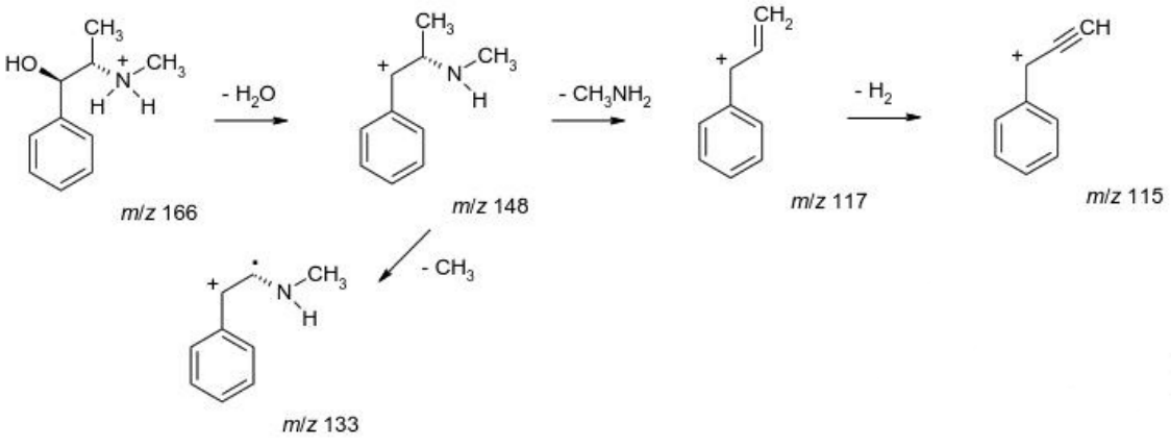


Abbildung S-3: Publizierte Fragmentierung für Ephedrin durch Thevis [8] (angelehnt an [66]).

Liste an selektierten und nicht selektierten Fragmenten durch `ChemFrag` während der Fragmentierung von Ephedrin

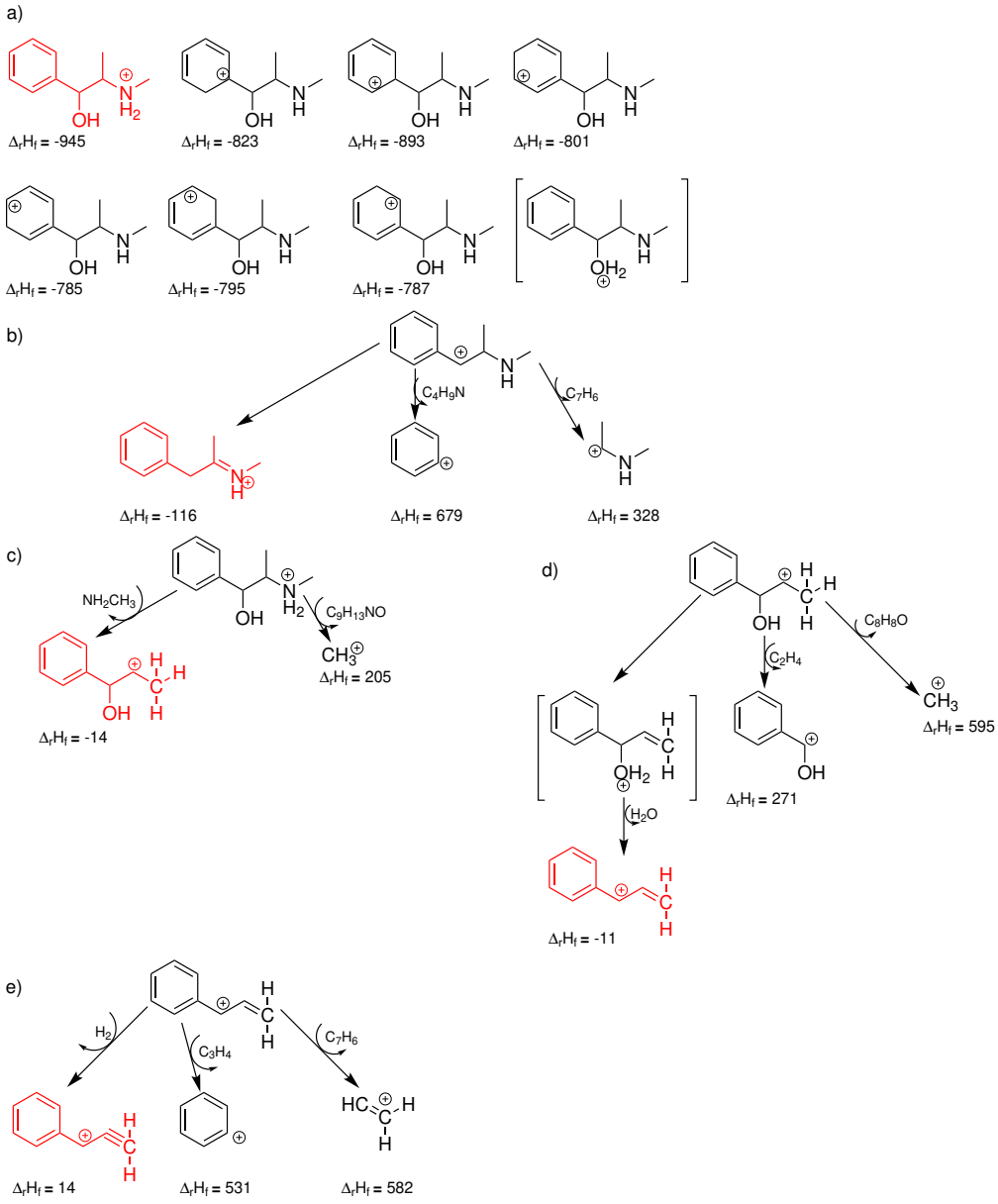


Abbildung S-4: Liste angewählten und nicht ausgewählten Fragmenten durch ChemFrag während der Fragmentierung von Ephedrin.
Abb. a) alle protonierten Moleküle, Intervall von $\Delta_f H_f$ = [-945,-895]
Abb. b) alle Fragmente mit Edukt E2, Intervall von $\Delta_f H_f$ = [-116,35]
Abb. c) alle Fragmente mit Edukt E6, Intervall von $\Delta_f H_f$ = [-14,136]
Abb. d) alle Fragmente mit Edukt E7, Intervall von $\Delta_f H_f$ = [-11,139]
Abb. e) alle Fragmente mit Edukt E9, Intervall von $\Delta_f H_f$ = [14,164]
Die ausgewählten Fragmente sind rot hervorgehoben.

C.2. Vergleich der Ergebnisse zur Annotation des Massenspektrums für Kokain

Publizierter Fragmentierungsweg von Kokain durch Thevis

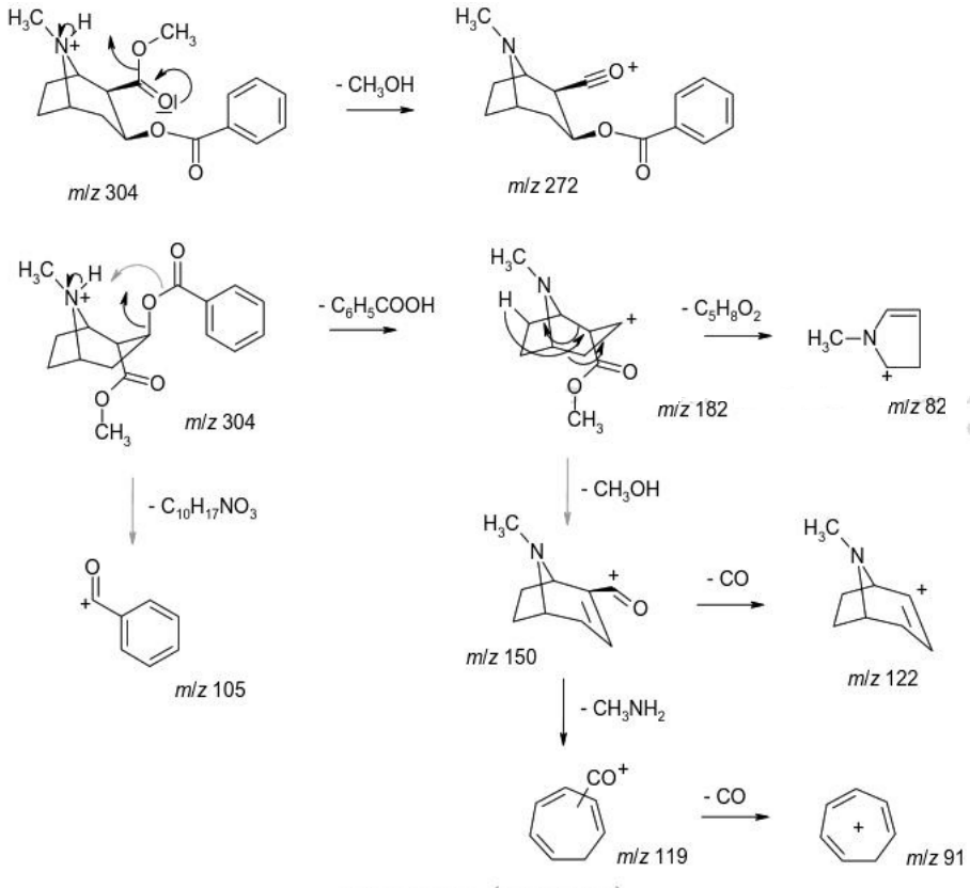


Abbildung S-5: Publizierte Fragmentierung für Kokain durch Thevis [8] (angelehnt an [66]).

C.3. Ergebnisse weiterer Annotationen von Massenspektren der Doping-Substanzen

Beispiel 3: Clenbuterol

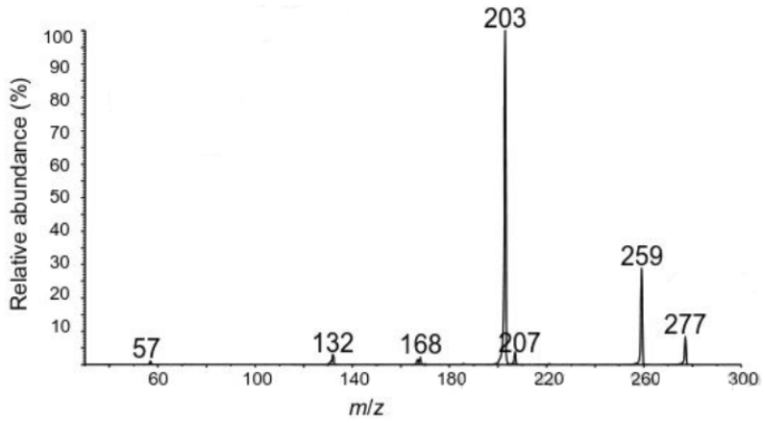
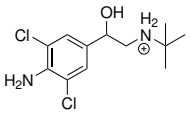
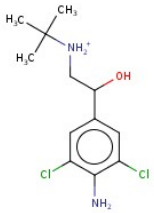
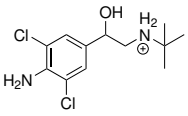
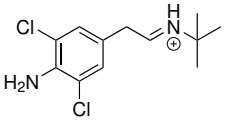
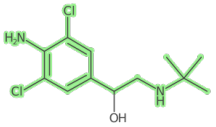
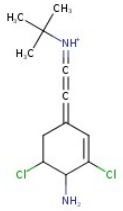
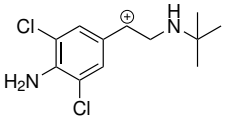
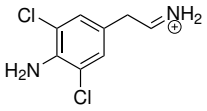
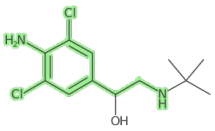
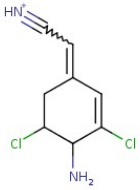
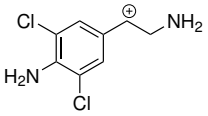
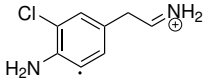
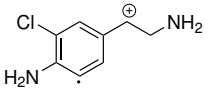
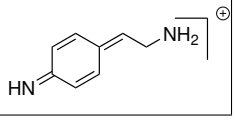


Abbildung S-6: Positiv ionisiertes ESI CID Massenspektrum von Clenbuterol aus Thevis [8] (angelehnt an [66]).

Tabelle S-2: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von Clenbuterol. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind (angelehnt an [66].)
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	ChemFrag	MetFrag *	CFM-ID *	Literatur [8]
277				
259		 $[C_{12}H_{17}Cl_2N_2]^+$		

m/z	ChemFrag	MetFrag *	CFM-ID *	Literatur [8]
203		 $[C_8H_7Cl_2N_2]+H^+$		
168				
132				

Beispiel 4: LDG-2226

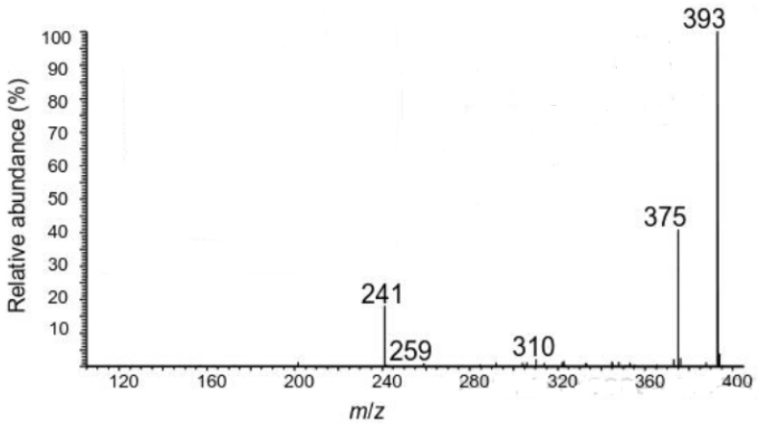
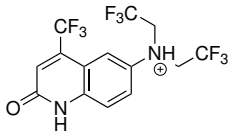
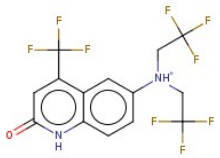
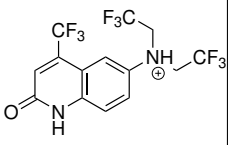
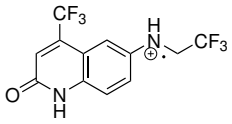
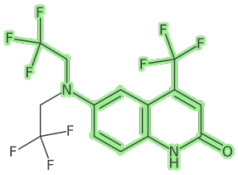
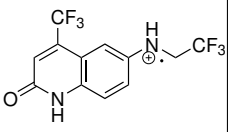
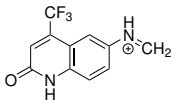
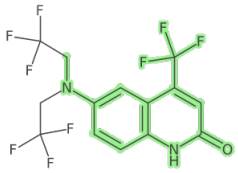
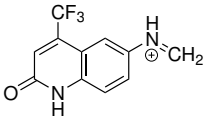
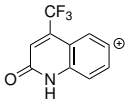
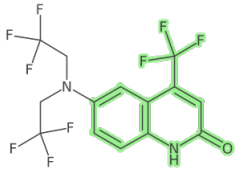
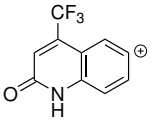


Abbildung S-7: Positiv ionisiertes ESI CID Massenspektrum von LDG-2226 aus Thevis [8] (angelehnt an [66]).

Tabelle S-3: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von LDG-2226. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind (angelehnt an [66]).
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	ChemFrag	MetFrag*	CFM-ID*	Literatur [8]
393				
310		 $[C_9H_9F_9N_2O]^+$		
241		 $[C_8H_9F_6N_2O]+H^+$		
213		 $[C_6H_9F_6N_2O]+H^+$		

Beispiel 5: Pethidin

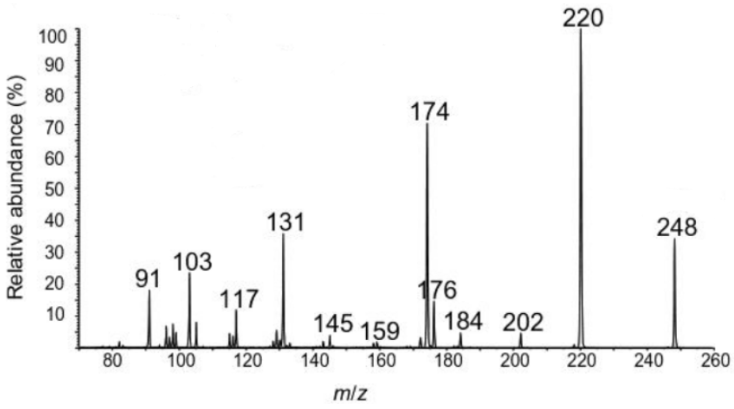
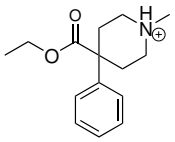
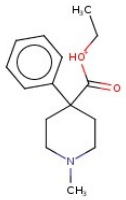
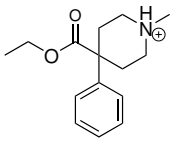
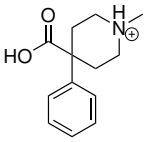
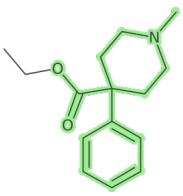
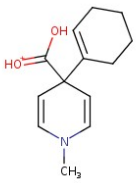
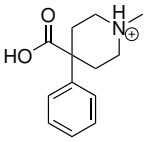
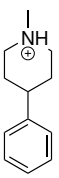
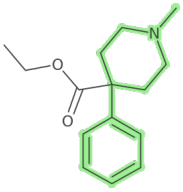
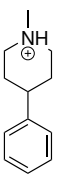
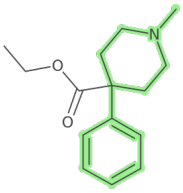
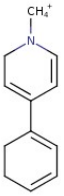
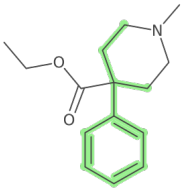
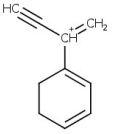
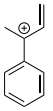


Abbildung S-8: Positiv ionisiertes ESI CID Massenspektrum von Pethidin aus Thevis [8] (angelehnt an [66]).

Tabelle S-4: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von Pethidin. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind (angelehnt an [66]).
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	ChemFrag	MetFrag*	CFM-ID*	Literatur [8]
248				
220		 $[C_{13}H_{15}NO_2+H]+H^+$		
176		 $[C_{10}H_{11}NO_2+H]+H^+$		

174		 $[C_{10}H_{11}NO_2]^+$		
131		 $[C_7H_6NO_2]^+$		

Beispiel 6: 5-Amini-4-Imidazolecarboxamid Ribonukleosid (AICAR)

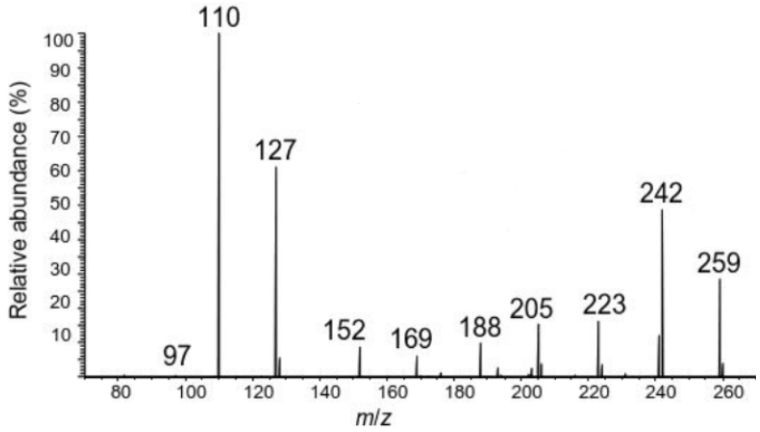
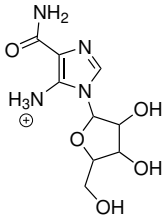
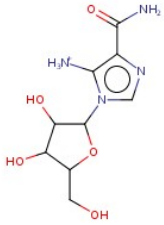
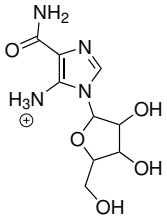
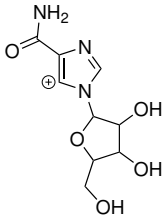
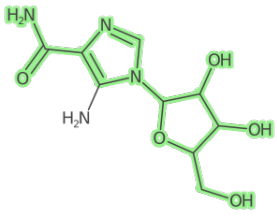
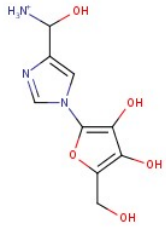
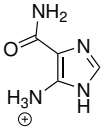
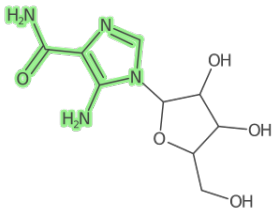
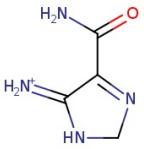
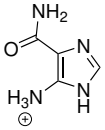
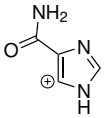
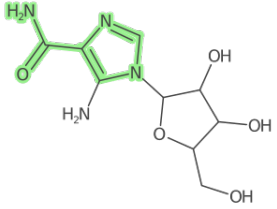
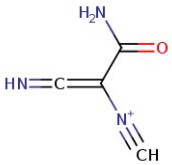
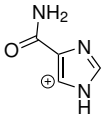


Abbildung S-9: Positiv ionisiertes ESI CID Massenspektrum von AICAR aus Thevis [8] und die zugehörige Molekülstruktur (angelehnt an [66]).

Tabelle S-5: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von AICAR. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind (angelehnt an [66]).
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	ChemFrag	MetFrag*	CFM-ID*	Literatur [8]
259				
242		 $[C_9H_{13}N_4O_4]^+$		
127		 $[C_5H_8O_4{+}H]{+}H^+$		
110		 $[C_5H_7O_3]{+}H^+$		

Beispiel 7: JTV-519

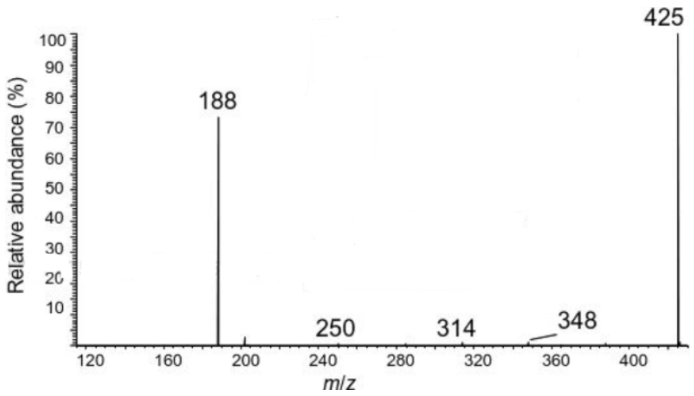
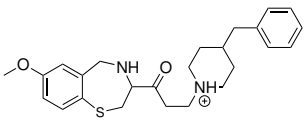
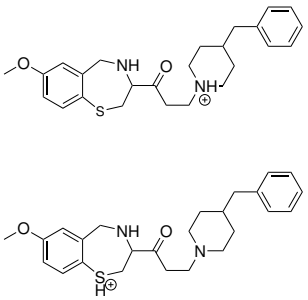
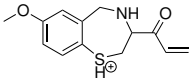
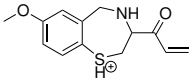
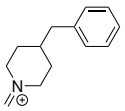
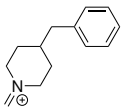


Abbildung S-10: Positiv ionisiertes ESI CID Massenspektrum von JTV-519 aus Thevis [8] und die zugehörige Molekülstruktur (angelehnt an [66]).

Tabelle S-6: Visualisierung der Ergebnisse von `ChemFrag`, `MetFrag` [30], `CFM-ID` [60] und den bekannten Strukturen aus der Literatur [8] zu den zugehörigen Peaks von JTV-519. Die Energieberechnung von `MetFrag` Strukturen ist nicht möglich, da die Positionen der positiven Ladung nicht erkennbar sind (angelehnt an [66]).
* Originalbilder aus der Programmausgabe
** semi-empirische Energieberechnung nicht möglich

m/z	ChemFrag	MetFrag*	CFM-ID*	Literatur [8]
425				
250				
188				

D. Analyse der Fragmentierungstiefe

Tabelle S-7: Darstellung der vorkommenden Fragmentierungstiefen mit deren Häufigkeit für die Parameter -Tprot 50 -Tfrag 150 -Trearr 100 -TBO 1.0 -TD 200.

Baumtiefe	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Häufigkeit	4	2	9	16	38	38	37	24	22	20	14	7	7	4	5	1	1

Tabelle S-8: Darstellung der vorkommenden Fragmentierungstiefen mit deren Häufigkeit für die Parameter -Tprot 50 -Tfrag 200 -Trearr 150 -TBO 0.8 -TD 200.

Baumtiefe	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	19
Häufigkeit	1	4	4	14	25	44	29	38	26	22	10	12	10	5	2	3

E. Anwendbarkeit der Regeln

Tabelle S-9: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Umlagerungsregeln für den Parametersatz 2.

Regel	Matchings	Vorkommen in Fragment-Ionen
Hydrid-Shift:	77128	52273
ProtonShift:	55904	104713
OCCRearr:	26414	19717
NCCProtShift:	8386	0
HShiftRad:	6014	1652
CCFeElRearr:	4718	3401
CORearr:	3296	0
AllylShift:	2047	2756
CNFeElRearr:	1484	441
NposRad:	1178	0
Cyclisation:	255	29
COFeElRearr:	49	0
ThionoThioloRearr:	35	0
SulfideCyclisation:	17	4
AnionBackw:	0	0
AnionRearr:	0	0
CNRearr:	0	0
COCreation:	0	0
PericyclicShift:	0	0

Tabelle S-10: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Spaltungsregeln für den Parametersatz 2.

Regel	Matchings	Vorkommen in Fragment-Ionen
homo_bond_cleavage	98318	203849
RemH2General	21694	12145
hetero_bond_cleavage	14538	13177
COReduction	3191	16877
ChlorCleavage	2931	3839
LCleavage	1456	543
AllylRule	722	171
RemovalHrad	594	170
AromaticElimination	388	181
EthenRemoval	363	3890
InductiveCleavage	362	47
McLaffertyRearrangement	236	396
CarboxylCleavage	219	1274
RDAReaction	157	41
OriginH20	101	0
PhosphoRemoval	89	2
OrthoRemoval	79	490
RemH2_C2C4	77	0
NHTransition	33	9
SOCleavage	25	0
H2OSulfonCleavage	22	31
CONCleavage	17	0
SO2Removal	7	2
AmmoniumRemoval	3	0
BenzylRule	0	0
EsterRule	0	0
OniumRule	0	0
QinoneCleavage	0	0
ReduC2H2Aro	0	0
RetroeneReaction	0	0
SOCleavQinone	0	0
RemOH	23744	31284
RemoveH2O	16692	49757

Tabelle S-11: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Umlagerungsregeln für den Parametersatz 3.

Regel	Matchings	Vorkommen in Fragment-Ionen
Hydrid-Shift	131295	85894
ProtonShift	97236	181568
OCCRearr	43214	28505
NCCProtShift	15569	0
HShiftRad	11094	2595
CCFeElRearr	8496	14507
CORearr	5891	0
AllylShift	3564	5059
CNFeElRearr	3239	649
NposRad	2542	0
Cyclisation	293	126
COFeElRearr	95	0
ThionoThioloRearr	35	0
SulfideCyclisation	22	3
AnionBackw	0	0
AnionRearr	0	0
CNRearr	0	0
COCreation	0	0
PericyclicShift	0	0

Tabelle S-12: Übersicht zur Anwendung und Erläuterung der Fragment-Ionen durch die implementierten Spaltungsregeln für den Parametersatz 3.

Regel	Matchings	Vorkommen in Fragment-Ionen
homo_bond_cleavage	160824	350768
RemH2General:	37758	21256
hetero_bond_cleavage:	23044	21248
COReduction:	5482	35902
ChlorCleavage:	4717	5698
LCleavage:	2637	718
AllylRule:	1425	229
RemovalHrad:	1158	402
InductiveCleavage:	696	54
EthenRemoval:	614	16978
AromaticElimination:	506	230
McLaffertyRearrangement:	420	415
CarboxylCleavage:	358	2558
RDAReaction:	237	89
OriginH20:	174	0
RemH2_C2C4:	124	0
PhosphoRemoval:	120	1
OrthoRemoval:	118	870
NHTransition:	46	12
H2OSulfonCleavage:	39	42
SOCleavage:	29	0
CONCleavage:	28	0
SO2Removal:	7	9
AmmoniumRemoval:	4	0
BenzylRule:	0	0
EsterRule:	0	0
OniumRule:	0	0
QinoneCleavage:	0	0
ReduC2H2Aro:	0	0
RetroeneReaction:	0	0
SOCleavQinone:	0	0
RemOH:	38947	51156
RemoveH2O:	37862	96416

F. Analyse für weitere Naturstoffe

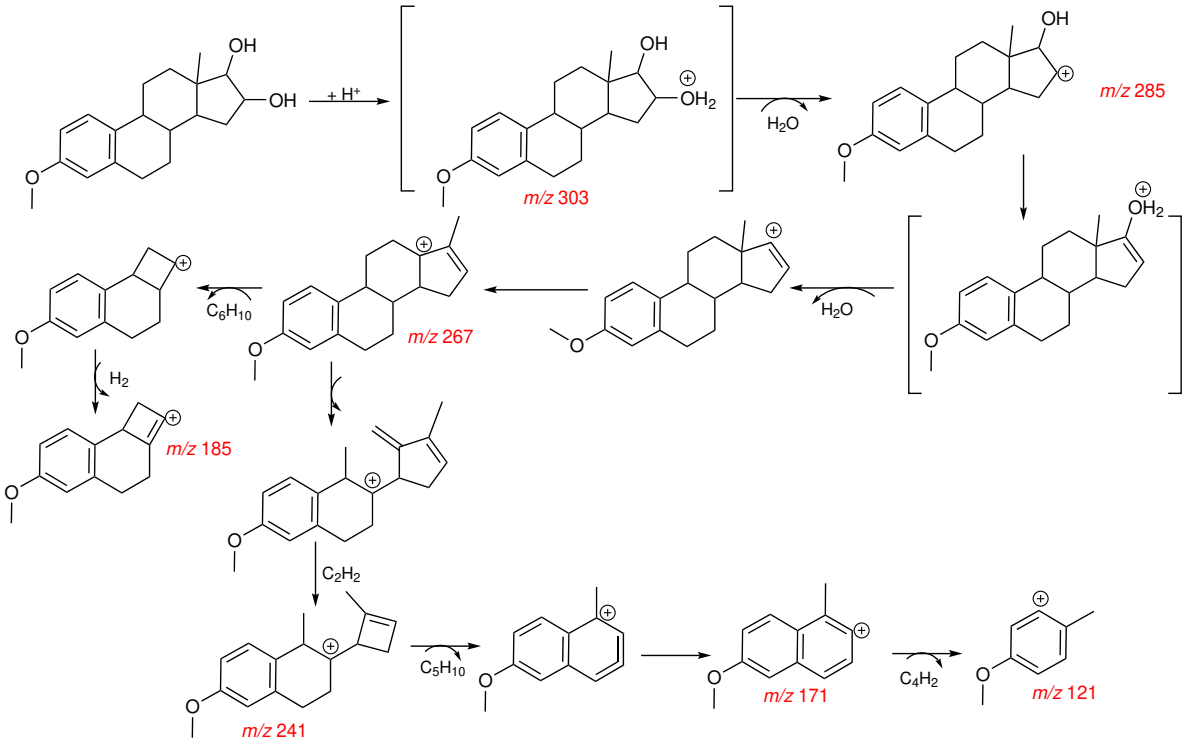


Abbildung S-11: Fragmentierungsweg Estradiol-3-methylether.

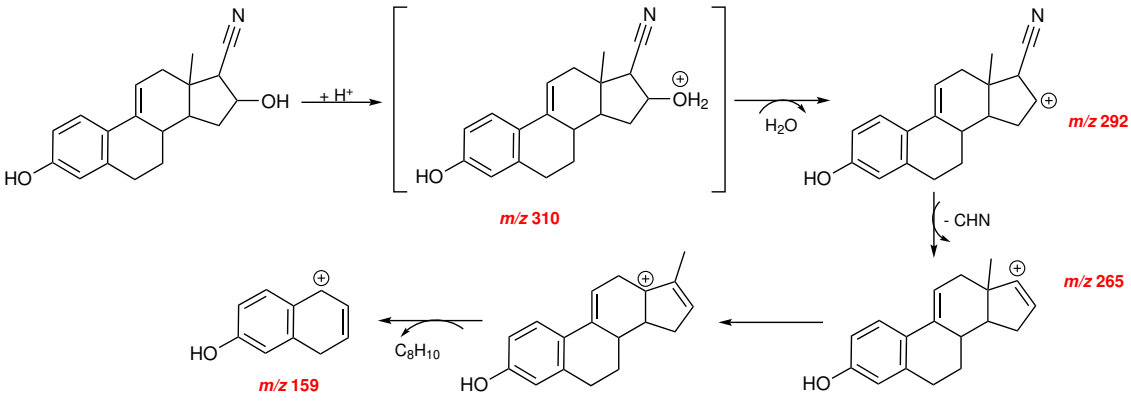


Abbildung S-12: Fragmentierungsweg von Dehydrocyanomethylestradiol.

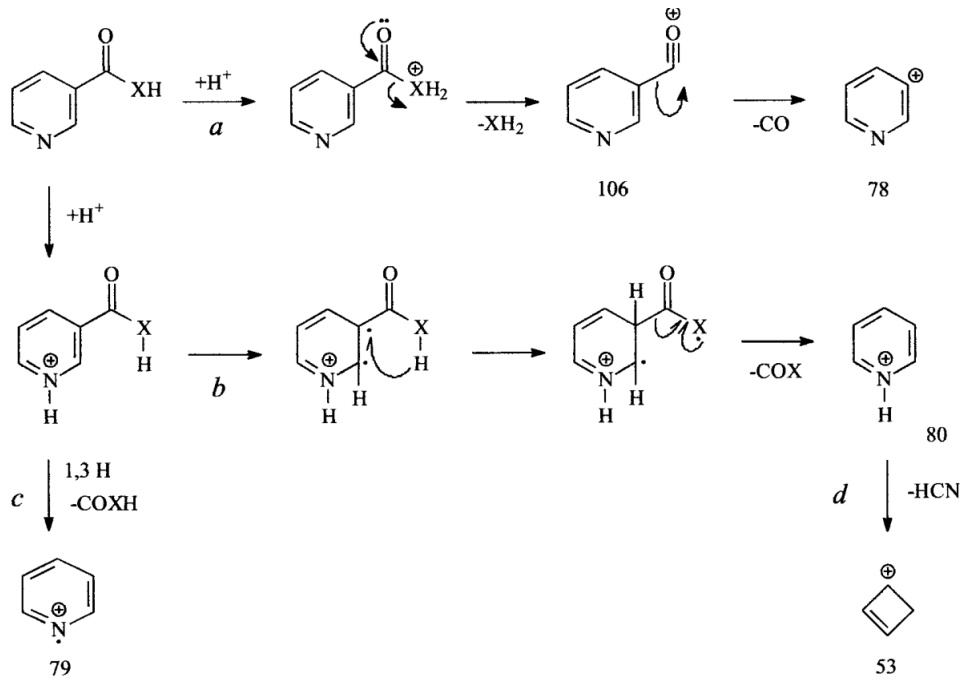


Abbildung S-13: Fragmentierungsweg von Nikotinamid nach Hau *et al* [83]. Der Rest X ist durch ein Sauerstoffatom oder ein Stickstoffatom zu ersetzen.

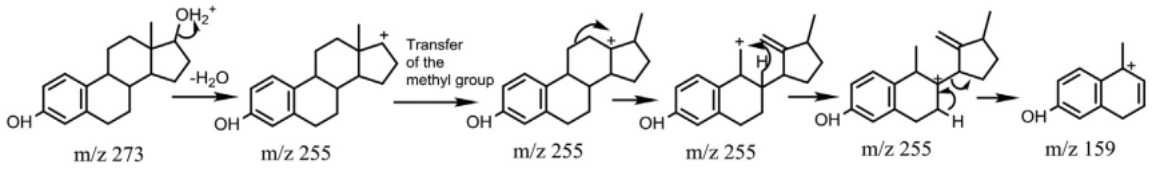


Abbildung S-14: Fragmentierungsweg von 17-β-Estradiol nach Ma *et al* [81].

G. Laufzeitanalyse für verschiedene Nachbarschaftsdeskriptoren

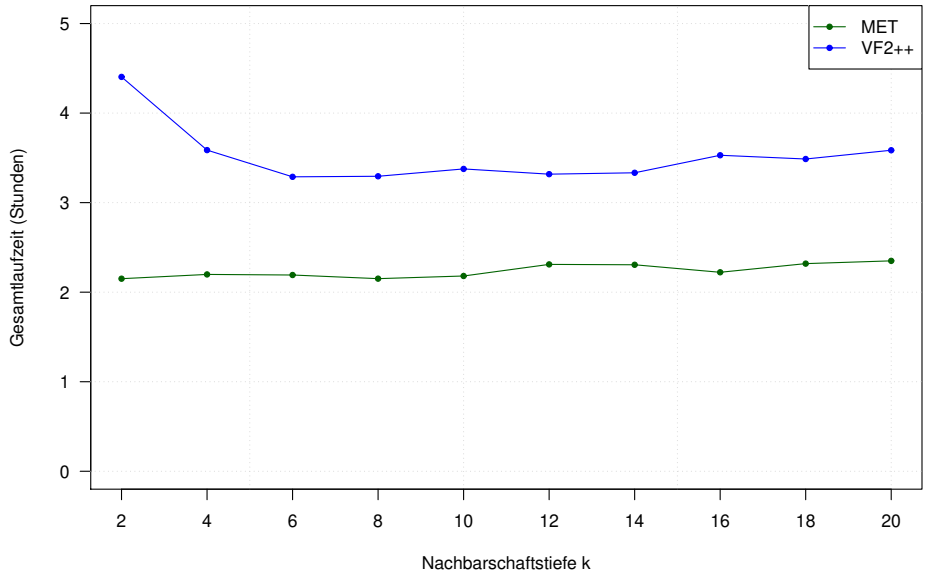


Abbildung S-15: Die Abbildung zeigt für $1 \leq k \leq 20$ die Laufzeitanalyse der Gruppe 2 .

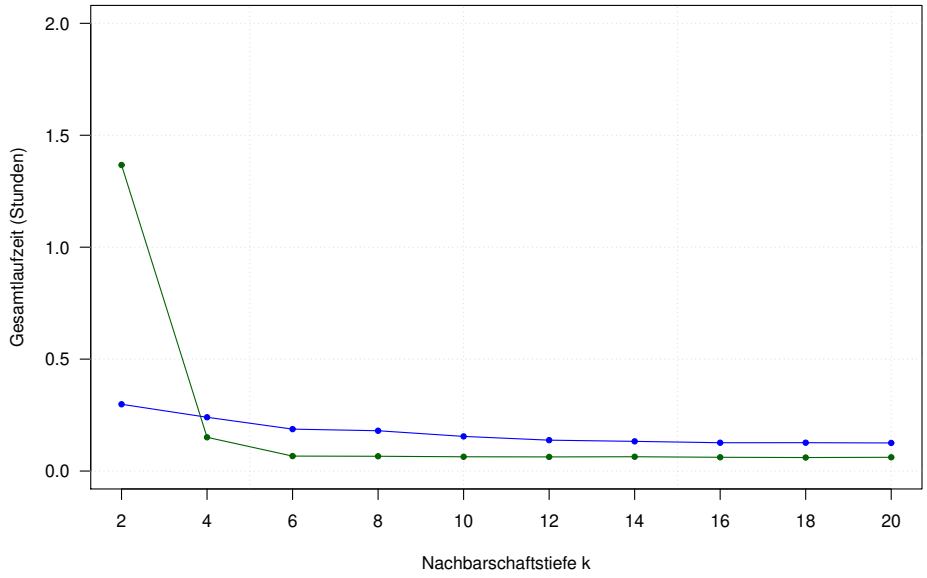


Abbildung S-16: Die Abbildung zeigt für $1 \leq k \leq 20$ die Laufzeitanalyse der Gruppe 4.

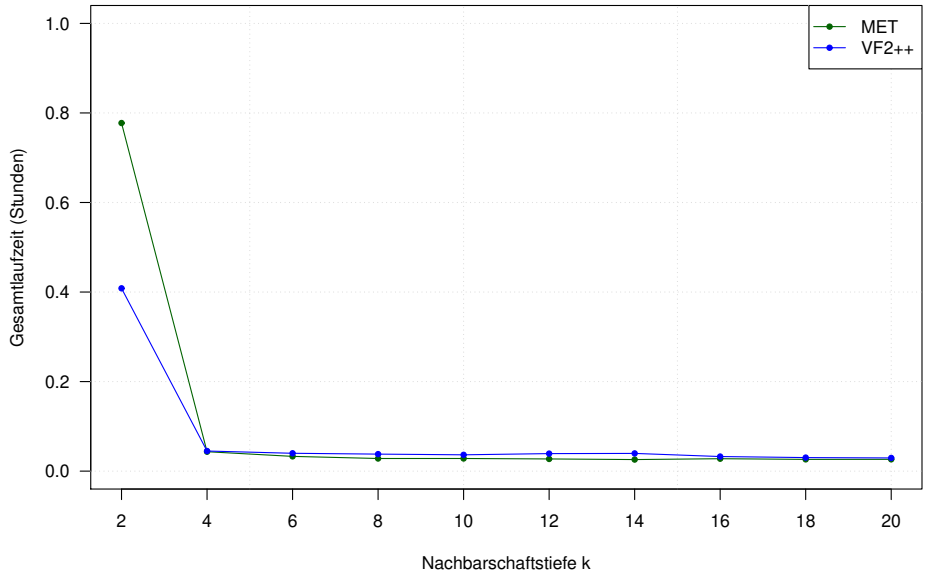


Abbildung S-17: Die Abbildung zeigt für $1 \leq k \leq 20$ die Laufzeitanalyse der Gruppe 5 .

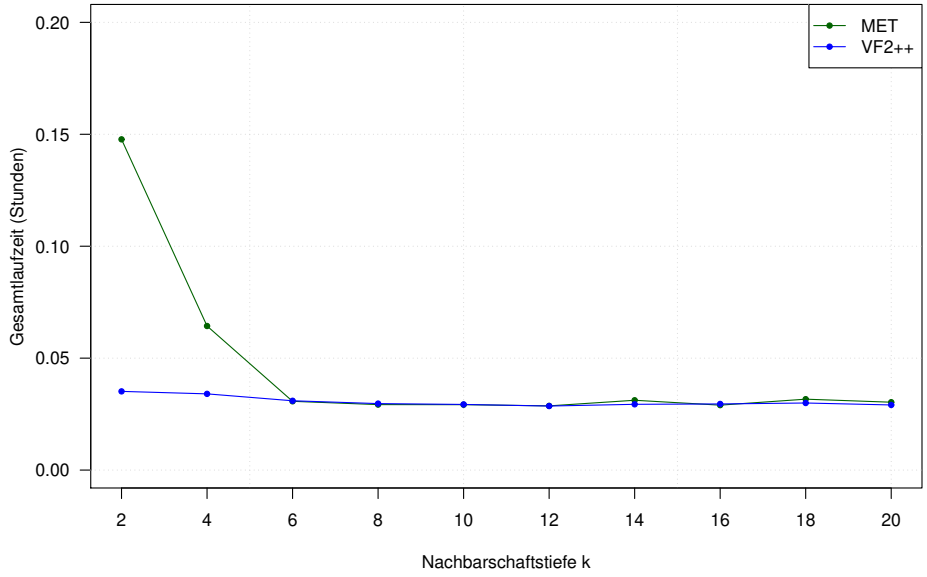


Abbildung S-18: Die Abbildung zeigt für $1 \leq k \leq 20$ die Laufzeitanalyse der Gruppe 6.

H. Partition Analyse

H.1. Auflistung der Fingerprints der Äquivalenzfamilie mit Größe 6

Für die Gruppe 1 (Atomanzahl maximal 25) existieren 45 Äquivalenfamilien, die jeweils sechs Äquivalenzgruppen enthalten. Die Tabelle enthält die Fingerprints der 45 Repräsentanten.

Tabelle S-13: Auflistung der 45 Repräsentanten der Äquivalenzfamilien, die jeweils sechs Äquivalenzklassen enthalten.

108_0_0_0_0_0_0_0_1359030320	68_0_0_0_0_0_4_0_1515642536
45_10_4_0_7_0_0_0_-54801393	38_10_2_0_8_0_0_0_1904215408
60_14_4_0_6_0_0_0_-2529466837	95_0_0_0_0_0_0_0_-657707374
76_24_0_0_20_0_0_0_864149084	45_12_2_0_9_0_0_0_2768025037
39_10_2_0_7_0_0_0_3091911216	91_20_6_2_21_0_0_0_-3854155605
36_6_6_0_3_2_1_0_1729390708	121_14_6_0_5_0_0_0_-802784314
40_8_4_0_4_0_0_0_5751747890	49_12_4_0_11_0_0_0_-2754079188
39_12_0_0_8_0_0_1_-1697509330	39_8_4_0_5_0_0_0_269084786
113_18_6_0_1_0_0_0_3211058136	61_16_6_0_11_0_0_0_-1067362705
39_12_0_0_8_0_0_1_2597457966	61_16_6_0_11_0_0_0_1420952954
60_14_6_0_14_0_0_0_1177560808	163_44_4_0_43_0_0_0_-4136346322
48_12_0_0_8_0_0_0_1543025825	123_38_0_0_37_0_0_0_3923236618
47_10_2_0_7_0_0_0_-2467496413	39_10_2_0_6_0_0_1_-225368464
54_0_0_0_0_0_0_0_-871797841	74_18_8_0_12_0_0_0_-6106763578
41_8_2_0_4_0_0_1_3129044678	43_10_4_0_9_0_0_0_-2430193009
114_0_0_0_0_0_0_0_-450152011	88_0_0_0_0_0_0_0_-3076485586
60_12_8_0_12_0_0_0_2649701674	48_8_4_0_4_0_0_0_4487307557
59_12_6_0_7_0_0_0_2827067143	77_0_0_0_0_0_6_0_1727607124
116_0_0_0_0_0_0_0_-584165517	41_6_4_0_2_0_0_1_306218248
57_0_0_0_0_0_0_0_-1072818100	53_12_2_0_9_0_0_0_1503584704
91_0_0_0_0_0_0_0_1017461451	80_0_0_0_0_0_0_0_-1133289749
73_16_10_0_11_0_0_0_-3663550611	47_8_4_0_5_0_0_0_3299611749
39_8_4_0_4_0_0_1_-3048194894	70_0_0_0_0_0_4_0_1381629030

H.2. Auflistung der Größen der Äquivalenzfamilien

Auflistung der Größen und deren Häufigkeiten der Äquivalenzfamilien, die auf unterschiedlichen Darstellungen der Summe der Nachbarschaftsdeskriptoren beruhen.

Tabelle S-14: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors erfolgte eine long-Darstellung.

Größe	1	2	3	4	5	6-10
Häufigkeit	80433434	352055	10061	9376	1720	4836

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	1536	475	165	111	32	53

Tabelle S-15: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors wurde die Primzahl 1073741741 verwendet.

Größe	1	2	3	4	5	6-10
Häufigkeit	81116990	122906	7748	3209	1461	2388

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	731	330	113	73	53	18

Tabelle S-16: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors wurde die Primzahl 1073741987 verwendet.

Größe	1	2	3	4	5	6-10
Häufigkeit	81136012	111943	7572	3280	1489	2384

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	833	324	129	77	52	22

Tabelle S-17: Übersicht zu den Größen der Äquivalenzfamilien und deren Häufigkeit. Für die Berechnung des Hash-Wertes des Nachbarschaftsdeskriptors wurde die Primzahl 2147483647 verwendet.

Größe	1	2	3	4	5	6-10
Häufigkeit	81090343	133819	7342	3463	1602	2606

Größe	11-20	21-40	41-60	61-100	101-150	>150
Häufigkeit	850	304	111	66	45	35

Eidesstattliche Erklärung / *Declaration under Oath*

Ich erkläre an Eides statt, dass ich die Arbeit selbstständig und ohne fremde Hilfe verfasst, keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

I declare under penalty of perjury that this thesis is my own work entirely and has been written without any help from other people. I used only the sources mentioned and included all the citations correctly both in word or content.

Halle, den 13.09.2022

Datum/Date	Unterschrift des Antragstellers/Signature of the applicant

Persönliche Daten

Vor- und Nachname	Jördis-Ann Schüler
Geburtsdatum und	04.02.1991
Geburtsort	Dessau
Staatsangehörigkeit	Deutsch

Bildungsweg

10/2012 - 01/2015	**M. Sc. Bioinformatik**, *Martin-Luther-Universität Halle-Wittenberg* Masterarbeit: *„Moleküldesign von Stilbenderivaten als Inhibitoren für Acetylcholinesterase und Butyrylcholinesterase und systematische Untersuchungen zur Berechnung von pK_i-Werten mittels Methoden der Computerchemie“* Betreuer: PD Dr. Wolfgang Brandt, Prof. Dr. René Csuk
10/2009 - 10/2012	**B. Sc. Bioinformatik**, *Martin-Luther-Universität Halle-Wittenberg* Bachelorarbeit: *„Gewinnung und Gehaltsbestimmung von Schisandrol A aus verschiedenen Sorten von Schisandra chin“* Betreuer: Prof. Dr. René Csuk, Dr. Renate Schäfer
08/2003 - 07/2009	**Abitur**, *Gymnasium „Walter-Gropius“ Europaschule Dessau-Roßlau*

Berufliche Stationen

04/2015 - 03/2017	**Graduiertenstipendium des Landes Sachsen-Anhalts** *Martin-Luther-Universität Halle-Wittenberg*
seit 04/2017	**Wissenschaftliche Mitarbeiterin** *Institut für Informatik der Martin-Luther-Universität Halle-Wittenberg* Tätigkeit in Forschung, Lehre, Qualitätssteigerung im Studiengang Bioinformatik, Bewerbung des Studiengangs Bioinformatik **Unterbrechungen**: 04/2019 - 03/2020: Mutterschutz und Elternzeit

Veröffentlichungen

08/2018	Schüler, J.-A., Neumann, S., Müller-Hannemann, M., Brandt, W.: Chemfrag: Chemically meaningful annotation of fragment ion mass spectra. Journal of Mass Spectrometry **53** (2018), 1104–1115
12/2020	Schüler, J.-A., Rechner, S., Müller-Hannemann, M.: MET: A Faster Java Package for Molecule Equivalence Testing. Journal of Cheminformatics **12** (2020), 73-85

Halle, den 13.09.2022

Datum/Date

Unterschrift des Antragstellers/Signature of the applicant