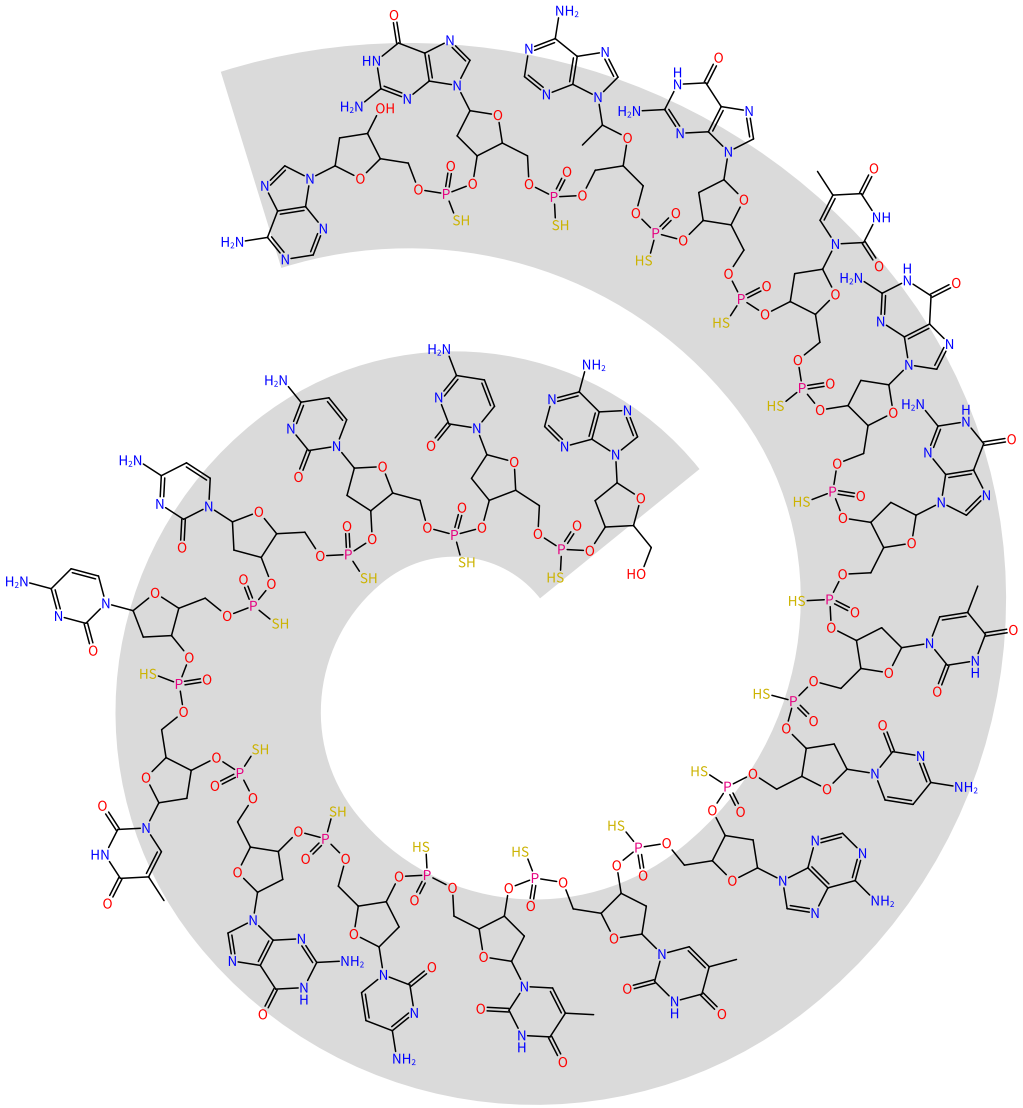
Grafisches Ausrichten von Strukturdiagrammen und interaktive Navigation großer Moleküldatensätze



Dissertation
zur Erlangung des Grads
Dr. rer. nat.
an der Fakultät
für Mathematik, Informatik and Naturwissenschaften der Universität Hamburg
eingereicht beim Fach-Promotionsausschuss Informatik von
Matthias Hilbig
aus Münster
August 2019

Gutachter:
Prof. Dr. Matthias Rarey
Prof. Dr. Johannes Kirchmair

Tag der Disputation:
20.1.2020

Zusammenfassung

In vielen Anwendungsfeldern der Lebenswissenschaften ist das Arbeiten mit einer großen Anzahl kleiner organischen Moleküle zur Notwendigkeit geworden: Umfangreiche Datenbanken mit organischen Molekülen müssen erstellt, verwaltet und miteinander verglichen werden.

In Rahmen dieser Dissertation wurde das Programm Mona entwickelt, das einen einfachen intuitiven und interaktiven Umgang mit Molekülmengen erlaubt. Mona benötigt keine aufwendige Installation oder Vorbereitung, sondern ermöglicht es, sofort Molekülmengen mit bis zu einer Million Molekülen explorativ zu erkunden. Moleküle werden von Mona in Mengen organisiert, die betrachtet, gefiltert und verglichen werden können. Anhand von Fallbeispielen wird gezeigt, wie man das Programm möglichst effizient einsetzt.

Eine grundlegende Funktionalität von Mona ist dabei die Anzeige von Strukturdiagrammen kleiner Moleküle. Hierfür wurde der neue Algorithmus Naomi_{2D} entwickelt, der leicht auch für andere Anwendungsfälle parametrisierbar ist und möglichst gute Strukturdiagramme liefert. Der Algorithmus wurde auf einem großen Datensatz mit 147 Mio. Moleküle getestet und nach objektiven Kriterien verifiziert. Im direkten Vergleich mit anderen Layoutalgorithmen für 2D-Strukturdiagramme generiert Naomi_{2D} weniger Diagramme mit Kollisionen und verzerrten Winkeln. Die Flexibilität des Algorithmus wird anhand einer Erweiterung demonstriert, die das Ausrichten von Strukturdiagrammen ermöglicht. Hierbei wählt der Benutzer ein Molekül als Vorlage und Mona richtet die Strukturdiagramme aller anderen Moleküle danach aus. Damit erhält man einen schnellen Überblick über die Ähnlichkeiten und Unterschiede in einer größeren Menge von Molekülen. Das Verfahren wird hinsichtlich der Güte des verwendeten Matching-Verfahrens und des Layouts bewertet.

Abstract

Working with a large number of small organic molecules has become a necessity in the life sciences: Extensive databases with organic molecules have to be created, managed and compared.

In the context of this dissertation, the program Mona was developed, which enables a simple, intuitive and interactive handling of molecule sets. Mona does not require any complex installation or preparation, but allows to explore molecule sets of up to one million molecules immediately. Molecules are organized by Mona into sets that can be viewed, filtered and compared. Case studies show how to use the program as efficiently as possible.

A basic functionality of Mona is the display of structure diagrams of small molecules. For this purpose the new algorithm Naomi_{2D} was developed, which can easily be parameterized for other applications and provides high quality structure diagrams. The algorithm was tested on a large data set with 147 million molecules and verified with objective criteria. In direct comparison with other 2D structure diagram layout algorithms, Naomi_{2D} generates fewer diagrams with collisions and distorted angles. The flexibility of the algorithm is demonstrated by an extension that allows the alignment of structure diagrams. Here the user selects one molecule as a template and Mona aligns the structure diagrams of all other molecules accordingly. This gives a quick overview of the similarities and differences in a large set of molecules. The procedure is evaluated with regard to the quality of the layouting and matching procedure used.

Danksagung

Ich möchte mich bei Prof. Dr. Matthias Rarey für das überaus spannende Thema und die tolle Arbeitsumgebung bedanken. Erstaunlich viele Dinge, die ich in den Jahren am ZBH gelernt habe, verwende ich noch immer.

Im Besonderen möchte ich mich bei allen ehemaligen Kollegen am ZBH bedanken. Ihr habt maßgeblich dafür gesorgt, dass die Arbeit und die Freizeit immer Spaß gemacht haben. Seien es die Diskussionen in der Kaffeeküche oder bei gemeinsamen Doppelkopfabende: Die besten Ideen kamen immer dort zustande. Ihr seid auch dafür verantwortlich, dass ich nach vier Jahren immer noch wehmütig an die schöne Zeit in Hamburg denke.

Viele Leute haben mich bei der Arbeit unterstützt, ein ganz großes Dankeschön an euch: Therese Inhester hat geholfen die PropertyDB umzuschreiben und war die beste Bürokollegin, die man sich vorstellen kann[1]. Der Naomi Trupp Sascha Urbaczek, Robert Fischer, Tobias Lippert und Adrian Kolodzik sorgte für frischen Wind auf der Arbeit[2] und in der Freizeit. Lennart Heinzerling für die jahrelange Büropartnerschaft und Unternehmungen mit interessanten Gesprächen und Käse. Angela Henzler für die vielen Diskussionen sowohl auf der Arbeit als auch mit Gin & Tonic. Marcus Gastreich schaute immer wieder im Detail auf die generierten Strukturdiagramme. Benjamin Schulz programmierte die erste Version des Ringsystemlayouters.

Meinen Eltern danke ich vor allem für ihre Geduld und meinem Vater besonders für die viele Arbeit, die er in das Korrekturlesen gesteckt hat. Ich danke auch Therese Inhester und Silke Rakow für das Finden von Fehlern in der Arbeit. Der endgültige Punktestand ist damit wie folgt:

Name		Punkte		
Erhard Hilbig	1337	Kommafehler und	101	Bindestriche
Therese Inhester	54	inhaltliche Kommentare und	130	S-Bahn Fahrten
Silke Rakow	42	Wortwiederholungen und	163	cm Größe

Ganz besonders möchte ich mich bei Silke bedanken, vor allem für dein Verständnis und deine Unterstützung, wenn ich meine Freizeit anstatt auf Mittelaltermärkten mit Schreiben verbracht habe. Du bist der Grund für die schönste Zeit meines Lebens.

[1] Das gilt natürlich auch für Lennart Heinzerling.
[2] u.a. für den Umstieg von C auf besseren C++ Code.

Inhaltsverzeichnis

1. Einleitung

Empirische Naturwissenschaft besteht zu einem großen Teil aus dem Katalogisieren, Verwalten und Vergleichen von Daten. Fortschritte ergeben sich immer dann, wenn aus diesen Daten Zusammenhänge erkennbar sind und rationale Schlussfolgerungen Dinge erklären, die vorher unbekannt waren. Auch die Chemie ist hier keine Ausnahme: Aus der gewaltigen Anzahl von vermutlich mehr als 10^{60} möglichen für Arzneimittel geeigneten organischen Molekülen [9], die Moleküle zu finden, die die gewünschten Eigenschaften haben, ist kein leichtes Unterfangen. Es stellt sich die Frage:

Wie kann man die Arbeit mit vielen kleinen Molekülen am Computer für Menschen einfacher und damit fehlerfreier und effizienter gestalten?

Bei der Arbeit mit organischen Molekülen kommt man schnell in die Verlegenheit, sich sowohl nur eine Handvoll als auch ganze Kataloge von Molekülen anzusehen, zu sortieren und einzelne zu verwerfen. Typische Tätigkeiten sind z. B.:

- Ein medizinischer Chemiker, der ein neues Medikament entwickeln möchte, benötigt einen Katalog mit möglichen Kandidaten, in dem keine doppelten Moleküle vorhanden sind.
- Ein Chemiker soll bei einer Handvoll sehr ähnlicher Moleküle entscheiden, welches Molekül synthetisiert wird.
- Ein Chemieinformatiker erstellt eine Testdatenbank mit Molekülen aus unterschiedlichen Quellen, die als Eingabe für einen neu entwickelten Algorithmus dienen soll.

Das Hauptproblem bei all diesen Anwendungen ist immer das gleiche: Um Entscheidungen schnell und richtig treffen zu können, muss sowohl das Vergleichen von einzelnen Molekülen als auch das Vergleichen von großen Mengen von Molekülen effizient möglich sein.

Es ist natürlich möglich, diese Aufgabe dem Computer zu überlassen. Nicht umsonst gibt es zahllose Maße, um die Ähnlichkeit von Molekülen zu berechnen. Aber in letzter Instanz ist immer das Expertenwissen einzelner Personen gefragt, die anhand der vorliegenden Indikatoren (Scoring-Werte, Ähnlichkeiten) die richtigen Entscheidungen treffen sollen.

In dieser Arbeit werden vor allem zwei Methoden vorgestellt, die bei dieser Fragestellungen helfen:

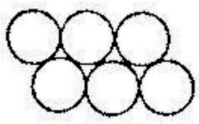
1. Wie lassen sich chemische Strukturdiagramme unterschiedlicher Moleküle möglichst ähnlich zeichnen, um damit ihre Gemeinsamkeiten visuell hervorzuheben?
2. Wie verwaltet und analysiert man Tausende oder Millionen von Molekülen?

1.1. Wie werden Moleküle dargestellt?

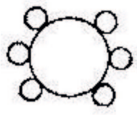
Chemische Strukturdiagramme sind die Sprache der Chemiker, um sich über organische Moleküle zu unterhalten: Anstatt die reale 3D-Konformation der Moleküle auf einem Blatt Papier abzubilden, wird das Molekül als schematisches Diagramm bestehend aus Atomen und Bindungen gezeichnet. Dies ermöglicht es, bekannte Strukturen in Molekülen einfach zu erkennen und wiederzufinden. Vom Prinzip her ähnelt dies den heute überall üblichen U-Bahn- und Nahverkehrsplänen. Die ikonische Karte der Londoner U-Bahn wurde 1931 von Henry Beck erstellt [59] und verbreitete sich im letzten Jahrhundert schnell auch in andere Städte. Dies geschah vor allem, weil sie die typische Benutzung berücksichtigte: Ein Reisender möchte nicht wissen, ob sein Ziel genau 13,7 km oder 14,4 km entfernt ist. Es interessiert ihn lediglich, welche Linien er nehmen muss, um sein Ziel zu erreichen.

1861 - Joseph Loschmidt

Constitutions-Formeln der organischen Chemie in graphischer Darstellung



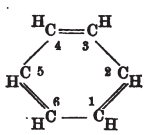
Schema 182



Schema 185

1872 - August Kekulé

Ueber einige Condensationsproducte des Aldehyds



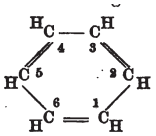


Abb. 1.1. Die ersten Strukturdiagramme von Molekülen finden sich in den Arbeiten von Joseph Loschmidt [43] und August Kekulé [3].

Die erste Benutzung von Strukturdiagrammen in der organischen Chemie lässt sich auf die Zeit um 1860 festlegen (s. Abb. 1.1). Damals verbreitete sich die Erkenntnis, dass Moleküle aus mehr Leere als aus Materie bestehen. Joseph Loschmidt formulierte dies 1861 in seiner Arbeit „Chemische Studien – Constitutions Formeln der organischen Chemie in graphischer Darstellung“ [43] folgendermaßen:

> „Die Chemie hat nach Liebig’s Vorgange die Annahme acceptirt, dass das Volumen des Materiellen, selbst in einem festen oder flüssigen Körper, verschwindent klein sei gegen die leeren Zwischenräume, welche die kleinsten Theile der Materie von einander trennen, und dass daher dieselben – die Atome – nur per distans durch Anziehungs- und Abstossungkräfte auf einander wirken. Man kann diese Constitution sinnreich mit der unseres Sonnensystems verglichen, in welchem die interplanetaren Räume in einem ähnlichen Verhältnisse zu dem Volumen der Sonne und der Planeten stehen.“

Folglich bestehen die grafischen Darstellungen der Moleküle in der Arbeit vor allem aus Ringen, die die Abstoßungen der Atome untereinander andeuten. Wie genau der

Abb. 1.2. Das Bild „L'Ortolano" (1587–1590) des Renaissancemalers Guiseppe Arcimboldo in normaler Orientierung und auf dem Kopf. Erst durch genaues Hinschauen erkennt der Betrachter, dass es sich um dasselbe Bild handelt.

Kern des Benzols aussah, war Loschmidt noch unbekannt. Er vermutete eine Stapelung der Kohlenstoffatome, kam dann aber zu dem Schluss, den Kern als einen ganzen Ring darzustellen und an diesen die Wasserstoffe anzuhängen. Vier Jahre später (1865) veröffentlichte August Kekulé das erste Mal seine im Traum erschienene Erklärung der Struktur des Benzols [4]. In einer weiteren Arbeit von 1872 finden sich dann die ersten Strukturdiagramme für Benzol nahezu in der heute noch gebräuchlichen Form [3].

Ausgehend von diesen Anfängen verbreiteten sich Strukturdiagramme rasch als universelle Methode zur Visualisierung von organischen Molekülen.

In den 1970er Jahren begann man, Moleküle mit dem Computer zu verarbeiten. Aufgrund der eingeschränkten grafischen und rechnerischen Fähigkeiten dieser Rechner beschränkte sich die Repräsentation auf ASCII Zeichen [12]. Mit dem Aufkommen von grafischen Benutzerschnittstellen in den 1980er Jahren, näherten sich die vom Computer gezeichneten Strukturdiagramm immer mehr den von Illustratoren mit Schablonen gezeichneten Diagrammen an [23].

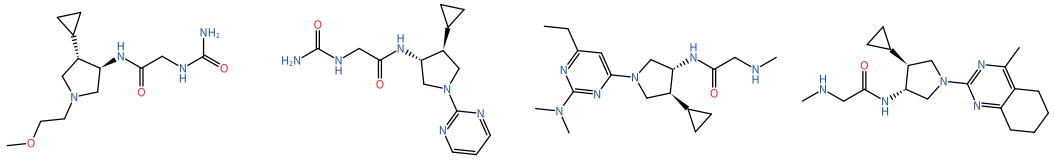
Heutzutage sind am Rechner dargestellte Strukturdiagramme nicht mehr von gedruckten Exemplaren in Publikationen zu unterscheiden.

Dieser Ausflug diente dazu, die Bedeutung von Strukturdiagrammen in der organischen Chemie hervorzuheben. Mehrere Generationen von Chemikern sind nun bereits mit dieser Darstellungsform in Berührung gekommen und nach 150 Jahren erfüllt sie immer noch ihren Zweck, sich einfach und unkompliziert über kleine organische Moleküle zu verständigen.

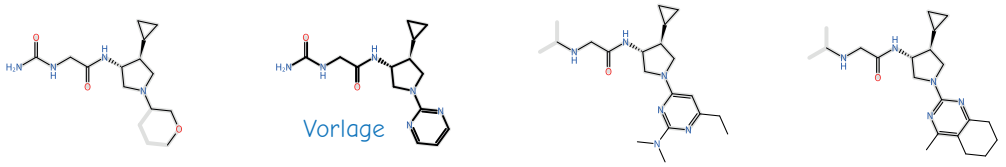
1.2. Wie vergleicht man Moleküle?

Da, wie im vorherigen Kapitel erwähnt, das Strukturdiagramm die Hauptvisualisierung der Chemiker von Molekülen ist, werden diese auch verwendet, um Unterschiede und Gemeinsamkeiten von Molekülen herauszuarbeiten. Allerdings fällt es Menschen schwerer, Gemeinsamkeiten und Unterschiede in Bildern zu erkennen, wenn diese nicht in der gleichen Orientierung vorliegen. Andererseits sind Menschen sehr gut darin, Muster in Dingen wiederzuerkennen, wenn sie ähnlich aussehen. Ein schönes Beispiel dafür sind die Bilder des Renaissancemalers Guiseppe Arcimboldo (1527–1593), in denen man je nach Orientierung vorwiegend Pflanzen oder Gesichter erkennt. Wenn man dasselbe Bild in unterschiedlicher Ausrichtung wie in Abb. 1.2 nebeneinanderlegt, ist es daher schwer zu erkennen, dass es sich um dasselbe Bild handelt.

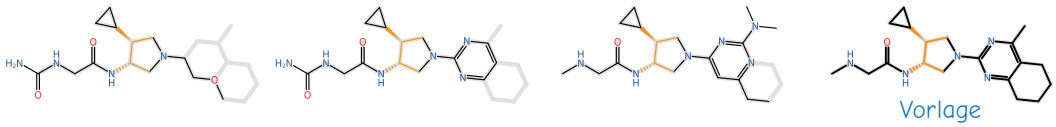
Das Gleiche gilt für Strukturdiagramme von Molekülen. Der Chemiker hat gelernt, Muster in den Diagrammen zu erkennen. Es wird deutlich übersichtlicher, wenn man zwei ähnliche Moleküle auch auf ähnliche Art und Weise zeichnet. Welche Gemeinsamkeiten haben diese vier Moleküle?



Wenn man alle Moleküle ähnlich zum zweiten zeichnet, ist der gemeinsame Teil leicht zu erkennen.



Ebenso wird deutlich sichtbar, dass alle Stereozentren der Moleküle gleich orientiert sind.



In beiden Fällen wurden die Moleküle mit dem im ersten Teil dieser Arbeit vorgestellten Algorithmus ausgerichtet.

Wenn man sich von einzelnen Molekülen löst und Sammlungen von Molekülen ansieht, kommt man zu den folgenden Fragestellungen:

- Was sind die Unterschiede dieser beiden Molekülmengen?
- Welche Moleküle haben sie gemeinsam?
- Gibt es irgendwelche Muster in der Verteilung einer bestimmten physikochemischen Eigenschaft?

Mit diesen Fragestellungen beschäftigt sich das Tool Mona, dass im zweiten Teil der Arbeit vorgestellt wird. Mona ist ein Desktopprogramm, dessen Ziel es ist, dem Benutzer explorative Arbeitsabläufe bei der Verwaltung von Molekülsammlungen zu ermöglichen. Der grundlegende Ansatz von Mona ist es, Moleküle in Mengen ohne Duplikate zu verwalten. Unterschiedliche Operationen auf diesen Mengen ermöglichen einen einfachen Einsatz des Programms, da alle Operationen leicht verständlich und performant im Hintergrund ausgeführt werden können.

2. Grundlagen

Dieses Kapitel dient dazu, die Arbeit in den aktuellen Kontext einzuordnen und die Herausforderungen und Vorarbeiten in den einzelnen Teilbereichen zu erläutern.

2.1. Zeichnen von Strukturdiagrammen

Praktisch alle chemischen Strukturdiagramme für Publikationen werden heutzutage über chemische Zeichenprogramme oder automatische Layoutalgorithmen mit Computern erzeugt.

Der Benutzer kann mit spezialisierten Zeichenprogrammen nahezu beliebige chemische Zeichnungen anfertigen. Programme wie ChemDraw [61] vereinen dafür Vektorzeichenprogramme mit geeigneten chemischen Vorlagen, die das Erstellen von Zeichnungen im chemischen Kontext erleichtern. Da Strukturdiagramme die wichtigste Diagrammart für Chemiker sind, wird vor allem das Erstellen dieses Diagrammtyps möglichst stark unterstützt. So ist es z. B. schnell möglich, Zeichnungen von Ketten oder kondensierte Ringsysteme aus einzelnen Ringen (Cycloalkane) zu erstellen. Durch ihren allgemeinen Ansatz sind chemische Zeichenprogramme aber nicht nur auf Strukturdiagrammen beschränkt, sondern es können typischerweise auch beliebige andere Zeichnungen erstellt werden. Im chemischen Kontext sind das vor allem Reaktionen, und im biochemischen Kontext sind es Stoffwechselwege. Trotz dieser Hilfestellungen ist das manuelle Erstellen von Strukturdiagrammen sehr aufwendig, um große Mengen Moleküle zu visualisieren. Diese Arbeit überlässt man daher besser dem Computer. Im Grunde handelt es sich beim automatisierten Zeichnen von Strukturdiagrammen um ein Graphzeichnen-Problem: Der Graph von Molekülen soll möglichst übersichtlich dargestellt werden. Die Herausforderungen für ein automatisiertes Computerprogramm (und zum Teil auch für Menschen) sind dabei die folgenden:

- Linienüberkreuzungen sollten minimiert werden,
- Symmetrien des Graphen hervorgehoben und
- der Wiedererkennungswert für Chemiker maximiert werden.

Dafür gibt es eine Reihe von festen Randbedingungen und über die Jahre entwickelten Konventionen, die beachtet werden müssen, um für Chemiker verständliche Strukturdiagramme zu berechnen. Diese Konventionen sind mittlerweile in zwei IUPAC (International Union of Pure and Applied Chemistry) Veröffentlichungen zusammengefasst. Die erste [42] handelt dabei von allgemeinen Konventionen bei der Erstellung

von Strukturdiagrammen und umfasst Regeln beginnend bei der zu wählenden Schriftart bis zu Methoden der Kollisionsvermeidung. Regeln sind in Form von zahlreichen Positiv- und Negativbeispielen dargestellt. An mehreren Stellen dieser Dissertation wird direkt auf bestimmte Regeln verwiesen, diese Verweise beginnen mit einem vorgestelltem GR, z. B. GR-0.1. Die zweite Publikation [41] umfasst Regeln zur Darstellung von Stereokonfigurationen von Molekülen. Direkte Verweise auf diese Regeln beginnen mit einem vorgestelltem ST.

Um das Layout von allgemeinen Graphen zu berechnen, gibt es eine Vielzahl von verfügbaren Algorithmen. Eine Übersicht bietet [6]. Die meisten dieser Algorithmen lassen sich nicht direkt zum automatischen Zeichnen von Strukturdiagrammen verwenden, da sie die in Strukturdiagrammen gebräuchlichen Regeln nicht einhalten. Eine Ausnahme bilden Kraftfeld-basierte Ansätze, die zum Zeichnen von komplizierten Ringsystemen verwendet werden können.

Stukturdiagramme werden typischerweise mit kombinatorischen Algorithmen berechnet. Die grundlegende Funktionsweise verschiedener Layoutalgorithmen ist in [32] beschrieben. Eine aktuellere Beschreibung über die Vorgehensweise des im Programm MOE [14] benutzten Layouters findet sich in [15]. Dieser Algorithmus partitioniert den Molekülgraphen in unterschiedliche Bereiche, berechnet für jeden Bereich ein geeignetes Layout und setzt die einzelnen Teile am Ende wieder zusammen. Eine andere Herangehensweise wählt [25]. Hier wird das komplette Diagramm mithilfe eines *simulated annealing* Ansatzes und eines Kraftfeldes berechnet. Dies führt vor allem bei großen Ringsystemen zu besseren Ergebnissen als bei Vergleichsprogrammen. Auch die globale Behandlung von Kollisionen führt zu symmetrischeren Layouts für Moleküle, die eigentlich nicht ohne Kollisionen gezeichnet werden können. Eine alleinige Verwendung dieses Algorithmus ohne eine Ringdatenbank führt jedoch dazu, dass erwartete Konventionen gerade von kleinen überbrückten Ringsystemen nicht eingehalten werden.

Das Zeichnen von Strukturdiagrammen für ähnliche Moleküle wurde bisher nicht ausführlich behandelt. In [10] benutzt der vorgestellte Layoutalgorithmus einen *supertree*, um das Layout von ähnlichen Diagrammen für mehrere Moleküle zu zeichnen. Der *supertree* ist ein Baum, der die Gemeinsamkeiten von zwei oder mehr Molekülen speichert. Ausgehend von den festgelegten Wurzelknoten zweier Moleküle wird der gemeinsame *supertree* anhand des jeweiligen Breitensuchbaumes der Moleküle berechnet. Dabei wird ein bipartites Matching benutzt, um möglichst viele gemeinsame Kanten übereinander zu legen. Beim Zeichnen der Moleküle werden die Positionsentscheidungen der Atome mithilfe des *supertree* getroffen. Eine Einschränkung der Methode ist, dass mindestens ein gemeinsames Atom des gemeinsamen Bereiches der Moleküle bekannt sein muss, damit man die Wurzelknoten definieren kann. Der in [27] beschriebene Algorithmus benutzt lokale Annotationen, um bestimmte Richtungen für Bindungen beim Layout zu bevorzugen. Das globale Layout muss sich aber nicht an alle Richtungsvorgaben halten. Für eine automatische Vorgabe der lokalen Richtungen wird exemplarisch der *feature tree* Algorithmus [66] benutzt.

2.2. Moleküldatenbanken

Das Feld der Datenbanken ist ein umfangreiches Feld in der Informatik. Daher sei hier vor allem auf die einschlägige Literatur verwiesen [18]. Datenbanken gehören seit jeher zum Grundstock der Informationsverarbeitung. Heutzutage sind vor allem relationale (SQL)-Datenbanken verbreitet und ausgereift. In den letzten Jahren haben sich auch andere sogenannte NoSQL-Datenbanken für Spezialzwecke verbreitet. Hier sind vor allem die Key-Value-Datenbanken vertreten, die riesige Datenmengen verteilt auf sehr viele Rechnern speichern können [13]. Es gibt aber auch noch eine Reihe weiterer Datenbanken für Spezialzwecke z. B. Graphen-Datenbanken, die effizient mit großen Graphen umgehen können.

Computer sind vor allem dann prädestiniert für eine Aufgabe, wenn es um die Organisation von großen Datenbeständen geht. Es ist daher naheliegend, auch zur Katalogisierung von organischen Molekülen Computer zu verwenden. Die grundlegendste Anwendung ist dabei die Speicherung von Molekülen in Dateien. Hierfür eignen sich die unterschiedlichen in der Chemieinformatik verbreiteten Dateiformate für Moleküldateien. Zum einen sind das Formate, die auf Verbindungstabellen des Molekülgraphen basieren und zum anderen kompakte zeilenbasierte Serialisierungen des Molekülgraphen. Diese sind vor allem für die Speicherung in Datenbanken geeignet. Der wichtigste Vertreter dieser Art ist SMILES [84].

Um Suchanfragen für Moleküle in Datenbanken zu formulieren, gibt es mehrere Methoden. Die Datenbank kann direkt benutzt werden, um Moleküle mit einer bestimmten ID, einem bestimmten Namen oder annotierten Eigenschaften effizient zu suchen. Da es sich bei Molekülen um Graphen handelt, sind spezielle Verfahren nötig um nach identischen bzw. ähnlichen Molekülen oder Substrukturen des Moleküls zu suchen.

Die Suche nach einem möglichst ähnlichen Molekül oder Teilen des Moleküls findet zum einen über Ähnlichkeitsmaße (s. Kap. 2.2.1) und zum anderen über eine Substruktursuche statt. Bei den Ähnlichkeitsmaßen dient ein Anfragemolekül als Muster, um alle Moleküle in der Datenbank zu finden, die eine möglichst große Ähnlichkeit zu diesem Molekül haben. Die Definition des Ähnlichkeitsmaßes bestimmt dabei, welche Moleküle besonders ähnlich zum Anfragemolekül sind. Bei der Substruktursuche dient dagegen eine Subgraph als Muster. Dieser wird typischerweise entweder über einen Moleküleditor erstellt oder über eine spezielle Sprache wie SMARTS [19] definiert. Beispielsweise sucht man mit dem SMARTS Muster `c1ccccc1` nach allen Phenylringen in einem Molekül.

2.2.1. Molekülidentität und Ähnlichkeit

Wann sind zwei Moleküle genau gleich? Eine Ja/Nein-Antwort hierauf führt direkt zum Graphisomorphie-Problem [2]. Zwei allgemeine Graphen sind dann isomorph, wenn es eine bijektive Abbildung zwischen den Knotenmengen der beiden Graphen gibt, die die Nachbarschaft der Knoten in beide Richtungen erhält. Oder umformuliert: Erstelle für jeden Graphen G eine kanonisierte Benennung und Reihenfolge der Knoten und Kanten, sodass jeder Graph F der isomorph zu G ist, dieselbe Benennung und Reihenfolge

besitzt. Diese Umformulierung ist äquivalent, da zwei Graphen genau dann isomorph sind, wenn ihre kanonischen Beschreibungen gleich sind. Die kanonische Beschreibung kann in polynomieller Zeit überprüft werden. Damit ist die Kanonisierung von Graphen von vergleichbarer Komplexität wie die Entscheidung, ob zwei Graphen isomorph zueinander sind.

Vom Graphisomorphie-Problem ist nicht bekannt, ob es sich in der Komplexitätsklasse P befindet. Es ist aber sehr wahrscheinlich auch nicht NP-vollständig [2]. Für Graphen mit beschränktem Knotengrad (und damit auch für Molekülgraphen) existiert jedoch ein Algorithmus [51] mit polynomialer Laufzeit. Zur Kanonisierung von Molekülen wird in dieser Arbeit der CANON-Algorithmus [85] benutzt, der ursprünglich auf dem Morgan-Algorithmus [55] basiert.

In delokalisierten Systemen von Molekülen ist die Position von Doppelbindungen und Ladungen nicht eindeutig bestimmt. Wenn man sich von exakter Ähnlichkeit hin zu Ähnlichkeitsmaßen bewegt, erreicht man daher zuerst die Frage: Was ist mit unterschiedlichen Tautomer- und Protonierungszuständen von Molekülen? Sollten diese nicht auch als dieselben Moleküle erkannt werden? Abhängig vom Kontext ist die Antwort auf diese Frage entweder ja oder nein. In [79] werden die in dieser Arbeit verwendeten Methoden erläutert, um kanonische Tautomer- und Protonierungszustände zu berechnen. Diese werden benutzt, um je nach Anwendung verschiedene Tautomere bzw. Protonierungszustände eines Moleküls als identisch zu erkennen.

Wenn es um eine ungefähre Ähnlichkeit von Molekülen geht, kommen Ähnlichkeitsmaße ins Spiel [53] . Hier gibt es viele verschiedene Varianten, abhängig davon, welche Eigenschaften bei der Ähnlichkeit die größte Rolle spielen. Sehr verbreitet haben sich in den letzten Jahren *fingerprint* Ansätze wie der ECFP [67]. Der ECFP-Algorithmus beschreibt von jedem Atom ausgehend die Umgebung des Atoms mithilfe einer Zahl. Abhängig von der Anzahl der Iterationen (meistens 4-6) wird ein immer weiterer Radius um jedes Atom herum erfasst und mithilfe einer Hashfunktion in einer Zahl zusammengefasst. Je mehr der durch die Hashfunktion entstandenen Zahlen zweier Moleküle am Ende gleich sind, desto ähnlicher sind sich die beiden Moleküle.

2.3. Pipeline und Visualisierungstools

Es gibt unterschiedliche Ansätze und Werkzeuge zur Verwaltung von Molekülen. Dabei haben sich zwei Fokusbereiche herausgezeichnet:

Werkzeuge, wie KNIME [8] und Pipeline Pilot [1], setzen auf Flexibilität und einen möglichst großen Funktionsumfang. Beide Tools sind zur Analyse von beliebigen Daten gedacht, enthalten aber auch umfangreiche Methoden aus der Chemieinformatik. Um Daten zu analysieren, wird ein Analyseworkflow programmiert, bei dem die Ein- und Ausgänge der entsprechenden Funktionsblöcke visuell miteinander verknüpft werden (s. Abb. 2.1). Wie man diese Funktionsblöcke und in welcher Reihenfolge man sie verwendet, ist Aufgabe des Benutzers. Die Funktionalität kann durch neue Funktionsblöcken sehr schnell erweitert werden. Der Nachteil dieser Tools besteht darin, dass ein explorativer Ansatz, bei dem der Benutzer viele verschiedene Sachen ausprobiert, zeit-

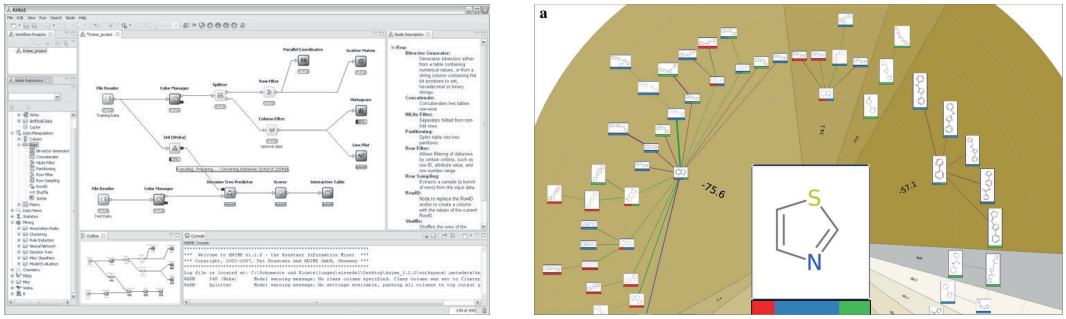


Abb. 2.1. Auf der linken Seite ist ein beispielhafter Analyseworkflow des Programms KNIME zu sehen (Bildquelle: [8]). Auf der rechten Seite sieht man einen Teil des *scaffold trees* von Scaffold Hunter (Bildquelle: [70])

aufwendig ist, da die einzelnen Funktionsblöcke immer wieder umkonfiguriert werden müssen.

Der zweite Fokusbereich ist das visuelle Analysieren von Moleküldatenbanken. Hier sind vor allem die Ansätze von Scaffold Hunter [70], Data Warrior [69] und StarDrop [58] zu nennen. Scaffold Hunter enthält mehrere Visualisierungsansätze, um Moleküle darzustellen: Der *scaffold tree* erlaubt es, Moleküle anhand ihrer Kernstrukturen zu gruppieren (s. Abb. 2.1). Von einigen wenigen Teilen der Moleküle fächert sich so der gesamte Baum der Moleküle auf. Weiterhin ist auch eine Dendrogramm-Visualisierung und die *molecular cloud* -Visualisierung von Peter Ertl [22] im Tool vorhanden. Die Visualisierungen basieren dabei auf einer vorberechneten Clusterung der Moleküle. Das Erkunden der entsprechenden Visualisierung geschieht interaktiv. In StarDrop ist es möglich, Moleküle visuell mithilfe von Karteikarten zu organisieren. Pro Molekül existiert eine Karteikarte, auf der das Strukturdiagramm des Moleküls und vom Benutzer wählbare Zusatzinformationen abgebildet werden. Data Warrior ist schließlich ein hauptsächlich an die Bedürfnisse von Chemieinformatikern angepasstes Programm. Das Programm stellt eine große Menge Verfahren bereit, die in der Chemieinformatik entwickelt wurden. Zu jedem Verfahren enthält es passende Visualisierungen. So ist es möglich, eine Ähnlichkeitsanalyse von Molekülen oder die Ergebnisse einer Substruktursuche visuell zu betrachten.

2.4. Das Naomi Molekülmodell

Alle in dieser Dissertation beschriebenen Methoden greifen auf die Verfahren der Naomi Molekülbibliothek zurück. Naomi selbst ist in [80] ausführlich beschrieben. Hier folgt nur ein Überblick über alle für das Verständnis dieser Arbeit nötigen Konzepte und Methoden in Naomi.

2.4.1. Molekültopologie und Konformation

Für den Chemiker ist ein Molekül dann klar beschrieben, wenn er es eindeutig anhand seiner chemischen Eigenschaften identifizieren kann. Ihm ist es wichtig zu erkennen, aus welchen Atomen ein Molekül besteht und wie diese zueinander angeordnet sind: Wo gibt es kovalente Bindungen? Handelt es sich um die Cis- oder die Trans-Variante einer Doppelbindung? Da Moleküle sich bewegen, sind für ein Molekül beliebig viele Positionen im Raum möglich. In der Chemieinformatik ist es daher sinnvoll, die Struktur eines Moleküls losgelöst von seiner absoluten Position zu betrachten. Als Molekültopologie oder vereinfacht Topologie wird in dieser Arbeit die Struktur des Moleküls bezeichnet. Diese besteht aus der Graphenstruktur des Moleküls, die sowohl die Atomtypen als auch die Bindungstypen enthält. Weiterhin zählen auch die strukturellen Informationen der Stereozentren und der Ringsysteme zur Topologie.

Nicht zu der Topologie des Moleküls zählt die Konformation. Die Konformation eines Moleküls ist eine Momentaufnahme der exakten räumlichen Positionen aller Atome. Diese kann entweder experimentell gemessen werden, z. B. über NMR-Spektroskopie, wenn ein Molekül als Ligand in einem Protein-Ligand-Komplex liegt. Alternativ können Konformationen auch näherungsweise erzeugt werden: Die Positionen von Atomen sind teilweise eingeschränkt, z. B. durch die typische Länge von kovalenten Bindungen oder die möglichen Torsionswinkel von drei aufeinanderfolgenden Atomen. Ob durch Experimente oder einen Konformationsgenerator berechnet, besteht die Konformation eines Moleküls am Ende immer aus einem Satz von 3D-Koordinaten. Zum Zeichnen von Strukturdiagrammen benötigt man entsprechend 2D-Koordinaten des Moleküls.

2.4.2. Atome, Bindungen und Ringe

Das Konzept der Naomi-Bibliothek ist vor allem das Konzept einer gemeinsamen, chemisch möglichst korrekten Molekül-Datenstruktur und der dazugehörigen dateibasierten Ein- und Ausgabe der Moleküle. Naomi besitzt klar definierte, schlanke und möglichst vielen Anforderungen gewachsene Molekül- und Protein-Datenstrukturen.

Die Molekülklasse besteht vor allem aus den einzelnen Atomen und Bindungen. Die chemischen Eigenschaften der Atome und Bindungen sind direkt in den entsprechenden Klassen hinterlegt, sodass der Zugriff leicht möglich ist. Aufbauend auf dieser Basis gibt es weitere Klassen, die zusätzliche Informationen enthalten.

Moleküle können mehrere Ringsysteme enthalten. Graphentheoretisch sind Ringsysteme alle zweifach zusammenhängenden Komponenten des ungerichteten Molekülgraphen. Dies bedeutet, dass man aus einem Ringsystem mindestens zwei Knoten entfernen muss, um den Zusammenhang zu lösen. Innerhalb jedes Ringsystemes sind alle relevanten Zyklen als Ringe abgespeichert. Um die relevanten Zyklen chemisch bedeutsam und vor allem auch effizient beschreiben zu können, führt Naomi das Konzept der *unique ring families* (URF) ein [47]. Eine URF enthält alle relevanten Zyklen [82], die nur Variationen voneinander sind. Im Folgenden bezeichnet $E(C)$ die Menge aller Bindungen eines Zyklus C. Es werden immer dann relevante Zyklen C_1, C_2 eines Mole-

küls M zu einer Familie zusammengefasst, wenn diese

1. dieselbe Größe haben: $|C_1| = |C_2|$
2. Kanten gemeinsam haben: $E(C_1) \cap E(C_2) \neq \emptyset$
3. und C_1 sich aus C_2 zusammen mit einer Kombination kleinerer Ringe ergibt: Es existiert eine Menge von Ringen $\{c_1, \ldots, c_n\}$ in M mit $|c_i| < |C_1|$, sodass:

$$C_1 = C_2 \oplus c_1 \ldots \oplus c_n$$

Der Vorteil dieser Darstellung ist, dass sie Fälle in einer Familie zusammenfasst, bei denen es sonst exponentiell viele relevante Zyklen gäbe.

2.4.3. Stereodeskriptoren

Eine weitere wichtige Annotation des Moleküls sind die Stereodeskriptoren an den Atomen. Während der Initialisierung wird mithilfe der CIP Regeln [64, 11] die Stereoinformationen für tetraedrische Kohlenstoffe und Doppelbindungen berechnet. Um die Stereodeskriptoren berechnen zu können, ist entweder eine 3D-Konformation des Moleküls nötig oder zusätzliche Informationen im Eingabedateiformat. Für tetraedrische Kohlenstoffe besteht der Stereodeskriptor aus den Prioritäten der vier benachbarten Atome und einem Typen. Wenn das Stereozentrum definiert ist, enthält der Typ die Drehrichtung (R oder S) um die Achse des Nachbarn mit der höchsten Priorität. Ein undefiniertes Stereozentrum wird durch einen eigenen Typ gekennzeichnet. Für Doppelbindungen sieht der Stereodeskriptor ähnlich aus: Neben den Prioritäten der vier benachbarten Atome enthält es den Typ Z oder E. Z bedeutet die Atome mit der höchsten Priorität liegen zusammen auf derselben Seite der Doppelbindung. Bei Typ E liegen die Atome mit der höchsten Priorität auf entgegengesetzten Seiten.

2.4.4. Symmetrien

Damit die Symmetrien von Molekülgraphen bestimmt werden können, enthält Naomi eine Methode, um alle Automorphismen des Molekülgraphen zu berechnen. Ein Automorphismus ist eine bijektive Abbildung auf sich selber. Für einen Graphen bedeutet das, dass alle Knoten und Kanten des Graphen und seines Bildes eindeutig zugeordnet werden. Die Identitätsabbildung ist dabei immer trivialerweise auch ein Automorphismus. Ein Cyclohexan kann z. B. inklusive der Identität auf 12 verschiedene Arten auf sich selbst abgebildet werden. In diesem Fall entspricht die Gruppe der Automorphismen der Diedergruppe D_6 mit 6 Rotationen und 6 Spiegelungen. Im weiteren Verlauf dieser Dissertation werden auch noch weitere Gebiete der Algebra vor allem Permutationen und Symmetriegruppen eine Rolle spielen. Eine Einführung in die in dieser Arbeit verwendeten Notation und die Theorie dahinter findet sich in [71].

3. Strukturdiagramme berechnen

Chemiker greifen typischerweise auf Strukturdiagramme zurück, wenn sie über organische Moleküle diskutieren. Dafür sollten die Diagramme nach festen Regeln aufgebaut sein [42].

Ein Ziel dieser Arbeit war es, einen Algorithmus zu entwickeln, der möglichst viele dieser Konventionen einhält und der gleichzeitig flexibel genug ist, die Layoutgenerierung nach zusätzlichen beliebigen Randbedingungen beeinflussen zu können.

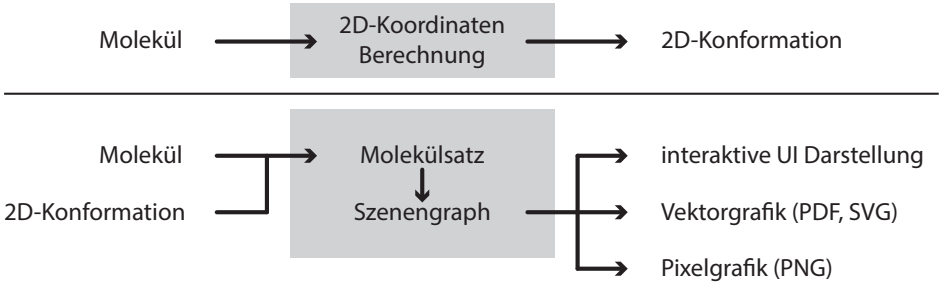


Abb. 3.1. Die Naomi$_{2D}$-Bibliothek besteht aus zwei voneinander getrennten Teilen: Mit der 2D-Koordinatenberechnung lässt sich eine 2D-Konformation für beliebige Moleküle berechnen. Der Molekülsatz sorgt für die grafische Darstellung eines Moleküls mit seiner zugehörigen Konformation. Als Ausgabeformat werden neben fixen Vektor- und Pixelgrafiken auch interaktive Darstellungen mithilfe eines Szenengraphen unterstützt.

Ein Überblick der Softwarearchitektur findet sich in Abb. 3.1. Wichtig ist hierbei, dass es eine klare Trennung zwischen der Berechnung der 2D-Koordinaten und der Darstellung der Strukturdiagramme, dem Molekülsatz, gibt.

Dies führt zu einer besseren Kapselung und sorgt dafür, dass die Bibliotheken einfacher und flexibler benutzt werden können: 2D-Koordinaten von Molekülen können getrennt vom Molekülsatz vorausberechnet und effizient zwischengespeichert werden (im Speicher oder auch in einer Datenbank). Das Setzen der Moleküle ist wiederum nicht auf 2D-Koordinaten der Naomi$_{2D}$-Bibliothek festgelegt: Koordinaten können aus beliebigen anderen Quellen stammen, sei es ein weiterer Algorithmus oder auch direkt aus den 2D- oder 3D-Koordinaten der eingelesenen Moleküle.

In diesem Kapitel werden zuerst die grundlegende 2D-Koordinatenberechnung und der anschließende Molekülsatz beschrieben. Im letzten Teil geht es um eine Erweiterung der Koordinatenberechnung, die Ähnlichkeiten zwischen Molekülen beachtet.

3.1. Berechnung von 2D-Molekülkoordinaten

Die Berechnung von 2D-Koordinaten ist in 3 Phasen eingeteilt. Zuerst werden die Koordinaten aller Ringsystemen berechnet, danach findet der Optimierungsschritt statt, in dem aus allen möglichen Strukturdiagrammen ein bestes herausgesucht wird. Im Nachbearbeitungsschritt finden auf diesem eine Reihe von Verfeinerungen statt.

Bevor wir zu der genaueren Beschreibung der Phasen kommen, starten wir mit der grundlegenden Frage, wie die 2D-Koordinaten von Molekülen dargestellt werden. In unserem Kontext werden die 2D-Positionen der Atome eines Moleküls auf zwei unterschiedliche jedoch äquivalente Arten beschrieben: Durch absolute Koordinaten (jedes Atom erhält eine feste 2D-Position) und durch relative Koordinaten (jedes Atom besitzt Entfernung und Winkel zum Vorgängeratom).

3.1.1. Absolute und relative Koordinaten

Um die Position der Atome eines Moleküls im zweidimensionalen Raum zu beschreiben, sind absolute Koordinaten die naheliegendste Möglichkeit: Jedem Atom des Moleküls wird eine feste 2D-Koordinate zugeordnet. Im Rechner erfolgt die Speicherung mithilfe von zwei Gleitkommazahlen doppelter Genauigkeit, die in einem dynamischen Array abgelegt sind.

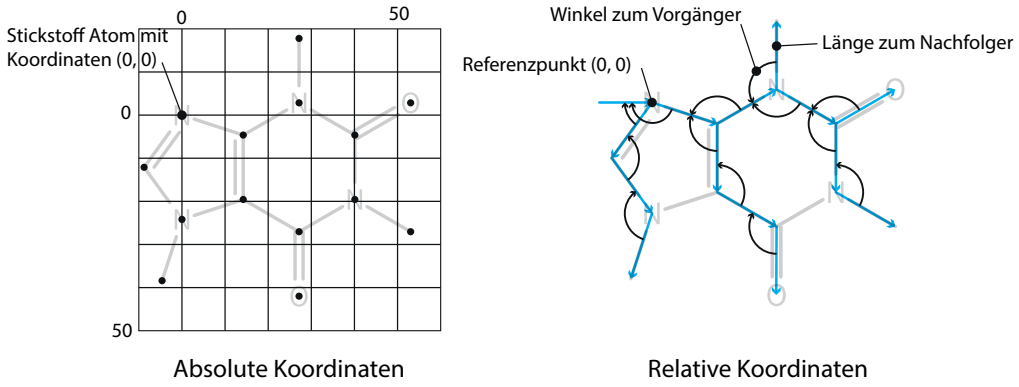


Abb. 3.2. Die Naomi$_{2D}$ Bibliothek verwendet je nach Anwendungsfall entweder absolute Koordinaten oder relative Koordinaten. Absolute Koordinaten enthalten pro Atom die absolute 2D-Position. Relative Koordinaten betrachten für jedes Atom den Winkel zum Vorgängeratom und die Distanz zum Nachfolgeatom.

Eine weitere Möglichkeit besteht darin, relative Koordinaten zu benutzen (s. Abb. 3.2). Dazu wird auf dem Molekülgraphen ausgehend von einem Referenzatom eine Breitensuche durchgeführt. Alle Atome außer dem Referenzatom haben in dem dabei entstandenem Breitensuchbaum genau einen Vorgänger. Die relativen Koordinaten bestehen für jedes Atom aus der Distanz zum Vorgängeratom und für jede Bindung aus dem Winkel zur Vorgängerbindung. Als Datenstruktur kommt ein Baum zum Einsatz, in

dem pro Knoten die Distanz und der relative Winkel gespeichert sind. Da Winkel in Strukturdiagrammen sehr regelmäßig sind und häufig aus Vielfachen von 60 bestehen (die Winkel 120, 240, 60 und 90 kommen am häufigsten vor), wird Grad anstatt Radiant als Einheit benutzt. Damit werden Rundungsfehler der Fließkommazahlen bei Änderungen auf den relativen Winkeln zu vermieden. Jedem Atom des Moleküls wird in dieser Darstellung ein Baumknoten eindeutig zugeordnet, für die Bindungen gilt dies im Allgemeinen nicht.

Beide Koordinatendarstellungen sind ineinander überführbar: Um aus den absoluten Koordinaten relative zu berechnen, wählt man ein Referenzatom, von dem aus eine Breitensuche durchgeführt wird. Anhand des Breitensuchbaums berechnet man für jedes Atom die Distanz und den relativen Winkel mithilfe der absoluten Koordinaten. Die umgekehrte Richtung erhält man nach der Wahl einer beliebigen absoluten Koordinate[1] für das Referenzatom, die restlichen ergeben sich durch rekursives Abarbeiten der relativen Winkel und Distanzen im Baum.

Absolute Koordinaten werden hauptsächlich in der Ringsystemphase verwendet, z. B. wenn ein Kraftfeld benutzt wird, um die Koordinaten eines Ringsystems über mehrere Iterationen hinweg zu berechnen. In der Optimierungsphase werden dagegen viele lokale Änderungen durchgeführt. Relative Koordinaten sparen dabei unnötige Berechnungen: Bei der Änderung einer Bindungslänge in der Mitte des Moleküls muss mit relativen Koordinaten nur ein Wert umgesetzt werden, mit absoluten Koordinaten müsste mindestens die Hälfte aller Koordinaten neu berechnet werden.

Schlussendlich sind relative Koordinaten allerdings nur eine temporäre Repräsentation, das Ergebnis der Koordinatenberechnung wird immer als 2D-Konformation mit absoluten Koordinaten zurückgegeben.

3.1.2. Phase 1 – Berechnung von 2D-Ringsystemkoordinaten

Das Berechnen von guten Ringsystemkoordinaten hat einen großen Einfluss auf das Aussehen der Strukturdiagramme. Etwa 97 % der gebräuchlichen Moleküle enthalten entweder keine Ringsysteme oder nur Ringe mit 3 bis 9 Atomen (s. Abb. 3.3). Dennoch ist es wichtig, für die fehlenden 3 % gute Vorgaben zu haben. In Abb. 3.4 ist exemplarisch zu sehen, welche negativen Auswirkungen ein unsymmetrisches Layout auf die Übersichtlichkeit eines komplizierten Ringsystems hat.

Es gibt viele Konventionen, wie bestimmte Arten von überbrückten Ringen oder größere Ringsystem gezeichnet werden sollen (GR-3.3 in [42]). Diese Konventionen sind über viele Jahrzehnte entstanden, um häufig vorkommende Ringsysteme möglichst übersichtlich und symmetrisch darzustellen. Vor allem erlauben sie aber Chemikern das einfache Wiedererkennen dieser Ringstrukturen.

Da diese Konvention keinem festen Schema folgen, lassen sie sich am besten über ein flexibel zu erweiterndes Vorlagensystem umsetzen. Um dabei die Anzahl der Vorlagen zu minimieren, werden alle Ringsysteme zuallererst in möglichst kleine Blöcke zerlegt. Danach werden für jeden Block eigene absolute Koordinaten berechnet, die am Ende

[1] im vorliegenden Programm immer $(0, 0)$

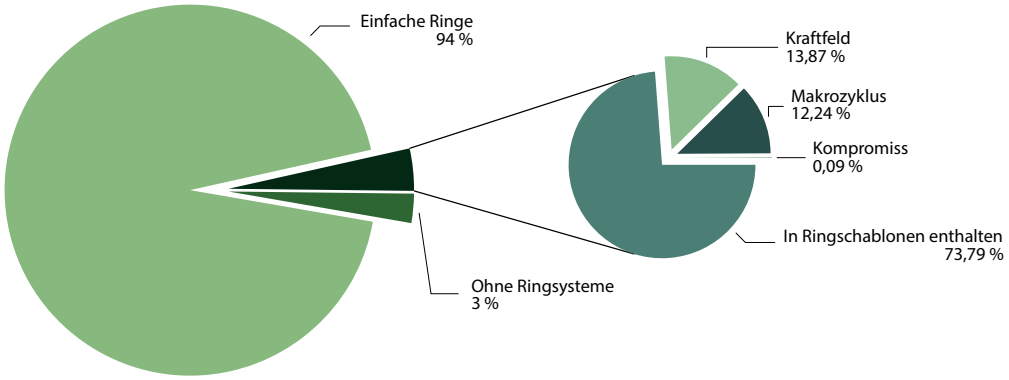


Abb. 3.3. Im Diagramm ist der Anteil unterschiedlicher Arten von Ringsystemen aller Moleküle der PubChem Substance Datenbank [46] dargestellt (s. Tab. 4.1). Die PubChem Substance Datenbank enthält 147 Mio. Moleküle (s. Kap. 4.2.1). Am häufigsten bestehen Moleküle ausschließlich aus Ringen mit 3-9 Atomen oder besitzen keine Ringsysteme. Für die meisten anderen wurde eine Schablone gefunden. Die restlichen Moleküle bestehen überwiegend aus Makrozyklen oder mussten per Kraftfeld generiert werden.

wieder zu einem kompletten Layout für das Ringsystem zusammengesetzt werden.

Vorbereitung der Ringsysteme

Als Methode zum Zerteilen des Ringsystems in einzelne Blöcke wird eine Verallgemeinerung des in [15] beschriebenen *terminal ring peeling* Verfahrens benutzt. Hierbei werden Ringsysteme nicht ausschließlich an terminalen Stellen unterteilt. Das Ringsystem kann dadurch in kleinere Blöcke zerteilt werden, wodurch wieder weniger Vorlagen nötig werden.

Damit nach dem Aufteilen das Ringsystem möglichst einfach wieder zusammengesetzt werden kann, darf dieses nur an bestimmten Stellen unterteilt werden. Intuitiv ist immer dann eine Unterteilung erlaubt, wenn zwei Ringe eines Ringsystems nur eine Bindung oder ein Atom gemeinsam haben.

Die Koordinaten der Ringsysteme werden in vier Schritten berechnet (s. Abb. 3.5):

Schritt 1 Die Ringe des Ringsystems werden anhand ihrer *unique ring family* aufgeteilt (s. Kap. 2.4.2).

Schritt 2 Daraus wird ein Graph der Nachbarschaftsbeziehungen zwischen den Familien aufgebaut. Zwei Familien sind dabei genau dann benachbart, wenn sie Bindungen oder Atome gemeinsam haben.

Schritt 3 Mithilfe einer Tiefensuche werden auf dem Nachbarschaftsgraphen alle zweifachen Zusammenhangskomponenten [17] berechnet. Potenzielle Stellen zur Aufteilung des Ringsystems in Blöcke sind alle Kanten zwischen zwei verschiedenen

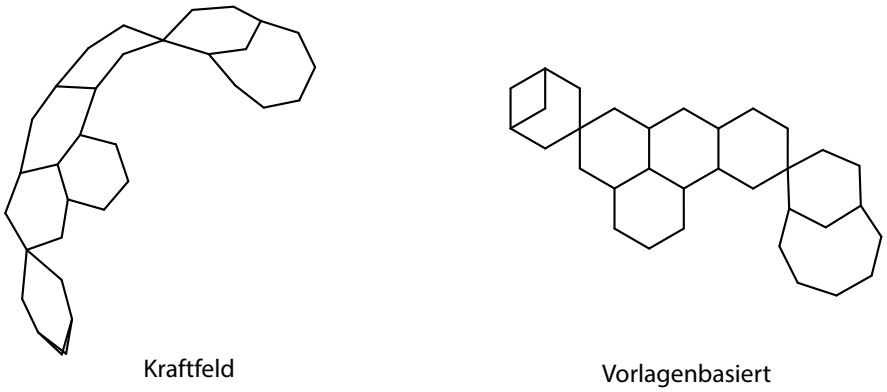


Abb. 3.4. Das Ringsystem des Moleküls auf der linken Seite wurde mit einem Kraftfeld gezeichnet, für dasselbe Molekül auf der rechten Seite wurden stattdessen Vorlagen benutzt.

Komponenten. Aus diesen Kanten werden diejenigen als Schnittkanten ausgewählt, bei denen die Nachbarschaft aus genau einem gemeinsamen Atom oder einer gemeinsamen Bindung besteht.

Schritt 4 Abschließend werden die Blöcke topologisch sortiert. Diese Sortierung gibt vor, in welcher Reihenfolge die Blöcke aneinandergesetzt werden. Vor dem Zusammenbau werden allerdings zuerst absolute Koordinaten für jeden einzelnen Block berechnet.

Berechnung von Blockkoordinaten

In Naomi_{2D} sind vier unterschiedliche Methoden enthalten, um Koordinaten für die einzelnen Ringblöcke zu berechnen:

Direkt berechnet Koordinaten trigonometrisch für beliebig große einzelne Ringe.

Datenbank benutzt eine Vorlagendatenbank, um die richtigen Koordinaten zuzuweisen.

Makrozyklus berechnet Koordinaten für beliebig große Makrozyklen.

Kraftfeld generiert Koordinaten mit einem Kraftfeld.

Es ist nicht immer möglich, pro Block von Anfang an zu entscheiden, welche Methode die richtige ist. Daher berechnet die Funktion RINGSYSTEM-LAYOUT für jeden Block eines Ringsystems einen Satz von Koordinaten C (s. Abb. 3.6). Beim Zusammenbau des Ringsystems werden die einzelnen Koordinatensätze bewertet und der Koordinatensatz benutzt, der nach dem Zusammenbau zu den wenigsten Überlappungen führt.

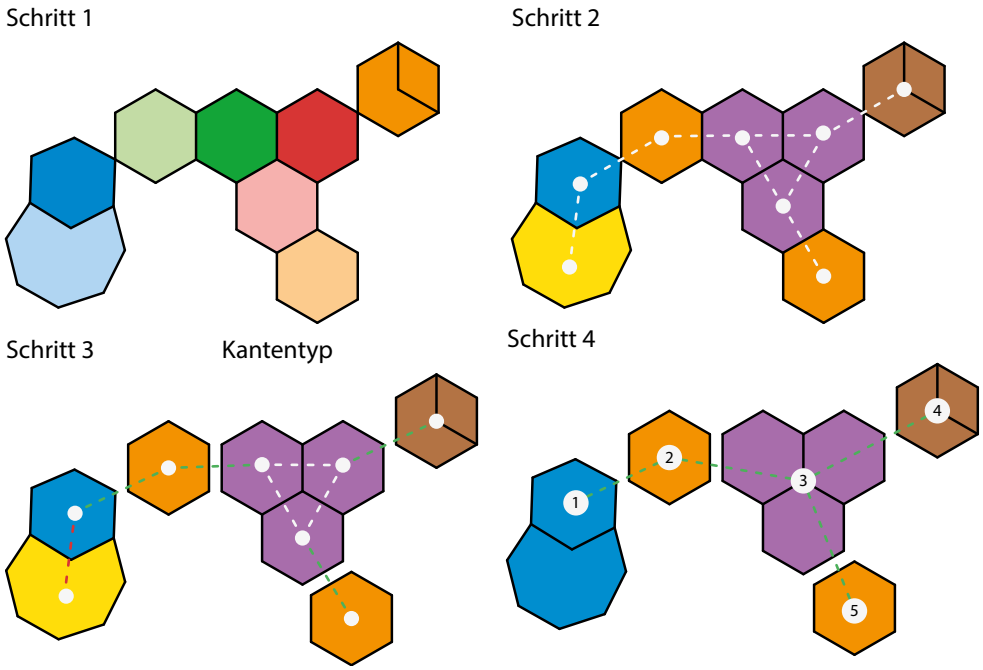


Abb. 3.5. Aus Ringsystemen entstehen in drei Schritten Ringblöcke: Im ersten Schritt wird ein Graph der Ringfamilien des Ringsystems aufgebaut. Im zweiten Schritt werden die Nachbarschaftsbeziehungen der einzelnen Familien untereinander ermittelt und die zweifach zusammenhängenden Komponenten dieses Graphen berechnet. Danach wird der Graph zwischen allen Komponenten geteilt, die maximal eine Bindung oder ein Atom gemeinsam haben. Das Resultat ist eine topologisch sortierte Aufteilung des Ringsystems in Blöcke.

Im Folgenden werden die einzelnen verfügbaren Methoden zur Generierung von Blockkoordinaten genauer beschrieben.

Methode 1: *Direkt* Die einfachste Methode ist gleichzeitig auch die am häufigsten eingesetzte: Wenn ein Block nur aus einem einzelnen Ring der Größe 3 bis 9 besteht, können die Koordinaten direkt trigonometrisch berechnet werden. Die Formel für einen Ring der Größe n für die 2D-Koordinaten des Eckpunktes k_i mit $i \in \{1, \dots, n\}$ ist:

$$k_i = s \begin{pmatrix} \cos\left(i\frac{2\pi}{n}\right) \\ \sin\left(i\frac{2\pi}{n}\right) \end{pmatrix}$$

Dabei stellt der Faktor

$$s = \frac{L}{2} \sin\frac{\pi}{n}$$

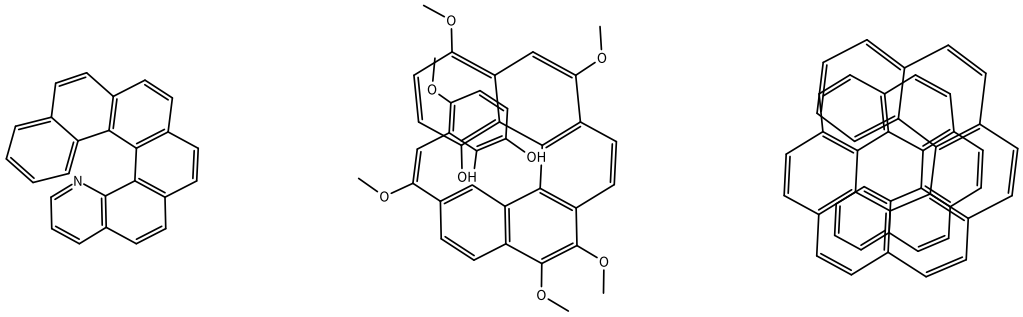
sicher, dass die Kanten des Ringes direkt auf die Länge L skaliert werden.

```
RINGSYSTEM-LAYOUT(R)
    C = ∅

    // Can we find and apply a template for the whole ring system?
    if R.rings.length > 1 and IN-DATABASE(R.atoms)
        C[1] = DATABASE-COORDINATES(R.atoms)
        return C

    blocks = PREPARE-BLOCKS(R)
    for i = 1 to blocks.length
        block = blocks[i]
        if block.rings.length == 1
            if block.rings[0].size < 9
                C[i] = DIRECT-COORDINATES(block.atoms)
            else
                C[i] = MACROCYCLE-COORDINATES(block.atoms)
                // add trigonometric coordinates as fallback
                C[i] = C[i] ∪ DIRECT-COORDINATES(block.atoms)
        else
            if IN-DATABASE(block.atoms)
                C[i] = DATABASE-COORDINATES(block.atoms)
            else
                if IS-MACROCYCLE(block.atoms)
                    C[i] = MACROCYCLE-COORDINATES(block.atoms)
                // add forcefield coordinates as fallback
                C[i] = C[i] ∪ FORCEFIELD-COORDINATES(block.atoms)
    return C
```

Abb. 3.6. Der RINGSYSTEM-LAYOUT Algorithmus berechnet absolute Koordinaten für die kompletten Ringsysteme der Moleküle. Dabei kommen vier unterschiedliche Methoden zum Einsatz: DATABASE-COORDINATES zieht eine Datenbank mit vorgegebenen Koordinaten zurate, DIRECT-COORDINATES berechnet die Koordinaten für einzelne Ringe trigonometrisch, MACROCYCLE-COORDINATES setzt die Koordinaten von großen Makrozyklen und FORCEFIELD-COORDINATES verwendet ein Kraftfeld, um die Koordinaten zu berechnen.



PubChem SID: 111345814 PubChem SID: 23077771 PubChem SID: 104406320

Abb. 3.7. Verschiedene Helicene aus der PubChem Datenbank. Zusammensetzen von trigonometrisch berechneten Ringen führt bei diesen Molekülen zu direkt übereinanderliegenden Bindungen. Durch die Verwendung von leicht verzerrten Ringen, lässt sich das Molekül trotz Bindungsüberschneidungen besser erkennen.

Beim Zusammenbau eines Ringsystems aus mehr als fünf Einzelringen können potenziell Ringe komplett überlappen (s. Abb. 3.7). Dies tritt unter anderem bei Helicenen auf, also Spiralen aus kondensierten Benzolringen. Um die Generierung von einfachen Ringsystemen nicht zu verlangsamen, wird erst bei Ringsystemen mit mehr als 5 Blöcken jedem Einzelring zusätzlich eine leicht verzerrte Variante der berechneten Koordinaten als Alternative mitgegeben. Diese Variante wird aus den trigonometrisch berechneten Koordinaten durch eine Scherung der Koordinaten k_i mit der affinen Transformation

$$k_i' = \begin{pmatrix} 1 & 0{,}1 \\ 0 & 1 \end{pmatrix} k_i$$

erzeugt. Bei Helicenen führt diese Alternative zu den in Abb. 3.7 gezeigten Layouts. Anstatt komplett überlappender Ringe erkennt man die einzelnen Ringe.

Methode 2: *Datenbank* Bei der Datenbankmethode wird ein kompletter Block in der Vorlagendatenbank gesucht und die Koordinaten der Vorlage direkt auf diesen angewendet. Diese Methode eignet sich vor allem für überbrückte Ringsysteme oder Ringsysteme, die nach bestimmten Konventionen gezeichnet werden (GR-3.2.2 in [42]). Um Blöcke zuverlässig in einer Vorlagendatenbank wiederzufinden, wird eine Vergleichsfunktion benötigt, die isomorphe Graphen erkennt. Hierzu wird ein modifizierter kanonisierter[2] SMILES-Ausdruck des Ringblocks (im Folgenden als Ringblock-ID bezeichnet) als Schlüssel einer Hashtabelle benutzt. Bei der Suche sollen sowohl die chemischen Elemente der Atome als auch die Bindungsordnungen ignoriert werden, da diese in den meisten Fällen keinen Einfluss auf das Aussehen des Ringsystems haben. Daher werden die Ringblock-IDs so generiert, als wenn das Ringsystem nur aus Kohlenstoffen und Einfachbindungen bestände.

[2] Die Kanonisierung wird mit dem CANON Algorithmus in Naomi berechnet (s. Kap. 2.2.1).

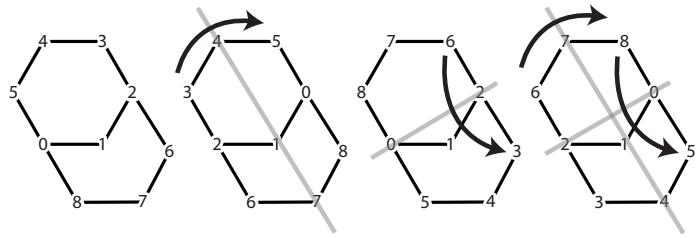


Abb. 3.8. Alle vier Automorphismen einer Ringvorlage. Das Molekül besitzt zwei Symmetrieachsen, um die unabhängig voneinander gespiegelt werden kann.

Alle Ringvorlagen von Naomi$_{2D}$ werden beim Programmstart aus einer SD-Datei in eine Hashtabelle geladen. Die Schlüssel der Hashtabelle sind dabei die Ringblock-IDs der einzelnen Ringblöcke. Als Werte sind direkt die Atomkoordinaten des Ringblocks hinterlegt. Diese sind anhand der Reihenfolge der Atome in der Ringblock-ID sortiert. Pro Ringblock können auch mehrere Koordinatensätze vorhanden sein. Zur Laufzeit führt Naomi$_{2D}$ pro Ringblock eine Suche in dieser Tabelle durch und erstellt aus den hinterlegten Koordinatensätzen verschiedene Varianten. Die richtige Koordinate wird den Atomen mithilfe der Ringblock-ID zugewiesen.

Vorbereiten der Ringvorlagen-Datenbank

Die Ringvorlagen-Datenbank wird in einem zweistufigen Prozess erstellt. Im ersten Schritt werden die Ausgangsdateien im SD-Format mit dem externen Programm `ringtemplate_builder` konvertiert. Im zweiten Schritt wird diese konvertierte Datei direkt von der Bibliothek Naomi$_{2D}$ benutzt. Dies ermöglicht es, aus beliebigen Moleküldateien mit 2D-Koordinaten neue Vorlagen für die Datenbank zu erstellen. Außerdem können so potenziell rechenintensive Schritte, wie das Erzeugen aller Ringblock-Automorphismen vorausberechnet werden. `ringtemplate_builder` zerlegt zuerst die Ringsysteme aller Moleküle der Ausgangsdateien in Ringblöcke. Als Koordinaten dieser Blöcke werden die 2D-Koordinaten benutzt, die in den Ausgangsdateien gespeichert sind. Danach werden diese Blöcke anhand der Ringblock-ID sortiert, so können auch mehrere Koordinaten-Varianten pro Ringblock-ID entstehen. Der Layoutalgorithmus wählt aus diesen Varianten die geeignetste aus.

Ein Graphenautomorphismus ist eine bijektive Abbildung der Knotenmenge des Graphen auf sich selbst, bei der alle bestehenden Kanten erhalten bleiben. Automorphismen beschreiben die Symmetrien eines Graphen (s. Abb. 3.8). Die Ausnutzung aller möglichen Symmetrien erlaubt es, die Anzahl der vorzugebenden Ringvorlagen minimal zu halten. Die Anzahl der Automorphismen steigt exponentiell bei sehr symmetrischen Strukturen, wie einfachen Ringen oder größeren ringähnlichen Gebilden. Daher werden nur für Ringblöcke, die keine einfachen Ringe sind, für jeweils alle Koordinatensätze alle Automorphismen berechnet und als zusätzliche Varianten abgelegt. Durch die Berechnung aller Automorphismen vergrößert sich die Datenbank. Im Fall von Naomi$_{2D}$ entstehen momentan aus den 174 am häufigsten in der PubChem Daten-

bank vorkommenden Ringblöcken auf diese Art 487 Vorlagen. Es ist im Ringsystemlayout nicht vorgesehen, dass für Ringblöcke aus der Datenbank weitere Kompromissvarianten vorsichtshalber berechnet werden. Es ist aber möglich, Alternativen explizit in der Datenbank abzulegen. Alle Alternativen eines Blockes werden der Reihe nach bewertet und durchprobiert. Eine weitere Information, die in den Vorlagen steckt, ist die bevorzugte Ausrichtung der Vorlagen. Typischerweise gibt das größte Ringsystem in einem Molekül die Ausrichtung für das komplette Molekül vor (GR-3.4.2 in [42]). Auch spezielle Ringsysteme wie Steroide werden immer auf dieselbe Art und in derselben Ausrichtung gezeichnet (GR-3.6 in [42]). Dazu ist es möglich, in der Ringvorlagen-Datenbank pro Vorlage eine bevorzugte Abweichung und die Priorität der Vorlage anzugeben. Beide Eigenschaften sind als SD-Eigenschaften umgesetzt, die von `ringtemplate_builder` automatisch bei der Konvertierung erzeugt werden. Als präferierte Ausrichtung wird dabei die Ausrichtung der Ausgangskoordinaten verwendet, die Priorität einer Vorlage ist ihre Größe. Dies führt bei der Koordinatengenerierung dazu, dass der größte Ringblock des Moleküls die Ausrichtung vorgibt und genau wie in der Ausgangsdatei ausgerichtet ist. Als Ergebnis des Vorbereitungsschritts schreibt `ringtemplate_builder` eine SD-Datei, die dann direkt von Naomi$_{2D}$ verwendbar ist.

Momentane Einschränkungen der Datenbankmethode Die Datenbankmethode kann noch an einigen Stellen verbessert werden, indem das Layout von Ringblöcken wie folgt abgeändert wird:

- Dreifachbindungen in Ringen sollten nach (GR-3.3.2 in [42]) wie bei den Ketten linear gezeichnet werden. Diese Forderung wird momentan von Naomi$_{2D}$ nicht umgesetzt. Die in der Datenbankmethode verwendete SMILES-Repräsentation lässt sich einfach mit den Bindungsordnungen erweitern. Dadurch wäre es möglich entsprechende Ringvorlagen mit Dreifachbindungen zur Vorlagendatenbank hinzuzufügen. Für die Makrozyklen- und Kraftfeldmethode sind die benötigten Änderungen dagegen aufwendiger.
- Momentan können nur Ringblöcke oder komplette Ringsysteme durch Vorlagen ersetzt werden. Ein Steroid, an dem noch ein weiterer Ring hängt, würde daher nicht die aktuelle Steroidvorlage benutzen, sondern in einzelne 6 und 5 Ringe zerlegt werden. Dieses führt zu einem korrekten Layout, nur die waagerechte Ausrichtung des Steroids (GR-3.6 in [42]) wird dabei nicht unbedingt beachtet. Außerdem sind bei der Ausrichtung mehrere bevorzugte Richtungen pro Ringblock momentan nicht vorgesehen, und die Spiegelung der Vorlage wird nicht berücksichtigt. Eine konsequente Behandlung der Ringblock-Ausrichtung in allen Layoutphasen, um möglichst viele und auch widersprüchlichen Ausrichtungen zu erfüllen, würde diese Probleme lösen.
- In der Ringdatenbank sind nur die häufigsten Vorlagen für PubChem-Ringsysteme enthalten. Die Längen und Winkel von einigen Vorlagen sind außerdem noch nicht perfekt. Durch ein entsprechendes Editierprogramm lässt sich das manuelle Einpflegen von neuen Vorlagen stark vereinfachen.

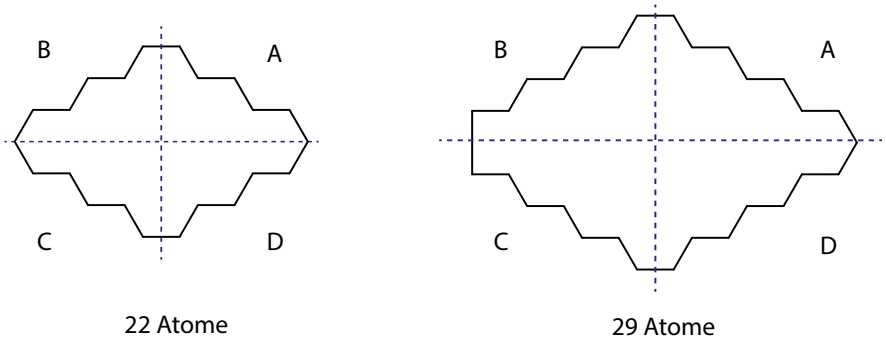


Abb. 3.9. Koordinaten für Makrozyklen unterschiedlicher Größe werden durch Spiegelungen erzeugt. Dazu wird Viertel A durch Spiegelung an der y-Achse zu B. Die darauffolgende Spiegelung an der x-Achse erzeugt C und D.

- Eine weitere Einschränkung ist die Verzögerung von bis zu einer Sekunde, die momentan bei jedem Programmstart benötigt wird. Dies liegt daran, dass die etwa 500 Moleküle der Ringdatenbank bei der Initialisierung der Naomi$_{2D}$-Bibliothek eingelesen und verarbeitet werden müssen. Ein geeignetes Binärformat der Datenbank würde es ermöglichen, die Initialisierung der Moleküle vorauszuberechnen und damit den Prozess zu beschleunigen.

Methode 3: *Makrozyklen* Der Makrozyklus-Algorithmus berechnet für große Ringe ab 10 Atome ein Layout, das sich möglichst auf den Linien eines Hexagonalgitters befindet. Da auf Hexagonalgittern nur bestimmte Ringe mit einer geraden Anzahl von Kanten möglich sind, müssen für Ringe mit einer ungeraden oder ungeeigneten geraden Anzahl von Bindungen zusätzliche Bindungen eingefügt werden. Der Algorithmus baut einen Makrozyklus der Größe n aus vier gespiegelten Teilen zusammen. Der erste Schritt besteht darin, aus $t = \lfloor \frac{n}{4} \rfloor$ Atomen Teil A des Makrozyklus zu erstellen (s. Abb. 3.9). Teil B entsteht durch Spiegelung an der y-Achse und die Teile C und D schließlich als Spiegelung von A und B an der x-Achse. Ist die Anzahl der Ringatome nicht exakt durch 4 teilbar, werden die übrigen Atome passend eingefügt. Hierbei können auch Bindungen entstehen, die nicht der Standardbindungslänge entsprechen. Eine Erweiterung des Makrozyklus-Algorithmus besteht darin, auch kleine, eingebettete Ringe zu behandeln. Besteht der gesamte Ringblock neben dem Makrozyklus nur aus einfachen Ringen mit maximaler Größe 9, wird versucht, diese kleineren Ringe in den Makrozyklus einzubetten. Dies führt allerdings in einigen Fällen zu sich überlappenden Bindungen. Um diese Fälle zu minimieren, wird jedes Mal, wenn der Makrozyklus-Algorithmus läuft, auch noch eine vom Kraftfeld (s. Kap. 3.1.2) generierte Kompromissvariante erzeugt. Falls der Makrozyklus mit den eingebetteten Ringen überlappende Bindungen oder Atome hat, wird die vom Kraftfeld generierte Variante bevorzugt.

Methode 4: *Kraftfeld* Die letzte Methode, um Ringblockkoordinaten zu berechnen, ist ein Kraftfeld. In dieser Arbeit wurde die in [26] beschriebene Methode verwendet. Da das Kraftfeld auf beliebigen Graphen läuft, kommt diese Methode immer dann zum Einsatz, wenn keine der anderen Methoden angewendet werden kann. Außerdem dient das Kraftfeld als Ersatzlösung für Methode 3: Bei bestimmten Konstellationen der eingebetteten Ringe und der Substituenten generiert der Makrozyklus-Algorithmus überschneidende Atome oder Bindungen.

Das Kraftfeld ist so parametrisiert, dass Ringsysteme mit möglichst wenig Überschneidungen erzeugt werden. Insgesamt wird das Kraftfeld zehnmal auf dem Ringsystem mit unterschiedlichen Zufallswerten laufen gelassen. Aus den Ergebnissen wird das Layout mit der kleinsten Anzahl an Überschneidungen ausgewählt.

Zusammensetzen der Blöcke Nachdem für alle Ringblöcke geeignete Koordinaten aus Vorlagen geladen oder berechnet wurden, werden die Blöcke wieder anhand der gemeinsamen Bindungen beziehungsweise der gemeinsamen Atome bei Spiroverbindungen zusammengesetzt. Um die beste Zusammensetzung zu finden, werden zwei Bewertungsfunktionen benutzt:

1. Die Akzeptanzfunktion bewertet, ob das zusammengesetzte System akzeptabel ist. Ein System ist dann akzeptabel, wenn keine Atome und keine Bindungen exakt übereinanderliegen. Dies gilt sowohl für die Atome und Bindungen des Ringsystems als auch für die der Substituenten. Bei einer akzeptablen Lösung sind Kollisionen der Bindungen und Atome erlaubt, die bis zu 7,5 % der Bindungslänge nah nebeneinander liegen.

2. Die Präferenzfunktion benutzt drei weitere Kriterien, um unter allen akzeptablen Lösungen eine präferierte Variante auszuwählen. Diese Kriterien sind:

   a) die minimale Anzahl der Kollisionen im Ringsystem: Hierzu zählen neben den exakten Überlagerungen auch Atome, die weniger als eine halbe Bindungslänge voneinander entfernt sind, und Bindungen, die sich überschneiden.

   b) die maximale Anzahl der Heteroatome, die sich an Bindungen im Innern des Ringes befinden.

   c) die maximale Anzahl der Substituenten, die den Block nach außen verlassen.

Existiert keine einzige akzeptable Bewertung, werden mit der Präferenzfunktion die entsprechend gekennzeichneten Kompromissvarianten für die einzelnen Blöcke bewertet.

Pro Block können mehrere Varianten vorhanden sein. Es ergeben sich also potenziell exponentiell viele Möglichkeiten, ein Ringsystem wieder zusammenzusetzen. Gerade bei symmetrischen Ringen steigt die Anzahl der Ringblock-Varianten sehr stark an (s. Abb. 4.3). Da die Anzahl der Varianten der einzelnen Blöcke bekannt ist, lässt

sich die Gesamtzahl der möglichen Varianten direkt berechnen. Wenn diese Zahl kleiner als 500 ist, werden alle Kombinationen bewertet und die beste benutzt. Wenn die Anzahl größer ist, werden insgesamt 500 zufällige Kombinationen gleichverteilt ausgewählt und diese bewertet. Von diesen wird die am besten bewertete für das komplette Ringsystem benutzt.

Das gesamte Ringsystem wird daraufhin in der 2D-Konformation abgelegt und geeignete Längen und Winkel werden daraus mit relativen Koordinaten berechnet. Diese Prozedur wird für jedes Ringsystem des Moleküls einzeln durchgeführt, sodass am Ende alle Ringsysteme Koordinaten besitzen, die danach intern nicht mehr verändert werden.

3.1.3. Phase 2 – Die Suche nach dem besten Diagramm

Die zweite Phase der Koordinatenberechnung besteht aus einem Optimierungsalgorithmus, der lokale Änderungen auf das Strukturdiagramm anwendet. Die Grundlage bilden dabei die benutzten Qualitätsbewertungskriterien des Layouts und die Beschreibung aller möglichen lokalen Änderungen. Die Qualitätsbewertungskriterien sind eine Erweiterung der schon aus dem vorherigen Abschnitt bekannten Kriterien für die Ringsysteme.

Qualitätsbewertungskriterien

Kollisionen Ein wichtiges allgemeines und naheliegendes Qualitätsmaß von Strukturdiagrammen ist die Anzahl der kollidierenden Atome und Bindungen in der Darstellung. Ein Diagramm mit möglichst wenig Kollisionen ist dabei einem Diagramm mit mehr Kollisionen vorzuziehen. Aber welche Arten von Kollisionen sind möglich?

Zum einen können die Atome des Strukturdiagrammes sehr nahe beieinanderliegen oder sogar exakt übereinanderliegen. Zum anderen ist es auch möglich, dass sich Bindungen in der Mitte überschneiden. Die Atome befinden sich zwar eine halbe Bindungslänge voneinander entfernt, es gibt aber trotzdem eine Kollision im Diagramm.

Um nicht Atome und Bindungen getrennt voneinander betrachten zu müssen, wird zur Bestimmung der Kollisionsrate ausschließlich die minimale geometrische Distanz zweier Liniensegmente betrachtet (s. Abb. 3.10). Eine Distanz von 0 bedeutet dabei, dass entweder zwei Atome übereinanderliegen, sich ein Atom auf dem anderen Segment befindet, oder dass sich beide Liniensegmente schneiden. Eine Distanz $d > 0$ bedeutet, dass alle Punkte des einen Liniensegmentes sich geometrisch mindestens d entfernt von allen Punkten des zweiten Liniensegmentes befinden. Insbesondere heißt das also für einen Abstand $d_k = S/2$, wobei S die Standardbindungslänge des Strukturdiagramms ist, dass sich diese beiden Bindungen nicht überschneiden und sich die Atome jeweils nicht näher als eine halbe Bindungslänge kommen. Wenn dieser Abstand d_k unterschritten wird, zählen die entsprechenden beiden Liniensegmente als potenzielle Kollision.

Beim Kollisionstest werden mit einem *scanline*-Verfahren zuerst einmal alle Bindungen ermittelt, die den Abstand d_k unterschreiten. Wenn dies für zwei Bindungen b_1 und

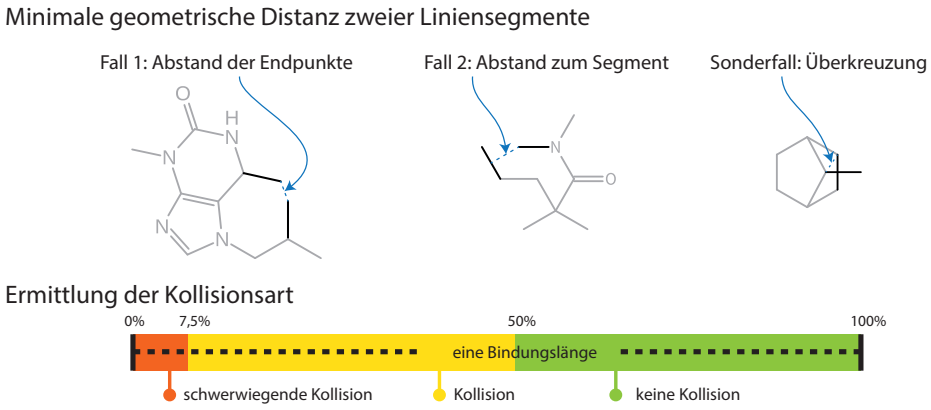


Abb. 3.10. Bei der Ermittlung der minimalen geometrischen Distanz von zwei Liniensegmenten sind mehrere Fälle zu beachten: Entweder ist die Distanz zweier Endpunkt die minimale Distanz oder die Distanz eines Endpunktes zu einem beliebigen Punkt des anderen Segmentes ist die minimale Distanz. Ein Sonderfall tritt ein, wenn sich die Segmente überkreuzen und die Distanz damit 0 beträgt. Hier wird die minimale Distanz der Endpunkte zur Bestimmung der Kollisionsart herangezogen. Anhand der minimalen geometrischen Distanz wird die Schwere einer Kollision ermittelt: Als Kollision werden alle Fälle betrachtet, bei denen die minimale Distanz kleiner als eine halbe Bindungslänge ist. Wenn die Distanz weniger als 7,5 % der Bindungslänge beträgt, liegt eine schwerwiegende Kollision vor.

b_2 der Fall ist, wird die exakte Kollisionsart näher klassifiziert. Es gibt dabei noch einen Spezialfall: Wenn b_1 und b_2 im Molekül topologisch benachbart sind und die Bindungen nicht komplett übereinanderliegen, handelt es sich um keine Kollision, da benachbarte Bindungen sich in genau einem Atom überlappen. Ansonsten werden die Kollisionen nach Distanz der Bindungen und deren Art unterteilt.

Ist die Distanz zweier Atome der Bindungen kleiner als 7,5 % der Bindungslänge, gilt dies als ein Fall ununterscheidbarer Atome. Wenn die Bindungen exakt übereinanderliegen, ist es ein Fall von ununterscheidbaren Bindungen. Diese beiden Fälle sind unbedingt zu vermeiden, da sie die chemische Aussage des Diagramms verfälschen.

Gelten diese beiden Fälle nicht, kann die Distanz der Liniensegmente trotzdem 0 betragen, wenn sich die beiden Segmente überschneiden. Diese Fälle werden als Überschneidung gewertet. Alle anderen Fälle werden als Kollisionen betrachtet. Sowohl Kollisionen als auch Überschneidungen hängen von der Art der beteiligten Bindungen ab: Wenn es sich bei b_1 und b_2 um Ringbindungen desselben Ringsystems handelt, ist es eine ringsysteminterne Kollision. Handelt es sich bei einer der Bindungen um eine Ringbindung und bei der anderen um einen direkten Substituenten dieses Ringsystems, ist die Kollision eine Ringsystem-/Substituenten-Kollision. Und die letzte Art der Kollision sind alle restlichen Fälle: Also Kollisionen von Kettenatomen, Kollisionen verschiedener Ringsysteme, usw. Diese Art der Kollision gilt es vornehmlich zu vermeiden.

Zusammenfassend lassen sich also alle Kollisionen nach ihrer Wichtigkeit sortiert in die vier in Tab. 3.1 beschriebenen Kategorien A, B, C und D einteilen. Kategorie A verän-

Tab. 3.1. Kollisionen im Layout von Strukturdiagrammen lassen sich nach Wichtigkeit in die vier Kategorien A, B, C und D einteilen. Kollisionen der Kategorie A werden in allen Phasen minimiert, Kollisionen der Kategorie B in der zweiten Phase (Optimierung) und Kollisionen der Klassen C und D in der ersten Phase (Ringysteme).

Kategorie	Name	Beschreibung
A immer	Ununterscheidbare Bindungen	Bindungen liegen exakt übereinander
A immer	Ununterscheidbare Atome	Atome sind maximal 7,5 % der Bindungslänge voneinander entfernt
B Phase 2	Kollision Ketten	Zwischen den Atomen befindet sich mindestens eine Kettenbindung.
B Phase 2	Schnitt Ketten	Zwischen den Bindungen befindet sich mindestens eine Kettenbindung.
C Phase 1	Kollision Ringsystem & Substituent	Kollision zwischen Atom eines Ringsystems und Atom eines Substituenten desselben Ringsystems.
C Phase 1	Schnitt Ringsystem & Substituent	Schnitt zwischen Bindung eines Ringsystems und einem Substituenten desselben Ringsystems.
D Phase 1	Kollision Ringsystem	Kollision zweier Atome innerhalb desselben Ringsystems.
D Phase 1	Schnitt Ringsytem	Schnitt zweier Bindungen innerhalb desselben Ringsystems.

dert dabei die chemische Bedeutung des dargestellten Moleküls. Kategorie B ist immer unerwünscht, da sie das Diagramm schwerer lesbar machen. Sie verändert aber nicht die Bedeutung des dargestellten Moleküls. Kategorien C und D können unerwünscht sein, sind aber durchaus in vollkommen validen und gut lesbaren Diagrammen vorhanden.

Gestreckte Ketten Ein weiteres wichtiges Kriterium betrifft die Form der Ketten des Moleküls. Um die Länge von Ketten in Molekülen leichter erkennen zu können, sollten diese einem Zickzack-Muster folgen (GR-3.2, GR-13 in [42]). Ketten sind flexibler als Ringsysteme und durch geschicktes Ausnutzen dieser Flexibilität lassen sich viele unlösbar erscheinende Kollisionen vermeiden. Für die Streckung der Ketten wird daher ein Maß benötigt, sodass man die bestmögliche Streckung der Ketten unter Vermeidung möglicher Kollisionen bewerten kann.

In Naomi$_{2D}$ werden zwei Maße A und B benutzt, die beide auf derselben geometrischen Idee beruhen: Verbindet man die Mittelpunkte gezackter Segmente, wird ihre

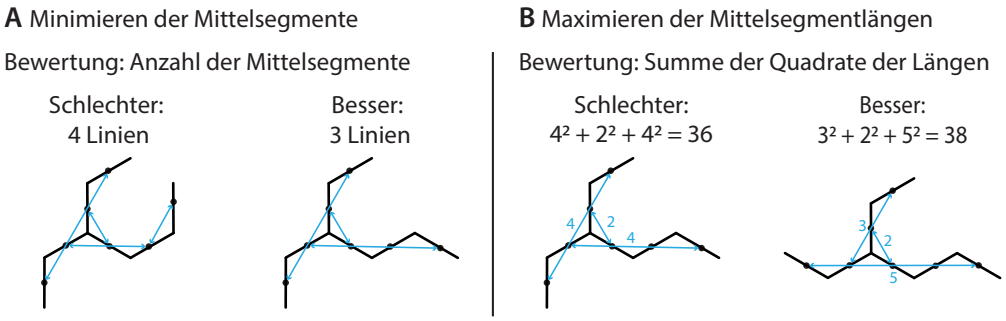


Abb. 3.11. Das Diagramm zeigt die beiden zur Bewertung der Kettenstreckung eingesetzten Maße. Maß A minimiert die Anzahl der Mittelsegmente in einem Diagram, Maß B maximiert die Länge des längsten Mittelsegmentes.

Gestrecktheit und ihre Länge messbar. Je weniger Knicke die durch die Mittelpunkte verlaufende Linie hat, desto mehr entspricht sie der idealen Zickzack-Form. Die Berechnung der kollinearen Segmente erfolgt durch einen geometrischen Algorithmus: Für jede Bindung des Moleküls wird zunächst der Mittelpunkt berechnet. Danach werden kurze Segmente gebildet, indem die Mittelpunkte von benachbarten Bindungen miteinander verbunden werden. Die Segmente werden anhand ihrer Steigung sortiert und mit benachbarten Segmenten mit derselben Steigung sukzessive zu längeren Segmenten zusammengefügt. Maß A ergibt sich aus der Anzahl der am Ende übrigbleibenden Mittelsegmente.

Ein Problem bei diesem Kriterium ist, dass es einige Fälle gibt, die durch die Anzahl der Mittelsegmente alleine nicht entschieden werden können: Im Fall des Moleküls aus Abb. 3.11 wird erst durch Maß B (die Maximierung des längsten Mittelsegmentes) das beste Moleküllayout gefunden.

Um das Kriterium der kollinearen Segmente auch für diesen Fall benutzen zu können, müssen auch die Längen der entstandenen Segmente betrachtet werden. Das zweite Maß B besteht daher aus der Summe aller quadrierten Segmentlängen S_i

$$B = \sum_{i=1}^{n} (S_i)^2$$

Dies führt dazu, dass längere Segmente ein deutlich höheres Gewicht als kürzere haben. Die Anzahl der Mittelsegmente muss minimiert (Maß A) und die Summe der Segmentlängen (Maß B) muss maximiert werden, um Diagramme mit möglichst langen Zickzack-Ketten aber dennoch möglichst wenig Abweichungen von der Zickzack-Form zu bekommen.

Distanzen Das letzte in Naomi_{2D} benutzte Kriterium ist die räumliche Ausdehnung, die das Gesamtmolekül einnimmt. Bei Kraftfeldern zum Berechnen von Graphen-Layouts ist es üblich, sowohl eine abstoßende Kraft als auch eine kontrahierende Kraft zu

benutzen, um eine Anordnung zu erzeugen, die weder zu weit noch zu kompakt ist [26]. Da Strukturdiagramme eine feste Kantenlänge haben, besitzen sie bereits eine minimale Ausdehnung. Es fehlt daher noch ein Term, um das Diagramm möglichst weit zu strecken und damit zu entzerren. Ein einfaches Kriterium ist die Summe der paarweisen Distanzen zwischen allen Atomen des Moleküls. Die Berechnung benötigt quadratische Zeit in der Anzahl der Atome. Dieses Maß betrachtet aber zu viele Distanzen: Viele Teile des Moleküls, wie die Ringsysteme, sind starr und die paarweisen Distanzen innerhalb dieses Systems ändern sich nicht. Es reicht also, die Distanzen zwischen einigen repräsentativen Atomen zu berechnen. Als Repräsentanten werden im Vorhinein ein Atom pro Ringsystems und das Endatom jeder Kette, deren Länge mindestens zwei ist, gewählt. Die Summe der paarweisen Distanzen zwischen diesen Atomen ist effizienter zu berechnen als für alle Atome des Moleküls.

Lexikographische Bewertung

Typischerweise bestehen Bewertungsfunktionen aus Summen, in die die Wertungsterme gewichtet eingehen. Dies erlaubt es, die Gewichtung der Terme dynamisch anzupassen, um so leichter das Optimum zu finden. Für Naomi_{2D} hat sich aber im Laufe der Entwicklung eine striktere Handhabung als besser herausgestellt: Diagramme sollten auf gar keinen Fall Kollisionen enthalten, erst wenn dies erfüllt ist, soll die Zickzack-Form der Ketten beachtet werden. Und die räumliche Ausdehnung des Diagramms ist erst dann wichtig, wenn es möglichst viele Zickzack-Ketten enthält. Als Form der Bewertung wurde daher eine lexikografische gewählt. Die Bewertungsfunktion besteht daher aus Unterwerten, die der Reihe nach verglichen werden. Nur wenn der Term mit der höchsten Priorität keinen Unterschied zeigt, wenn es also z. B. keine Kollisionen gibt, wird der nächst wichtigste Wert herangezogen.

Ein weiterer Vorteil der lexikografischen Bewertung ist ein Effizienzgewinn. Niedrig priorisierte Wertungsterme müssen gar nicht erst berechnet werden, wenn ein höher priorisierter Term schon zu einer Entscheidung geführt hat. Da die Bewertungsfunktion in der innersten Schleife der Optimierung abläuft, wirkt sich jede Beschleunigung direkt auf die Gesamtgeschwindigkeit der Koordinatenberechnung aus. Einige Wertungskriterien, wie die Anzahl der Kollisionen und die Berechnung der kollinearen Segmente, sind dabei durchaus aufwendig zu berechnen (Die Wertungsterme können bis zu $O(|E| \log |E|)$ Zeit[3] benötigen). Daher hat auch ein gitterbasierter Ansatz zur schnellen ($O(|V|)$) aber ungenauen Erkennung von Kollisionen die allerhöchste Priorität.

Lokale Änderungen

Grundlage der Suche nach dem besten Strukturdiagramm sind eine Reihe von lokalen Änderungen, die auf die relativen Koordinaten angewandt immer von einem gültigen

[3] Laufzeiten werden wie bei Graphen üblich angegeben: $|V|$ ist die Größe der Knotenmenge und $|E|$ ist die Größe der Kantenmenge. Auf den Molekülgraphen angewendet ist $|V|$ dann die Anzahl der Atome und $|E|$ die Anzahl der Bindungen.

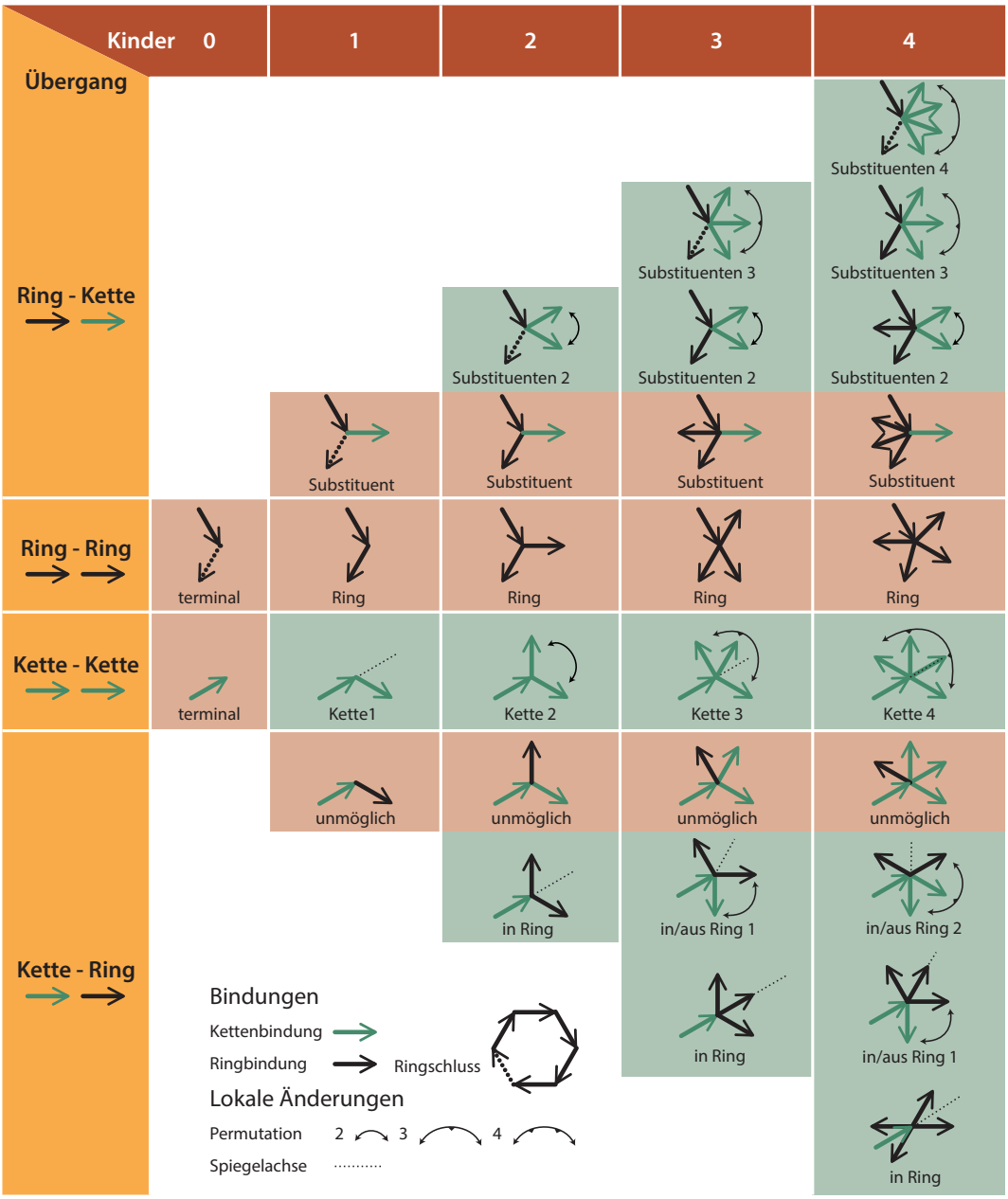


Abb. 3.12. Das Diagramm zeigt die möglichen lokalen Änderungen auf Atomen mit bis zu vier ausgehenden Bindungen. Lokale Änderungen arbeiten auf relativen Koordinaten, daher ist jedes Atom mit einer eingehenden Bindung und mehreren ausgehenden dargestellt. Vertikal sind die Fälle nach dem Übergang sortiert und horizontal nach der Anzahl der ausgehenden Bindungen. Rot unterlegte Fälle erlauben entweder keine Modifikation oder es ist ausgeschlossen, dass dieser Fall auftritt. Bei den grün unterlegten Fällen besteht die Möglichkeit von mindestens einer Änderung durch Spiegelung oder Permutation.

Tab. 3.2. In der Tabelle sind alle Bewertungskriterien für das 2D-Moleküllayout aufgeführt. Bei den Laufzeiten ist $|E|$ die Anzahl der Bindungen und $|V|$ die Anzahl der Atome des Moleküls. Die letzte Spalte enthält die Richtung, in die jedes Kriterium optimiert wird: Das Kriterium ist optimal, wenn der Wert entweder minimal oder maximal ist. Kriterien sind nach ihrer Priorität in der lexikografischen Bewertungshierarchie von oben nach unten sortiert. Die Vermeidung von Kollisionen hat damit die höchste Priorität.

Kriterium	Beschreibung	Laufzeit	Optimum
Gitterbasierte Atom Kollision	Ungefähre Anzahl der Atome die zu nahe beieinander liegen	$O(\|V\|)$	minimal
Ununterscheidbare Kollisionen	Anzahl der nicht unterscheidbaren Atome und Bindungen	$O(\|E\| \log \|E\|)$	minimal
Kollisionen	Anzahl der Bindungen, die zu nahe an anderen Bindungen liegen oder diese überkreuzen	$O(\|E\| \log \|E\|)$	minimal
Länge der Mittelsegmente	Gesamtlänge aller Zickzack-Ketten im Layout	$O(\|E\| \log \|E\|)$	maximal
Anzahl der Mittelsegmente	Anzahl der unterschiedlichen Zickzack-Ketten im Layout	$O(\|E\| \log \|E\|)$	minimal
Atom Distanzen	Die Summe der Distanzen aller endständigen Atome zueinander gibt an, wie gestreckt das Diagramm ist.	$O(\|V\|^2)$	maximal

Zustand des Strukturdiagramms zu einem anderen gültigen Zustand führen. Die Änderungen sind inspiriert von den in [10] vorgestellten *Drawing Conventions and Constraints*. Grundsätzlich gibt es drei Arten von Änderungen, die durchgeführt werden können: Bindungen in Ketten können gespiegelt werden, Ringsysteme sind als Ganzes rotierbar und mehrfache Substituenten an Ringsystemen oder Ketten können permutiert werden. Als einfachste und damit beste Möglichkeit hat sich eine atomweise Klassifizierung der Änderungen ergeben. Dabei wird für jede Bindung in den relativen Koordinaten und dem dazugehörigen Atom geschaut, welcher Fall eingetreten ist. In Abb. 3.12 sind alle möglichen Fälle für bis zu vier ausgehende Verbindungen bei Atomen dargestellt.

Es gibt die folgenden drei grundlegenden Änderungen auf dem Graphen:

Spiegelung von Kettenbindungen Ausgehende Kettenbindungen können gespiegelt werden. Diese Änderung ist für das Flippen von Zickzack-Bindungen zuständig. Wenn eine eingehende Bindung zwei ausgehende Bindungen besitzt, ist diese Änderung zu der Permutation der beiden Bindungen identisch. In einigen Fällen ist es notwendig, an einer beliebigen Achse spiegeln zu können. Die Spiegelung hat

immer zwei Zustände.

Spiegelung von Ringsystembindungen Ringsysteme können nur als Ganzes gespiegelt werden, ohne das Ringsystem zu verändern. Dazu werden alle in den relativen Koordinaten nachfolgenden Ringsystembindungen gleichzeitig gespiegelt. Diese Änderung hat zwei Zustände.

Permutation Bei mehr als zwei ausgehenden Bindungen können diese permutiert werden. Als Gesamtzahl der möglichen Zustände ergibt sich damit für den Fall von zwei ausgehenden Bindungen $2! = 2$, für drei ausgehende Bindungen sind es dann bereits $3! = 6$ Permutationen. Für den Fall von vier ausgehenden Bindungen werden, anstatt alle $4! = 24$ möglichen Permutationen anzubieten, nur die vier möglichen Permutationen durchgeführt, bei denen sich die Reihenfolge nicht ändert. Die Bindungen werden also quasi rotiert. Dies ist ein Kompromiss, um die Anzahl der möglichen Änderungen nicht zu stark zu erhöhen. Dasselbe gilt auch für den Fall von fünf ausgehenden Bindungen. Mehr als fünf ausgehende Bindungen kommen in organischen Moleküldiagrammen ohne Metallinteraktionen nicht vor.

Diese Änderungen können je nach Fall auch in Kombination auftreten: Ab drei ausgehenden Bindungen wird die Spiegelung für Ketten mit einer Permutation kombinierbar. Für den Fall „Kette 3“ ergeben sich $2 \cdot 6 = 12$ mögliche Zustände. Hier ist die Permutation unabhängig von der Spiegelung, sie können also in beliebiger Reihenfolge angewendet werden. Bei zwei ausgehenden Ringbindungen und einer ausgehenden Kettenbindung („in/aus Ring 1“ in Abb. 3.12) ist das nicht möglich. Die Spiegelachse für die Ringsystemspiegelung hängt davon ab, ob die Permutation des Ringes und des Substituenten zuerst ausgeführt wurde oder nicht.

Neben diesen Änderungen gibt es noch einige Spezialfälle, die die Freiheitsgrade erhöhen oder vermindern. Stereobindungen vermindern Möglichkeiten in Ketten. Sie werden über zwei voneinander abhängige Fälle modelliert: Der Zustand für die Doppelbindung selber hängt immer vom Zustand der eingehenden Bindung ab. Der Zustand der Doppelbindung ist dabei immer statisch, alle Veränderungen an der Doppelbindung werden durch die eingehende Bindung vorgenommen.

Ein weiterer Spezialfall sind Kettenatome mit vier ausgehenden Kettenbindungen. In speziellen Fällen werden diese nicht mit den Standardwinkeln 90°, 60°, 90° und 120° dargestellt, sondern es ist auch die rechtwinklige Variante 90°, 90°, 90° und 90° erlaubt (GR-4.1.3 in [42]). Die rechtwinklige Variante wird benutzt, wenn es sich beim Kettenatom um ein Sulfat oder Fluor handelt. Außerdem wird es benutzt, wenn alle vier ausgehenden Kettenbindungen nicht terminal sind. Dadurch kann der in Abb. 3.13 dargestellte Fall ohne Verzerrungen gelöst werden.

Der letzte Sonderfall sind lineare Bindungen. Dreifachbindungen oder zwei aufeinanderfolgende Doppelbindungen in Ketten werden, anstatt mit einem Knick von 120°, linear dargestellt, also mit einem Winkel von 180°.

Für die Optimierung ist es später wichtig, dass alle lokalen Änderungen beliebige Zustandswechsel erlauben. Für den Fall einer einfachen Spiegelung oder Permutation von

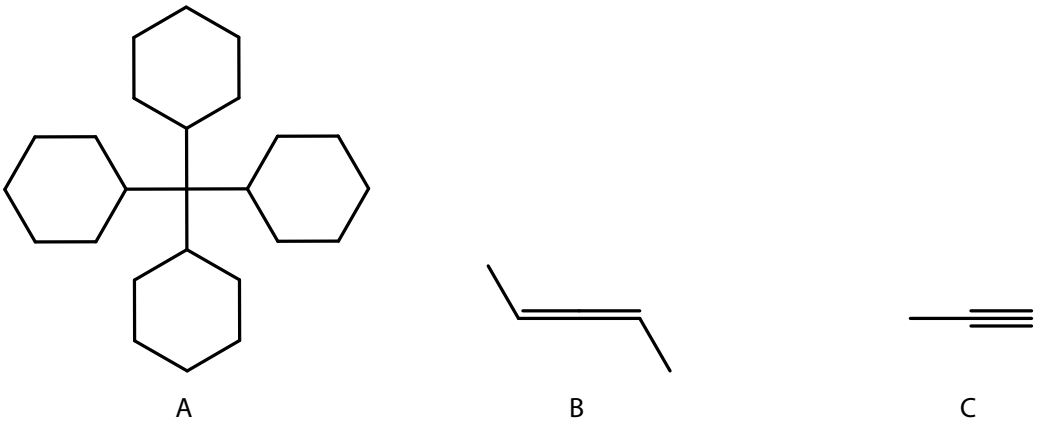


Abb. 3.13. Für die lokalen Änderungen bei Kettenatomen existieren drei Sonderfälle: In bestimmten Fällen ist für vier ausgehende Kettenbindungen die rechtwinklige Variante besser (Fall A). Zwei aufeinanderfolgende Doppelbindungen (Fall B) oder Dreifachbindungen (Fall C) werden immer linear dargestellt.

zwei Bindungen ist dieses trivial, da es nur zwei mögliche Zustände gibt. Jedoch schon ab der Permutation von drei Bindungen lohnt es sich, systematisch an dieses Problem zu gehen, um Fehler zu vermeiden. Allgemein bilden Permutationen von n Ziffern jeweils die endliche Gruppe S_n mit $n!$ Elementen. Alle Permutation lassen sich außerdem durch eine Abfolge von Vertauschungen von jeweils zwei Ziffern darstellen [71]. Beispielsweise kann die Permutation p aus S_4

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$

auch durch die Transpositionen

$$p = (1\,2)(2\,3)(3\,4)$$

dargestellt werden[4]. Um nun in einer beliebigen Gruppe von einem Zustand s_1 zu einem Zustand s_2 zu gelangen, wird das Gruppenelement d gesucht, sodass

$$s_1 \circ d = s_2$$

gilt. Dabei ist $\circ$ die Verknüpfungsfunktion der Gruppe. Umgeformt ergibt sich dann für

$$d = s_1^{-1} \circ s_2$$

Dafür muss die Gruppe nicht kommutativ sein. Mit einer Gruppenmultiplikationstabelle lässt sich d sehr effizient ermitteln. Es muss nun nur noch die passende Abfolge

[4] Wenn man als Vertauschungen nur benachbarte Elemente zulässt, erhält man die Grundlage eines sehr ineffizienten Sortieralgorithmus.

von Änderungen für das Gruppenelement d ausgeführt werden. Um z. B. in S_4 von der Permutation p zu der Permutation

$$q = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$

zu gelangen, berechnet man die Differenz d, um die nötigen Vertauschungen zu erhalten

$$d = p^{-1} \circ q = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix} = (1\,3)(2\,4)$$

Wenn man daher bei p Positionen 1 mit 3 und 2 mit 4 vertauscht, gelangt man zum Zustand q.

$$p \circ d = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix} = q$$

Für beliebige Zustandswechsel wird d berechnet und die sich daraus ergebenden Vertauschungen lokal ausgeführt. Lokale Änderungen, die aus einer Kombination von Spiegelung und Permutation bestehen, lassen sich ebenso über Gruppenoperationen beschreiben. Bei dem Fall „in/aus Ring 1“ aus Abb. 3.12 handelt es sich um die Kleinsche Vierergruppe V. V kann genau wie S_4 benutzt werden, um die für einen beliebigen Zustandswechsel benötigte Folge von Vertauschungen und Spiegelungen zu berechnen.

Insgesamt sieht man, dass es bei Ketten von Molekülen deutlich weniger Abweichungen von exakten 120° Winkeln gibt als in Ringsystemen. Jedoch ist die Anzahl der möglichen Diagramme exponentiell abhängig von der Anzahl der Kettenbindungen. Die meisten organischen Moleküle sind relativ klein und es können in vertretbarer Zeit alle möglichen Diagramme durchprobiert werden. Es ist aber dennoch nicht akzeptabel, wenn sich die Layoutzeit durch das Hinzufügen von nur einer Kettenbindung verdoppelt. Daher gibt es zwei Optimierungsmethoden: Bis zu einer maximalen Anzahl von 1000 Zuständen werden alle Möglichkeiten durchprobiert. Bei Molekülen mit mehr Möglichkeiten wird die folgende heuristische lokale Suche benutzt.

Heuristische Suche

Ziel der in Abb. 3.14 gezeigten Suche ist es, einen möglichst optimalen Zustand innerhalb aller möglichen Layouts eines Moleküls zu finden. Wenn man die oben beschriebenen lokalen Änderungen manuell ausführt, sieht man, dass häufig Änderungen an strategischen Bindungen auf einen Schlag zu sehr vielen Kollisionen führen oder diese beheben.

Diese Intuition steht hinter der Wahl eines *hill climbing* Suchverfahrens [36], bei dem in jedem Schritt eine Reihe von lokalen Änderungen durchprobiert wird. Diese Änderungen werden zufällig in der inneren Schleife ausgewählt mit dem Ziel, die Bewertung des Gesamtdiagramms möglichst stark zu verbessern. Eine weiterer wichtiger Punkt

```
HEURISTIC-SEARCH(diagram)
    maxLocalSlots = 50
    maxSteps = 100
    INITIALIZE-ZIG-ZAG-STATE(diagram)

    bestScore = CALCULATE-SCORE(diagram)
    bestState = GET-STATE(diagram)

    stateChanged = True
    step = 0

    while step < maxSteps and stateChanged == True
        step = step + 1

        C = GET-CHANGEABLE-STATES(diagram)
        Choose random subset L of C with |L| = min(|C|, maxLocalSlots)
        stateChanged = False
        for i = 0 to |L|
            APPLY-STATE(diagram, L[i])
            score = CALCULATE-SCORE(diagram)
            if score > bestScore
                bestScore = score
                bestState = L[i]
                stateChanged = True

    return bestState
```

Abb. 3.14. Wenn die Anzahl der möglichen Zustände eines Strukturdiagramms 1024 übersteigt, wird die heuristische Suche benutzt. Um ein möglichst gutes Layout zu finden, kommt ein *hill climbing* Verfahren [36] zum Einsatz. Dieses läuft $maxSteps$ Runden. In jeder Runde werden an maximal $maxLocalSlots$ Stellen zufällige Änderungen durchgeführt. Jede einzelne Änderung wird daraufhin bewertet und die beste Verbesserung wird genommen. Wenn durch keine der Änderungen die Bewertung verbessert wurde, wird die Suche vorzeitig abgebrochen.

ist der Initial-Zustand: Am Anfang werden alle Ketten des Strukturdiagramms in den Zickzack-Zustand versetzt. Dies stellt einen in vielen Fällen von vornherein optimalen Zustand dar.

3.1.4. Phase 3 – Nachbearbeitung

Nach dem Ringlayout und dem Layout der Ketten ist eine allgemeine Nachbearbeitung die letzte Phase. In dieser Phase findet eine Reihe von Schritten statt, um das Layout des Diagramms zu vervollständigen. Es fehlen noch die Koordinaten der Wasserstoffe, übriggebliebene Kollisionen werden, soweit möglich, durch Verzerrungen behoben, und das Diagramm wird im Gesamten ausgerichtet.

Wasserstoffkoordinaten

Wasserstoffe werden im Normalfall nicht explizit in Strukturdiagrammen gezeichnet, da ihre Position und Anzahl implizit bekannt ist: An jedem Schweratom sind so viele Wasserstoffe gebunden, wie es freie Valenzen besitzt. In einigen Ausnahmefällen müssen sie dennoch angezeigt werden: Zum einen bei Stereozentren mit zwei Wasserstoffen muss ein Wasserstoff explizit gemalt werden, um die Ausrichtung des Stereozentrums zu verdeutlichen (s. Kap. 2.4.3). Zum anderen bei der Darstellung von Wasserstoffbrücken in den von PoseView [76] modifizierten Strukturdiagrammen.

In Strukturdiagrammen werden Wasserstoffe mit 75 % der Standardbindungslänge entfernt vom Schweratom gezeichnet. Die Wasserstoffkoordinatenberechnung läuft daher einmal über alle Atome und berechnet geeignete Koordinaten für die Wasserstoffe von jedem Schweratom. Die zugrundeliegende Idee ist dabei einfach: Durch die eingehenden Bindungen wird der freie Bereich um jedes Schweratom herum in mehrere Teile geteilt. Von diesen Bereichen wählt man den Bereich aus, der für die Platzierung des Wasserstoffes am geeignetsten ist. Dafür wird für jeden Bereich berechnet, ob er sich innerhalb oder außerhalb eines Ringsystems befindet. Bevorzugt werden die Wasserstoffe in Bereichen außerhalb von Ringsystemen platziert. Außerdem muss berechnet werden, ob die Platzierung zu Kollisionen, oder in diesem Fall ausschließlich zu nah beieinanderliegenden Atomen führt. Die Platzierung geschieht durch einen *greedy* Ansatz, der Wasserstoffe bestmöglich platziert und bei jeder Platzierung die bereits vorher platzierten beachtet.

Diagrammorientierung

Chemiker erwarten Diagramme in einer bestimmten Orientierung, die von der Art des Moleküls und über die Jahre gebildete Konventionen abhängt. Unter anderem sollten Steroide immer auf eine bestimmte Art orientiert sein. Generell sollten Moleküle möglichst mit der längsten Achse horizontal ausgerichtet sein, und das wichtigste Ringsystem sollte sich rechts unten befinden (GR-3.1, GR-3.2 in [42]). In Naomi$_{2D}$ ist es über die Ringdatenbank möglich, Ringblöcken präferierte Ausrichtungen zuzuweisen. Diese Ausrichtungen werden am Ende gesammelt und die Ausrichtung mit der höchsten Priorität wird auf das Gesamtdiagramm angewendet. Wenn es keine präferierten Ausrichtungen gibt, wird mithilfe der in Kap. 3.1.3 beschriebenen Methode die Ausrichtung der längsten Zickzack-Kette ermittelt und das Gesamtdiagramm so rotiert, das diese Kette horizontal liegt. Da in der Ringdatenbank auch das Steroidgerüst mit der

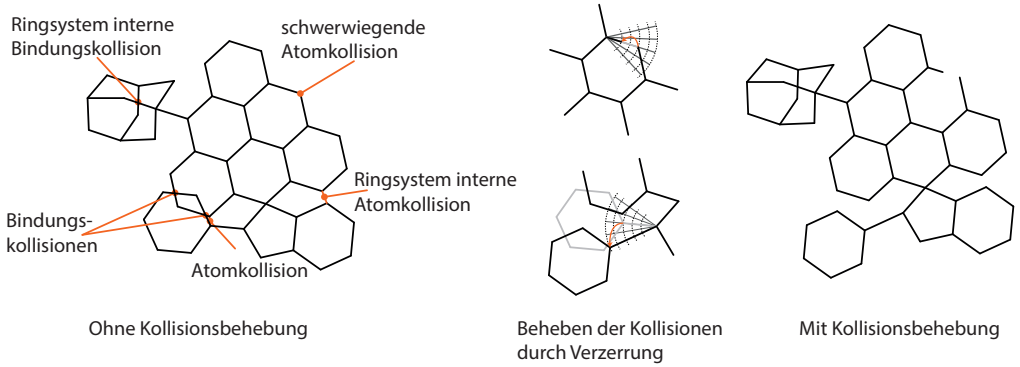


Abb. 3.15. Eine Reihe von unterschiedlichen Kollisionen werden in Molekülen entdeckt. Kollisionen, die sich nur innerhalb eines Ringsystems befinden, werden nicht behoben. Alle anderen Kollisionen werden möglichst vermieden, indem die Bindungen verbogen, verkürzt oder verlängert werden.

korrekten Ausrichtung vorgegeben ist, werden Steroide in den meisten Fällen richtig orientiert. Wenn das Steroid-Ringsystem gespiegelt wurde, wird das Gesamtdiagramm momentan nicht noch einmal im Gesamten gespiegelt, um die Spiegelung des Ringsystems erneut rückgängig zu machen. Auch existiert momentan keine Spezialbehandlung für Diagramme, die mehrere Steroide besitzen oder Ringsysteme, die sehr ähnlich zu Steroiden sind.

Verzerren

Nicht alle Moleküle lassen sich ohne Überschneidungen zeichnen. Gerade Ringsysteme enthalten zum Teil auch beabsichtigte Schnitte von Bindungen (s. Abb. 3.15). In Ketten oder zwischen unterschiedlichen Ringsystemen sind Überschneidungen jedoch niemals beabsichtigt. Dieser Nachbearbeitungsschritt versucht alle noch übriggebliebenen Kollisionen und Überschneidungen von Kettenatomen und Bindungen zu beheben.

Dazu wird zuerst einmal festgestellt, welche Atome Kollisionen besitzen. Kollisionen werden dabei immer als eine paarweise Beziehung zwischen zwei Atome aufgefasst. Wenn drei Atome nahe beieinander liegen, ergeben sich daher drei Kollisionspaare. Jedes Kollisionspaar des Diagramms wird der Reihe nach einzeln betrachtet. Der kürzeste Pfad über die Baumkanten der relativen Koordinaten zwischen den beiden Atomen der Kollision gibt die möglichen Kanten für die Kollisionsbehebung vor. Auf diesem Pfad sind prinzipiell alle Kettenbindungen veränderbar und alle Ringbindungen fixiert. Für jede veränderbare Kante auf dem Pfad werden nun spiralförmig verschiedene Längen und Winkelkombinationen durchprobiert. Nach jeder Änderung wird getestet, ob die Kollision behoben wurde oder nicht. Von allen möglichen Änderungen auf dem Kollisionspfad wird diejenige genommen, bei der die wenigsten Änderungen nötig sind. Dabei werden Änderungen nur an der Länge oder nur am Winkel höher gewichtet als

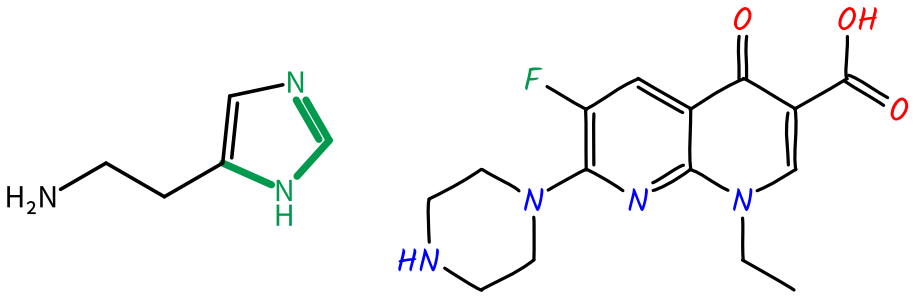


Abb. 3.16. Die Flexibilität des Molekülsatzes erlaubt es, die Parameter jedes einzelnen Elementes des Strukturdiagramms zu ändern. Linkes Molekül: Sowohl die Breite einer Bindung als auch ihre Farbe können angepasst werden. Dies wird in Mona verwendet, um die Treffer eines SMARTS -Ausdrucks hervorzuheben. Rechtes Molekül: Ein handgezeichnetes Aussehen lässt sich durch Austausch der Schriftart und zufällige Krümmung der Linien erzeugen.

Kombinationen der beiden. Gemäß den Regeln (GR-4.3.7, GR-4.3.8 in [42]) wird Längenänderungen der Vorzug vor Winkeländerungen gegeben.

3.2. Molekülsatz – Darstellen von Strukturdiagrammen

Die Koordinatenberechnung der Strukturdiagramme ist nur die halbe Strecke zum fertigen Bild. Der zweite Schritt besteht aus dem Molekülsatz, dem Erzeugen eines Bildes aus einem beliebigen Molekül mit 2D-Koordinaten. Die Herausforderungen hier sind eher ästhetischer als algorithmischer Natur. Es existieren hierbei keine klaren Antworten auf Fragen wie: Welche Schriftart ist zu bevorzugen? Welche Stärke wählt man für die Linien der Bindungen? Generell ist das Ziel immer ein möglichst gut verständliches Diagramm. Das oberste Designprinzip des hier vorgestellten Molekülsatzes ist daher Flexibilität, mit gut ausgewählten Vorgaben. Alles ist leicht austauschbar oder parametrisierbar: die Schriftart, die Stärke und Farbe der Bindungslinien. Dies ermöglicht z. B. das Hervorheben von einzelnen Bindungen durch Modifikation ihrer Stärke oder Farbe. Durch Änderungen, wie das Ersetzen von geraden Linien durch leicht gekrümmte, wird der ästhetische Eindruck des gesamten Diagramms stark verändert (s. Abb. 3.16).

Auch zusätzliche Elemente können im Diagramm platziert werden. PoseView [76] ist in diesem Zusammenhang sicherlich das komplexeste Beispiel.

3.2.1. Pixel und Vektoren

Wie in der Einleitung beschrieben (s. Kap. 1.1), hängt die Darstellung von Strukturdiagrammen am Computer immer auch von den vorhandenen technischen Möglichkeiten ab. Heutzutage sind eigentlich alle Ausgabegeräte rasterbasiert, das Bild wird also über ein Gitter von Pixeln dargestellt. Allerdings unterscheiden sich die Ausgabegeräte stark in der Granularität dieser Pixel: Ein Drucker hat immer noch eine höhere Auflösung

als ein Bildschirm, wobei der Trend hier ganz klar in die Richtung geht, Bildschirm und Druckerzeugnisse ununterscheidbar zu machen.

Damit der Molekülsatz diesen Anforderungen gerecht wird, muss eine Grafikbibliothek benutzt werden, die auflösungsunabhängige Zeichnungen erlaubt. Der verbreitetste Ansatz, vor allem bei 2D-Illustrationen, ist vektorbasiert. Hierbei werden alle Grafikelemente über nahezu beliebig geformte und gefüllte mathematisch beschriebene Pfade umgesetzt [39]. Die Umrechnung auf eine feste Auflösung übernimmt dann die verwendete Grafikbibliothek. Die von Naomi_{2D} unterstützte Grafikbibliothek ist dabei leicht austauschbar. Momentan werden sowohl cairo [60] als auch Qt [78] unterstützt. Der Nachteil dieser Methode ist die aufwendigere Rasterisierung für das Ausgabegerät, die momentan komplett auf der CPU berechnet werden muss. Ansätze, auch die GPU für diese Aufgabe zu benutzen ([50], [45]), sind noch nicht ausgereift, und es gibt vor allem keinen herstellerunabhängigen Standard.

Für einzelne Strukturdiagramme ist dieser zusätzliche Berechnungsschritt kein Problem, da die Diagramme typischerweise aus verhältnismäßig wenig und einfach zu berechnenden Pfadelementen bestehen. Wenn es aber wie bei Mona darum geht, die Strukturdiagramme von großen Mengen an Molekülen zu berechnen, wird dieses Problem akuter (s. Kap. 6.3.4). Allerdings geht die überwiegende Zeit der Berechnung in das Layout der Moleküle und nicht in den Molekülsatz.

Eine schnellere Alternative sind pixelbasierte Ansätze, bei denen alle Maße, wie die Linienstärke der zu zeichnenden Elemente exakt in Pixeln vorgegeben wird. Je nach Größe des Moleküls muss ein Strukturdiagramm dynamisch an den vorhandenen Platz angepasst werden. Das dafür benötigte stufenlose Skalieren ist aber mit pixelbasierten Ansätzen prinzipbedingt nicht möglich und daher kam dieser Ansatz nicht zur Anwendung.

Im restlichen Kapitel wird die grundlegende Funktionsweise des Molekülsatzes erläutert: Wie wird aus einem Molekül mit 2D-Koordinaten eine Grafik aus Pfadelementen?

3.2.2. Modell und Szenengraph

Der Molekülsatz ist in zwei Teile geteilt: In das Modell und den Szenengraphen.

Im Modell sind alle für die Darstellung wichtigen Eigenschaften von Atomen, Bindungen und Ringen gespeichert. Eine Aufzählung der verfügbaren Eigenschaften findet sich in Tab. 3.3. Diese Eigenschaften werden vom Szenengraphen benutzt, um das Aussehen zu bestimmen oder lokale Layoutentscheidungen zu treffen. Die Eigenschaft *Bezeichner-Richtungen* dient z. B. dazu festzulegen, ob das entsprechende Atomlabel vertikal oder horizontal gezeichnet werden soll.

Für cairo und Qt existieren zwei getrennte Szenengraphen, da die Qt API bereits einen 2D-Szenengraphen enthält, der nicht kompatibel zu dem für cairo entwickelten Szenengraphen ist. Dennoch ist die Funktionsweise im Prinzip dieselbe: der Szenengraph ist eine Datenstruktur, mit der sich eine grafische Szene hierarchisch in immer abstraktere Grafikelemente unterteilen lässt. Auf der untersten Ebene sind dies Elemente, um die Linien einer Bindung oder die einzelnen Buchstaben zu zeichnen. Die

Tab. 3.3. Die Eigenschaften des Molekülsatzmodelles lassen sich in chemische und grafische einteilen: Die chemischen Eigenschaften stammen aus den chemischen Daten des Moleküls, die grafischen kommen aus allgemeinen Stilvorgaben für Strukturdiagramme.

	Atom	Bindung	Ring
Chemisch			
	benachbarte Wasserstoffe Text des Bezeichners Text der Ladung	grafischer Bindungstyp in aromatischem Ring	
Grafisch			
	Sichtbarkeit	Sichtbarkeit Stärke der Linie	Stärke der Linie
	Farbe des Bezeichners Schriftart des Bezeichners	Farbe der Linie	
	Bezeichner-Richtungen		Mittelpunkt

mittlere Abstraktionsebene besteht aus Elementen, wie z. B. dem Bindungslayouter, der alle Bindungen des Strukturdiagramms zeichnet, oder dem Textlayouter, mit dem man die Bezeichner von beliebigen Atomen setzen kann. Auf der obersten Ebene liegt dann ein Grafikobjekt, das das komplette Strukturdiagramm zeichnet. Jede Hierarchieebene besitzt dabei ihr eigenes Koordinatensystem. Dies führt zu einer Gruppierung: Wenn man das Strukturdiagramm als Ganzes rotiert, werden alle in diesem Diagramm enthaltenen Elemente wie die Bindungen mitrotiert.

Diese Aufteilung hat auch Nachteile, die nicht offensichtlich sind: Wenn der Textlayout-Knoten nur sein eigenes Koordinatensystem kennt und nicht weiß, wie die Gesamtstruktur orientiert ist, ist es unmöglich zu bestimmen, in welche Richtung Text gesetzt werden muss, um am Ende waagerecht zu liegen. Dieses Problem ließ sich durch das Durchreichen eines Schwerkraft-Vektors durch die gesamte Hierarchie lösen.

Insgesamt macht es diese Struktur sehr einfach, einzelne Teile der Darstellung zu spezialisieren oder komplett neue Grafikelemente hinzuzufügen. Auch ist es leicht, mehrere Strukturdiagramme darzustellen und diese Diagramme über zusätzliche Grafikelemente in Beziehung zueinander zu setzen.

Am kompliziertesten wird der Molekülsatz dann, wenn Keilstrichformeln oder Texte gesetzt werden sollen. Daher werden diese beiden Bereiche in den nächsten beiden Abschnitten genauer beschrieben.

3.2.3. Keilstrichformeln

Um mehr Übersichtlichkeit zu erlangen, verbergen Strukturdiagramme einen Großteil der Dreidimensionalität der Moleküle. Die meisten Atomgeometrien kann ein geübter Chemiker aus dem Kontext der Atome schließen. Aber gerade bei Stereozentren ist es

nicht mehr eindeutig, wie sie orientiert sind (s. Kap. 2.4.3). In Strukturdiagrammen wird daher an geeigneten Bindungen eine Markierung gesetzt, um dessen 3D-Orientierung eindeutig zu machen. Als Markierung wird entweder ein Keil (die Bindung ragt aus der Ebene heraus) oder ein Strich (die Bindung verläuft in die Ebene hinein) benutzt (ST-0.3 in [41]). Die Markierung hängt dabei auch von der konkreten 2D-Konformation des Moleküls ab. Dies zeigt ein einfaches Gedankenexperiment: Wenn man ein Strukturdiagramm spiegelt, wechseln auch alle Stereonotationen. Denn eine Bindung, die zuvor ins Bild hineinragte, wird nach der Spiegelung aus dem Bild herausragen. Die Art der Stereomarkierung muss also von der 2D-Konformation des Moleküls abhängen.

Vor der Darstellung berechnet der Molekülsatz für jede Bindung des Moleküls, wie sie zu zeichnen ist. Neben den direkt von der Moleküldatenstruktur übernommenen Einfach-, Doppel- und Dreifachbindungen muss vom Molekülsatz für die Stereozentren entschieden werden, welche der angrenzenden Bindungen als Keile oder Striche gezeichnet werden sollen.

Bei einem Stereozentrum spannen die vier benachbarten Atome ein Tetraeder auf. Das Atom mit der höchsten Ordnung definiert dabei die Drehachse. Die restlichen drei Atome sind von dieser Drehachse aus gesehen ihrer Ordnung nach entweder im Uhrzeigersinn oder gegen den Uhrzeigersinn angeordnet. Diese Drehrichtung wird nicht verändert, wenn man das Tetraeder als Ganzes beliebig im Raum orientiert. Insbesondere wird sie also nicht verändert, wenn man die Eckpunkte des Tetraeders durch Drehungen des Gesamtkörpers aufeinander abbildet.

Alle möglichen Drehungen des Tetraeders, die die Eckpunkte aufeinander abbilden, ergeben die alternierende Gruppe A_4. A_4 ist eine Untergruppe der symmetrischen Gruppe S_4 und besteht aus genau den 12 Permutationen, die durch eine gerade Anzahl von Transpositionen erzeugt werden. Welche Drehungen finden sich in A_4? Zum einen lassen sich alle vier Flächen des Tetraeders um 120° und 240° drehen. Zum anderen gibt es noch drei Drehungen mit 180° um die Achse durch die Mittelpunkte von zwei jeweils gegenüberliegenden Kanten (s. Abb. 3.17). Alle Operationen von A_4 bestehen immer aus zwei aufeinanderfolgenden Vertauschungen der Eckpunkte des Tetraeders. Die Rotationen um die Seitenflächen sind dabei zwei Vertauschungen der drei unterschiedlichen Positionen i, j, k der Form $(i\ j\ k) = (i\ j)(j\ k)$. Die Drehungen um die Kantenmittelpunkte sind die zwei Vertauschungen $(i\ j)(k\ l)$ der vier unterschiedlichen Positionen i, j, k und l.

Die Idee bei der Bestimmung von Keilen und Strichen ist, mit den in A_4 vorhandenen Permutationen die vier benachbarten Atoms geeignet zu sortieren. Am Ende werden den vier sortierten Atomen dann entweder die grafischen Bindungstypen *Einfach, Einfach, Keil, Strich* oder *Einfach, Einfach, Strich, Keil* zugewiesen.

Der Algorithmus (s. Abb. 3.18) startet damit, jeder Bindung des Stereozentrums eine potenzielle Bindungsordnung zuzuweisen.

Fest Diese Bindung soll nach Möglichkeit als Einfachbindung gezeichnet werden. Dieser Fall gilt für alle Bindungen in Ringsystemen. Wenn zwei Stereozentren nur eine Bindung entfernt benachbart sind, gilt dieser Fall auch für diese Bindung. Die von den beiden Stereozentren zugewiesenen Bindungstypen würden sich an-

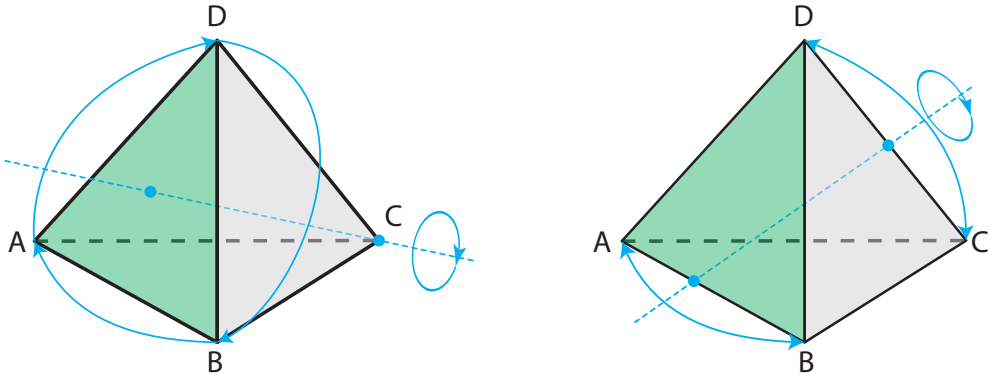


Abb. 3.17. Es gibt zwei verschiedene Rotationsachsen, um die Eckpunkte eines regelmäßigen Tetraeders aufeinander abzubilden: Die linke Achse rotiert die Eckpunkte einer Seitenfläche und die rechte Achse rotiert zwei gegenüberliegende Kanten.

sonsten gegenseitig überschreiben (ST-0.5 in [41]). Ein ästhetisches Kriterium ist, ob man dieses auch bei Stereozentren anwendet, die zwei Bindungen voneinander entfernt sind. Dies verkompliziert allerdings die Zuweisung der Bindungstypen, da aus einer Reihe von lokalen Entscheidungen ein Netz von Abhängigkeiten werden kann. Daher wird diese Möglichkeit momentan nicht angewendet.

Unsichtbar Diese Bindung ist normalerweise nicht sichtbar im Diagramm. Nur Wasserstoffbindungen an Stereozentren bekommen diesen Typen zugewiesen. Höchstens eine der vier Bindungen eines Stereozentrums kann eine Wasserstoffbindung sein, denn bei zwei Wasserstoffbindungen handelt es sich nicht mehr um ein Stereozentrum.

Variabel Die Darstellung dieser Bindung ist variabel. Sie kann also sowohl als Einfachbindung als auch als Keil oder Strich gezeichnet werden. Allen Bindungen, die nicht in die beiden anderen Kategorien fallen, wird dieser Typ zugewiesen.

Dabei kann der Typ *Unsichtbar* höchstens einmal vertreten sein, die Typen *Fest* und *Variabel* können jeweils bis zu viermal vorkommen. Diese vier Bindungen werden durch TETRAEDER-SORT so sortiert, dass alle festen Bindungen am Anfang stehen und die variablen danach. Eine unsichtbare Bindung wird dabei immer auf die letzte Position rotiert.

Das Vorgehen von TETRAEDER-SORT ist an Insertionsort angelehnt: Die initiale Reihenfolge entspricht den durch das Stereozentrum vorgegebenen Ordnungen. Die Bindung an der ersten Position ist also immer die Rotationsachse des Stereozentrums. Um die ersten beiden festen Bindungen auf die vorderen beiden Positionen zu rotieren, kommen Permutationen aus A_4 mit drei Elementen zum Einsatz. Um eine etwaige wei-

```
TETRAEDER-SORT(A)
    p = FIND-INDEX(Fixed, A[1..4])
    if p == 2
        APPLY-PERMUTATION((1 p 3), A)
    elseif 2 < p ≤ 4
        APPLY-PERMUTATION((1 p 2), A)

    p = FIND-INDEX(Fixed, A[2..4])
    if p == 3
        APPLY-PERMUTATION((2 p 4), A)
    elseif p == 4
        APPLY-PERMUTATION((2 p 3), A)

    if FIND-INDEX(Fixed, A[3..4]) == 4
        APPLY-PERMUTATION((1 2)(3 p), A)

    p = FIND-INDEX(Invisible, A[1..4])
    if 1 ≤ p ≤ 2
        APPLY-PERMUTATION((4 p 3), A)
```

Abb. 3.18. TETRAEDER-SORT sortiert die potenziellen Bindungstypen *Fest*, *Variabel* und *Unsichtbar* mithilfe von Vertauschungen, die die Stereochemie nicht ändern. Nach der Sortierung befinden sich alle festen Bindungen am Anfang und die potenziell vorhandene unsichtbare auf Position 3 oder 4 von A.

tere feste Bindung auf die dritte Position zu bringen, muss die vierelementige Vertauschungsoperation (1 2)(3 4) durchgeführt werden.

Nach diesen Schritten befinden sich auf jeden Fall alle festen Bindungen am Anfang. Wenn es jedoch keine oder nur eine feste Bindung gibt, kann sich eine unsichtbare Bindung auf den Positionen 0 oder 1 befinden. Dieses führt zu falschen Diagrammen: Es würde ein Stereozentrum bestehend aus 3 Bindungen gemalt, bei dem eine Bindung normal, eine als Keil und eine als Strich dargestellt ist (ST-1.2.2, ST-1.2.10 in [41]). Die unsichtbare Bindung muss daher zwingend an die letzte oder vorletzte Position geschoben werden. Dies geschieht durch die Permutation (i 3 4), wobei i die Position der unsichtbaren Bindung ist.

Nach der Sortierung sind die Bindungen in der richtigen Reihenfolge, damit den ersten beiden Bindungen normale Einfachbindungen und den letzten beiden hoch/runter oder runter/hoch zugewiesen werden kann. Welche der beiden Varianten gewählt wird, hängt von zwei Faktoren ab: Der erste Faktor ist, ob das Stereozentrum die R oder die S Variante ist. Der zweite Faktor ist die geometrische Anordnung der Atome im 2D-

Layout um das Stereozentrum herum: Sind die drei Bindungen mit der höchsten Ordnung geometrisch im Uhrzeigersinn oder gegen den Uhrzeigersinn um das Stereozentrum herum angeordnet?

Ein Sonderfall bleibt noch übrig: In Ringsystemen existieren häufig Stereozentren, an denen drei Ringbindungen und eine Wasserstoffbindung hängen. Dies führt zu drei festen Bindungen und einer variablen oder unsichtbaren. Hierbei werden alle Ringbindungen als Einzelbindung gezeichnet und der Substituent oder der eigentlich unsichtbare Wasserstoff wird mit einer Stereomarkierung versehen. Eine leichter verständliche Variante für überbrückte Ringe sind verdickte Ringbindungen im Ringsystem, dennoch ist die verwendete Darstellung eindeutig (ST-1.3.3 in [41]). TETRAEDER-SORT kann unverändert benutzt werden, um zu entscheiden, welche Ringbindung keilförmig dargestellt wird. Der Implementierungsaufwand der grafischen Darstellung ist nur deutlich aufwendiger als die momentan umgesetzte Methode, die Ring- und Kettenbindungen grafisch gleich behandelt.

3.2.4. Layout und Ausrichten der Atombezeichner

Eine immer wiederkehrende Aufgabe beim Visualisieren von Strukturdiagrammen ist das Setzen von Atombezeichnern. Der Molekülsatz enthält eine verallgemeinerte Methode, um Atombezeichner mit beliebigen Schriftarten korrekt auszurichten.

Die Atombezeichner in Strukturdiagrammen folgen denselben Regeln wie die Bezeichner in chemischen Summenformeln. Der Bezeichner besteht immer aus dem abgekürzten Namen des Atomelementes. In organischen Molekülen ist dies häufig nur ein Buchstabe, der Name kann aber bis zu zwei Buchstaben besitzen. In Strukturdiagrammen wird die Formalladung der Atome durch einen hochgestellten Multiplikator und ein + oder − dargestellt, z. B.: N^{2+}, N^{2-} oder N^{+}. Wenn ein Atom Wasserstoffbindungen besitzt, wird die Anzahl der Wasserstoffe an den Elementnamen angehängt: $N^{2+}H$ oder NH.

Im Unterschied zu chemischen Summenformeln werden die Atombezeichner nicht nur horizontal, sondern je nach Platz in Strukturdiagrammen auch vertikal gesetzt. Für die Bezeichner sind insgesamt vier Richtungen möglich: horizontal vorwärts, horizontal rückwärts, vertikal hoch und vertikal runter.

Um diese Layoutvorgaben möglichst allgemein umzusetzen, hantiert der Textlayouter mit einzelnen Textgruppen (s. Abb. 3.19). Jede Textgruppe besteht aus einem Text, an den wahlweise Zeichen hoch- oder tiefgestellt angehängt sind. Die Texte werden dabei nach den Abstandsvorgaben der Schriftart gesetzt, dabei wird auch die Unterschneidung (*kerning*) der Schriftart beachtet. Das Textlayout selber ist zeilenbasiert, d. h. es besteht aus beliebig vielen Textgruppen, die an beliebigen Stellen von Zeilenumbrüchen unterbrochen sind. Der Zeilenabstand ist dabei variabel und für die Atombezeichner abhängig von der Höhe des Buchstaben H gewählt. Um die Textgruppen korrekt zu platzieren, ist es möglich, beliebige Gruppen als Anker zu definieren. Die von diesen Gruppen aufgespannte Ankerbox dient dazu, das gesamte Textlayout zu platzieren. Einzelne Zeilen können außerdem an der Ankerbox ausgerichtet werden. So kann beim vertikalen Layout der Wasserstoff genau zum Element zentriert werden, das als Anker

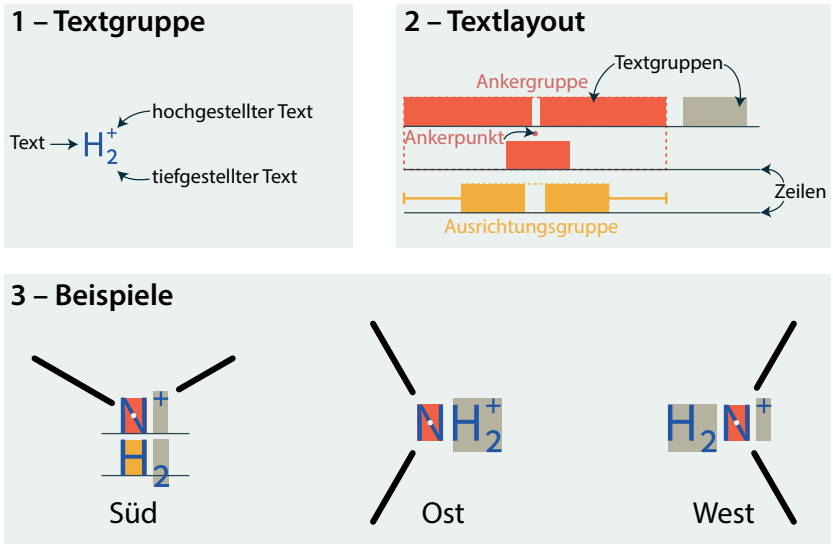


Abb. 3.19. Atombezeichner werden mithilfe von Textgruppen und Textlayouts gesetzt. Jedes Textlayout besteht dabei aus einzelnen Textgruppen, die zeilenweise nach den Abstandsregeln der verwendeten Schriftart angeordnet werden. Die Positionierung des gesamten Layouts erfolgt anhand des Mittelpunkts der Ankergruppe. Einzelne Zeilen können zu dieser Ankerbox ausgerichtet werden, indem sie in einer Ausrichtungsgruppe zusammengefasst sind. Im unteren Teil befinden sich drei konkrete Beispiele, wie Atombezeichner in Molekülen in einzelne Textgruppen aufgeteilt werden.

definiert wurde.

Welche der vier möglichen Richtungen am besten ist, ergibt sich aus den optimalen Bezeichner-Richtungen, die im Modell abgelegt sind (s. Tab. 3.3). Um die optimalen Bezeichner-Richtungen zu ermitteln, werden bei jedem Atom alle ausgehenden Bindungen betrachtet: Für die größten freien Bereiche wird jeweils der mittlere Winkel ermittelt. Der Textlayouter überprüft dann, mit welcher Layoutrichtung sich die minimale Abweichung zu einer der optimalen Richtungen ergibt. Die meisten Atombezeichner können in alle vier Richtungen gesetzt werden. Atome, die sich an den Enden von Ketten befinden, können dabei nur horizontal gezeichnet werden (GR-2.1.6, GR-2.1.7 in [42]).

3.3. Über das Ausrichten von Molekülen

Das Ziel der Molekülausrichtung besteht darin, Strukturdiagramme zweier unterschiedlicher Moleküle möglichst ähnlich zu zeichnen, um die Unterschiede und Gemeinsamkeiten der Moleküle sichtbar zu machen. Im Folgenden werden die beiden der Ausrichtung zugrundeliegenden Moleküle immer als M und M_V bezeichnet. M_V ist dabei die Vorlage für das Molekül M.

Die dahinterstehende Methode besteht aus zwei Teilen: Der erste Teil ist ein mög-

lichst effizient zu berechnendes Molekülmatching und der zweite Teil besteht dann aus der Adaptierung von Naomi$_{2D}$, um die Molekülvorlage in Phase 2 der Koordinatengenerierung (s. Kap. 3.1.3) zu beachten.

3.3.1. Matching

Die grundlegendste Frage beim Matching von Molekülen ist, welche Eigenschaften der Moleküle einander zugeordnet werden: Man kann z. B. direkt Atome oder Bindungen einander zuordnen oder die Abstraktionsebene erhöhen und stattdessen funktionelle Gruppen der Moleküle betrachten. Bindungen sind die markantesten grafischen Elemente in Strukturdiagrammen und da das Ziel ist, grafische Ähnlichkeiten in Strukturdiagrammen hervorzuheben, eignen sich Matchings auf Bindungsebene am besten.

Für den Matching-Algorithmus gibt es mehrere Möglichkeiten: *Maximum common subgraph (MCS) isomorphism* Algorithmen können benutzt werden, um den größten gemeinsamen Teilgraphen zweier Moleküle zu berechnen. Das MCS-Problem ist jedoch NP-vollständig, da man mit polynomiell vielen Aufrufen von MCS das Cliquenproblem lösen kann. Für viele organische Moleküle eignen sich exakte Verfahren trotzdem, wenn der MCS-Algorithmus auf diesen Spezialfall getrimmt ist und z. B. den geringen Knotengrad von organischen Molekülen geschickt ausnutzt. Allerdings gibt es immer wieder Ausnahmefälle, die die Laufzeit exponentiell steigen lassen.

Heuristische Methoden verzichten auf Genauigkeit, haben aber dafür eine Laufzeit, die nicht exponentiell mit der Anzahl der Atome steigt und damit zu keinen Überraschungen bei großen Datensätzen führt. Die am weitesten verbreitete Methode zum Erkennen von Ähnlichkeiten in Molekülen sind fingerprint-basierte Ansätze. Für das grafische Matching zum Ausrichten der Moleküle wurde daher eine auf ECFP basierte Methode gewählt. Diese ist von der Genauigkeit gut genug und führt vor allem mit einer quadratischen Laufzeit zu keinen Laufzeitausreißern bei komplizierten Molekülen.

3.3.2. Grafisches Matching

Das für das Ausrichten benötigte grafische Matching basiert auf ECFP Fingerprints [67], die typischerweise als Ähnlichkeitsmaß für Moleküle eingesetzt werden (s. Kap. 2.2.1).

Anstatt wie beim ECFP die Anzahl der Iterationen festzulegen, wird beim grafischen Matching der Fingerprint-Algorithmus iterativ und gleichzeitig auf den zwei zu vergleichenden Molekülen M und M_V laufen gelassen (s. Abb. 3.20).

Zu Beginn des Algorithmus wird jeder Bindung und jedem Atom aus M und M_V ein Hashwert zugewiesen. Als Hashfunktion wird immer die 32Bit-Version von CityHash verwendet [62]. Das Kriterium für Bindungen ist die Bindungsordnung: Diese kodiert in einer Zahl, ob die jeweilige Bindung eine Einfach-, Doppel- oder Dreifachbindung ist oder ob sie sich in einem delokalisierten System befindet. Für Atome werden zwei Kriterien kodiert: die Anzahl der benachbarten Schweratome und ob sich das Atom in einem Ring befindet. Der Verzicht auf das chemische Element bei den Atomkriterien sorgt für eine topologische anstatt einer chemischen Zuordnung der Moleküle.

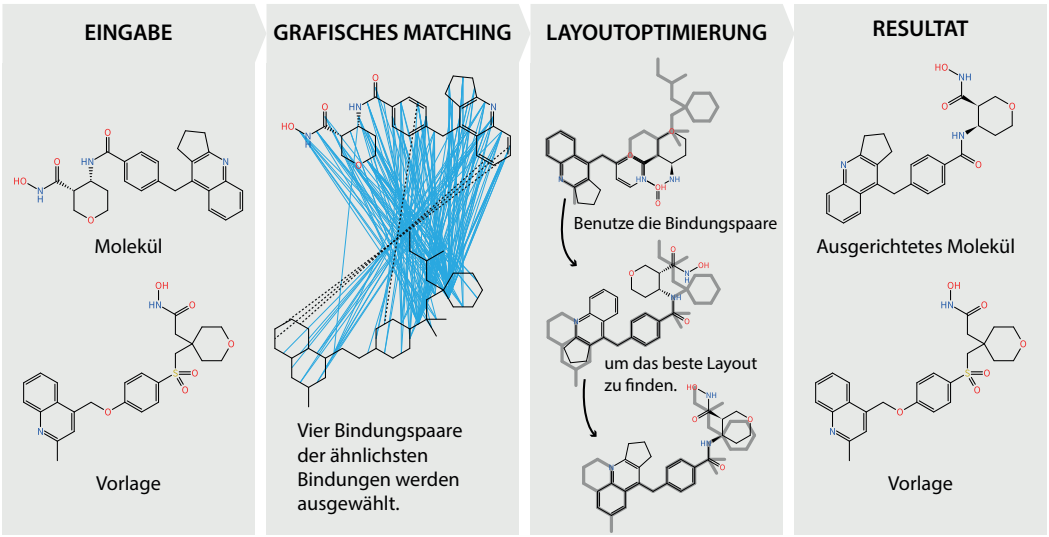


Abb. 3.20. Um das Strukturdiagramm eines Moleküls an einer Vorlage auszurichten, werden zunächst mit dem grafischem Matching die vier ähnlichsten Bindungspaare ausgewählt. Diese Paare und die Vorlage werden in der Layoutoptimierung benutzt, um das Strukturdiagrammlayout zu bewerten. Das Resultat der Optimierung ist ein an der Vorlage ausgerichtetes Strukturdiagramm des Moleküls.

In jeder Iteration werden ähnlich zum ECFP die Hashwerte aller Atome beider Moleküle M und M_V neu berechnet: Für jedes Atom a wird aus den Hashwerten der benachbarten Bindungen und angrenzenden Atomen ein neuer Hashwert durch Konkatenation der bestehenden Hashwerte und anschließendes Hashen berechnet. Atompaare für die Atome aus M werden erzeugt, indem jedem Atom a alle Atome $\hat{a}_i$ aus M_V mit demselben Hashwert zugeordnet werden. Sobald es kein Atom aus M_V mit demselben Hashwert wie a mehr gibt, werden Atompaare aus den in der Iteration zuvor zugeordneten Atomen erzeugt: Jedes Atompaar enthält Atom a und eines der in der Iteration davor zugeordneten Atome $\hat{a}_1, \ldots, \hat{a}_n$. Außerdem enthält jedes Atompaar noch eine Ähnlichkeitskennzahl, bei der die Nummer der aktuellen Iteration eingetragen wird. Je höher diese Zahl ist, desto größer ist der umliegende Bereich um die beiden Atome, der in beiden Molekülen ähnlich ist.

Auf exakt dieselbe Weise werden parallel Atompaare für die Atome aus M_V zu den Atomen aus M berechnet.

Das Verfahren läuft solange, bis allen Atomen aus M und M_V ein oder mehreren Atomen des jeweils anderen Moleküls zugeordnet wurde. Der maximale Suchradius ist erreicht, wenn so viele Iterationen durchgeführt wurden, wie maximal Atome in den Molekülen M und M_V existieren. Der Algorithmus kann dann abgebrochen werden. Dieses Abbruchkriterium wird dann erreicht, wenn die Moleküle M und M_V den Hashfunktionen nach exakt gleich sind.

Nach dem grafischen Matching werden aus den Atompaaren Bindungspaare erzeugt:

Zu jeder Bindung b aus M werden alle Bindungen $\hat{b}$ aus M_V gesucht, für die gilt, dass die entsprechenden Endatome einander zugeordnet wurden. In allen Atompaaren müssen sich entweder die Paare $(b_{\text{start}}, \hat{b}_{\text{start}})$ und $(b_{\text{end}}, \hat{b}_{\text{end}})$ oder die Paare $(b_{\text{start}}, \hat{b}_{\text{end}})$ und $(b_{\text{end}}, \hat{b}_{\text{start}})$ befinden. Im zweiten Fall (das Startatom von b wurde dem Endatom von $\hat{b}$ zugeordnet) wird am Bindungsmatching vermerkt, dass es über Kreuz angewendet werden muss. Als Ähnlichkeitskennzahl der Bindungspaare wird die Summe der zwei Ähnlichkeitskennzahlen der beiden ursprünglichen Atompaare genommen.

Um die Anzahl der entstandenen Bindungspaare zu reduzieren, wurden drei unterschiedliche Ansätze implementiert und miteinander verglichen (s. Kap. 4.5).

Direkt nimmt keine Reduktion der Bindungspaare vor.

Maximales Matching berechnet ein maximales bipartites Matching auf den Bindungspaaren und entfernt alle Paare, die nicht zum Matching gehören. Dadurch entscheidet man sich für die Bindungspaare, die zusammengenommen einen möglichst großen Bereich beider Moleküle abdecken und dabei die höchsten Ähnlichkeitskennzahlen besitzen.

Ohne Duplikate entfernt alle Bindungspaare, sodass keine Bindungen der Moleküle M und M_V mehr doppelt in den Bindungspaaren auftauchen. Dazu werden die folgenden vier Schritte durchgeführt:

1. Sortiere alle Bindungspaare nach der Bindungs ID aus Molekül M.
2. Tritt dieselbe Bindung b aus M mehrfach hintereinander auf, behalte nur das erste Bindungspaar, das b enthält.
3. Sortiere alle verbliebenen Bindungspaare nach der Bindungs ID aus Molekül M_V.
4. Tritt dieselbe Bindung b aus M_V mehrfach hintereinander auf, behalte nur das erste Bindungspaar, das b enthält.

Aus den verbleibenden Bindungspaaren müssen nun die k Bindungspaare ausgewählt werden, die als Grundlage der Bewertungsfunktionen bei der Layoutoptimierung des Strukturdiagramms dienen. Dafür ist es nötig, die Bindungspaare zu wählen, die in beiden Molekülen möglichst ähnliche Umgebungen besitzen. Nachdem die Bindungspaare absteigend nach ihrer Ähnlichkeitskennzahl sortiert wurden, stehen Bindungspaare am Anfang, bei deren Atomen die größte Umgebung um die Atome herum während des grafischen Matchings gleich war. Bindungspaare mit jeweils gleicher Ähnlichkeitszahl werden vor der Wahl in eine zufällige Reihenfolge gebracht. Dies sorgt dafür, dass die ausgewählten Bindungen nicht von der Eingabereihenfolge der Bindungen und Atome im Molekül abhängen. Die ersten k Bindungen aus den sortierten Bindungspaaren werden als Eingabe für die Layoutoptimierung benutzt.

3.3.3. Vorlagenbasierte Berechnung von 2D-Molekülkoordinaten

Um 2D-Koordinaten anhand einer Vorlage zu berechnen, werden dieselben Methoden wie zur normalen 2D-Koordinatenberechnung verwendet (s. Kap. 3.1). Die einzige not-

wendige Adaption findet sich in Phase 2. Anstatt die Güte des Layouts alleine an der Anzahl der Kollisionen und gestreckten Ketten festzumachen, wird hier zusätzlich bewertet, wie gut die Koordinaten zur Vorlage passen. Grundsätzlich wird die heuristische Suche in der Art geändert, dass nicht pro Schritt eine 2D-Konformation, sondern k 2D-Konformationen bewertet werden. Dazu werden alle k Bindungspaare aus dem grafischen Matching auf die aktuelle 2D-Konformation als affine Transformation angewendet. D. h. die aktuelle Konformation wird so verschoben, dass sie möglichst deckungsgleich auf der Konformation der Vorlage liegt. Diese transformierten Konformationen werden mit einer Reihe von speziellen Bewertungskriterien bewertet (s. Tab. 3.4). Die kollisionsrelevanten Kriterien haben weiterhin die höchste Priorität.

Tab. 3.4. In der Tabelle sind alle Bewertungskriterien für das Ausrichten von Molekülen aufgeführt. Die kursiv markierten Elemente wurden unverändert vom 2D-Moleküllayout übernommen. Die nicht kursiven Kriterien bewerten die Güte der Ausrichtung verglichen zur Vorlage. Die Laufzeiten aller Kriterien sind in der bei Graphen gebräuchlichen Form geschrieben. Auf den Molekülgraphen angewandt ist $|E|$ die Anzahl der Bindungen und $|V|$ die Anzahl der Atome. Die letzte Spalte enthält die Richtung, in die jedes Kriterium optimiert wird: Das Kriterium ist optimal, wenn der Wert entweder minimal oder maximal ist. Die Kriterien sind nach ihrer Priorität in der Bewertungshierarchie von oben nach unten sortiert. Die Vermeidung von Kollisionen hat damit auch beim Ausrichten die höchste Priorität.

Kriterium	Beschreibung	Laufzeit	Optimum
Gitterbasierte Atom Kollision	*Ungefähre Anzahl der Atome, die zu nahe beieinander liegen*	$O(\|V\|)$	*minimal*
Ununterscheidbare Kollisionen	*Anzahl der nicht unterscheidbaren Atome und Bindungen*	$O(\|E\| \log \|E\|)$	*minimal*
Kollisionen	*Anzahl der Bindungen, die zu nahe an anderen Bindungen liegen oder sich überkreuzen*	$O(\|E\| \log \|E\|)$	*minimal*
Überdeckte Bindungen	Anzahl der von Molekül M überdeckten Bindungen der Vorlage M_V	$O(\|E\|^2)$	maximal
Überdeckte Ringsysteme	Anzahl der komplett überdeckten Ringsysteme der Vorlage	$O(\|V\|^2)$	maximal
Passende Atome	Anzahl der Atome, die auf einem Atom der Vorlage mit demselbem Element liegen	$O(\|V\|^2)$	maximal
Überdeckte Ringe	Anzahl der Ringe, die auf einem gleichen Ring der Vorlage liegen	$O(\|V\|^2)$	maximal
Überdeckte Ketten	Anzahl der Kettenatomen, die auf Kettenatomen der Vorlage liegen	$O(\|V\|^2)$	maximal
Atomdistanz	Summe aller minimalen Atomdistanzen zwischen Molekül und Vorlage	$O(\|V\|^2)$	minimal
Länge der Mittelsegmente	*Gesamtlänge aller Zickzack-Ketten im Layout*	$O(\|E\| \log \|E\|)$	*maximal*
Anzahl der Mittelsegmente	*Anzahl der unterschiedlichen Zickzack-Ketten im Layout*	$O(\|E\| \log \|E\|)$	*minimal*

4. Validierung von Strukturdiagrammen

Um den Strukturdiagramm-Layouter zu validieren, wurden sowohl manuelle als auch automatische Verfahren benutzt. Der Vorteil manueller Verfahren ist, dass man Unstimmigkeiten findet, die man nicht objektiv bewerten kann. Automatische Verfahren können dagegen große Datenmengen verarbeiten.

Das Erzeugen von Moleküldarstellungen ist bei allen in dieser Arbeit verglichenen Programmen in zwei getrennte Schritte geteilt: Zuerst werden 2D-Koordinaten erzeugt, die im zweiten Schritt (dem Molekülsatz) in ein Bild umgewandelt werden. Für eine automatische Auswertung eignen sich die 2D-Koordinaten am besten, während die Bilder besser für eine stichprobenartige Validierung durch Menschen geeignet sind. Die automatische Auswertung testet einen Großteil der Komplexität, vor allem die Qualität der Methode aber auch deren Robustheit.

4.1. Automatische Validierung

Für die automatische Validierung interessieren uns vor allem die folgenden Qualitätsmaße:

1. Wie viele Kollisionen von Atomen und Bindungen befinden sich im Diagramm?
2. Welche Bindungswinkel im Diagramm weichen von regelmäßigen Winkeln ab?
3. Wie viele Bindungen weichen von der optimalen Länge ab?

Für die automatische Validierung werden diese drei Maße wie folgt angewendet:

Kollisionen In Strukturdiagrammen gibt es unterschiedliche Arten von Kollisionen. Für die Bewertung werden die Kollisionen in drei unterschiedliche Klassen eingeteilt. Diese Klassen ergeben sich direkt aus den im Methodenteil definierten vier Kollisionskategorien A, B, C und D (s. Tab. 3.1).

Klasse I Am unvorteilhaftesten ist es, wenn man beim Betrachten der Diagramme chemisch falsche Informationen bekommt. Das tritt auf, wenn Bindungen oder Atome exakt übereinanderliegen. Hierbei spielt es keine Rolle, ob sie sich in einem Ringsystem oder einer Kette des Moleküls befinden. (Kategorie A)

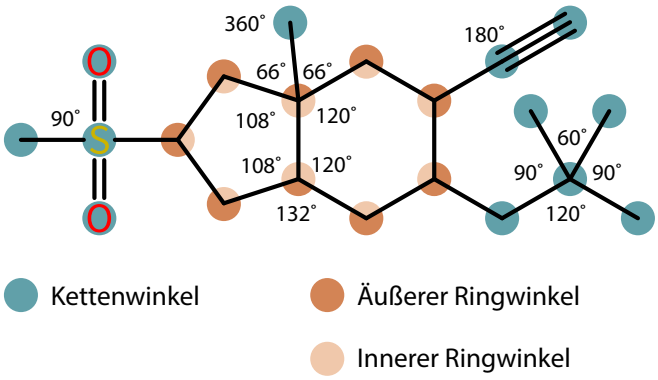


Abb. 4.1. In der Abbildung sieht man die Einteilung von Winkeln an Atomen in unterschiedlichen Kategorien. Typische Winkel sind exemplarisch dargestellt.

Klasse II Wenn Atome sehr nah zusammen liegen oder sich Bindungen überkreuzen, wird dadurch die Lesbarkeit des Diagramms negativ beeinflusst. (Kategorien B, C)

Klasse III Ausgenommen hiervon sind Bindungsüberkreuzungen und Kollisionen, die sich innerhalb eines Ringsystems des Moleküls befinden. Bei vielen Ringsystemen ist eine perspektivische Darstellung mit Linienüberkreuzungen übersichtlicher, obwohl das Ringsystem an sich planar ist und überkreuzungsfrei gezeichnet werden könnte. Cuban ist hierfür das bekannteste Beispiel. Außerdem gibt es Ringsysteme, die nicht planar sind, und sich daher auch nicht überkreuzungsfrei im zweidimensionalen zeichnen lassen. (Kategorie D)

Winkel Die Vorstellung, Strukturdiagramme befinden sich komplett auf den Kanten eines Hexagonalgitters, ist zumindest für die Ketten von Molekülen nahezu richtig. Verkompliziert wird die Sache allerdings durch Ringsysteme, die typischerweise aus zusammengesetzten regelmäßigen Polygonen bestehen. Anstatt alle Atome des Moleküls mit denselben Regeln zu testen, werden Winkel um Atome in drei Kategorien K, R_I, R_A eingeteilt (s. Abb. 4.1). Jede Kategorie erhält unterschiedlich strenge Winkelkriterien. Der Kategorie K werden alle Kettenatome zugeordnet, d. h. genau die Atome, die nicht in einem Ringsystem liegen. Winkel der Kategorie R_I und R_A befinden sich in Ringsystemen. R_I enthält dabei die Innenwinkel und R_A die Außenwinkel bezüglich des Ringsystems.

Für die Winkel der Kategorie K gelten die strengsten Kriterien: Es sollten in validen Strukturdiagrammen nur die Winkel 60°, 90°, 120°, 180°, 240° und 360° vorkommen. Die Winkel bei Ringsystemen sind im Prinzip willkürlich, gerade bei überbrückten Ringsystemen ist alles erlaubt, dass den Konventionen entspricht oder ordentlich genug aussieht. Am häufigsten enthalten übliche Moleküle jedoch nur Ringsysteme, die aus Kombinationen regelmäßiger Polygone mit maximal 9 Eckpunkten bestehen

(s. Tab. 4.1). Für die inneren Ringwinkel R_I wurden daher die Innenwinkel regelmäßiger Polygone erlaubt, die sich mit der Formel

$$I(n) = 180 - \frac{360}{n}$$

und der Anzahl der Eckpunkte $n \in \{3, \ldots, 9\}$ ergeben. Damit alle Ringformen gültig sind, die sich auf einem hexagonalen Gitter befinden, fehlt abschließend noch der Winkel 240°. Das Kriterium für Kategorie R_I besteht damit aus 8 gültigen Winkeln. Die direkten äußeren Winkel ergeben sich durch

$$A_1(n) = 360 - I(n) = 180 + \frac{360}{n}$$

Außerdem sind Außenwinkel erlaubt, die sich aus der Kombination zweier regelmäßiger Polygone mit entsprechend n und m Eckpunkten ergeben

$$A_2(n, m) = 360 - (I(n) + I(m)) \text{ mit } n, m \in \{3, \ldots, 9\}$$

Jeder dieser Außenwinkel kann durch die Substituenten des Ringsystems in zwei, drei oder vier gleichgroße Winkel geteilt werden. All diese Außenwinkel werden in einer Liste gesammelt und Duplikate wurden entfernt. Damit ergeben sich 109 unterschiedliche Winkel für das Kriterium von Kategorie R_A. Da gerade die Winkel bei Ringsystemen sehr variabel sind, werden für die Bewertung die Winkel für Ketten und die für Ringe getrennt betrachtet. Die Anzahl der Kettenatome, die nicht den strikten Vorgaben der Kategorie K entsprechen, sind dabei aussagekräftiger als die Anzahl der Ringatome, die nicht den Kategorien R_A oder R_I entsprechen. Der Grund hierfür ist, dass die Kriterien für R_I und R_A nicht alle denkbaren validen Winkel enthalten und somit falsch-negative Fälle möglich sind. Alle Winkel wurden mit zwei unterschiedlichen Toleranzen überprüft: Die erste erlaubt 0,1° Abweichung vom optimalen Winkel und ist damit nicht sichtbar. Die zweite Toleranz beträgt 1° und ist kaum sichtbar.

Bindungslänge Deutlich einfacher als die Betrachtung der Winkel ist die Betrachtung der Bindungslängen in Strukturdiagrammen. Für ein möglichst übersichtliches Layout, ist es wichtig, dass alle Bindungen die gleiche Länge haben. Bindungen mit mehr als 1 % bzw. mit großer Toleranz 5 % Abweichung von der Standardbindungslänge werden als fehlerhaft betrachtet. Wiederum ist die Variabilität in Ringsystemen höher als in Ketten. In Ringsystemen entsprechen dabei auch einige verzerrte Bindungen den Konventionen. Diese falsch-negativen Fälle spielen allerdings keine Rolle beim Vergleich mehrerer Methoden. Man muss sich nur im Klaren sein, dass man nicht alle falschen Bindungslängen in Ringsystemen eliminieren kann. Da Kettenbindungen keine Variabilität haben und immer gleich lang sein sollten, wird die Anzahl der falschen Ketten- und Ringbindungen getrennt gezählt und falsche Kettenbindungen werden als stärkere Indizien für falsche Strukturdiagramme gewichtet.

Ästhetische Kriterien Eine Reihe von Kriterien vor allem ästhetische lassen sich nicht durch automatische Tests überprüfen. Hier fanden zwei manuelle Validierungen statt, in denen Marcus Gastreich von der BiosolveIT sich einen Satz von Diagrammen aus der Perspektive eines Chemikers angesehen hat und ein ausführliches Feedback lieferte. Weiterhin hat der Autor dieser Arbeit sowohl alle Diagramme aus dem PubChem-Testdatensatz mit nicht unterscheidbaren Bindungen manuell überprüft als auch automatisch alle Moleküle ermittelt, die Ringsysteme enthielten, bei denen das Kraftfeld als Kompromiss eingesetzt wurde. Vorlagen für die häufigsten Fälle wurden danach der Ringdatenbank hinzugefügt.

Zusammengefasst ergaben sich daraus die folgenden Qualitätsmerkmale, die zwar wichtig sind, jedoch nicht automatisiert überprüft werden konnten:

- Welche Vorlagen für ein Ringsystem zur Verfügung stehen, hängt zu einem großen Teil von der Größe und der Qualität der Ringdatenbank ab, die im entsprechenden Programm eingebaut ist. Unterschiede in der Qualität der Ringsystemdarstellung der verschiedenen Programme lassen sich häufig auf das Vorhandensein einer Vorlage zurückführen. Diese Unterschiede lassen sich durch Hinzufügen der Vorlage zur Ringdatenbank beheben. An der Verwendung einer Ringdatenbank führt kein Weg vorbei, da viele kleine überbrückte Ringsysteme spezielle den Konventionen der Chemikern entsprechende Darstellungen haben, die sich nicht automatisch berechnen lassen. Für größere unbekannte Ringsysteme ist es dagegen wiederum interessant, unterschiedlichen Layoutalgorithmen zu vergleichen.

- Die Ausrichtung der Diagramme spielt zwar für die chemische Bedeutung keine Rolle, dem Chemiker erleichtert sie allerdings die Interpretation der Strukturen. Hierbei gilt, dass im Allgemeinen Moleküle horizontal ausgerichtet sein sollten, und dass das primäre Ringsystem des Moleküls sich rechts unten befinden sollte (GR-3.1, GR-3.2 in [42]). Man kann alle Algorithmen um einen Nachbearbeitungsschritt erweitern, der eine korrekte Ausrichtung vornimmt. Im Layoutalgorithmus ist es lediglich wichtig, möglichst wenige sich widersprechende Ausrichtungen zu erzeugen. Keines der getesteten Programme beachtet momentan die korrekte Ausrichtung multipler Ringsysteme.

- Ästhetische Merkmale des Molekülsatzes, wie die benutzte Schriftart oder die Stärke der Bindungslinien spielen eine wichtige Rolle bei der Verständlichkeit des Diagramms (GR-0.3, GR-0.4 in [42]). Für den Vergleich der Layoutalgorithmen sind diese Merkmale jedoch unerheblich, da sie ausschließlich durch den Molekülsatz bestimmt werden.

Mit den automatisch bestimmbaren Qualitätskriterien wurden drei verschiedene Experimente durchgeführt. Das erste Experiment diente dazu, die Robustheit der Methode zu validieren, das zweite Experiment vergleicht die Qualität der Methode mit anderen Programmen und das letzte Experiment vergleicht die Geschwindigkeit von Naomi_{2D} mit anderen Programmen.

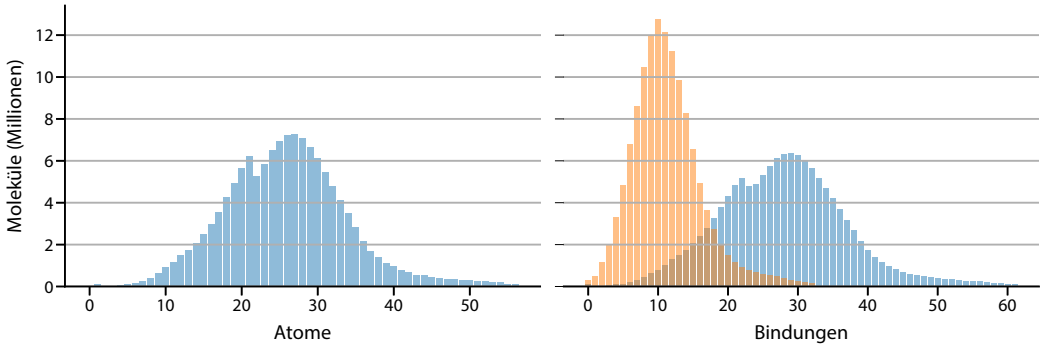


Abb. 4.2. Die Histogramme zeigen die Verteilung der Anzahl der Atome und Bindungen der PubChem-Datenbank. Im linken Diagramm ist die Anzahl der Moleküle im PubChem-Datensatz aufgetragen, die die entsprechende Anzahl an Atomen besitzen. Im rechten Diagramm sieht man in blau die Anzahl der Moleküle, die die entsprechende Anzahl an Bindungen besitzen. In Orange ist die Verteilung für die Anzahl der Bindungen ausschließlich aus Ketten dargestellt.

4.2. Experiment 1: Robustheit

Das Ziel des ersten Experiments ist die Validierung der Robustheit der Methode. Für eine möglichst realistische und große Auswahl an Eingabemolekülen muss die Methode valide 2D-Koordinaten für alle Moleküle berechnen.

4.2.1. Datensatz

Die PubChem-Datenbank [46] ist die zurzeit größte öffentlich verfügbare Sammlung kleiner Moleküle. Die Datenbank besteht aus den drei Teilen BioAssay, Compound und Substance. In BioAssay sind Assay Daten gesammelt, es handelt sich dabei um Ergebnisse von experimentellen Screenings. PubChem Substances enthält alle Moleküle, die von Drittanbietern über entsprechende Interfaces bereitgestellt wurden. In diesem Datensatz befinden sich auch Duplikate und Molekülgraphengerüste, die in der Natur nicht möglich sind. In Compounds wurde diese Datenmenge durch automatische Skripte bereinigt und doppelte Moleküle entfernt.

Um die Robustheit des Strukturdiagramm-Layouters zu validieren, bietet sich daher vor allem der PubChem Substances Datensatz an, da sich in diesem alle Arten von gebräuchlichen Molekülen und auch ungewöhnliche Molekülgerüste befinden. Zum Downloadzeitpunkt (Mitte 2014) enthielt der PubChem Substances Datensatz 147 Mio. Strukturen.

Die Moleküle der PubChem bestehen vor allem aus kleinen Molekülen mit einer durchschnittlichen Atomzahl von 26 und einer maximalen Anzahl von 992 (s. Abb. 4.2).

Tab. 4.1. Die Tabelle enthält die Resultate von Experiment 1. Der erste Teil der Tabelle enthält die Anzahl der Moleküle der PubChem Substance Datenbank, die fehlerfrei eingelesen werden konnten. Der zweite Teil schlüsselt auf, wie viele der fehlerfrei eingelesenen Moleküle möglichst kollisionsfrei gezeichnet werden konnten und der dritte Teil gibt an, welche Ringsystem-Algorithmen für die einzelnen Moleküle verwendet wurden (s. Abb. 3.3).

Einlesen	Moleküle	Moleküle(%)
fehlerfrei	133 822 889	90,98
leer	11 246 735	7,65
Syntaxfehler	1 250 987	0,85
Metallfehler	504 088	0,34
Valenzfehler	258 900	0,18
Kollisionen	**Moleküle**	**Moleküle(%)**
kollisionsfrei	130 616 368	97,60
Klasse III	2 023 779	1,51
Klasse II	1 180 981	0,88
Klasse I	1761	0,0013
Ringsysteme	**Moleküle**	**Moleküle(%)**
keine Ringsysteme	3 403 610	2,54
trigonometrisch	125 497 601	93,78
Datenbank	3 631 871	2,71
Kraftfeld	682 868	0,51
Makrozyklen	602 619	0,45
Kompromiss	4320	0,0032

4.2.2. Durchführung

Der PubChem Datensatz liegt in einzelne Dateien zerteilt vor. Für alle Teile wurden 2D-Koordinaten parallel auf einem Cluster berechnet und pro Datei Statistiken erstellt und gemeinsam ausgewertet.

4.2.3. Ergebnis

Von den 147 Mio. Molekülen des Datensatzes konnte Naomi 134 Mio. fehlerfrei einlesen und 13 Mio. Moleküle nicht einlesen (s. Tab. 4.1). Da einmal vergebene PubChem Substance IDs kein zweites Mal benutzt werden können, dienen Moleküle ohne Atomeinträge dazu, verschobene, ersetzte oder zurückgezogene Moleküle zu kennzeichnen. Dies traf auf den überwiegenden Teil (11 Mio. Moleküle, 84,81 %) der nicht einlesbaren Moleküle zu. Wirkliche Fehler während der Initialisierung der Moleküle gab es daher nur bei 2 Mio. Molekülen (1,37 % des Gesamtdatensatzes). Bei diesen Molekülen schlug entweder die syntaktische Analyse fehl (1,25 Mio., 62,12 % der Initialisierungsfehlschlä-

ge), das Molekül enthielt kovalent gebundene Metalle (0,50 Mio., 25,03 % der Initialisierungsfehlschläge) oder es wurden keine passenden Valenzzustände gefunden (0,25 Mio., 12,85 % der Initialisierungsfehlschläge).

Durch die Größe der Datenbank konnten während des Experimentes einige seltene Fehler identifiziert und behoben werden:

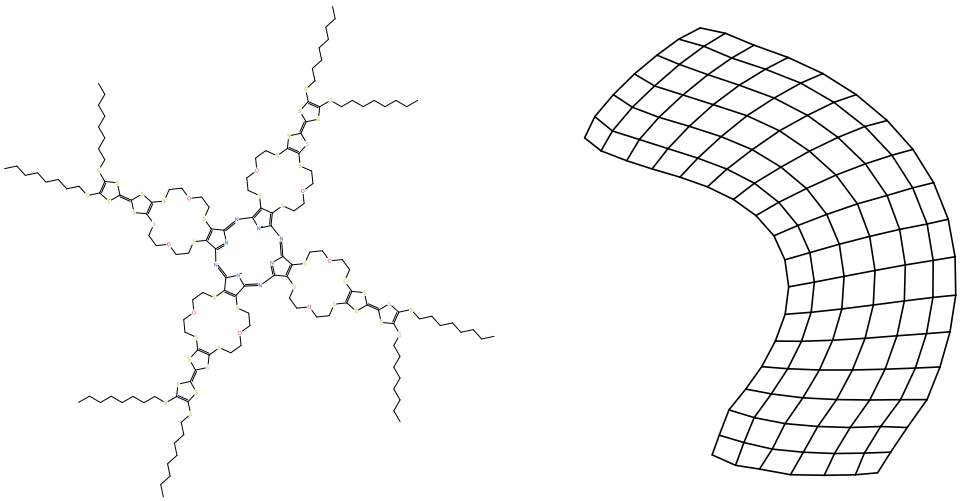
Einige Moleküle in der PubChem sind in Wirklichkeit Tabellen (s. Abb. 4.3), die mit einem Molekülzeichenprogramm erstellt wurden. Mit hoher Wahrscheinlichkeit kommen diese Moleküle nicht in der Natur vor und sind auch nicht synthetisierbar. Typischerweise zeichnen sich diese Moleküle durch eine sehr große Anzahl an relevanten Zyklen aus. Um auch solche Moleküle einlesen zu können, mussten Vorkehrungen in Naomi getroffen werden, nicht alle (potenziell exponentiell vielen) relevanten Zyklen zu berechnen. Außerdem muss im Layout- und Zeichenalgorithmus an allen Stellen, bei denen Entscheidungen anhand der Ringe getroffen werden, darauf geachtet werden, dass diese Entscheidung nicht von der Anzahl der relevanten Zyklen abhängt und dass stattdessen die Ringfamilien benutzt werden.

Ein weiterer Fall von Langläufern ergab sich ebenfalls im Kontext der Ringsysteme: Beim Kombinieren und Bewerten der einzelnen Teile eines Ringsystems im Ringsystem-Layouter kann es prinzipiell zu exponentiell vielen Kombinationsmöglichkeiten kommen. In diesem Fall wird die wahrscheinlichste Möglichkeit und eine begrenzte zufällige Auswahl an Alternativen getestet.

Der häufigste und schwerwiegendste Fall der Klasse I Kollisionen waren Helicene. Diese bestehen aus kondensierten Benzolringen, die zusammengesetzt eine Spirale ergeben. Das Ringsystemlayout wurde umgeschrieben, sodass auch verzerrte einfache Ringe in Betracht gezogen werden (s. Kap. 3.1.2). Diese Änderung führt dazu, dass bei Helicenen erkennbar wird, aus wie vielen Ringen sie bestehen.

Als Kollisionsrate für Naomi_{2D} ergibt sich insgesamt: 97,6 % (130 Mio.) der Moleküle enthalten keine Kollisionen. Ausschließlich Kollisionen in Ringsystemen (Klasse III) enthalten weitere 1,5 % (2 Mio.) der Moleküle. Die restlichen 1,2 Mio. Moleküle enthalten schwerwiegendere Kollisionen der Klasse I und II. Klasse II macht dabei den Hauptteil mit 0,8 % (1,2 Mio.) der Moleküle aus. Klasse I Kollisionen sind sehr selten: 1505 Moleküle des gesamten Datensatzes enthalten überlappende Atome und 256 Moleküle enthalten überlappende Bindungen. Im Anhang in Abb. A.5 und Abb. A.6 befindet sich eine Auswahl aller Klasse I Kollisionen.

Die meisten Ringsysteme der Moleküle sind sehr einfach: Keinerlei Ringsysteme enthalten 3,4 Mio. Moleküle (2,5 % des Gesamtdatensatzes). Der absolut überwiegende Teil des Datensatzes mit 125 Mio. Molekülen (93,78 % des Gesamtdatensatzes) enthält ausschließlich einfache Ringe der Größe 3–9 oder Ringsysteme die ausschließlich aus einfachen Ringen bestehen. Weitere 3,6 Mio. Moleküle (2,71 %) des Datensatzes bestehen aus einfachen Ringen oder überbrückten Ringen, für die eine Vorlage in der Ringdatenbank gefunden wurde. Jeweils etwa ein halbes Prozent der Moleküle enthielt Ringe, die entweder vom Kraftfeld oder dem Makrozyklus-Algorithmus berechnet werden mussten. Bei 4320 Molekülen des Datensatzes musste auf die Kompromissvariante der Ringe zurückgegriffen werden: Dies umfasst zum einen alle Helicene, bei denen die verzerrten Sechsringe benutzt werden. Zum anderen erhalten alle Makrozyklen auch noch einen



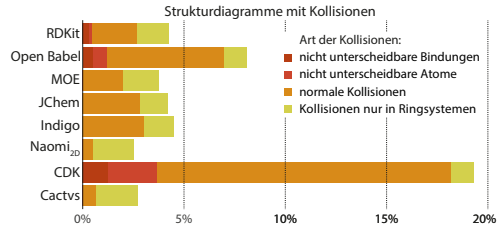
PubChem SID 136364209 PubChem SID 139437363

Abb. 4.3. Beim Molekül auf der linken Seite ergeben sich exponentiell viele Möglichkeiten beim Zusammensetzen des zentralen Ringsystems aus den einzelnen Ringblöcken. Für jeden der vier angehängten 18-Ringe existieren mehrere Möglichkeiten in der Ringdatenbank. Auf der rechten Seite ist eines der zahlreichen Beispiele abgebildet, bei dem ein Ringsystem hauptsächlich aus Rechtecken besteht. Die Vermutung liegt hier nahe, dass ein Molekülzeichenprogramm benutzt wurde, um eine Tabelle zu erstellen.

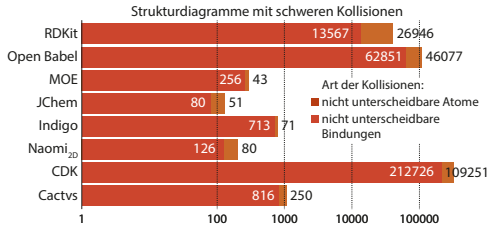
trigonometrisch berechneten Ring derselben Größe als Kompromiss.

4.2.4. Folgerung

Der in dieser Arbeit vorgestellte Algorithmus zur Berechnung von 2D-Molekülkoordinaten ist in der Lage, für die absolut überwiegende Menge der von Chemikern benutzten Moleküle kollisionsfreie Layouts zu erstellen. Die Resultate des Experimentes 1 geben Aufschluss darüber, an welchen Stellen weitere Verbesserungen erfolgen können: Moleküle mit kovalent gebundenen Metallen können nicht eingelesen werden und dadurch auch nicht gezeichnet werden. Die Unterstützung von kovalent gebundenen Metallen würde jedoch sowohl große Änderung an der Naomi als auch an der Naomi_{2D}-Bibliothek erfordern, da sich metallische Bindungen grundsätzlich anders verhalten als kovalente. Eine weitere naheliegende Verbesserung ist die Verringerung von schwerwiegenden Kollisionen. Die häufigsten Ursachen für übereinanderliegende Atome oder Bindungen sind im Ringsystemlayoutalgorithmus begründet. Bei diesem ist es zum einen die Behandlung von Makrozyklen, die überarbeitet werden muss, zum anderen sind es Spiroverbindungen, die für viele Kollisionen verantwortlich sind. Ein ähnlicher Ansatz wie beim Verzerren von Kettenbindungen ist auch am Spiroatom nötig, um überschneidungsfreie Layouts zu generieren (GR-4.3.4, GR-4.3.5 in [42]).



(a) Im Diagramm ist der Prozentsatz aller Moleküle mit einer oder mehr Kollisionen für alle acht Tools aufgetragen. Die Farbe der Balken gibt Aufschluss, wie kritisch eine Kollision ist. Kollisionen ausschließlich innerhalb der Ringsysteme sind nicht so kritisch wie Kollisionen, bei denen Atome übereinanderliegen.



(b) Das Diagramm zeigt pro Tool die Anzahl der Moleküle mit Klasse I Kollisionen, d.h. Strukturdiagramme, bei denen Atome oder Bindungen übereinanderliegen. Die x-Achse ist logarithmisch skaliert, um den Wertebereich aller Tools besser darstellen zu können.

Abb. 4.4. Die Diagramme zeigen die Anzahl der Moleküle mit Kollisionen der Klasse I bis Klasse III für alle acht verglichenen Tools in der Stichprobe des PubChem Substance Datensatzes.

4.3. Experiment 2: Qualitätsvergleich

Beim zweiten Experiment geht es darum, Naomi_{2D} mit 8 weiteren Programmen zur Berechnung von 2D-Molekülkoordinaten zu vergleichen. Dabei soll anhand möglichst objektiver Bewertungskriterien herausgefunden werden, welche Methode die lesbarsten Diagramme liefert.

4.3.1. Datensatz

Für den Vergleichstest wurde eine Stichprobe von 10 Mio. Molekülen aus dem Datensatz PubChem Substances ausgewählt. Die Moleküle der PubChem kommen aus verschiedenen Sammlungen, die häufig blockweise hinzugefügt werden. Daher ist es wichtig, die Stichprobe für den Test uniform und zufällig aus der Gesamtmenge auszuwählen. Als Zufallsquelle kam dabei ein Pseudozufallsgenerator zum Einsatz (Mersenne Twister 19937)[54]. Da das Skript die Moleküle in der Reihenfolge der ursprünglichen Moleküle zurücklieferte, wurde die Stichprobe daraufhin noch in eine zufällige Reihenfolge gebracht. Daher sind auch alle kleineren zusammenhängenden Teile (z. B. die ersten 1000 Moleküle) der 10 Mio. Moleküle eine gleichverteilte Stichprobe der Gesamtmenge.

4.3.2. Durchführung

Naomi_{2D} wurde mit einer Reihe von externen Programmen verglichen, die ebenfalls 2D-Koordinaten von Molekülen berechnen. Im Einzelnen waren das:

Cactvs ist eine in C geschriebene Chemieinformatik-Programmsammlung der Firma Xemistry. Unter anderem werden durch Cactvs-Skripte Moleküle der PubChem Substance zu PubChem Compound Molekülen aufbereitet. Auch die 2D-Koordinatenberechnung für Compound Moleküle wird von Cactvs bereitgestellt. Für die Vergleichsrechnungen wurde die Version 3.425 benutzt.

CDK ist eine Open-Source-Bibliothek für strukturelle Algorithmen aus der Chemie und Bioinformatik. Die Bibliothek ist seit 2007 frei verfügbar und wurde von Steinbeck et. al initiiert. Als Version wurde 1.5.10 vom 30.12.2014 benutzt.

Indigo ist eine ursprünglich von der Firma GGA entwickelte frei verfügbare Chemieinformatik-Bibliothek. Als Version wurde 1.1.12 vom 24.12.2013 benutzt.

JChem Suite von ChemAxon ist die Basisbibliothek der Tools von ChemAxon. Das in der Bibliothek enthaltene Programm `molconvert` wurde für die Vergleichsrechnungen benutzt. Als Version wurde 15.2.9 vom 9.2.2015 benutzt.

MOE ist eine kommerzielle Chemieinformatik Suite der Firma Chemical Computing Group. MOE wurde in der Version 2013.09 benutzt.

Open Babel ist ein Sammlung von Programmen zum Konvertieren unterschiedlicher Chemie-Dateiformate. Mit enthalten ist auch eine 2D-Koordinatenberechnung für Moleküle. Als Version wurde 2.3.2 vom 11.2.2015 benutzt.

RDKit ist ein Open-Source-Toolkit der Chemieinformatik, das von Greg Landgrum initiiert wurde. Als Version wurde 2015.03.1 vom 9.5.2015 verwendet.

Die Vergleichbarkeit der Ergebnisse wird gewährleistet, indem alle Programme mit denselben Eingabedateien liefen und die Ausgabe durch die gleiche Testroutine überprüft wurde. Dazu wurde das Naomi_{2D} Kommandozeilenprogramm um die Möglichkeit erweitert, anstatt der internen 2D-Koordinatenberechnung ein extern aufgerufenes Programm zu benutzen. Alle Moleküle wurden dem externen Programm einzeln als SDF-Eingabedatei bereitgestellt. Das externe Programm versah dieses Molekül mit 2D-Koordinaten und schrieb das Ergebnis in eine definierte Ausgabedatei. Auf diese Art ließ sich für jedes prozessierte Molekül auf exakt die gleiche Art ermitteln, welche Qualitätsmaße eingehalten, welche verletzt wurden und wie lange das externe Programm insgesamt für die Berechnung benötigte. Um die Vergleichbarkeit mit Naomi_{2D} zu gewährleisten, wurden dabei nur diejenigen Moleküle betrachtet, die sich von Naomi einlesen lassen. Alle Ergebnisse wurden in einer SQLite-Datei pro Eingabedatei protokolliert, deren Auswertung für die Diagramme und Statistiken automatisch erfolgte.

4.3.3. Ergebnis

Insgesamt betrachtet generiert Naomi_{2D} mehr Diagramme ohne Kollisionen als alle anderen getesteten Programme (s. Abb. 4.4a). Das einzige Programm mit einer ähnlich guten Rate ist Cactvs. Wenn man Diagramme, die nur ringsysteminterne Kollisionen

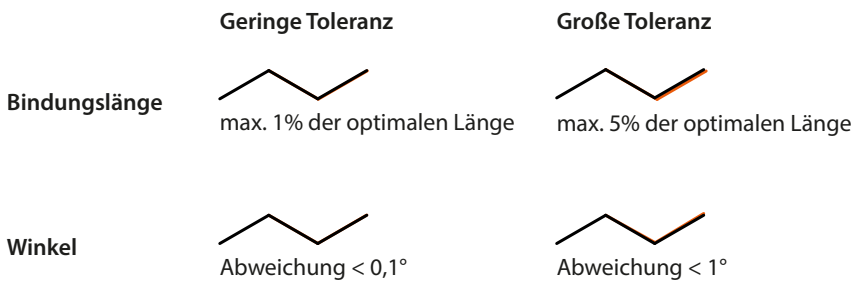
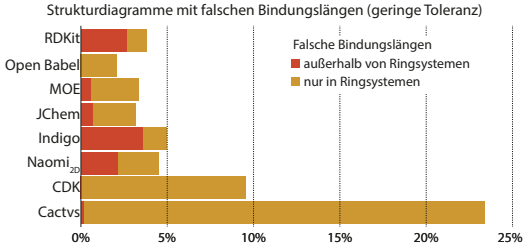


Abb. 4.5. Im Bild sind die zwei unterschiedlichen Toleranzen dargestellt, die für die Validierung benutzt wurden. Die geringe Toleranz ist sowohl bei Bindungslänge und Winkel kaum sichtbar, die große Toleranz ist dagegen deutlicher sichtbar. Dabei kommt es natürlich auf die Vergrößerung an, mit denen die Strukturdiagramme dargestellt werden.

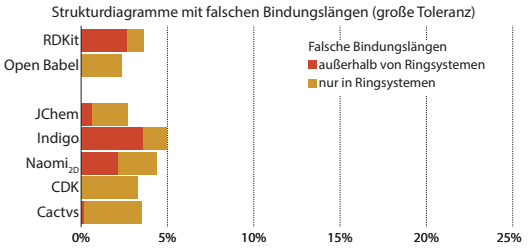
enthalten, außer Acht lässt, sind sowohl Naomi$_{2D}$ als auch Cactvs deutlich besser als alle anderen Programme.

Beim Vermeiden von Kollisionen der Klasse I ist dagegen JChem das beste Programm (s. Abb. 4.4b). Lediglich 131 Moleküle der 10 Mio. Moleküle des Datensatzes wurden mit überlappenden Bindungen oder Atomen gezeichnet. Naomi$_{2D}$ liegt an zweiter Position mit 206 Molekülen. Ganz weit abgeschlagen sind Open Babel und CDK, bei denen 1 % bzw. fast 5 % aller Moleküle schwere Kollisionen enthalten.

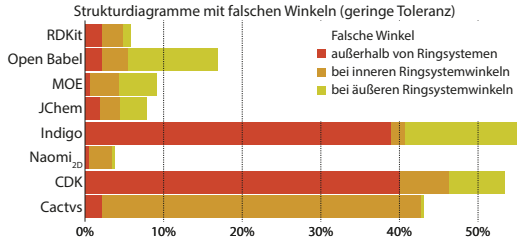
Die Anzahl der Kollisionen alleine ist kein aussagekräftiges Maß: Um die Anzahl der Kollisionen auf ein Mindestmaß zu reduzieren, könnte man schließlich einfach alle Gepflogenheiten ignorieren und Layouts mit beliebigen Bindungslängen und Winkeln generieren. Daher wurden alle Strukturdiagramme auch noch auf die Länge der Bindungen und die Größe der Winkel untersucht. Wie zu Beginn des Kapitels beschrieben folgen die Bindungen und Winkel Regeln, deren Einhaltung automatisch überprüft werden kann (s. Kap. 4.1). Das einfachste Kriterium ist die Länge der Bindungen. Diese sollte immer gleich sein, wobei Längenunterschiede der Bindungen in Ringsystemen nicht so schlimm sind wie Längenunterschiede in Ketten. Die Experimente wurden mit zwei unterschiedlichen Toleranzen durchgeführt (s. Abb. 4.5). In Abb. 4.6a sind die Ergebnisse für alle acht Tools mit geringer Toleranz aufgetragen. Im Experiment stellte sich heraus, dass die ursprünglich gewählte Toleranz von 1 % der optimalen Bindungslänge zu gering gewählt war: Das Programm Cactvs erstellt für nahezu 25 % der Moleküle Layouts mit falschen Bindungslängen, die sich ausschließlich in Ringsystemen befinden. Der Grund dafür ist, dass Cactvs ein systematischer Fehler bei kondensierten Ringsystemen unterläuft (s. Abb. 4.7). Sowohl die Winkel als auch die Längen des kondensierten Ringes sind leicht verzerrt. Durch direktes Übereinanderlegen des korrigierten Ringes fällt der Unterschied auf, bei normaler Betrachtung dagegen nicht. Mit einer größeren Toleranz von 5 % sowohl für die Bindungslängen als auch Winkel verschwinden diese Ausreißer für Cactvs. Die Werte der restlichen Tools bis auf CDK bleiben im selben Bereich (s. Abb. 4.6b). Die Resultate mit der größeren Toleranz wurden für das Programm MOE kein zweites Mal berechnet, weil die Berechnungen zu lange dauer-



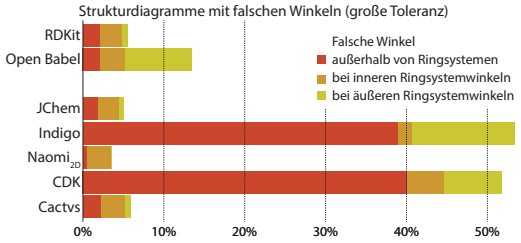
(a) Im Diagramm ist der Prozentsatz aller Moleküle mit mindestens einer falschen Bindungslängen für die acht verglichenen Programme aufgetragen. Als Toleranz wurde 1 % der optimalen Bindungslänge gewählt.



(b) Mit einer höheren Toleranz von 5 % der optimalen Bindungslänge verbessern sich vor allem die Werte von Cactvs und CDK. Aufgrund der langen Dauer der Berechnung mit MOE wurden die Werte mit der höheren Toleranz nicht nochmal für MOE berechnet.



(c) Im Diagramm ist der Prozentsatz aller Moleküle mit mindestens einem falschen Winkel für unterschiedliche Tools aufgetragen. Als Datensatz wurde eine zufällig ausgewählte Untermenge der PubChem mit 10 Mio. Molekülen benutzt.



(d) Bindungen in Ringen sind beim Tool Cactvs systematisch leicht zu lang. Durch Wahl einer größeren Toleranz werden diese Fälle nicht mitgezählt. Die Werte für MOE wurden nicht noch einmal mit einer höheren Toleranz berechnet.

Abb. 4.6. Die Diagramme zeigen die Anzahl der Moleküle mit mindestens einer falschen Bindungslänge oder einem falschen Winkel für alle acht verglichenen Tools.

ten: Für MOE stand nur eine geringe Anzahl an Lizenzen zur Verfügung und dadurch konnten die Koordinaten auf dem Rechnercluster nicht ausreichend parallel berechnet werden. Da die Werte für MOE bereits im strikten Fall sehr gut waren, wird sich, wie bei den anderen Programmen, an diesen mit geringerer Toleranz nichts ändern.

Beim Betrachten der Ergebnisse mit geringer Toleranz (s. Abb. 4.6b) fällt auf, dass Naomi_{2D} im hinteren Feld bei der Einhaltung der Bindungslängen landet. Der Grund dafür sind die Prioritäten bei der Kollisionsbehebung. Naomi_{2D} verändert bei Kettenbindungen zur Kollisionsvermeidung eher die Länge als die Winkel. Dies führt in vielen Fällen zu schöneren Diagrammen (GR-4.3.7 in [42]). Insgesamt liegt die Anzahl der Diagramme mit falschen Bindungslängen für alle Programme unter 5 % des Gesamtdatensatzes. Daraus kann man folgern, dass Längenänderungen nur für die Vermeidung von Kollisionen eingesetzt werden.

Bei der Bewertung der eingehaltenen Winkel zeigt sich nun, dass Naomi_{2D} bei der Kollisionsvermeidung das Einhalten der Winkel priorisiert (s. Abb. 4.6d). Die Winkel

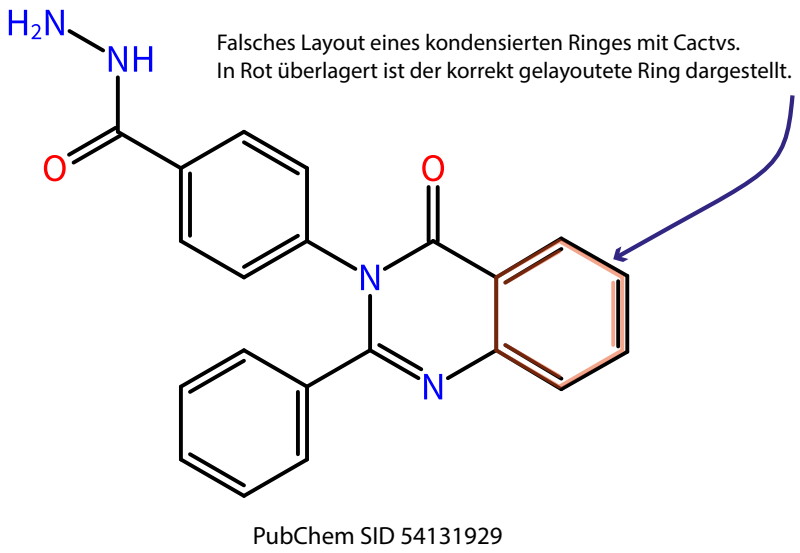


Abb. 4.7. Beim Zusammenfügen von kondensierten Ringsystemen unterläuft Cactvs ein systematischer Fehler: Angehängte Ringe sind leicht gestreckt. Beim normalen Betrachten der Diagramme fällt diese kleine Ungenauigkeit nicht auf, der direkte Vergleich mit dem korrekten Layout (im Bild in Orange) macht den Fehler jedoch sichtbar.

bei Kettenbindungen werden von Naomi$_{2D}$ am besten von allen Programmen eingehalten. Wenn man den systematischen Fehler bei Cactvs ignoriert, sieht man, dass bis auf Indigo und CDK alle Programme Winkelanpassungen wiederum ausschließlich für die Behebung von Kollisionen einsetzen. Für die Programme Indigo und CDK zeigt sich hier, dass sie Winkel nicht allzu ernst nehmen. Für über 50 % des Datensatzes generieren beide Programme Strukturdiagramme mit falschen Winkeln. Dabei sind vor allem die Winkel in Ketten nicht korrekt.

4.3.4. Folgerung

Im direkten Vergleich mit anderen Programmen ist Naomi$_{2D}$ sowohl bei der Anzahl der Kollisionen als auch bei Einhaltung der Winkel besser als die anderen Programme. Verbesserungspotenzial besteht vor allem bei der Vermeidung schwerer Kollisionen und bei der Vermeidung von Bindungslängenänderungen.

4.4. Experiment 3: Geschwindigkeit

Das dritte Experiment vergleicht die Geschwindigkeit von Naomi$_{2D}$ mit anderen Programmen. In zwei Versuchen wird sowohl die Geschwindigkeit bei der Verarbeitung einzelner Moleküle als auch die benötigte Zeit für das Konvertieren eines ganzen Datensatzes gemessen.

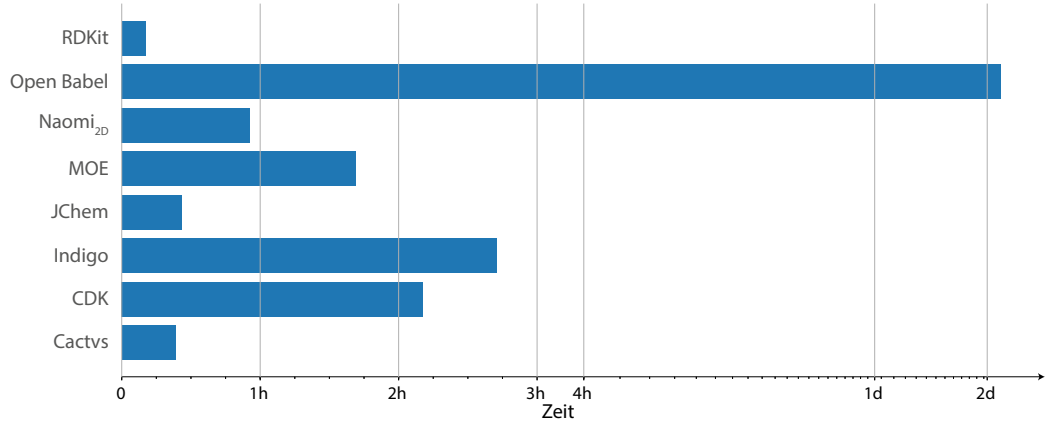


Abb. 4.8. Das Diagramm zeigt die benötigte Zeit, um den NCI DTP Datensatz zu konvertieren. Für alle getesteten Programme wurde der Median aus drei Durchläufen als Repräsentant gewählt. Die relative Abweichung zum Median der zwei anderen Durchläufe beträgt für alle Tools weniger als 1%. Um einen Gesamtüberblick zu ermöglichen, ist die Zeitachse bis drei Stunden linear und für höhere Werte logarithmisch skaliert.

4.4.1. Datensatz

Als Datensatz wird wiederum die Stichprobe von 10 Mio. Molekülen aus dem PubChem Substances Datensatz benutzt (s. Kap. 4.3.1) und als weiterer Datensatz werden Moleküle des Developmental Therapeutics Program des National Cancer Instituts (NCI DTP) verwendet [21]. Der Datensatz besteht aus 257 513 Molekülen. Alle diese Moleküle befinden sich auch im PubChem-Datensatz. Für Naomi_{2D} wurden außerdem die Zeiten für den gesamten PubChem Substances Datensatz ermittelt.

4.4.2. Durchführung

Bei der Durchführung von Experiment 2 auf dem PubChem Substances Datensatz konnten die benötigten Zeiten zur Berechnung von 2D-Koordinaten für die Programme CDK, Cactvs, Naomi_{2D} und RDKit direkt gemessen werden. Das Messen der gesamten Ausführungszeit der externen Aufrufe stellte sich als nur schwer vergleichbar heraus: Die Programme verwenden unterschiedliche Programmierumgebungen und dadurch unterscheidet sich die Zeit für die Initialisierung der Programme stark voneinander: Ein in C++ geschriebenes Programm startet schneller als ein Java Programm.

Besser vergleichen lässt sich die Zeit, die die Programme benötigen, um 2D-Koordinaten für eine komplette Datei zu berechnen. Hierfür wurde der NCI DTP Datensatz mit 257 513 Molekülen benutzt. Die Zeiten für jeweils drei Konvertierungen pro Programm wurden gemessen.

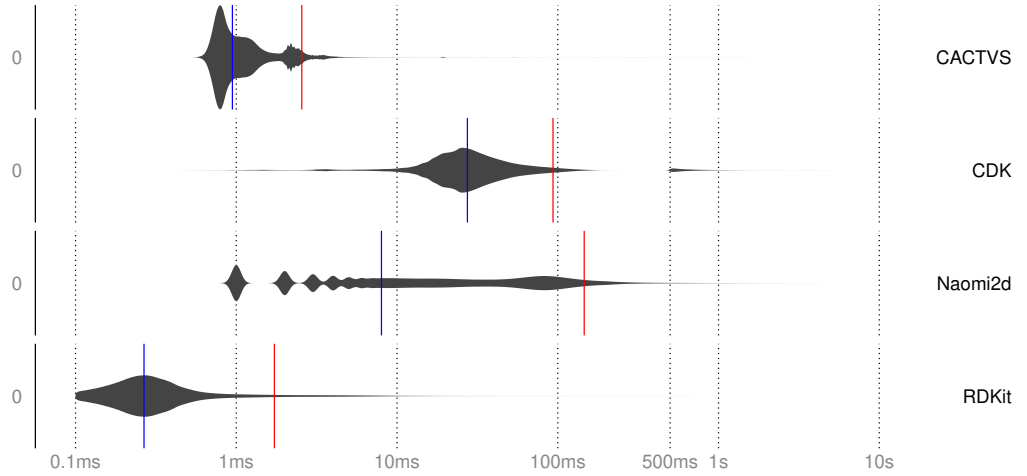


Abb. 4.9. Vergleich aller Programme bei denen eine direkte Zeitmessung pro Molekül durchgeführt werden konnte. Die Verteilung der einzelnen Zeiten wurde durch eine Kerndichteschätzung ermittelt. Der blaue Strich markiert den Median und der rote Strich das 95 % Quantil der ermittelten Zeiten.

4.4.3. Ergebnis

Bei der Berechnung von 2D-Koordinaten für den NCI DTP Datensatz liegt Naomi_{2D} im Vergleich zu den anderen Programmen im Mittelfeld. Die Ergebnisse sind in Abb. 4.8 als Diagramm dargestellt. Von den drei Durchläufen wurde dabei für jedes Programm jeweils der Durchlauf mit der mittleren Zeit als Repräsentant genommen. Die einzelnen Laufzeiten aller Programme weichen weniger als 1 % von der jeweiligen mittleren Laufzeit ab. Die Zeitmessungen sind also reproduzierbar und wurden nicht durch nebenläufige Prozesse auf dem Rechner verfälscht. Die genauen Zeiten sind in diesem Zusammenhang eher uninteressant, interessanter sind die Verhältnisse der verschiedenen Programme zueinander. Am schnellsten ist RDKit mit 10 Minuten. JChem und Cactvs liegen bei etwa der doppelten Zeit mit 25 Minuten bzw. 23 Minuten. Wiederum etwa doppelt so lange benötigt dann Naomi_{2D} mit 55 Minuten. MOE benötigt etwa das Doppelte mit 101 Minuten und Indigo und CDK fast das Dreifache mit 130 und 162 Minuten. Das Schlusslicht bildet Open Babel, dass mehr als zwei Tage für die Berechnung benötigt.

Für den PubChem Substances Datensatz war es möglich, für vier der Tools genauere Messwerte pro Molekül zu bekommen. Dies ermöglicht einen besseren Blick auf die Verteilung der einzelnen Zeiten. In Abb. 4.9 sind die Verteilungen für alle 10 Mio. Moleküle des Datensatzes abgebildet. Einzelne Zeiten pro Molekül konnten dabei mit den Programmen Cactvs, CDK, Naomi_{2D} und RDKit gemessen werden. Der blaue Strich im Diagramm markiert dabei jeweils den Medianwert und der rote Strich markiert das 95 % Quantil. Die Positionierung der Programme ist dabei auch bei diesem Datensatz dieselbe wie beim NCI DTP Datensatz.

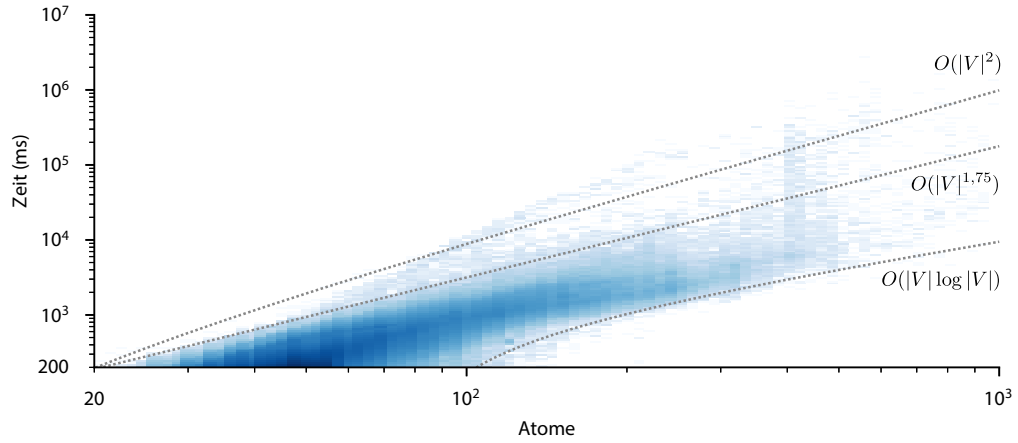


Abb. 4.10. Das Diagramm zeigt die benötigte Zeit der Layoutberechnung verglichen mit der Anzahl der Atome der jeweiligen Moleküle. In Blau sind die Häufigkeiten der Laufzeiten für alle Moleküle des PubChem Substances Datensatzes zu sehen, die für die Koordinatenberechnung mehr als 200 ms benötigen. Dies war bei 3 582 779 Molekülen der Fall, keines der Moleküle enthielt weniger als 20 Atome.

Neben den Zeiten für die PubChem Substances Stichprobe sind für Naomi_{2D} auch die gemessenen Zeiten für den kompletten PubChem Substances Datensatz verfügbar (s. Tab. 4.2). Diese Zeiten wurden in mehrere Klassen gruppiert. Die Layouts für die meisten Moleküle (88,57 %) können augenblicklich, also innerhalb von 100 ms berechnet werden. Weniger als eine Sekunde benötigen insgesamt 99 % der Koordinatenberechnungen. Nur sehr wenige Moleküle (0,15 %) benötigen länger. In Abb. 4.10 ist die Verteilung der Laufzeit abhängig von der Anzahl der Atome des jeweiligen Moleküls zu sehen. Dabei wurden nur die Moleküle betrachtet, deren Laufzeit größer als 200 ms ist. Man sieht, dass die Laufzeiten der Moleküle zwischen $O(|V|\log|V|)$ und $O(|V|^2)$ liegen. Dies stimmt mit der theoretischen Betrachtungen in der Tabelle mit den Bewertungskriterien[1] überein (s. Tab. 3.2). Also ist vor allem für große Moleküle die Zeit für die Kettenberechnung ausschlaggebend, da der überwiegende Teil der Moleküle keine Kollisionen enthält. Die Zeit für das Layout der Ringsysteme spielt eine untergeordnete Rolle. Ausreißer ergeben sich durch Moleküle mit vielen Kollisionen, bei denen die nachträgliche Kollisionsbehebung läuft.

4.4.4. Folgerung

Im Vergleich zu den anderen Programmen sind die Laufzeiten von Naomi_{2D} im Schnitt mehr durch Langläufer geprägt. Dies liegt an drei Faktoren:

[1] Bei organischen Molekülen ist die Anzahl der Bindungen pro Atom auf sechs begrenzt. Die Anzahl der Bindungen ist also maximal $O(|V|)$ anstatt $O(|V|^2)$. Damit entspricht die Laufzeitklasse $O(|E|\log|E|)$ der Klasse $O(|V|\log|V|)$.

Tab. 4.2. In der Tabelle sind die benötigten Zeiten von Naomi_{2D} für alle Moleküle des PubChem Substances Datensatzes zusammengefasst. Der Median der Zeiten beträgt 7 ms und das Maximum 2744 s.

Zeit	Moleküle	Moleküle(%)
< 100 ms	118 529 109	88,57
< 1 s	15 094 820	11,28
< 10 s	196 160	0,15
< 1 min	2168	0,0016
< 10 min	606	0,000 45
> 10 min	26	0,000 019

- Der erste Faktor ist die Parametrisierung der Layoutoptimierung: Naomi_{2D} durchsucht bei großen Molekülen nicht den gesamten Raum der Möglichkeiten, sondern bricht vorher ab. Die Abbruchkriterien lassen sich leicht anpassen auf Kosten der Qualität der Strukturdiagramme.

- Der zweite Faktor ist die Laufzeit für die Bewertung eines Layouts. Hier betragen die Laufzeiten typischerweise $O(|E| \log |E|)$ und können bis zu $O(|V^2|)$ betragen (s. Tab. 3.2). Eine Verbesserung dieser Zeiten wird sich direkt auf die Gesamtzeit der Optimierung auswirken.

- Der dritte Faktor ist die Laufzeit für die nachträgliche Kollisionsbehebung. Da alle Kollisionen einzeln betrachtet werden, dauert es sehr lange, wenn das Molekül groß ist und nach der Optimierung noch viele Kollisionen enthielt (s. Abb. A.4).

Allerdings muss man auch beachten, dass der überwiegende Teil der Moleküle innerhalb von 100 ms berechnet werden kann und nur ein Bruchteil der Moleküle länger als eine Sekunde benötigen. Die Geschwindigkeit ist damit ausreichend für das interaktive Zeichnen von Molekülen, vor allem da die Berechnung für mehrere Moleküle trivial parallelisierbar ist.

4.5. Experiment 4: Selbstausrichtung

Das letzte Experiment validiert die Strukturdiagrammausrichtung mithilfe einer Selbstausrichtung. Bei der Selbstausrichtung dient jedes Molekül sich selbst als Vorlage. Dies ist eigentlich ein Sonderfall, der in der Praxis nicht benötigt wird. Zum Validieren der Ausrichtung ist er jedoch nützlich, da das optimale Ergebnis (das Molekül erhält die gleichen Koordinaten wie die Vorlage) bekannt ist. Vor allem kann die Güte und der Einfluss des grafischen Matchings beurteilt werden, indem dieses mit der Nullhypothese (Zufallsmatching) und dem Optimum (Identitätsmatching) verglichen wird.

4.5.1. Datensatz

Als Datensatz für dieses Experiment diente eine Untermenge der PubChem Substance Stichprobe mit 125 000 Molekülen.

4.5.2. Durchführung

Für jedes Molekül aus dem Datensatz wurden zuerst normale 2D-Koordinaten berechnet. Lediglich die Kollisionsbehebung durch Verbiegen und Verlängern von Bindungen war deaktiviert. Danach wurde die Reihenfolge aller Atome und Bindungen zufällig geändert. Dies verhindert, dass die Reihenfolge der Atome und Bindungen zwischen Vorlage und Molekül gleich ist und möglicherweise zu einem künstlich verbesserten grafischen Matching führt. Auf dem neu geordneten Molekül wurden mit dem ursprünglichen Molekül als Vorlage 2D-Koordinaten berechnet. Diese sollten den 2D-Koordinaten der Vorlage exakt entsprechen. So kann man im Nachhinein anhand des Anteils der exakt übereinanderliegenden Bindungen bewerten, wie gut das Matching und der Ausrichtealgorithmus funktioniert haben. Wenn beide perfekt funktionieren, stimmen bei der Selbstausrichtung alle Bindungen überein und der Anteil überdeckter Bindungen ist 100 %.

Es wurden drei unterschiedliche Methoden verwendet, um 2D-Koordinaten anhand der Vorlage zu berechnen. Die Methoden unterscheiden sich nur in der Matching-Phase, die Layoutgenerierung mit dem augmentierten Layoutverfahren (s. Kap. 3.3.3) ist immer dieselbe.

Alle drei Verfahren benutzen das in Kap. 3.3.2 beschriebene grafische Matching. Zum Einsatz kam dabei jeweils einer der drei verschiedenen Nachbearbeitungsschritte „Direkt“, „Maximales Matching“ oder „Ohne Duplikate“ um die Anzahl der Bindungspaare zu reduzieren.

Die drei Verfahren wurden untereinander und mit den folgenden zwei Extremen verglichen:

- Bei der Selbstausrichtung ist nicht nur das optimale Ergebnis bekannt, auch das optimale Matching ist bekannt: Das erste Extrem, die Identitätsabbildung, ordnet jeder Bindung dieselbe Bindung der Vorlage zu.
- Das zweite Extrem ergibt sich aus der Frage: Was würde passieren, wenn das Matching-Verfahren komplett versagt? Findet in dem Fall das Layoutverfahren trotzdem ein akzeptables Layout? Um diese Fragen zu beantworten wird ein zufälliges Matching benutzt. Dabei wird jeder Bindung eine zufällige Bindung der Vorlage zugeordnet.

4.5.3. Ergebnis

Das Ergebnis des Experiments ist in Abb. 4.11 zu sehen. Die Ergebnisse der fünf verwendeten Verfahren wurden dafür nach dem Anteil der überdeckten Bindungen sortiert.

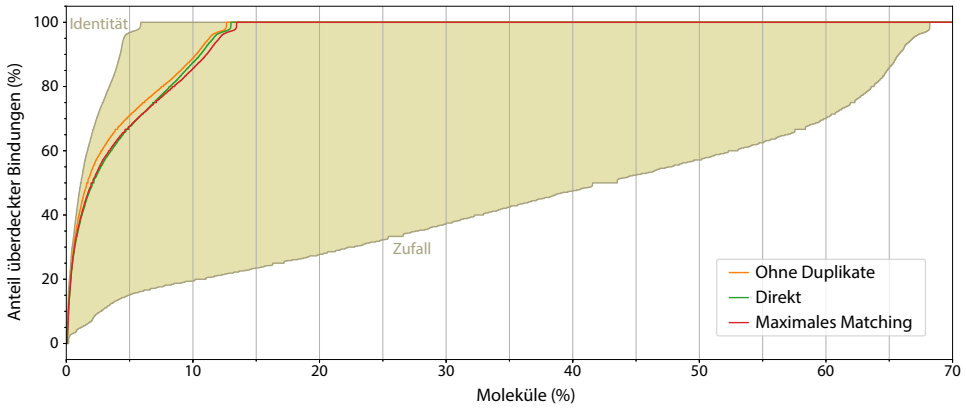


Abb. 4.11. Das Diagramm vergleicht die Güte der drei Nachbearbeitungsverfahren „Ohne Duplikate“, „Direkt“ und „Maximales Matching“ mit dem optimalen Verfahren („Identität“) und dem schlechtest möglichen Verfahren („Zufall“). Das grafische Matching verhält sich mit allen drei Nachbearbeitungsschritte ähnlich. Am besten ist das Verfahren „Ohne Duplikate“. Insgesamt kommt man mit allen Verfahren dem optimal möglichen Ergebnis sehr nahe.

Auf der x-Achse ist der Prozentsatz aller Moleküle sichtbar, die weniger als den gezeigten Anteil an überdeckten Bindungen haben. Das Identitätsverfahren erreicht bei 5,87 % der Moleküle den Stand, dass alle Bindungen perfekt überdeckt wurden. Dies ist vor allem auf die heuristische Komponente des Layoutalgorithmus zurückzuführen, da bei großen Molekülen nicht mehr alle möglichen Strukturdiagrammvarianten durchprobiert werden. Mit einem bestmöglichen Matching-Algorithmus würde also dennoch in 5,87 % der Fälle kein korrektes Layout gefunden werden. Die drei grafischen Matching-Verfahren liegen dabei von den Werten zwischen dem optimalen und dem zufälligen Matching. Der Unterschied der einzelnen Verfahren ist relativ gering: Das maximale Matching steigt etwas schneller als das direkte Matching, erreicht dann aber die komplette Bindungsüberdeckung bei 13,45 % der Moleküle entgegen den 13,00 % beim direkten Matching. Am besten ist das Verfahren „Ohne Duplikate“. Dieses erreicht das Optimum bei 12,65 % der Moleküle.

Das zufällige Verfahren funktioniert nur für etwa 31,8 % der Moleküle und erreicht das Optimum daher erst bei 68,2 %.

4.5.4. Folgerung

Die Ergebnisse des Experimentes zeigen, dass auf jeden Fall ein Matching-Verfahren notwendig ist, um die Eingabe für den Layoutalgorithmus zu liefern. Sowohl beim Matching-Verfahren als auch beim Layoutalgorithmus bestehen noch Verbesserungsmöglichkeiten: Der Layoutalgorithmus erreicht momentan bei 5,87 % der Moleküle das Optimum nicht und das Matching-Verfahren ist momentan 6,78 % vom optimalen Matching entfernt.

5. Moleküldatenbanken

Moleküldatenbanken sind in der Chemieinformatik weit verbreitet. Vor allem in Form von großen Sammlungen aller bekannten Moleküle [46] oder Proteine [7]. Diese dienen hauptsächlich dazu, in der großen Anzahl aller natürlich vorkommenden Moleküle den Überblick zu behalten. Für diese Anwendung ist es daher wichtig, dass sich Moleküle eindeutig identifizieren und effizient wiederfinden lassen. Die für diese Dissertation entworfene Moleküldatenbank verfolgt neben diesem aber noch einen weiteren Designaspekt: Das Tool Mona soll große Mengen von Molekülen effizient verwalten können. Hierbei ist es wichtig, dass alle Operationen auf diesen Mengen, wie Iterieren, Vereinigen oder auch der Schnitt selbst mit einer großen Anzahl an Molekülen in vertretbarer Zeit durchführbar sind.

In diesem Kapitel werden zunächst die der MoleculeDB und PropertyDB zugrundeliegenden Entscheidungen erläutert und danach auf Implementierungsdetails eingegangen.

5.1. Molekülidentität

Die wichtigste Designentscheidung der Datenbank ist die Frage nach der Identität von Molekülen. Wann kann man zwei Moleküle als gleich betrachten?

5.1.1. Kanonisierung von Molekülen

Bei der Kanonisierung von Molekülen werden die Atome und Bindungen eines Moleküls in eine kanonische Reihenfolge gebracht [55]. Wenn zwei Moleküle identisch sind, besitzen sie auch dieselbe kanonische Reihenfolge. Für die Kanonisierung von Molekülen wird Naomi benutzt (s. Kap. 2.2.1). Da Naomi Abbruchkriterien bei der Kanonisierung von hoch symmetrischen Molekülen enthält, führt die Kanonisierung nicht immer zu einem exakten Ergebnis. Was bedeutet das für die praktische Anwendung? Es ist möglich, dass zwei Moleküle als unterschiedlich erkannt werden, obwohl sie eigentlich isomorph sind. Der gegenteilige Fall, zwei unterschiedliche Moleküle werden als gleich erkannt, kann nicht auftreten. Zwei Moleküle sind immer isomorph, sobald auch nur eine exakt gleiche Beschreibung existiert.

Die Kanonisierung muss für alle Zufallsreihenfolgen der Atome und Bindungen des Moleküls immer dieselbe Reihenfolge berechnen. Wenn unterschiedliche Reihenfolgen für dasselbe Molekül berechnet werden, ist die Kanonisierung fehlgeschlagen. Dieser Test wurde mit allen 147 Mio. Molekülen aus der PubChem Substances Datenbank (s. Kap. 4.2.1) durchgeführt. Der Test war für fast alle Moleküle erfolgreich. Lediglich

17 Moleküle wurden identifiziert, bei denen die Kanonisierung nicht funktionierte. Alle 17 Moleküle besaßen große Ringsysteme und waren hochgradig symmetrisch. Bei jedem Molekül wurde die Kanonisierung abgebrochen, da eines der Abbruchkriterien von Naomi erreicht wurde.

5.1.2. Molekülidentität

Auf welche Art und Weise Moleküle als identisch erkannt werden ist eine wichtige konzeptuelle Frage der hier vorgestellten Moleküldatenbank. Sind zwei Moleküle identisch, wenn ihre Graphenstruktur isomorph ist? Wenn sie aus denselben Ursprungsdaten erstellt wurden? Oder müssen zusätzlich auch die 3D-Koordinaten übereinstimmen? Was ist mit zusätzlichen Eigenschaften des Moleküls, wie dem Namen oder der Bindungsaffinität, die über eine SDF-Eigenschaft in die Datenbank importiert wurde.

Es fällt schnell auf, dass eine eindeutige Übereinstimmung von Molekülen nur auf topologischer Ebene eindeutig und intuitiv verständlich ist. Wenn man als zusätzliches Kriterium die Originaldaten oder die Ursprungsdatei heranzöge, würde es schnell unübersichtlich: Das Umbenennen einer Moleküldatei oder das Hinzufügen oder Entfernen von Wasserstoffen führt zu einem Molekül, das nicht mehr identisch zum ursprünglichen Molekül ist. Das Gleiche gilt für die Konformation eines Moleküls: Um identische Moleküle zu finden, müssten die Koordinatensätze der Moleküle überlagert werden. Dies liefert dann kein eindeutiges Kriterium mehr sondern eine Wahrscheinlichkeit, mit der zwei Moleküle identisch sind.

Jedoch gerade für den Anwendungsfall eines intuitiv zu bedienenden Programms, dessen Funktionalität sowohl nützlich als auch leicht nachvollziehbar ist, ist ein möglichst einfaches Identitätsmaß sehr wichtig. Daher sind im Fall der MoleculeDB zwei Moleküle identisch, sobald ihre Topologie (s. Kap. 2.4.1) übereinstimmt. Weder spielt die Konformation eine Rolle noch der Name des Moleküls oder die Dateien, aus denen es stammt.

Standardmäßig sind zwei Moleküle identisch, wenn ihre komplette Topologie exakt übereinstimmt. Diese umfasst ebenfalls den konkreten Tautomerzustand und das Stereoisomer. In der Realität kann ein Molekül verschiedene Tautomerzustände annehmen die von Mona gleich behandelt werden sollen. Ebenfalls spielt es nicht immer eine Rolle, ob ein Molekül in (E)-Anordnung oder (Z)-Anordnung vorliegt. Es gibt daher drei Abstufungen der Identitätsfunktion: Zum einen können auch alle Tautomere eines Moleküls als identisch betrachtet werden. Zum anderen kann der Typ eines Stereozentrums bei der Identität ignoriert werden. Als letzte Abstufung ist es noch möglich, Moleküle mit unterschiedlichen Ladungen an denselben Atomen als gleich zu betrachten. Dies dient dazu, konjugierte Systeme unabhängig von den aktuellen Ladungen der Atome als identisch zu behandeln.

Nachdem die Identität von Molekülen geklärt ist, ist die Frage, wie unterschiedliche Ausprägungen desselben Moleküls gehandhabt werden. Die Kombination der Molekültopologie, Konformation und beliebig vieler zusätzlicher Daten wird im konkreten Datenbankkontext Molekülinstanz oder kurz Instanz genannt. Jede Molekülinstanz in der Datenbank ist damit eindeutig über seine InstanzID identifiziert, und es gibt per De-

finition keine Duplikate. Jede dieser Instanzen enthält aber einen Verweis (anhand der MoleculeID) auf das zugehörige Molekül. Moleküle können daher mehrfach verwendet werden, Instanzen dagegen nicht.

Der nächste Abschnitt betrachtet die Struktur der MoleculeDB und der PropertyDB. Wie werden Moleküle abgespeichert und geladen? Wie wird die Identitätsfunktion konkret umgesetzt? Und wie kann man mithilfe von InstanzID und MolekülID Mengen definieren.

5.2. Struktur der MoleculeDB und PropertyDB

Bei der Wahl des Datenbanksystems lag vor allem eine einfache Einbindung und Benutzung der Datenbank im Vordergrund. Diese Anforderung erfüllen vor allem eingebettete Datenbanken. Diese Datenbanken benötigen keinerlei Konfiguration und keine eigenen Server, um lauffähig zu sein. Da sie im selben Prozess wie die Anwendung laufen, sind sie jedoch nicht so skalierbar, wenn es darum geht, möglichst viele Anfragen parallel zu bearbeiten. Mona wurde jedoch als Einzelanwendungsprogramm konzipiert, sodass dieses keine große Einschränkung ist.

Eine weitere Entscheidung betrifft die Art des Datenbanksystems: Herkömmliche SQL-Datenbanken [18] sind gut geeignet, viele Daten zu verarbeiten, sind aber auf eine Organisation der Daten nach dem *entity-relationship* Modell, also auf eine Tabellenform, beschränkt. Dies heißt vor allem, dass Graphen nicht direkt von den Datenbankoperationen unterstützt werden, da diese sich nur schwer in eine effiziente Tabellenform bringen lassen. Spezialisierte Datenbanksysteme für Graphen sind wiederum auf das Ablegen eines riesigen allgemeinen Graphen spezialisiert und nicht auf viele Millionen kleiner molekularer Graphen. Da das Speichern einer großen Anzahl von Molekülen wesentlich wichtiger war als der direkte Zugriff auf die Graphenstruktur der Moleküle, wurde eine herkömmliche SQL-basierte Datenbank benutzt. Die Datenbankschicht zwischen Mona und der Datenbank dient dabei explizit dem Zweck, die verwendete Datenbank zu abstrahieren, um sie gegebenenfalls austauschen zu können. Momentan werden sowohl SqLite [35] als auch rudimentär PostgreSQL [63] von der Datenbankschicht unterstützt. Eine andere mittlerweile verbreitete Art von Datenbanken, die NoSQL-Datenbanken, sind vor allem für das Speichern von unstrukturierten Daten in Form von Schlüssel-Wert Paaren prädestiniert. Hier gab es zum Zeitpunkt der Entwicklung keinen weit verbreiteten stabilen Vertreter, der einfach in das eigene Anwendungsprogramm eingefügt werden konnte.

Zu Anfang der Entwicklung von Mona bestand die gesamte Datenbankschicht aus einer Bibliothek. Dieses Konzept stellte sich jedoch nicht als flexibel genug heraus, sodass die Datenbankschicht in zwei getrennte Schichten unterteilt wurde, um auch für andere Projekte granular verwendbar zu sein. Die MoleculeDB speichert die Daten von Molekülen und Instanzen. Die PropertyDB annotiert beliebige Eigenschaften sowohl an Instanzen als auch an Molekülen und erlaubt die mathematisch bekannten Mengenoperationen auf Mengen von Molekülen.

Der folgende Abschnitt beschreibt zunächst die Funktionalität der MoleculeDB: Wie

werden Moleküle in der Datenbank abgelegt? Und wie findet die Zuordnung von Molekülen und Instanzen statt? Der darauf folgende Abschnitt beschreibt die PropertyDB, die die Funktionalität der MoleculeDB benutzt, um Mengen von Molekülen in der Datenbank abzubilden.

5.2.1. Serialisierung von Molekülen in Naomi

Um Moleküle in einer SQL-Datenbank zu speichern, muss der Molekülgraph in Tabellenform umgewandelt werden. Man hat hier die Wahl zwischen zwei Vorgehensweisen: Zum einen können die Molekülgraphen verteilt über mehrere Tabellen gespeichert werden, sodass Datenbankoperationen auch direkt auf den Graphen zugreifen können. Zum anderen kann man den Graphen serialisieren und ihn komplett in einem eigenen Eintrag speichern. Wie die Überschrift bereits verrät, wurde für die MoleculeDB der zweite Ansatz gewählt. Datenbankoperationen auf den Molekülgraphen sind nur dann nötig, wenn man möglichst schnell Moleküle per Ähnlichkeitsmaß oder Substruktursuche finden möchte. Und selbst dann ist es fragwürdig, ob SQL-Datenbankoperationen hier große Vorteile bringen können oder ob es nicht effizienter ist, einen zusätzlichen Index zu erstellen, der auf genau diese Anwendung spezialisiert ist. Weiterhin stand dieser Anwendungsfall nicht im Fokus von Mona. Vielmehr ging es vor allem um die einfache und effiziente Handhabung von möglichst großen Mengen organischer Moleküle. Alle Moleküle werden daher in ein platzsparendes Binärformat umgewandelt und als einzelne Einträge gespeichert. Wenn in Mona der Bedarf besteht, auf den Molekülgraphen zuzugreifen, werden die Moleküle im Speicher wieder aufgebaut.

Die Anforderungen an die Serialisierung sind daher die folgenden:

- Die Binärform der Moleküle sollte möglichst platzsparend sein. Dies hält die Datenbank klein.
- Redundanzen in der Binärform sollten möglichst minimal sein, um Konvertierungsfehler und andere Inkonsistenzen beim Wiederaufbau zu vermeiden.
- Das Speichern und Laden von Molekülen muss möglichst zeiteffizient erfolgen.

In gewisser Weise widersprechen sich diese Anforderungen: Damit das Speichern und Laden von Molekülen möglichst schnell geht, müssen Informationen, die aus den Rohdaten aufwendig berechnet werden, wie die Ringsysteme und Ringfamilien, zusätzlich im Binärformat gespeichert werden.

Oberste Priorität hatte bei der MoleculeDB die Korrektheit: Im Binärformat werden daher kaum redundante Informationen gespeichert. Um trotzdem eine möglichst zeiteffiziente Verwaltung der Moleküle zu ermöglichen, wurden die direkt von Naomi ermittelten Atom- und Bindungstypen im sogenannten MolString kodiert. Dieser Zeichenketten basierte MolString wird vor dem Speichern in der Datenbank noch mit der verlustfreien Kompressionsmethode DEFLATE der *zlib* [20] komprimiert.

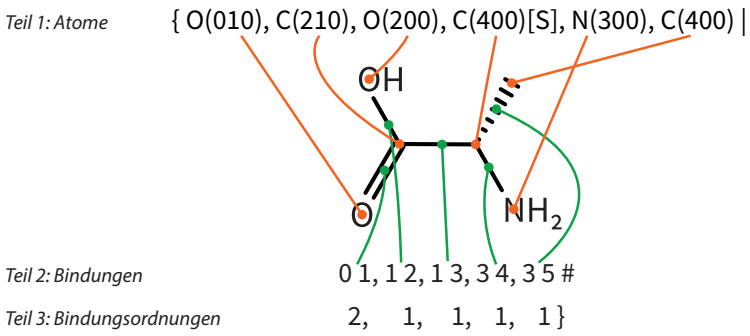


Abb. 5.1. Der MolString für L-Alanin besteht aus drei Teilen: Die Beschreibung der Atomtypen, Definition der Bindungen und die Ordnungen der Bindungen.

Das MolString Format Ziel des MolString Formates war es, die interne Molekülrepräsentation von Naomi möglichst direkt in einem zeilenbasierten Format ähnlich zu SMILES [84] und Inchl [31] abzubilden. Das MolString Format besteht aus drei Teilen (s. Abb. 5.1):

Atome Der erste Teil enthält die von Naomi benutzten Atomtypen und die Richtung der Stereozentren. Beispielsweise sieht dieser Abschnitt für L-Alanin (mit der SMILES-Repräsentation: `O=C(O)[C@@H](N)C`) so aus:
`O(010),C(210),O(200),C(400)[S],N(300),C(400)`

Bindungen Der zweite Teil enthält die Bindungsdefinitionen. Jede Definition besteht aus 2 Zahlen. Diese referenzieren die entsprechenden Endatome aus der Atomliste über ihre jeweilige Position. Der Bindungsabschnitt für L-Alanin sieht folgendermaßen aus: `0 1,1 2,1 3,3 4,3 5`

Bindungsordnungen Der dritte Teil enthält die Bindungsordnungen für alle Bindungen. Für L-Alanin ergibt sich: `2,1,1,1,1`

Die drei Teile sind durch | bzw. # voneinander getrennt. Insgesamt sieht der komplette MolString für L-Alanin damit wie folgt aus:

```
{ O(010),C(210),O(200),C(400)[S],N(300),C(400) | ↵
0 1,1 2,1 3,3 4,3 5 # 2,1,1,1,1 }
```

Beim Einlesen dieses Formats mit Naomi müssen die Atomtypen nicht mehr ermittelt werden. Da die Konformation des Moleküls nicht im MolString gespeichert wird, kann diese nicht benutzt werden, um die Art der Stereozentren zu berechnen. Es ist daher erforderlich, die Stereozentren-Information zusätzlich im MolString zu speichern.

Die Initialisierung in Naomi überprüft nach dem Erstellen des Molekül-Grundgerüstes die Plausibilität, d. h. ob alle Bindungen zu den angegebenen Atomtypen passen. Danach müssen noch die Initialisierungsschritte ausgeführt werden, die die Aromatizität, die Ringsysteme und Ringe berechnen. Dieses Vorgehen dient auch der Korrektheit von Naomi: Nur wenn die Eingabedaten aus verschiedenen Molekülformaten durch diesel-

ben Initialisierungsschritte gehen, kann man einfach sicherstellen, dass alle Moleküle auf die gleiche korrekte Art initialisiert wurden.

Behandlung von Wasserstoffen Im MolString-Format werden die Wasserstoffe nicht explizit abgespeichert. Dadurch reduzieren sich die im MolString kodierten Atome und Bindungen um etwa die Hälfte. Dies führt zu einer deutlichen Datenreduktion, verkompliziert allerdings auch das Speichern und Laden von Molekülen: Konformationen in der Datenbank enthalten neben den Koordinaten der Schweratome auch die Koordinaten der Wasserstoffe. Wenn also Wasserstoffe nicht mehr explizit im MolString abgespeichert werden, müssen diese über eine definierte Prozedur angefügt werden, um auch die Wasserstoffkoordinaten der Konformation korrekt zuordnen zu können.

Dazu werden beim Speichern der Moleküle die Atome umsortiert: Alle Schweratome werden an den Beginn des Moleküls verschoben und die Wasserstoffatome werden anhand der Reihenfolge, in der sie an die Schweratome gebunden sind, dahinter aufgereiht. Die Konformation des Moleküls wird anhand dieser Atomreihung erstellt. Beim Laden der Moleküle werden die Wasserstoffe, entsprechend der Reihenfolge der Schweratome, am Ende eingefügt. Die abgespeicherten Wasserstoffkoordinaten werden auf diese Art wieder den richtigen Wasserstoffatomen zugeordnet.

5.2.2. Moleküle und Instanzen

Wie in Kap. 5.1.2 beschrieben, speichert die MoleculeDB die Topologie der Moleküle getrennt von der konkreten Ausprägung, der Instanz. Dies geschieht in zwei getrennten Tabellen: Die `MoleculeDB_molecule` Tabelle enthält den MolString aller in der Datenbank vorhandenen Moleküle zusammen mit einer MoleculeID. Instanzen werden in der `MoleculeDB_instances` Tabelle gespeichert, diese enthält neben der Konformation und dem Namen des Moleküls auch die MoleculeID des passenden Moleküls aus der `MoleculeDB_molecule` Tabelle.

Um feststellen zu können, zu welchem Molekül eine bestimmte Instanz gehört, müssen Moleküle in der Datenbank identifizierbar sein. Dies geschieht mithilfe der in der Naomi vorhandenen Möglichkeit, Moleküle zu kanonisieren. Wenn ein Molekül zum ersten Mal in der Datenbank gespeichert wird, wird aus der kanonisierten MolString-Repräsentation ein Hashschlüssel berechnet, der zur Identifikation in der `MoleculeDB_molecule` Tabelle mitgespeichert wird. Der Hashschlüssel wird dabei mit dem SHA256-Verfahren berechnet, sodass Kollisionen sehr unwahrscheinlich sind.

Beim Hinzufügen einer weiteren Instanz dieses Moleküls zur Datenbank, wird ebenfalls zuerst der Identifikationsschlüssel der kanonisierten MolString Repräsentation berechnet. Anhand des Schlüssels lässt sich das bereits in der Datenbank vorhandene Molekül finden. In diesem Fall muss für die zusätzliche Instanz nur ein weiterer Eintrag mit den Instanzen spezifischen Daten in der `MoleculeDB_instances` Tabelle hinzugefügt werden.

Durch die Kanonisierung kann es passieren, dass ein Molekül, das eigentlich schon in der Datenbank vorhanden ist, nicht gefunden wird. Dieser Fall ist nicht tragisch, da das Molekül in dem Fall einfach ein zweites Mal in der Datenbank gespeichert wird. Durch

die Hashberechnung kann eine Hashkollision jedoch dazu führen, dass zwei Moleküle fälschlich als identisch angesehen werden. Dieser Fall ist schwerwiegender, da die beiden Moleküle eine unterschiedliche Anzahl von Atomen haben könnten. Höchstwahrscheinlich führt dieses Ereignis zu einem Absturz von Mona. Durch die Verwendung einer kryptografischen Hashfunktion mit 256 Bit ist dieses Ereignis sehr unwahrscheinlich und bisher nie aufgetreten.

Die verschiedenen Identitätsstufen werden über eine Vorprozessierung der Moleküle abgebildet. Um den Tautomerzustand zu ignorieren, wird das Molekül in einen kanonischen Tautomerzustand versetzt und von diesem der Identifikationsschlüssel abgeleitet. Die Stereochemie eines Moleküls kann ignoriert werden, indem alle Stereodeskriptoren auf den undefinierten Zustand gesetzt werden. Auf die gleiche Weise wird die Ladung ignoriert, wenn man das Molekül in einen neutralen Zustand versetzt. Da alle diese Zustände unabhängig voneinander durchgeführt werden können, sind sie beliebig miteinander kombinierbar. Man kann also gleichzeitig sowohl die Stereochemie als auch die Tautomerzustände ignorieren.

Diese Identitätsstufen sind für Anwendungsfälle in Mona hilfreich und erleichtern es vor allem dem Chemiker, Tautomerzustände zu ignorieren. Allerdings muss dem Benutzer bewusst sein, dass die Moleküle verändert werden. Diese Funktionen sollten also nicht benutzt werden, wenn man die Datenbank als reine Aufbewahrungslösung ansieht.

5.2.3. Speichern von Mengen und Eigenschaften

Aufbauend auf der MoleculeDB existiert die PropertyDB, die die Moleküle und Instanzen aus der MoleculeDB mit annotierten Daten versieht und Moleküle und Instanzen in Mengen organisiert. Die PropertyDB arbeitet mit den in der MoleculeDB oder anderen Datenbanken definierten Schlüsseln: Aus der MoleculeDB stammen die MoleculeId und die InstanceId, aus anderen am ZBH entwickelten Datenbanken wie der ProteinDB können ebenfalls die entsprechenden IDs in der PropertyDB benutzt werden.

Bevor man eine Eigenschaft verwenden kann, muss diese zuerst bei der Datenbank registriert werden. Dabei erhält sie einen Namen und einen Schlüssel, den PropertyKey. Die Eigenschaften selbst werden in der `PropertyDB_moleculeproperties` bzw. `PropertyDB_instanceproperties` Tabelle gespeichert. Beide Tabellen enthalten drei Spalten: die ID der Instanz oder des Moleküls, den Schlüssel der Eigenschaft und den Wert der Eigenschaft selbst. Diese Art der Speicherung erlaubt es, beliebig viele Eigenschaften für beliebig viele Moleküle in einer Tabelle zu speichern. Trotzdem ist die Verwendung dieser Daten weiterhin effizient möglich. Es befinden sich Datenbank-Indizes auf dem Schlüssel der Eigenschaft, der ID und dem Wert. Alle k Werte zu einem festen Schlüssel lassen sich in Zeit $O(\log N + k)$ ermitteln, wobei N die Anzahl aller in der Datenbank gespeicherten Eigenschaften ist. Dasselbe gilt für alle k Werte einer Eigenschaft. Ebenso ist das Filtern nach Wertebereichen mit k Ergebnissen in der Zeit $O(\log N + k)$ möglich. Nur wenn man alle Werte einer festen Eigenschaft für k verschiedene Schlüssel ermitteln möchte, benötigt man $O(k \log N)$ Zeit. Diese Operation wird von Mona jedoch nicht direkt benutzt.

Neben beliebigen Eigenschaften aus externen Quellen gibt es eine Reihe interner Eigenschaften, die beim Hinzufügen der Moleküle zur Datenbank berechnet werden: Eine Reihe von physikochemischen Eigenschaften steht zur Verfügung. Beispiele hierfür sind das Molekulargewicht oder der LogP eines Moleküls. Außerdem werden die in jedem Molekül vorhandenen chemischen Elemente und häufigsten funktionalen Gruppen als Bitvektoren gespeichert. Dies erlaubt es, chemische Elemente und funktionelle Gruppen in Molekülen zu suchen, ohne die Moleküle selbst initialisieren zu müssen. Dabei stellt SQLite in den numerischen Typen maximal 51 Bit zur Verfügung (dies entspricht der Größe der Mantisse einer IEEE 754 Fließkommazahl doppelter Genauigkeit). Chemische Elemente mit einer Ordnungszahl größer als 51 werden daher nicht unterstützt. Jedes einzelne chemische Element oder jede funktionelle Gruppe als booleschen Wert zu speichern, hätte jedoch den Speicherbedarf in der SQLite-Datenbank enorm erhöht.[1]

Mengen werden in der PropertyDB auf eine ähnliche Art wie die Eigenschaften abgespeichert. Die Tabelle `PropertyDB_moleculesubsets` enthält eine Spalte mit dem Schlüssel der Menge und eine Spalte mit den Mitgliedern der Menge. Damit wird eine n:n-Beziehung zwischen den Schlüsseln der Mengen und den Mitgliedern modelliert. Es befindet sich ein Datenbankindex auf beiden Spalten. Dadurch sind die typischen Operationen wieder effizient möglich: Der Zugriff auf alle k Mitglieder einer Menge benötigt Laufzeit $O(\log N + k)$. Mengenoperationen lassen sich ebenfalls effizient ausführen: In diesem Fall benötigt die Vereinigung von zwei Mengen mit den Größen k_1 und k_2 die Laufzeit $O(\log N + k_1 + k_2)$. Die Laufzeit kommt dadurch zustande, dass alle Mitglieder der beiden Mengen in sortierter Reihenfolge aus der Datenbank geholt werden müssen. Durch den Index ist die Mitgliederspalte aber immer sortiert, wenn auch die Schlüsselspalte sortiert ist. Daher ist ein explizites Sortieren der Mitglieder nach der ersten Anfrage nicht nötig.

An den Laufzeiten sieht man, dass die Größe der Ausgabe k in den obigen Laufzeitbetrachtungen eine große Rolle spielt. Gerade beim Hantieren mit großen Mengen oder vielen Eigenschaften macht sich der lineare Faktor bemerkbar. Die Gesamtgröße der Datenbank spielt dagegen eine geringere Rolle. Messwerte für typische Aktionen auf verschiedenen Mengen finden sich im nächsten Kap. 6.5.

[1] Da SQLite variable Datentypen für Spalten verwendet, wird für jeden booleschen Wert mindestens 1 Byte benötigt. Anstatt 8 Byte für eine Fließkommazahl doppelter Genauigkeit würden insgesamt 51 Eigenschaften benötigt. Das ergibt einen Speicherbedarf von mindestens $51 \cdot 9$ Byte $= 459$ Byte pro Molekül, dabei bestehen die 9 Byte aus vier Byte *Property Key*, vier Byte *Instance Key* und einem Byte für den Wert.

6. Mona

Mona ist ein Anwendungsprogramm, das es erlaubt, komfortabel mit großen Mengen kleiner Moleküle zu arbeiten. Entstanden ist es aus dem Bedürfnis heraus, ein einfach zu bedienendes Werkzeug zu haben, das einem immer wiederkehrende Arbeitsschritte in der Chemieinformatik abnimmt. Im Gegensatz zu Pipeline Tools, wie KNIME oder auch selbstgeschriebenen Skripten ist Mona für Arbeitsabläufe gedacht, die aus explorativen Arbeitsschritten bestehen. Hierfür ist es wichtig, dass Mona intuitiv bedienbar ist und damit dem Benutzer die Zeit erspart, sich in eine komplizierte Umgebung einzuarbeiten.

Das Verwalten von Molekülen aus den verschiedensten Eingabedateien ist dabei die elementare Funktionalität, die Mona bietet. Moleküle werden dafür in Form von Mengen organisiert. Wie bereits in Kap. 5.1.2 vorweggenommen, spielt das Konzept der Molekülidentität hierbei eine wichtige Rolle und wird daher im nächsten Abschnitt in den Kontext von Mona eingeordnet.

6.1. Moleküle und Instanzen

Für ein vollständiges Verständnis von Mona ist es wichtig, die genaue Handhabung von Molekülen und Instanzen zu kennen. Die Identität der Moleküle ergibt sich dabei aus ihrer Topologie, eine Instanz ist eine konkrete Ausprägung eines Moleküls (s. Kap. 5.1.2). Prinzipiell werden alle Moleküle, die Mona lädt, als Instanzen in der Datenbank abgelegt. Es ist also immer möglich, die Originalmoleküle inklusive ihrer Konformation und den annotierten SDF-Eigenschaften exakt wiederherzustellen. Innerhalb von Mona wird aber nur mit den aus den Instanzen extrahierten Molekülen gearbeitet. In den Molekülmengen existieren daher keine Duplikate. Erst dadurch sind mathematische Mengenoperation anwendbar und intuitiv nachvollziehbar. Beim Abspeichern von Molekülmengen aus Mona werden die Instanzen wieder hergestellt. Daher ist es notwendig, vor dem Abspeichern beim Benutzer zu erfragen, aus welchen Ursprungsquellen die zu den Molekülen der Menge passenden Instanzen wiederhergestellt werden sollen.

6.2. Historie

Die erste Version von Mona wurde im Rahmen einer Projektgruppe 2011 am ZBH konzipiert und programmiert. Als Vorlage diente der Naomi-Converter, der bereits die häufigsten Operationen mit einer minimalen GUI umsetzte. Die Aufgabe bestand darin, ein GUI Tool zu erstellen, das aufbauend auf der Naomi-Bibliothek Konvertierungs-,

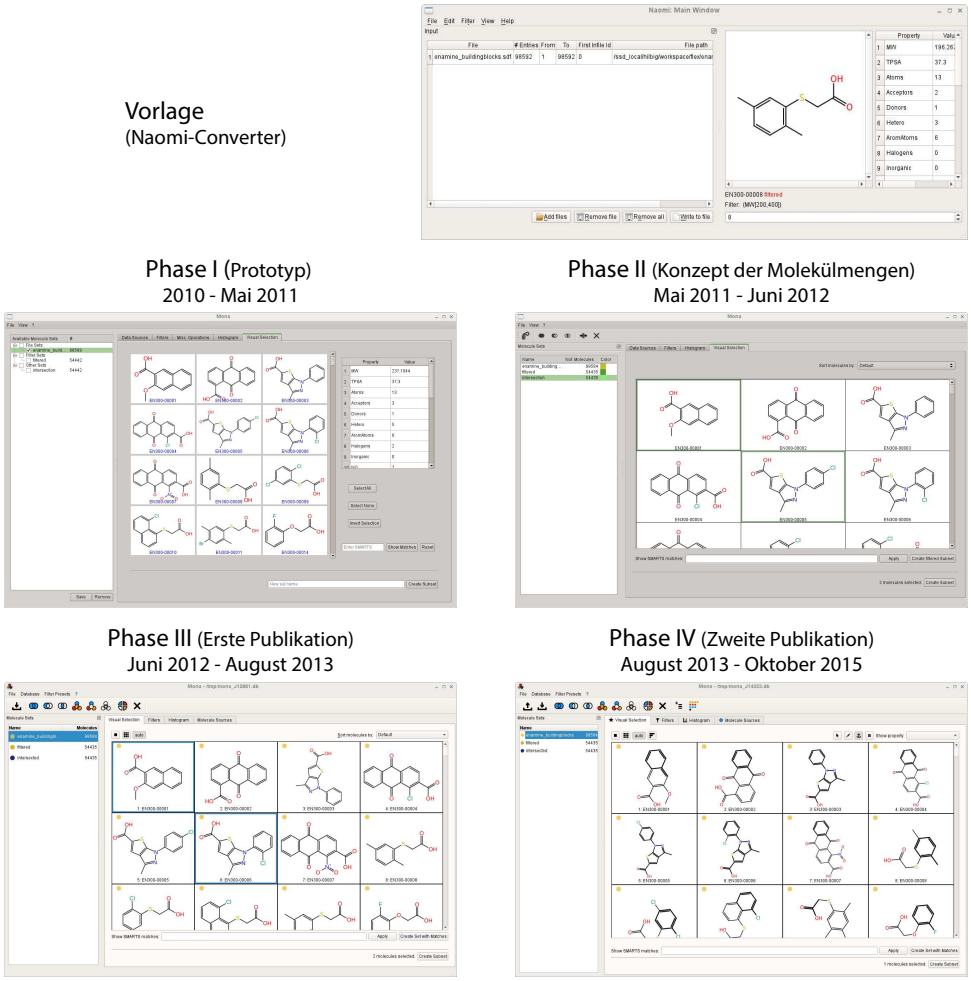


Abb. 6.1. Die vier Entwicklungsphasen von Mona sind anhand von jeweils einem repräsentativen Bildschirmfoto illustriert. Visuell hat sich vor allem die Aufteilung der Anwendung und die Darstellung der Molekülauswahl in den vier Phasen verändert.

Darstellungs- und Filteroperationen auf Dateien erlaubt. Mathias v. Behren, Andreas Heumeier und Thomas Otto entwarfen ein Instanzenmengen-Konzept und setzten einen ersten Prototypen um (s. Phase I in Abb. 6.1). Dieser erste Prototyp unterschied Moleküle anhand ihrer Position in den Eingabedateien und sah eine Unterteilung der Sets in unterschiedlicher Arten vor.

In Phase II (Konzept der Molekülmengen) wurde das Mengen-Konzept bedeutend vereinfacht, indem Moleküle anhand ihrer Topologie identifiziert werden. Außerdem wurde der Prototyp komplett umgeschrieben, um eine höhere Modularisierung zu erreichen. Dabei wurde intern eine klare Trennung zwischen Datenbankschicht, Abstraktion der Molekülmengen sowie der Operationen darauf und der Darstellungsschicht umgesetzt (s. Abb. 6.4).

Die Gestaltung der GUI durchlief ebenfalls mehrere Iterationen in Phasen I bis IV:

Das Grundkonzept mit der Mengenansicht auf der linken Seite und der Gruppierung gleicher Arbeitsschritte auf der rechten Seite, ist über die Versionen hinweg weitestgehend gleich geblieben. Die einzelnen Bestandteile wurden alle angepasst und umgestaltet: Operationen auf Molekülmengen wurden in die Symbolleiste ausgelagert, die *Visual Selection* erlaubt ein flüssiges Navigieren der Mengen und das Erstellen von Filtern wurde vereinfacht. Häufige Tests mit Benutzern führten dazu, dass die GUI weiterhin einfach zu bedienen und möglichst selbsterklärend ist.

In der letzten Phase IV wurden die in den vorherigen Phasen begonnenen Konzepte abgerundet, indem eine Reihe von Erweiterungen vorgenommen wurden: Die Datenbankschicht wurde umstrukturiert, um beliebige Eigenschaften an Molekülen zu erlauben. Weiterhin wurde Mona um eine Clusterung erweitert, mit der es möglich ist, Mengen zu clustern und das Ergebnis übersichtlich in der *visual selection* darzustellen. Sowohl in der Clusteransicht als auch in der normalen Ansicht wird in der jetzigen Version das Ausrichten der Strukturdiagramme ähnlicher Moleküle benutzt (s. Kap. 3.3).

Im Rückblick hat sich diese iterative Vorgehensweise bewährt, da auf diese Art und Weise eine neue Funktionalität schnell getestet und auch wieder verworfen werden konnte, wenn sie nicht intuitiv verständlich war.

6.3. Funktionalität

Die in Mona verfügbare Funktionalität lässt sich in unterschiedliche Bereiche kategorisieren: Die grundlegende Funktionalität besteht aus dem Importieren und Exportieren von Molekülen aus möglichst vielen unterschiedlichen Dateiformaten. Alle in Mona importierten Moleküle und Instanzen verfügen über beliebige Eigenschaften. Ein Hauptbereich von Mona sind sicherlich die unterschiedlichen Operationen zum Verwalten von Molekülmengen. Ein weiterer Hauptbereich ist die Visualierung der Molekülmengen. Welche Moleküle enthält eine Menge? Hierbei sind Strukturdiagramme zur Darstellung der Moleküle das Mittel der Wahl. Der letzte Bereich ist die Analyse von Molekülmengen. Histogramme ermöglichen es dabei, den Überblick über beliebige Eigenschaften der Moleküle zu behalten. In Abb. 6.2 ist die Hauptansicht von Mona mit den wichtigsten möglichen Operationen dargestellt.

6.3.1. Importieren und Exportieren von Molekülen

Es ist möglich, in Mona alle Moleküldateiformate einzulesen, die von Naomi unterstützt werden. Neben SDF, MOL2 und SMILES können auch alle Liganden aus PDB-Dateien eingelesen werden [81]. Alle Moleküle werden dabei von Naomi in ein einheitliches Molekülformat gebracht, das von der Moleküldatenbank in eine SQLite-Datenbank geschrieben wird (s. Kap. 5.2.1). Je nach Ursprungsformat stehen unterschiedliche Informationen zur Verfügung. Diese sind in Tab. 6.1 aufgelistet.

Beim Einlesen der Moleküle kann man einstellen, wie unterschiedliche Tautomere, Protonierungszustände oder Ladungen der Moleküle interpretiert werden sollen (Für die konkrete Umsetzung s. Kap. 5.2.2). Standardmäßig werden unterschiedliche Tauto-

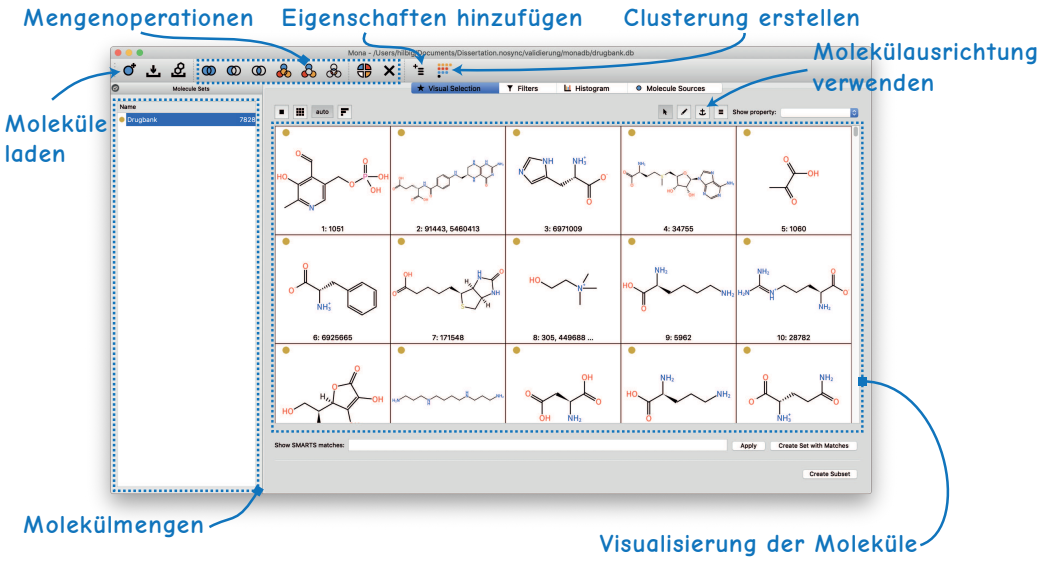


Abb. 6.2. Die Hauptansicht von Mona zeigt die Molekülmengen und die darin enthaltenen Moleküle. Von hier aus lassen sich alle wichtigen Operationen in Mona durchführen.

merformen eines Moleküls als zwei unterschiedliche Moleküle betrachtet. Nach dem Aktivieren der Tautomererkennung werden alle Tautomere eines Moleküls für die importierten Moleküle als ein und dasselbe Molekül erkannt. Alle Einstellungen beim Importieren der Moleküle sind dabei nicht reversibel. Es ist aber möglich dieselbe Ursprungsdatei mehrfach mit unterschiedlichen Einstellungen zu importieren. Wenn man die Entscheidung flexibel treffen möchte, wie Tautomere, Protomere und Stereozentren behandelt werden sollen, sind stattdessen entsprechenden Mengeneigenschaften geeigneter (s. Kap. 6.3.2).

Dateien, die in Mona importiert wurden, werden in den Molekülquellen abgelegt. Aus diesen Quellen lassen sich jederzeit neue Molekülmengen erstellen. Da die Mole-

Tab. 6.1. Mona unterstützt die gebräuchlichsten Moleküldateiformate. Jedes Dateiformat enthält leicht unterschiedliche Daten, die von Mona eingelesen und interpretiert werden.

Dateiformat	Konformation	Molekülname	Externe Eigenschaften
SDF	2D/3D	ja	beliebige Schlüssel-Wert Paare
PDB	3D	ja	wichtige PDB-Header[a]
MOL2	2D/3D	ja	nein
SMILES	keine	ja[b]	nein

[a] Aus der PDB-Datei werden der Titel, die vierstellige PDB-ID, die Auflösung des Experimentes, alle EC Nummern und der Organismus extrahiert.
[b] Namen von Molekülen werden in SMILES-Dateien über Leerzeichen vom SMILES-String getrennt erkannt.

küle dabei bereits in den Datenbank enthalten sind, werden diese Mengen unmittelbar erstellt.

Alle in Mona importierten Moleküle lassen sich auch wieder exportieren. Beim Exportieren von Molekülmengen stehen MOL2, SDF und SMILES als Ausgabedateiformate zur Verfügung. Als einziges dieser Formate unterstützt SDF beliebige Eigenschaften. Moleküle in SDF-Dateien werden mit allen Eigenschaften der Originaldaten (nur bei PDB- und SDF-Quelldateien) und zusätzlich mit allen in Mona hinzugefügten Eigenschaften gespeichert. Die von Mona hinzugefügten Eigenschaften sind dabei an den Präfixen `naomi` oder `mona` vor dem Namen erkennbar. Wenn das Ausgabeformat Konformationen unterstützt und das Quellmolekül eine Konformation besitzt, wird die Konformation unverändert vom Quellmolekül übernommen. Die Atome des Ausgabemoleküls befinden sich immer in der kanonischen Reihenfolge. Diese kann daher anders sein als die Reihenfolge der Atome in der Quelldatei. Weitere Unterschiede können durch Valenzfehler im Quellmolekül verursacht werden: Diese Fehler werden durch Naomi gefunden und, wenn möglich, behoben [80]. Die Verwendung von Naomi ermöglicht außerdem eine konsistente Konvertierung zwischen allen unterstützten Dateiformaten.

6.3.2. Eigenschaften von Molekülen und Instanzen

Die Architektur der Moleküldatenbank erlaubt es, sowohl Moleküle als auch Instanzen mit beliebigen Eigenschaften zu versehen (s. Kap. 5.2.3). In Mona wird dieses Mittel benutzt, um drei verschiedene Arten von Eigenschaften zu speichern:

Interne Eigenschaften Diese Eigenschaften werden von Naomi intern anhand der Molekültopologie berechnet. Diese umfassen typische physikochemische Eigenschaften wie das Molekulargewicht, die Anzahl der Atome oder eine Schätzung des logP Wertes [86]. Im Molekül vorhandene ausgewählte funktionelle Gruppen werden als Bitvektor gespeichert. Außerdem ist es möglich, eindeutige Bezeichner für die kanonische Tautomerdarstellung und Protonierungsform des Moleküls zu berechnen. Diese Bezeichner definieren eine Äquivalenzrelation auf einer Molekülmenge, die die Menge in die einzelnen Tautomer- oder Protonierungsklassen unterteilt.

Externe Eigenschaften Hierzu zählen alle Eigenschaften, die aus externen Quellen stammen. Mona erkennt externe Eigenschaften aus SDF- und PDB-Dateien. Weitere externe Eigenschaften sind der Name des Moleküls, der Dateiname der Ursprungsdatei und die Position in dieser Datei.

Datenbank-Eigenschaften Die letzte Klasse von Eigenschaften ergibt sich aus dem aktuellen Zustand der Datenbank. Es ist möglich, die Anzahl aller aktuell zu einem Molekül abgespeicherten Instanzen (Duplikate) als Moleküleigenschaft zu erzeugen. Dabei ist zu beachten, dass Datenbank-Eigenschaften nur zum Zeitpunkt der Erstellung gültig sind. Die Eigenschaften werden nicht aktualisiert,

wenn zu einem späteren Zeitpunkt weitere Instanzen zur Datenbank hinzugefügt werden, die den Wert dieser Eigenschaft verändern.

Alle internen und Datenbank-Eigenschaften gelten dabei immer für das komplette Molekül. Externe Eigenschaften gelten dagegen immer für einzelne Instanzen. Mona kann Eigenschaften für einzelne Instanzen anzeigen, aber nicht direkt mit diesen arbeiten: Mengen nach Eigenschaften zu sortieren oder zu filtern, funktioniert nur mit Moleküleigenschaften. In Mona ist es daher möglich, beliebige Instanzeigenschaften in Moleküleigenschaften umzuwandeln. Dazu muss eine Methode gewählt werden, mit der mehrere Instanzeigenschaften eines Moleküls gehandhabt werden. Die folgenden beiden Methoden sind für Eigenschaften beliebigen Typs verfügbar:

Wähle Erstes Diese Methode benutzt die Eigenschaft der ersten Instanz in der Datenbank als Moleküleigenschaft.

Wähle Alle Alle Instanzeigenschaften werden durch Kommata getrennt hintereinander angeordnet.

Für numerische Instanzeigenschaften sind drei weitere Methoden verfügbar. Diese berechnen das Minimum, Maximum oder den Durchschnitt über die Eigenschaftswerte aller zu einem Molekül registrierten Instanzen. Falls eine dieser drei Methoden auf nicht numerische Instanzeigenschaften angewandt wird, werden alle ungültigen Eigenschaften ignoriert.

Alle in der Datenbank hinterlegten Molekül- und Instanzeigenschaften lassen sich in der Detailansicht für jedes Molekül inspizieren.

6.3.3. Arbeiten mit Molekülmengen

Beim Importieren von Moleküldateien erstellt Mona immer Mengen aus den in der Datei vorhandenen Molekülen. Diese lassen sich mit den aus der Mathematik bekannten Mengenoperationen verknüpfen.

Im Einzelnen werden die drei Operationen Vereinigung, Differenz und Schnitt unterstützt. Bei der Handhabung von vielen Mengen ist es für den Benutzer nützlich, die Operationen Vereinigung und Schnitt auf alle ausgewählten Mengen gleichzeitig anzuwenden. Für die Differenz ist dies nicht möglich, da die Reihenfolge der Operanden hier eine Rolle spielt. Bei der Differenz erscheint daher immer ein Dialog, in dem man die beiden Mengen spezifiziert, die voneinander abgezogen werden sollen. Für einige Anwendungsfälle ist es außerdem sehr hilfreich, alle paarweisen Schnitte mehrerer Mengen auf einmal zu erstellen. Es gibt daher jeweils eine Variante der Standardoperationen, die anstatt eine Molekülmenge zu erzeugen, die jeweilige Operation auf alle paarweisen Kombinationen der Eingabeparameter anwendet und auf diese Art genauso viele Ausgabemengen erzeugt.

Neben dem Verknüpfen von Mengen gibt es mehrere Operationen, die Untermengen erzeugen. Im Einzelnen sind das:

Teilen Molekülmengen können auf drei Arten geteilt werden: In eine feste Anzahl von Teilmengen, in Teilmengen einer festen Größe oder in zwei unterschiedliche große Mengen anhand einer Prozentangabe. Die Reihenfolge, in der die Moleküle in die Ausgabemengen verteilt werden, kann dabei von beliebigen Moleküleigenschaften abhängen. So ist es z. B. möglich, Moleküle nach dem Molekulargewicht sortiert in Untermengen einzufügen. Wenn diese Menge in 10 Teile geteilt wird, enthält danach die erste Ergebnismenge die 10 % leichtesten Moleküle und die letzte die 10 % schwersten Moleküle.

Filtern Filter auf beliebigen Moleküleigenschaften können auf Mengen angewendet werden, um nur Moleküle auszuwählen, die die Filtereinstellungen erfüllen.

Auswählen Moleküle in Mengen können in der Molekülansicht durch den Benutzer direkt markiert werden. Aus den ausgewählten Molekülen lassen sich jederzeit eigene Untermengen erstellen.

Clusterauswahl In der Clusteransicht lassen sich aus den einzelnen Clustern direkt Untermengen erstellen.

Bis auf die Filteroperationen sind alle anderen Operationen direkt anwendbar, ohne dass es viele Wahlmöglichkeiten oder Einstellungen gibt.

Filteroperationen

Die Filteroperation selber bieten unterschiedliche Möglichkeiten, Filterketten zu spezifieren. Eine Filterkette besteht dabei aus beliebig vielen Filtern. Es gibt vier unterschiedliche Filtervarianten:

Eigenschaft (Typ A) Sowohl numerische als auch textbasierte Eigenschaften der Moleküle können als Kriterien in einem Filter spezifiziert werden. Für numerische Eigenschaften ist es möglich, den Bereich aus einem Histogramm auszuwählen oder direkt zu spezifizieren. Für textbasierte Eigenschaften kann man einen Suchbegriff eingeben, der in der Eigenschaft vorkommen muss.

Chemische Elemente (Typ B) Dieser Filter spezifiziert, welche chemischen Elemente ein Molekül enthalten muss. Die chemischen Elemente können direkt in einem Periodensystem ausgewählt werden.

Funktionelle Gruppen (Typ C) Typische funktionelle Gruppen der Moleküle sind als Bitvektoren in der Datenbank hinterlegt (s. Kap. 6.3.2). Dieser Filter erlaubt es, Moleküle nach den vorhandenen funktionellen Gruppen auszuwählen.

Substrukturfilter (Typ D) Der Substrukturfilter erlaubt es, Moleküle anhand einer Substruktursuche auszuwählen. Die Substruktur wird dabei anhand eines oder mehrerer SMARTS-Muster vorgegeben [19]. Ein Molekül erfüllt diesen Filter, wenn die vorgegebene Substruktur mindestens einmal im Molekül vorkommt.

Prinzipiell lassen sich alle Filter negieren. Die Filteraussage verkehrt sich dadurch jeweils ins exakte Gegenteil: Die Filter für chemische Elemente und funktionelle Gruppen akzeptieren alle Moleküle, die die gewählten Elemente oder funktionellen Gruppen nicht enthalten. Der Eigenschaftsfilter akzeptiert alle Moleküle, deren Eigenschaften nicht im spezifizierten Bereich liegen, und der Substrukturfilter akzeptiert alle Moleküle, die keine der angegebenen Substrukturen aufweisen.

Alle Filter in einer Filterkette sind durch logische Konjunktion verknüpft, d. h. damit ein Molekül von der gesamten Filterkette akzeptiert wird, muss jeder einzelne Filter für dieses Molekül erfüllt sein. Durch die Toleranzeigenschaft der Filterkette lässt sich dieses Kriterium aufweichen: Die Toleranz gibt die minimale Anzahl der Filter an, die für ein Molekül erfüllt sein müssen, damit dieses von der gesamten Filterkette akzeptiert wird. Wenn man daher als Toleranz 1 wählt, wird aus der Konjunktion der Filterkette eine Disjunktion, da nur ein beliebiger Filter der Kette erfüllt sein muss.

Die Laufzeit der unterschiedlichen Filter hängt vor allem vom Typ ab:

Am schnellsten, sowohl theoretisch als auch in der Praxis, sind die Eigenschaftsfilter, da diese unter Ausnutzung des Datenbankindex ausgeführt werden können. Die Laufzeit auf einer Molekülmenge der Größe N beträgt für die Anfrage $O(\log N)$. Die k resultierenden Molekülindizes müssen danach linear in eine eigene Menge kopiert werden. Bei einem Ergebnis mit mehreren Millionen Molekülen macht sich dieser letzte Schritt bemerkbar. Bei einigen tausend Molekülen fällt die Zeit zum Kopieren der Menge dagegen nicht auf.

Die Filter vom Typ B und C müssen alle Eigenschaften der Menge betrachten und benötigen daher $O(N)$ Laufzeit für die Anfrage, da N Bitoperationen von der Datenbank ausgeführt werden müssen, um die Existenz der funktionellen Gruppen oder chemischen Elemente zu testen.

Am längsten benötigt der Substrukturfilter. Da die Datenbank keinen Index zur Beschleunigung von Substrukturanfragen enthält, wird jedes Molekül der Menge initialisiert und auf diesem dann eine oder mehrere Substrukturanfragen durchgeführt. Hierbei ergibt sich wiederum $O(N)$ als Laufzeit für die Anfrage und N Molekülinitialisierungen und Substruktursuchen.

Beim Ausführen einer kompletten Filterkette ohne Toleranz werden alle Filter der Typen A, B und C mit einer gemeinsamen Anfrage ausgeführt. Substrukturfilter werden im Nachhinein auf dem Ergebnis der anderen Filter ausgeführt. Wenn die Filterkette eine Toleranz besitzt, müssen alle Filter einzeln ausgeführt werden. Für jeden Filter wird eine eigene Ergebnismenge erzeugt, die am Ende unter Beachtung des Toleranzkriteriums zum Gesamtergebnis zusammengefasst wird. Filtern mit Toleranzen dauert daher typischerweise wesentlich länger als einzelne Filter.

6.3.4. Visualisierung

Um Molekülmengen zu visualisieren, existieren zwei Ansichten in Mona: Die erste besteht aus einer klassischen tabellenbasierten Ansicht für Mengen. Die zweite Ansicht stellt die Cluster von Molekülmengen dar. Beide Ansichten sind exemplarisch in Abb. 6.3 dargestellt.

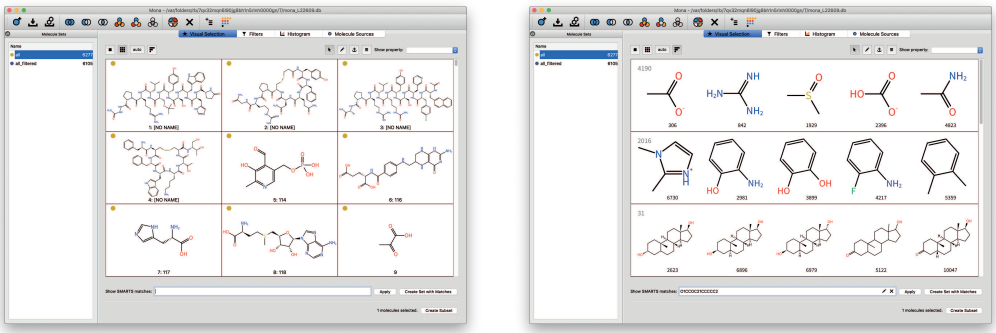


Abb. 6.3. Mona stellt zwei unterschiedliche Visualisierungmodi für Molekülmengen bereit: Auf der linken Seite ist die tabellenbasierte Darstellung und auf der rechten Seite ist die clusterbasierte Ansicht zu sehen.

Tabellenbasierte Darstellung

Beim tabellenbasierten Ansatz werden alle Moleküle der Menge als Strukturdiagramme in einer Tabelle dargestellt. In Mona wird die Anzeige der Strukturdiagramme abhängig vom Kontext beeinflusst. Wenn z. B. eine Menge nach der Anzahl der im Molekül vorhandenen Wasserstoffakzeptoren sortiert ist, werden in allen Strukturdiagrammen dynamisch die Wasserstoffakzeptoren hervorgehoben. Eine weitergehende Modifikation der Strukturdiagramme ist die Ankerfunktion: Durch Verankern eines Moleküls wird dieses als Vorlage für alle anderen Strukturdiagramme der Menge benutzt. Für das Ausrichten wird die in Kap. 3.3 beschriebene Methode verwendet. Die letzte mögliche dynamische Darstellung ist das farbliche Hervorheben von SMARTS-Treffern. Beim Durchblättern von Mengen ist es sehr hilfreich, wenn man schnell einen Überblick bekommt, welche Moleküle vom aktuellen SMARTS-Muster betroffen sind und welche nicht.

Die Herausforderung bei der tabellenbasierten Ansicht war es ein sofortiges flüssiges Scrollen zu erlauben, vor allem auch in großen Mengen mit Millionen von Molekülen. Dafür wurde eine Methode gewählt, die die benötigten Strukturdiagramme erst bei Bedarf berechnet. Ein Vorausberechnen von allen Bildern kam dabei vor allem wegen der fehlenden Flexibilität bei der Gestaltung der Bilder nicht zum Einsatz. Außerdem hätte es sehr viel Platz gekostet und wäre auch viel zu langsam gewesen. Die 2D-Koordinatenberechnung benötigt für die meisten Moleküle 10 ms. Auf die Bilder von 1 Mio. Molekülen wartet man dann fast 3 Stunden.

Um Strukturdiagramme der Moleküle möglichst schnell bei Bedarf zu berechnen, werden mehrere Methoden eingesetzt: Die Berechnung der 2D-Koordinaten erfolgt parallel auf möglichst vielen verfügbaren Prozessorkernen des Rechners. Da dieser Teil den größten Aufwand verursacht, werden die 2D-Koordinaten nach der Berechnung in der Moleküldatenbank gespeichert. Neue Koordinatenberechnungen werden dabei von der

Tabelle für die momentan sichtbaren Zellen angefordert. Alle angeforderten Berechnungen werden an das Ende einer Warteschlange angefügt. Wenn man sehr schnell durch die Menge scrollt, kann die Anzahl der angeforderten Koordinatenberechnungen sehr hoch sein und die Anzahl der unerledigten Aufgaben sehr schnell wachsen. Daher gibt es eine maximale Größe für die Warteschlange. Wenn mehr als 50 Aufgaben in die Warteschlange eingestellt werden, werden die ältesten Aufgaben abgebrochen. Aus den 2D-Koordinaten werden Strukturdiagramme erstellt, mit den angeforderten Markierungen versehen und die Vektorzeichnung abschließend als Pixelbild gerastert. Die letzten 1000 dieser erstellten Bilder werden ebenfalls zwischengespeichert. Dadurch wird ein Hoch- und Runterblättern um einige Seiten in der Tabelle sehr flüssig dargestellt.

Clusteransicht

Die Clusteransicht ist die zweite verfügbare Ansicht in Mona, die hilft einen Überblick über alle in einer Menge enthaltenen Moleküle zu bekommen. Um diese Ansicht verwenden zu können, muss zuerst eine Clusterung für eine Molekülmenge berechnet werden. Hierzu bietet Mona momentan zwei verschiedene Möglichkeiten:

Die erste Möglichkeit besteht darin, eine beliebige Eigenschaft zu clustern. Wenn die Werte einer Eigenschaft nur jeweils aus Texten, Binärdaten oder Ganzzahlen bestehen, werden genau die Moleküle demselben Cluster zugeordnet, die exakt denselben Eigenschaftswert besitzen. Bei numerischen Eigenschaften mit Gleitkommawerten werden die Werte Clustern der Größe b zugewiesen. Für b wird die am besten geeignete Größe berechnet, die einer Fünfer- oder Zehnerpotenz entspricht (... $1/10$, $1/5$, 1, 5, 10, 25 ...) und für etwa 50 verschiedene Cluster sorgt. Dies ergibt eine überschaubare Anzahl von Clustern mit leicht verständlichen Größen. Das Clustern von Eigenschaften ist sehr schnell ($O(N)$), da jedes Element der Molekülmenge nur genau einmal betrachtet werden muss.

Die zweite Möglichkeit besteht aus einem heuristischen k-medoid Clusterverfahren, das Moleküle nach ihrer Ähnlichkeit clustert [44]. Im Gegensatz zum k-means Verfahren ist das k-medoid Verfahren nicht auf ein euklidisches Distanzmaß angewiesen, da kein arithmetisches Mittel der Cluster berechnet wird. Die Anzahl der Cluster, das Ähnlichkeitsmaß und der Distanzoperator sind frei wählbar. Momentan werden als Ähnlichkeitsmaße ECFP, FCFP, TorsionFingerprint und eine ECFP-Variante für Strukturdiagramme unterstützt. Die Strukturdiagramm ECFP-Variante benutzt für Atome ausschließlich topologische Merkmale, dies führt zu einem Ähnlichkeitsmaß, das sich gut zum Ausrichten von Molekülen eignet (Für das Ausrichten von Molekülen wird dieselbe Variante benutzt, s. Kap. 3.3.2). Als Distanzmaße stehen alle üblichen Varianten zur Verfügung: Tanimoto, Cosine, Hamming, Euklidisch und Dice. Die Laufzeit dieses Clusterverfahrens ist deutlich höher als für die einfachen Eigenschaften: Das k-mediod Clusterverfahren läuft eine maximale Anzahl von Runden. In jeder Runde wird jedes Molekül dem nächsten Clusterrepräsentanten zugeordnet (Laufzeit: $O(N)$). Danach werden für jedes Cluster alle paarweisen Distanzvergleiche ($O(N^2)$) durchgeführt, um jeweils einen neuen Mittelpunkt des Clusters zu bestimmen.

Nachdem für eine Molekülmenge eine Clusterung berechnet wurde, kann die Men-

ge in der Clusteransicht dargestellt werden. Hierbei werden die Cluster der Größe nach sortiert zeilenweise angeordnet. In jeder Zeile befindet sich für jedes Cluster der Clusterrepräsentant und die ersten 100 Moleküle des Clusters. Es werden nur die ersten 100 Moleküle dargestellt, da mehr Moleküle zum Ruckeln führten und nicht mehr effizient von aktueller Hardware dargestellt werden konnten.

6.3.5. Analyse

Zur Analyse der Verteilung von beliebigen Eigenschaften in Mengen stellt Mona Histogramme zur Verfügung. In der Analyseansicht ist es möglich, Histogramme für beliebige numerische Eigenschaften zu erstellen. Die Histogramme werden laufend anhand der aktuell ausgewählten Mengen aktualisiert. Ist nur eine Menge selektiert, wird das Histogramm für die gewählte Eigenschaft mit der Farbe der ausgewählten Menge gezeichnet. Wenn mehrere Mengen selektiert sind, werden die entsprechenden Histogrammbalken mit den Farben und Werten aller selektierten Mengen nebeneinander gezeichnet. Dies erlaubt es, die Verteilung eines Wertes für zwei oder drei Mengen zu vergleichen. Wählt man mehr Mengen aus, wird diese Darstellung schnell unübersichtlich. Die Histogrammdarstellungen selbst sind interaktiv: Man kann einzelne Balken anklicken, um die exakte Höhe des Balkens zu bekommen und bei der Definition von Eigenschaftsfiltern werden die Histogrammdarstellungen benutzt, um komfortabel den Bereich der Eigenschaft auszuwählen. Hierbei ist zu beachten, dass die Histogramme als exemplarische Analysemöglichkeit umgesetzt wurden. Die Möglichkeiten zur Analyse insgesamt könnten mit unterschiedlichen Arten von interaktiven statistischen Visualisierung noch deutlich ausgebaut werden.

6.4. Anwendungsszenarien

Das Ziel von Mona ist es nicht, möglichst umfassende Funktionalität zu bieten, sondern die Funktionalität, die es besitzt, möglichst intuitiv zugänglich zu machen. Dabei soll Mona aber trotzdem universell eingesetzt werden können. Insbesondere müssen also in der Praxis übliche Anwendungsabläufe einfach möglich sein. In diesem Abschnitt werden die in [34] und in [33] beschriebenen Anwendungsabläufe wiederholt und zusammengefasst.

Die Anwendungsabläufe sind in drei Bereiche unterteilt:

Grundlegende Arbeitsschritte erläutern anhand von Beispielen einige grundlegende Fragestellungen, die mit Mona gelöst werden können.

Verwalten von Moleküldatenbanken Wenn die Anzahl der Moleküle ein paar Hundert übersteigt, ergibt es Sinn, sich Gedanken über deren Verwaltung zu machen. In diesem Bereich werden die Probleme beschrieben, die beim Verwalten von Moleküldatenbanken entstehen. Unter anderem wird die Verifizierung solcher Datenbanken anhand von Beispielen erklärt.

Vor- und Nachbearbeitung von Experimenten Sowohl für virtuelle als auch für reale Experimente muss häufig eine kleine Anzahl Moleküle aus einer großen Menge ausgewählt werden. Mona kann bei dieser Aufgabe unterstützen. Auch die Ergebnisse der Experimente können in Mona bearbeitet werden, indem man sie als SDF-Eigenschaften an den Molekülen annotiert.

6.4.1. Grundlegende Arbeitsschritte

Zur Demonstration der grundlegenden Funktionalität von Mona wird der LigandExpo-Datensatz [24, 49] benutzt. LigandExpo enthält alle kleinen Moleküle der Protein Data Bank (PDB) [7].

Fallbeispiel 1: Welche Moleküle befinden sich in der Datei?

Die ursprüngliche LigandExpo-Datei enthält 500 000 Liganden aus der PDB. Alle Moleküle besitzen 3D-Koordinaten. Mona ermöglicht es, all diese Instanzen zu laden. Dabei werden Duplikate erkannt, und eine Menge mit ∼19 000 unterschiedlichen Molekülen entsteht. In der tabellenbasierten visuellen Übersicht kann jedes einzelne Molekül dieser Menge betrachtet werden. Dafür sind keine Vorausberechnungen nötig, die Molekülbilder stehen sofort zur Verfügung. Es ist möglich, die Eigenschaften von bis zu 100 Instanzen eines Moleküls in der Instanzenansicht zu betrachten. Um von jedem Molekül die Anzahl der Duplikate zu ermitteln, fügt man den Molekülen der Menge die Eigenschaft *Duplikate* hinzu. Nun ist es auch möglich, in der tabellenbasierten visuellen Übersicht nach dieser Eigenschaft zu sortieren. Dadurch lassen sich auf einfache Weise die Liganden anzeigen, die am häufigsten oder seltensten in Proteinen vorkommen. Liganden mit sehr vielen Duplikaten sind nicht selektiv bei der Auswahl der Bindetasche, und Liganden mit wenig Duplikaten sind selektiv in ihrer Wahl. Die Verteilungen der Moleküleigenschaften für die komplette Menge lassen sich anhand von Histogrammen visualisieren.

Fallbeispiel 2: Wie konvertiert man eine große SMILES-Datei in mehrere kleine SD-Dateien?

In der Cheminformatik sind leicht parallelisierbare Vorgänge verbreitet. Berechnungen werden häufig auf jedem einzelnen Molekül einer Menge von Molekülen durchgeführt. Um diese Parallellisierbarkeit auszunutzen, ist es unerlässlich, große Eingabemengen in Teilmengen aufzuteilen. Mona ermöglicht es, auf unterschiedliche Arten Teilmengen von Molekülmengen zu bilden (s. Kap. 6.3.3). Die so erstellten Teilmengen können in Mona inspiziert werden und in einem Schritt in mehrere SD-Dateien exportiert werden. Beim Exportieren ist es wichtig zu wissen, dass die Ursprungsinstanzen wiederhergestellt werden. Daher muss der Benutzer vor dem Exportieren auswählen, aus welchen Molekülquellen die Instanzen verwendet werden. Lädt man z. B. die SD- und die SMILES-Datei der LingandExpo gleichzeitig in Mona, dann besitzen alle Moleküle mehrere Instanzen in der SD-Datei und in der SMILES-Datei. In diesem Fall gibt es zwei Möglichkeiten: Zum einen können die Moleküle mit den 3D-Koordinaten und Namen

der Instanzen aus der SD-Quelle exportiert werden. Zum anderen kann man die SMILES-Datei als Quelle auswählen und alle Moleküle werden ohne Koordinaten (da das SMILES-Format keine Konformation enthält) und mit den Namen der Moleküle in der SMILES-Datei exportiert.

Fallbeispiel 3: Wie findet man die Moleküle mit den gewünschten Eigenschaften?

Ein weiterer Standardanwendungsfall ist das Entfernen von Molekülen mit nicht erwünschten Eigenschaften aus Molekülmengen. Häufig ist dabei nicht von Beginn an klar, welche Kriterien für unerwünschte Moleküle gelten. Mona ermöglicht es, hier einen explorativen Ansatz zu verwenden. Angenommen der Benutzer möchte mithilfe der Moleküle aus der LigandExpo auf Interaktionsmuster schließen. Nach Betrachten der Verteilung der Molekulargewichte entscheidet er sich, dass nur Moleküle mit einem relativen Molekulargewicht von 200 bis 400 infrage kommen. Die Anwendung dieses Filters ergibt eine Menge von 8557 Molekülen. Weiterhin sollten die Moleküle zumindest ein Halogen, eine Ketongruppe und zwei Wasserstoffbrücken-Donoren enthalten. Die entsprechenden Mengen mit 4383, 3859 und 13 881 Molekülen lassen sich direkt mit Filteroperationen in Mona erstellen. Nach Betrachten der entstandenen Molekülmengen möchte der Benutzer wissen, auf welche Moleküle alle diese Eigenschaften zutreffen. Mithilfe der Mengenoperation Schnitt ergibt sich eine Molekülmenge mit 217 Molekülen.

Fallbeispiel 4: Wie sucht man nach Substrukturen in Molekülen?

Angenommen ein Pteridine-Ring hat sich als essenzielles Element für die Bindung eines Liganden herausgestellt. Interessant ist nun, welche anderen Liganden ebenfalls diese Substruktur enthalten. Ein SMARTS-Ausdruck anhand eines Strukturdiagrammes zu erstellen, verlangt auch von geübten Personen ein wenig Überlegung, vor allem bei Ringsystemen. Einfacher ist es, in Mona mit einem Mausklick zunächst den SMILES-Ausdruck eines Moleküls mit dem gewünschten Pteridine-Ring zu erstellen. Dieser Ausdruck lässt sich direkt editieren, oder man öffnet mit einem weiteren Mausklick den grafischen SmartsEditor. Im SmartsEditor entfernt man alle Teile außer dem Pteridine-Ring und filtert danach die Molekülmenge mithilfe des entstandenen SMARTS-Ausdrucks. Um die Moleküle in der entstandenen Menge einfacher vergleichen zu können, wird das erste Molekül als Anker benutzt. Die restlichen Moleküle werden daraufhin mit dem ersten als Vorlage gezeichnet, um die Gemeinsamkeiten und Unterschiede der Moleküle besser sichtbar zu machen.

Fallbeispiel 5: Wie lassen sich kleine Moleküle aus PDB-Dateien extrahieren?

Vereinfacht beschrieben enthalten PDB-Dateien sowohl die kompletten Proteine als auch die Liganden in Form einer großen Wolke von 3D-Atomkoordinaten. Das Erkennen der korrekten Bindungen ist fehleranfällig. Mona basiert auf Naomi und greift damit auf die in [81] beschriebenen Methoden zurück, um Liganden in Proteinen zu erkennen. Beim Importieren von PDB-Dateien werden von Mona alle kleinen Moleküle

in dem Protein erkannt und in einer gemeinsamen Molekülmenge abgelegt. Die Instanzen besitzen dabei die im Protein eingenommen 3D-Konformation. Typische im Kopf der PDB-Datei kodierte Eigenschaften, wie die PDB-ID, die Auflösung des Experiments oder die EC-Nummer, stehen nach dem Importieren in Mona als Instanzeigenschaften zur Verfügung.

6.4.2. Verwalten von Moleküldatenbanken

Das manuelle Verwalten von Moleküldatenbanken führt mit steigender Größe der Datenbank schnell zu schwierig zu findenden Inkonsistenzen. Mona kann dazu benutzt werden, sowohl Fehler und Inkonsistenzen in Sammlungen von Molekülen zu finden, als auch diese zu vermeiden. Molekülmengen mit einer Größe von bis zu einer Million Moleküle lassen sich auf diese Art komfortabel verwalten.

Fallbeispiel 6: Wie findet man Inkonsistenzen in Moleküldatenbanken?
Die DUD-E Datenbank [56] enthält aktive Moleküle und *Decoy*-Moleküle für 102 unterschiedliche Ziele in ebenfalls 102 unterschiedlichen Proteinen. Insgesamt enthält der Datensatz 1 172 433 Moleküle. Um Inkonsistenzen zu finden, gilt es, Annahmen zu finden, die sich mit Mengenoperationen schnell überprüfen lassen.

Die *Decoy*-Moleküle des Datensatzes sollten nicht gleichzeitig aktiv für eines der anderen Proteine sein. Auf welche Moleküle des DUD-E-Datensatzes trifft dies zu? Um diese Frage zu beantworten, werden zunächst jeweils die aktiven Moleküle und die *Decoy*-Moleküle aller Ziele als eigene Molekülmengen in Mona importiert. Ein Schnitt der beiden Mengen ergibt 123 Moleküle, die sowohl als *Decoy*-Molekül existieren als auch aktiv sind.

Gibt es *Decoy*-Moleküle, die bereits als Medikamente registriert sind? Hier liefert ein Schnitt mit der Drugbank [87] die Antwort: 40 Moleküle sind sowohl *Decoy*-Moleküle im DUD-E-Datensatz als auch bereits anerkannte Medikamente.

Welche der *Decoy*-Moleküle finden sich als Liganden in einem der Proteine der PDB wieder? Nach dem Laden aller LigandExpo-Moleküle [24, 49] werden diese mit der Molekülmenge der DUD-E-*Decoys* geschnitten. Dabei ergeben sich 187 Moleküle.

Potenzielle Inkonsistenzen innerhalb einer Molekülmenge lassen sich ebenfalls finden: Die Tautomer-Eigenschaft von Mona kann benutzt werden, um Moleküle zu finden, die Tautomere voneinander sind. Dazu fügt man der Molekülmenge die ID des kanonischen Tautomers als Eigenschaft hinzu. Moleküle, die dieselbe ID besitzen, sind unterschiedliche Tautomere[1] desselben Moleküls. Nach dem Hinzufügen dieser ID zum Drugbank-Datensatz und dem Clustern danach sind die Moleküle nach der Anzahl ihrer Tautomere sortiert. In der aktuellen Version 5.1.2 existieren zwei Moleküle, für die jeweils auch ein Tautomer im Datensatz vorhanden ist. In der vorherigen Version 4.1 war dies noch für 11 Moleküle der Fall.

[1] Rein technisch kann auch eine Kollision des kryptographischen SHA1-Hashwertes vorliegen, dies ist jedoch extrem unwahrscheinlich.

Fallbeispiel 7: Wie importiert man Moleküle aus Herstellerdatenbanken in die eigene interne Datenbank?
Angenommen der Hersteller von Molekülen gibt monatlich einen neuen Katalog der verfügbaren Moleküle heraus. Manuell ist es schwierig festzustellen, welche Moleküle hinzugekommen oder entfernt wurden. Mithilfe von Mona kann man die Differenz der internen Datenbank und der Datenbank des Herstellers leicht erstellen. Die Molekülmenge mit der Differenz enthält die Moleküle, die der Hersteller neu hinzugefügt hat. Mit Filtern lässt sich die Menge der neuen Moleküle weiter einschränken. Ein bekanntes Beispiel sind die PAINS SMARTS-Filter [5, 28], mit denen bekannte *frequent hitter* ausgefiltert werden können. Mona erlaubt es, die gesamte Liste von 482 SMARTS-Mustern in einem Filter auf die Menge anzuwenden. Wenn man die Moleküle dann nach Ähnlichkeit clustert, erkennt man, welche neuen Klassen von Molekülen in dem Herstellerdatensatz vorhanden sind. Mit dem Vereinigungsoperator wird die Herstellerdatenbank abschließend der internen Datenbank hinzugefügt.

6.4.3. Vor- und Nachbearbeitung von Experimenten

Nach der Durchführung von Experimenten gilt es, neue Erkenntnisse aus den Ergebnissen zu gewinnen. Mona kann hier mit den verfügbaren Methoden helfen: Die Clusteransicht zeigt die verschiedenen Klassen von Molekülen im Resultat, Histogramme zeigen die Verteilungen von Eigenschaften und die Ausrichtung von Molekülen hilft, kleine Unterschiede in Molekülen einfacher zu erkennen.

Eine weitere Quelle zum Erkenntnisgewinn ist die Kombination der Resultate mit anderen Datenbanken. Dazu wird das einzige immer verfügbare gemeinsame Merkmal zweier beliebiger Moleküldatenbanken als Schlüssel benutzt: das Molekül.

Fallbeispiel 8: Wie verbindet man die Informationen mehrerer Moleküldatenbanken miteinander?
Um die Drugbank-Datenbank mit der LigandExpo-Datebank zu kombinieren, werden beide Datenbanken zunächst in Mona geladen. Wichtig dabei ist es, möglichst die SD-Dateien zu verwenden, damit die an den Molekülen annotierten Eigenschaften erhalten bleiben. Nachdem beide Datensätze als eigene Molekülmengen in Mona vorhanden sind, wird die Schnittmenge berechnet. Diese besteht aus 4400 Molekülen. Für diese Moleküle sind die Eigenschaften sowohl der LigandExpo als auch der Drugbank verfügbar. Der Schnittmenge werden die Eigenschaften *DRUGBANK_ID* und *DRUG_GROUPS* aus der Drugbank und die Eigenschaft *Molecule Name* aus der LigandExpo hinzugefügt. Die Schnittmenge wird nun in eine SD-Datei exportiert. Diese Datei enthält die kombinierte Information beider Datensätze in Form von entsprechenden SD-Eigenschaften.

Fallbeispiel 9: Wie bereitet man Datensätze zum virtuellen Screening vor und analysiert die Ergebnisse?
Eine sehr verbreitete Aufgabe in der Chemieinformatik ist das Erstellen von Molekül-

sammlungen, um auf diesen ein virtuelles Screening durchzuführen. Alle Tools zum virtuellen Screening von Molekülen haben die Eigenschaft, dass sie aus einer riesigen Anzahl von Molekülen die besten Kandidaten auswählen. Mona erlaubt es, komfortabel mit Mengen bis zu 1 Mio. Moleküle zu arbeiten. Wenn für ein virtuelles Screening eine deutlich größere Anzahl an Molekülen benötigt wird, ergibt es daher Sinn, Mona nur zur Betrachtung von Stichproben dieser Menge zu verwenden und die Vorbereitung des Datensatzes mit anderen Programmen zu erledigen.

Das Vorbereiten eines virtuellen Screeningdatensatzes könnte wie folgt aussehen: Mehrere Hersteller von Molekülen, die ihre Kataloge dem ZINC-Datensatz [40] bereitstellen, werden ausgewählt. Die Kataloge aller Hersteller werden zu einer Menge von Molekülen vereinigt. Duplikate gibt es dank des Mengenkonzeptes von Mona in dieser Menge nicht. Um einen Überblick zu bekommen, ist es hilfreich, sich in Histogrammen die physikochemischen Eigenschaften aller Moleküle der Menge anzusehen. Unerwünschte Moleküle können anschließend mithilfe der Filteroperationen entfernt werden.

Das Ergebnis des virtuellen Screenings lässt sich ebenfalls mit Mona betrachten und nachbearbeiten: Die vielversprechendsten Moleküle sollen von den Herstellern bestellt werden. Dazu wird das Resultat in Mona geladen. Diese Menge wird nach Ähnlichkeit geclustert und aus jedem Cluster werden die am besten bewerteten Moleküle in einer neuen finalen Menge vereinigt. Für diese finale Menge kann der Dateiname der Quelle aller Instanzen als Eigenschaft hinzugefügt werden. So kann man sehen, bei welchen Herstellern jedes Molekül verfügbar ist. Wenn die ursprünglichen Herstellerkataloge auch die Preise für die Moleküle als Eigenschaft enthalten, kann dieser ebenfalls der Menge hinzugefügt werden. Da es sich bei dem Preis um einen numerischen Wert handelt, kann das Minimum, der Durchschnittswert und das Maximum aller Preise der einzelnen Instanzen aus den Herstellerkatalogen direkt verwendet werden. Wenn man die Menge nun nach dem minimalen Preis sortiert, erhält man die Moleküle, die am günstigsten zu beschaffen sind.

6.5. Geschwindigkeit

Eine wichtige Eigenschaft von Mona ist eine hohe Geschwindigkeit, um einen explorativen Ansatz zu ermöglichen. Die theoretischen Geschwindigkeiten der Datenbankschicht wurden bereits in Kap. 5.2.3 betrachtet. Die Operationen auf der Datenbank sind typischerweise linear abhängig von der Anzahl der Moleküle in der Menge. Systematische Messungen der Dauer von Mengenoperationen finden sich in [34] und [33]. In diesem Abschnitt werden die Ergebnisse aus [33] wiederholt und zusammengefasst. Typische Operationen in Mona wurden mit drei unterschiedlich großen Molekülmengen gemessen: Als kleine Menge wurde die LigandExpo [24, 49] mit 18 986 Molekülen verwendet. Eine mittelgroße Menge ergibt sich aus dem Enamine-Building-Block-Datensatz der ZINC [40] (148 216 Moleküle). Der große Datensatz enthielt alle 1 172 433 *Decoy*-Moleküle des DUD-E-Datensatzes [56].

Für alle drei Datensätze wurden die Zeiten für typische Operationen in Mona ge-

messen (s. Tab. 6.2): Die Datensätze wurden zunächst aus mehreren SMILES-Dateien importiert und anschließend in eine SD-Datei exportiert.

Filterketten in Mona haben unterschiedliche Laufzeiten abhängig von den Elementen in der Filterkette: Die einfachsten Filter filtern nur nach dem numerischen Wert einer Eigenschaft. In diesem Fall nach dem Molekulargewicht der Moleküle zwischen 200 und 400. Der Wirkstofffilter testet mit der verbreiteten *Rule-of-Five*[2] Moleküle auf ihre Eignung als Wirkstoff. Als Unterschied zum einfach Eigenschaftsfilter enthält die *Rule-of-Five* eine Toleranz von drei, d. h. es müssen für jedes Molekül nur zwei der fünf Kriterien erfüllt sein. Der Phenyl SMARTS-Filter akzeptiert alle Moleküle, bei denen der SMARTS-Ausdruck eines Phenyls gefunden wurde. Am langsamsten ist schließlich der PAINS SMARTS-Filter, bei dem jedes der Moleküle auf 482 verschiedene SMARTS-Ausdrücke getestet wird.

Mona stellt die beiden in Kap. 6.3.4 beschriebenen Clustermethoden bereit. Auf allen Mengen lässt sich nahezu augenblicklich nach einer Eigenschaft clustern ($< 4\,\mathrm{s}$). Das Clustern nach der Ähnlichkeit der Moleküle dauert dagegen aufgrund des verwendeten quadratischen Algorithmus deutlich länger.

Die wichtigste Operation in Mona ist das Verwalten von Molekülmengen. Da die Mengenoperationen stark von der Größe der Ergebnismenge abhängen, wurden sie wie folgt getestet: Die Vereinigung und der Schnitt wurden auf zwei identischen Eingabemengen durchgeführt. Dadurch ist die Größe der Ergebnismenge und der Eingabemenge gleich. Die Differenz wurde auf der Eingabemenge und der ersten Hälfte der Eingabemenge durchgeführt. Die Ergebnismenge hatte in diesem Fall die Größe der zweiten Hälfte der Eingabemenge. Insgesamt lassen sich Mengenoperationen auf kleinen und mittleren Mengen interaktiv verwenden. Nur auf der großen Menge nehmen sie bis zu 22 s Zeit in Anspruch.

Zusammenfassend lassen sich die meisten Operationen in Mona interaktiv auf kleinen und mittleren Molekülmengen ausführen. Die meiste Zeit benötigt bei den grundlegenden Operationen das Importieren und Exportieren von Molekülen. Mit Mona können typischerweise etwa 1000 bis 2000 Moleküle pro Sekunde eingelesen werden. Von den restlichen Operationen fallen vor allem das Clustern nach Ähnlichkeit und das Filtern nach 482 SMARTS-Ausdrücken aus dem Raster: Beide Operationen benötigen deutlich länger als alle restlichen. Beim Ähnlichkeitsclustern liegt es am quadratischen Algorithmus und beim SMARTS-Filter liegt es daran, dass jedes Molekül einzeln getestet wird. Geeignete Indizes könnten benutzt werden, um die SMARTS-Suche zu beschleunigen.

Ein wichtiger Punkt bei der Betrachtung der Geschwindigkeit und der Interaktivität ist, dass alle Operationen in Mona nebenläufig ausgeführt werden. Dadurch ist ein interaktives Weiterarbeiten mit dem Tool immer gewährleistet. Die Operationen laufen in eigenen Threads, wobei der Fortschritt immer über Fortschrittsbalken sichtbar ist.

[2] Zwei der folgenden Kriterien müssen für die *Rule-of-Five* erfüllt sein: MW 0-500, Akzeptoren 0-10, Donoren 0-5 und logP < 5.

Tab. 6.2. Die Tabelle zeigt die Dauer von typischen Operationen in Mona auf drei unterschiedlich großen Molekülmengen. Alle Messungen wurden auf einem Rechner mit einem Intel Core i7-4770 mit 3,4 GHz, 16 GB RAM und einer SSD gemacht. Bei allen Operationen mit einer kürzeren Dauer als 10 min gibt der gezeigte Wert den Durchschnittswert von fünf unabhängigen Messungen an.

	klein	mittel	groß
Größe	18 986	148 216	1 172 433
Import aus mehreren SMILES-Dateien	14,4 s	51,6 s	9:13 min
Export in SD-Datei	16,8 s	1:46 min	17:55 min
Einfacher Filter	0,1 s	1,3 s	10,7 s
Wirkstofffilter	1,2 s	10,9 s	1:29 min
Phenyl SMARTS-Filter	7,6 s	37,2 s	9:08 min
482 PAINS SMARTS-Filter	33:39 min	1:42 h	42:26 h
Cluster nach Eigenschaft	0,06 s	0,4 s	3,6 s
Cluster nach Ähnlichkeit	3:22 min	16:24 min	24:43 h
Vereinigung zweier Mengen	0,07 s	0,6 s	4,8 s
Differenz zweier Mengen	0,2 s	1,3 s	10,7 s
Schnitt zweier Mengen	0,3 s	2,5 s	21,9 s
Aufteilen einer Menge	0,1 s	1,1 s	8,9 s

6.6. Implementierung

Mona benutzt alle in den vorherigen Kapiteln beschriebenen Methoden. Zum einen wird eine Moleküldatenbank benutzt, um viele Moleküle effizient zu speichern und zu verarbeiten. Zum anderen ist eine schnelle Darstellung einzelner Moleküle nötig, um durch große Mengen von Molekülen navigieren zu können.

Um die inhärente Komplexität von UI-Programmen möglichst weit zu begrenzen, ist das Model-View-Architekturmuster in der UI-Programmierung sehr weit verbreitet. In Mona wurde versucht, die Darstellung soweit wie möglich von den Modellen zu trennen, konkret also die UI-Komponenten und die Behandlung von UI-Ereignissen von den Datenstrukturen zur Verarbeitung von Molekülmengen und den Operationen auf diesen Mengen.

6.6.1. Architektur und Erweiterungen

Mona ist wie in Abb. 6.4 dargestellt in vier unabhängige Schichten unterteilt. Das Programm wurde in C++ mithilfe der Bibliothek Qt [78] entwickelt. Mona kann dadurch auf den üblichen Plattformen Windows, macOS und Linux ausgeführt werden.

Auf der untersten Ebene befindet sich die Moleküldatenbank, die, wie in Kap. 5.2 beschrieben, als Sammlung von unabhängigen Bibliotheken organisiert ist.

Die Schicht darüber enthält die grundlegenden Datenstrukturen, die überall durch Mona gereicht werden. Die Klasse `Sets` abstrahiert eine Menge von Molekülen und die

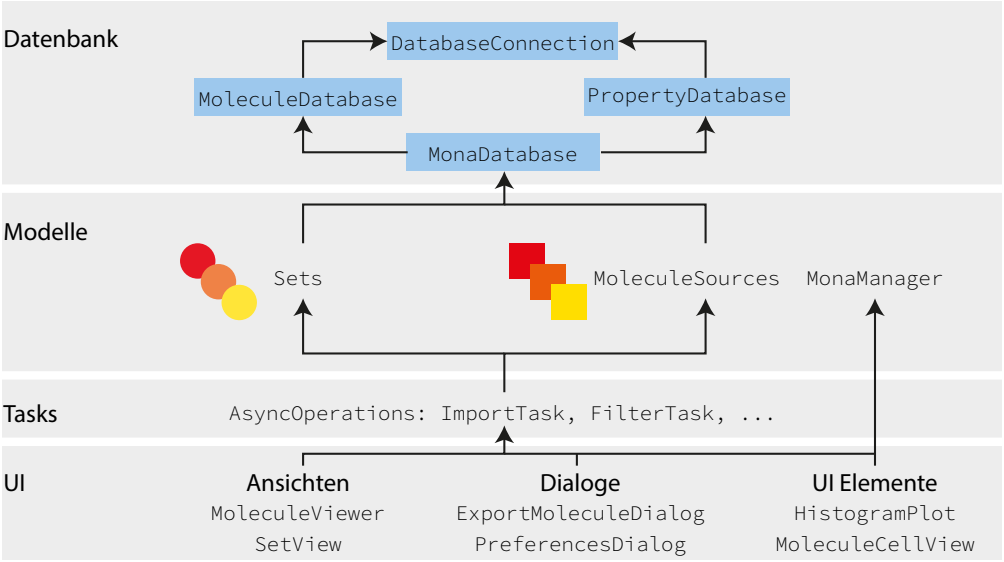


Abb. 6.4. Mona besteht aus den vier unabhängigen Schichten Datenbank, Modelle, Tasks und UI. Jede Schicht verwendet ausschließlich Elemente der darüberliegenden Schicht.

möglichen Operationen auf diesen. Die Klasse `MoleculeSources` bündelt eine Quelle von Molekülen. Die Klasse enthält eine Molekül- und eine Instanzmenge der Quelle. Beide Klassen sind *thread-safe*, um gleichzeitig von unterschiedlichen Mengenoperationen benutzt werden zu können. Die Klasse `MonaManager` dient dazu, Zugriffe auf alle Modelle von Mona wie die Sammlung von Mengen (`SetTree`) oder die aktuell geladene Datenbank (`MonaDatabase`) zu ermöglichen. Der `MonaManager` ist dabei absichtlich kein globales Objekt (*singleton*), um alle Zugriffe auf Modelle in Mona über den dadurch benötigten zusätzlichen Methodenparameter sichtbar zu machen. Trotzdem existiert während der gesamten Laufzeit von Mona nur genau eine Instanz dieser Klasse. Diese ist daher immer gültig und der `MonaManager` im gesamten Programm verfügbar.

Die Operationen auf Molekülmengen sind in der darüberliegenden Schicht gebündelt. Mengenoperationen werden in Mona typischerweise asynchron in einem eigenen nebenläufigen Thread ausgeführt.

Die oberste Schicht von Mona ist die UI. Diese ist in möglichst voneinander getrennte einzelne Komponenten unterteilt.

Die grundlegendsten eigenen UI-Komponenten sind dabei in einzelne Klassen gekapselt. Beispiele hierfür sind die tabellenbasierte Molekülansicht (`Widgets::MoleculeView`) oder die Filterkomponenten (`Mona::FilterList`). Sammlungen von mehreren UI-Komponenten sind mittels Komposition zu größeren Funktionseinheiten in einer Klasse zusammengefasst. Hier sind vor allem die Dialoge in Mona zu erwähnen, die jeweils einen Eingabedialog von Mona enthalten. Ein wichtiges Merkmal dieser Dialoge ist, dass sie keinerlei Logik enthalten, die über das Überprüfen der Eingabewerte hinaus geht. Alle Dialoge liefern als Ergebnis genau die Daten, die an der Stelle ihres Aufrufs

benötigt werden.

Ein weiterer Typ von Funktionseinheiten sind die Ansichten. Diese kapseln alle im Hauptfenster von Mona sichtbaren Bereiche und enthalten den Hauptteil der Darstellungslogik. Mona besteht aus den folgenden Ansichten:

`MonaGUI` enthält das komplette Hauptfenster von Mona. Alle anderen Ansichten sind in diesem Fenster enthalten.

`SetView` beinhaltet die aktuelle Liste der Molekülmengen. Alle Operationen auf Molekülmengen werden über Qt-Slots in dieser Klasse gestartet. Das zugrundeliegende Modell ist `Mona::SetTree`.

`MoleculeView` zeigt die in den momentan ausgewählten Mengen enthaltenen Moleküle in Form von chemischen Strukturdiagrammen an.

`MoleculeSourcesViewer` Die Verwaltung der Molekülquellen erfolgt in dieser Ansicht. Die zugrundeliegenden Daten sind im `DataSources` Modell enthalten.

`FilterViewer` dient der Erstellung von Filterketten zum Filtern von Molekülmengen.

`HistogramViewer` Die Eigenschaften von Molekülmengen können in Form von Histogrammen in dieser Ansicht erzeugt und betrachtet werden.

`InstanceViewer` Jedes Molekül in Mona kann aus verschiedenen Instanzen stammen. In der Instanzenansicht lassen sich die ursprünglichen Instanzen eines Moleküls mitsamt ihrer Eigenschaften betrachten.

`LogView` Wenn nicht alle Moleküle beim Laden initialisiert werden können, findet man in dieser Ansicht die entsprechenden Logeinträge. Außerdem kann hier die Historie von Mengenoperationen für jede Molekülmenge angezeigt werden.

An mehreren Stellen von Mona sind Möglichkeiten der Erweiterung vorgesehen.

Komplett eigene Ansichten lassen sich durch Komposition zur Klasse `MonaGUI` hinzufügen. Wenn diese über Änderungen in der Molekülmengenauswahl benachrichtigt werden sollen, muss das Qt-Signal `selectedSetsChanged` von `MonaGUI` mit einem Slot in der Ansicht verbunden werden. Wenn eine Ansicht außerdem Zugriff auf alle in Mona enthaltenen Modelle benötigt, wird typischerweise im Konstruktor eine Instanz der Klasse `MonaManager` übergeben.

Neue Dialoge in Mona sollten möglichst ohne Logik als einzelne Klassen erstellt werden, wobei nach Möglichkeit außerdem eine entsprechende eigene UI-Datei erstellt wird. Durch das GUI-Hilfsprogramm Qt Designer kann das Design von Dialogen mit UI-Dateien wesentlich schneller bearbeitet werden, als wenn das Aussehen im Programmcode spezifiziert ist. Außerdem sind Änderungen direkt sichtbar und benötigen keine Neukompilierung des gesamten Programms.

Neue Operationen auf Molekülmengen bestehen aus zwei Teilen: Die eigentliche nebenläufige Operation befindet sich in einer von `SetBaseTask` abgeleiteten Klasse. Der zweite Teil der Operation besteht aus dem Aufruf der asynchronen Operation, die

per Konvention möglichst einem normalen Methodenaufruf nachempfunden ist. D. h. wenn eine Mengenoperation mit dem Namen Vereinigung aus zwei Molekülmengen eine neue Menge erstellt, sieht die C++ Signatur des Aufrufs wie folgt aus:

```
ptr::Set Vereinigung(ptr::Set set1, ptr::Set set2);
```

`SetBaseTask` enthält zusätzlich zu `Base::Task` einige Hilfsfunktionen, die das Warten auf Molekülmengen vereinheitlichen und vereinfachen. Dadurch kann eine Operation in Mona aus mehreren Teiloperationen bestehen. Diese werden nebenläufig hintereinander ausgeführt. Ein Beispiel ist das Importieren von Molekülen. Diese Operation besteht aus zwei Teilen. Zuerst werden die Moleküle in die Datenbank importiert und danach wird aus den importierten Molekülen eine Molekülmenge im `SetView` erstellt. Wichtig bei eigenen Operationen ist außerdem, dass pro Thread immer eine eigene Verbindung zur Datenbank reserviert wird (`Mona::ReservedConnection`). Das Teilen von Verbindungen zwischen verschiedenen Threads ist nicht möglich, da diese nicht thread-safe sind.

Neue Moleküleigenschaften in Mona müssen an zwei Stellen hinzugefügt werden: Die Klasse `Mona::PropertyWriter` ist für das Bereitstellen aller Eigenschaften, sowohl beim direkten Einlesen aus Dateien als auch beim späteren Hinzufügen zu bereits eingelesenen Molekülen in der Datenbank verantwortlich. Informationen zu den einzelnen Eigenschaften in Mona müssen außerdem in der Klasse `Mona::MoleculeProperty Information` eingetragen werden. Dabei erfolgt die Identifizierung aller Eigenschaften innerhalb von Mona immer über eine eindeutige Zeichenkette der Form `Präfix.Name`. Beispielsweise identifiziert die Zeichenkette `naomi.Atoms` die Naomi-Eigenschaft *Anzahl der Atome pro Molekül*. Welche Eigenschaften zur Auswahl in der GUI angezeigt sind, bestimmen einmal die eingelesenen SDF-Eigenschaften der Eingabedateien und eine Reihe von vorgegebenen Naomi- und Mona-Eigenschaften, die in der Funktion `Mona::availableProperties()` spezifiziert sind.

7. Zusammenfassung und Ausblick

Das im Rahmen dieser Dissertation entstandene Programm Mona erlaubt es, auf intuitive und iterative Weise Moleküldateien zu verarbeiten. Vor allem das Konzept der Molekülmengen sorgt für einfache Arbeitsabläufe bei Routineaufgaben. Molekülmengen lassen sich einfach auf verschiedene Arten in Mona erstellen und mithilfe der üblichen mathematischen Mengenoperationen miteinander vergleichen. Eine performante Visualisierung sorgt dafür, dass auch in Mengen mit 1 Mio. Moleküle die Moleküle sofort dargestellt werden können. Alle Methoden der in Kap. 3 vorgestellten Bibliothek zum Erstellen von Strukturdiagrammen werden in Mona eingesetzt: 2D-Koordinaten von Molekülen werden erst dann berechnet, wenn sie nötig sind. Strukturdiagramme können als Vorlage für andere Strukturdiagramme dienen, indem sie miteinander verankert werden. Und einige molekulare Eigenschaften, wie die Anzahl der Akzeptoren oder die rotierbaren Bindungen, können durch den flexiblen Molekülsatz direkt in den Strukturdiagrammen angezeigt werden.

Insgesamt wurde die Software in einer Schichtenarchitektur aufgebaut, sodass auch zukünftige Erweiterungen leicht möglich sind. Allerdings sollte man diese Erweiterungen mit Bedacht angehen: Um weiterhin eine einfache Benutzung zu ermöglichen, müssen diese in das Konzept von Mona passen. Zum einen sollten Erweiterungen intuitiv sein. Ohne viele Erklärungen und Einstellmöglichkeiten sollen sie automatisch das Richtige tun. Zum anderen sollten alle Erweiterungen interaktiv benutzt werden können. Dies heißt vor allem, dass sie, ohne die Aufmerksamkeit des Benutzers zu verlieren (Dauer < 1 s) auf kleinen Molekülmengen mit einigen tausend Molekülen laufen.

Einige konkrete naheliegende Erweiterungen wären das Editieren von Moleküleigenschaften zu ermöglichen oder weitere Arten von Diagrammen und statistische Auswertungen hinzuzufügen. Nicht jede einzelne Funktionalität von Mona ist selbsterklärend und einfach: Die Behandlung von Molekül- und Instanzeigenschaften ergab sich vor allem aus technischen Gründen. Vielleicht ist hier noch eine elegantere Handhabung möglich? Ebenfalls sehr technisch ist die Verwendung von SMARTS-Ausdrücken. Ein häufiger Anwendungsfall von SMARTS-Ausdrücken ist die Suche nach reinen Substrukturen von Molekülen. Dieser Anwendungsfall ließe sich intuitiver gestalten, wenn man die zu suchenden Substrukturen direkt im Strukturdiagramm eines Moleküls auswählen könnte.

Komplizierter wird es, wenn man die Interaktivität von Mona auch mit größeren Datensätzen ermöglichen möchte. Ein Ansatz ist, die Datenbankschicht von Mona durch eine möglicherweise performantere verteilte Datenbank zu ersetzen. Dennoch müssten auch hier die Moleküldaten erst über das Netzwerk zur Datenbank übertragen werden, sodass beim Importieren von Molekülen kein Geschwindigkeitszuwachs zu erwarten ist.

Für Mona stellt momentan die tabellenbasierte Darstellung von Molekülmengen den besten Kompromiss aus Übersichtlichkeit und Geschwindigkeit dar: Es ist möglich, auch sehr große Molekülmengen augenblicklich zu visualisieren. Dies wird durch das selektive Zeichnen der aktuell benötigten Moleküle in einem eigenen Thread ermöglicht. Alternativen hierzu scheitern entweder an der Geschwindigkeit der Visualisierungsmethode, die die Daten von bis zu einer Million Moleküle aufbereiten muss oder sie scheitern an der Übersichtlichkeit, da schon wenige hundert Strukturdiagramme für ein undurchdringliches Liniengewirr sorgen. Weitere Visualisierungsmethoden von Molekülmengen ohne diese Probleme wären eine gute Erweiterung von Mona.

Die Clusteransicht ist eine solche alternative Visualisierung. Sie bietet eine gute Übersicht über eine Menge von Molekülen. Jedoch besteht ihr Nachteil darin, dass diese Ansicht nicht direkt verfügbar ist. Der Benutzer muss zunächst eine Clusterung der Molekülmenge durchführen. Der Benutzer muss sich also entscheiden, welche Art von Clusterung zu der Molekülmenge passt. Außerdem kann die Berechnung der Clusterung gerade bei den ähnlichkeitsbasierten Verfahren sehr zeitaufwendig werden. Erweiterungen dieses Modus, die diese Nachteile nicht besitzen, könnten dem Benutzer den Überblick über Molekülmengen weiter erleichtern. Eine Idee, dem Benutzer die Entscheidung über das Clusterverfahren abzunehmen, wäre eine lernende Clusterung. Dabei gibt der Benutzer anhand einiger Beispiele vor, wie die Moleküle in verschiedene Gruppen einzusortieren sind, und der Algorithmus führt die Sortierung danach fort.

Als unterste Schicht in Mona werkelt die Datenbank. Sie stellt die gesamte Funktionalität bereit, um Molekülmengen zu persistieren und zu verwalten. Für Mona war eine Anbindung an SQLite die naheliegendste Wahl. Allerdings beschleunigt SQLite nicht alle benötigten Operationen: Sowohl für die Substruktursuche mit SMARTS als auch für die Clusterung muss jedes Molekül der Molekülmenge einzeln wiederhergestellt werden, um die benötigten Berechnungen durchzuführen. Hier wären Methoden auf Datenbankebene hilfreich, um diese Operationen zu beschleunigen.

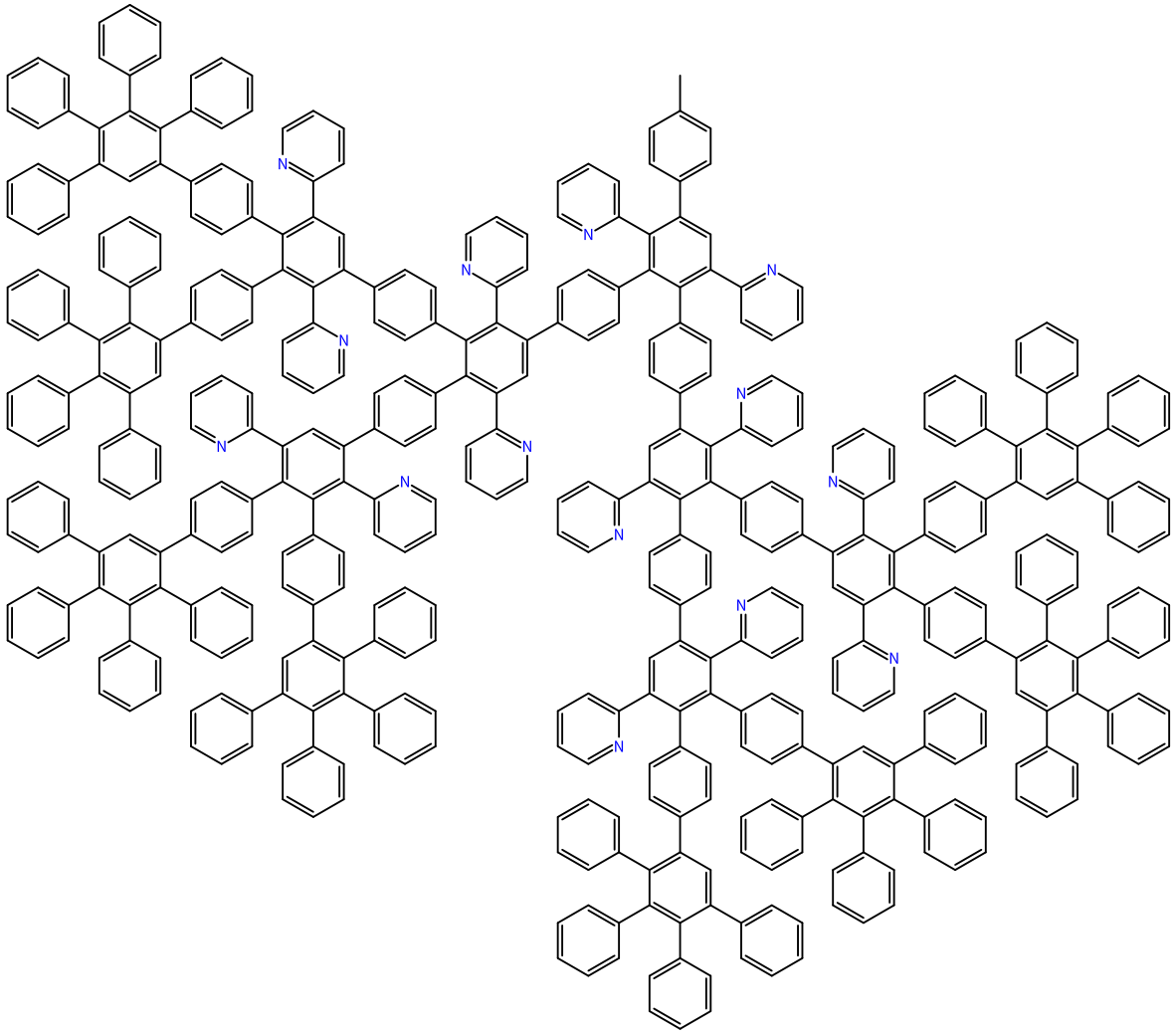
Zur Visualisierung von einzelnen Molekülen kommen in Mona immer Strukturdiagramme zum Einsatz. Die zugrundeliegende Bibliothek wurde ausführlich in dieser Arbeit beschrieben und validiert. Verglichen mit anderen Programmen enthalten die erzeugten Strukturdiagramme weniger Kollisionen und Fehler in Form von falsch gewählten Winkeln. Die Flexibilität der beschriebenen Layoutmethode wurde anhand der einfachen Unterstützung der Ausrichtefunktionalität von Strukturdiagrammen gezeigt. Naheliegende Erweiterungen der Layoutgenerierung sind die weitere Reduktion vor allem von schwerwiegenden Kollisionen, das Verbessern des Ringlayoutalgorithmus und die Beschleunigung des Algorithmus bei gleichbleibender Qualität. Bei der Verbesserung des Ringlayoutalgorithmus sollten Stereobindungen in Ringen beachtet werden (s. Kap. 3.1.2) und vor allem symmetrische Layouts für große Ringfamilien erstellt werden können. Beim Ausrichten der Moleküle besteht noch Potential im Verbessern sowohl des Matching-Algorithmus als auch der lokalen Optimierung der Molekülketten. Der Molekülsatz hält ebenfalls einige interessante Herausforderungen bereit: Stereobindungen in Ringsystemen werden häufig durch verdickte Bindungen innerhalb der Ringe gekennzeichnet. Diese automatisch korrekt zu zeichnen, ist vermutlich nicht trivial.

Die interessanteste Erweiterung der Strukturdiagrammbibliothek ist aber sicherlich die Entwicklung weiterer Visualisierungsarten, die auf Strukturdiagrammen basieren: Mit Poseview [75] lassen sich die Aminosäuren von Proteinbindungstaschen als 2D-Bilder visualisieren. Und der SmartsEditor [72] erlaubt es, chemische Muster direkt anhand von Strukturdiagramm zu erstellen.

Wie kann man die Flexibilität des Layouters verwenden, um ähnliche hilfreiche Visualisierungen zu generieren? Lassen sich perspektivisch korrekt gezeichnete 3D-Strukturdiagramme verwenden, um grundlegend neue hilfreiche Arten von Protein- oder Molekülvisualisierungen zu generieren?

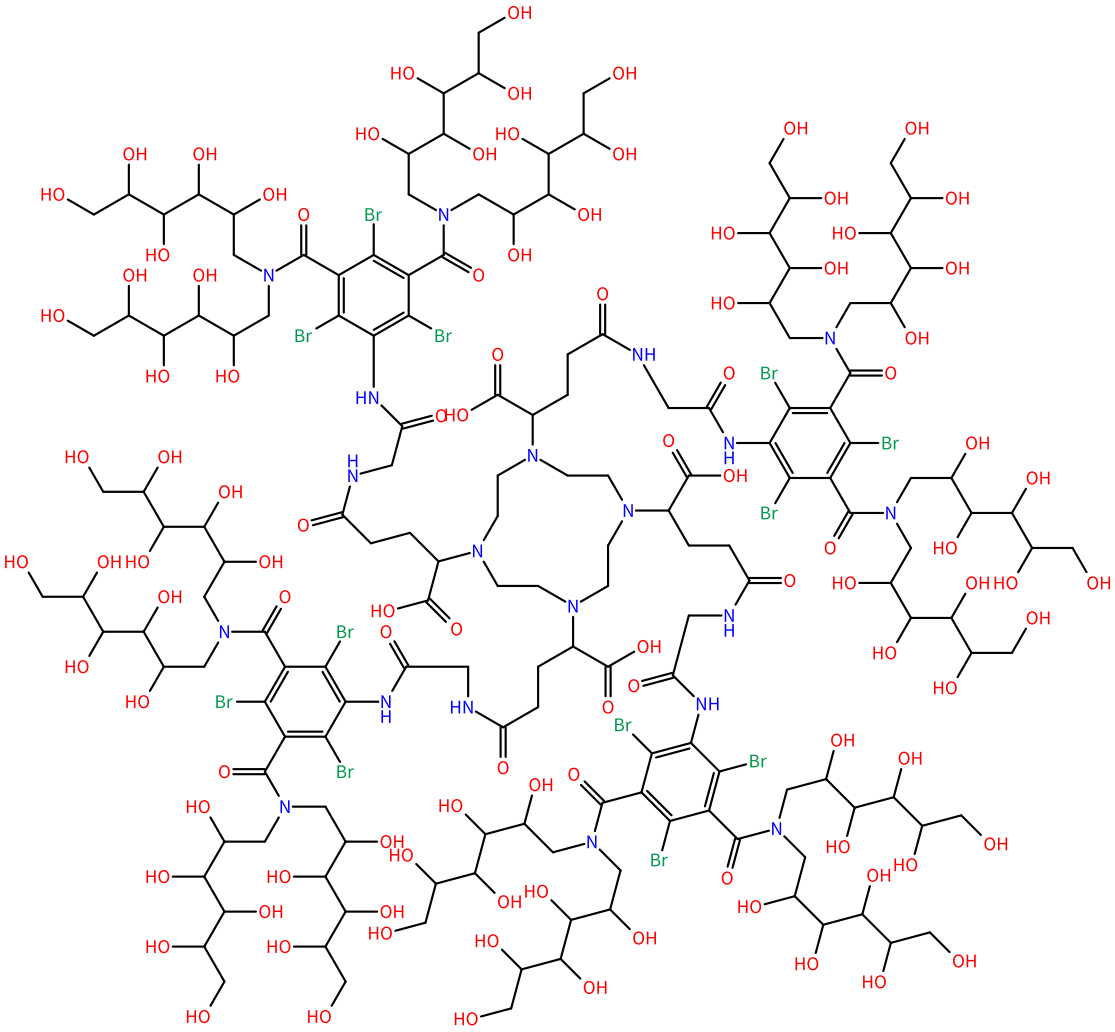
A. Galerie

Dieser Abschnitt enthält Strukturdiagramme von Beispielmolekülen und Fälle ununterscheidbarer Kollisionen. Alle Bilder wurden mit Naomi$_{2D}$ berechnet und gesetzt.



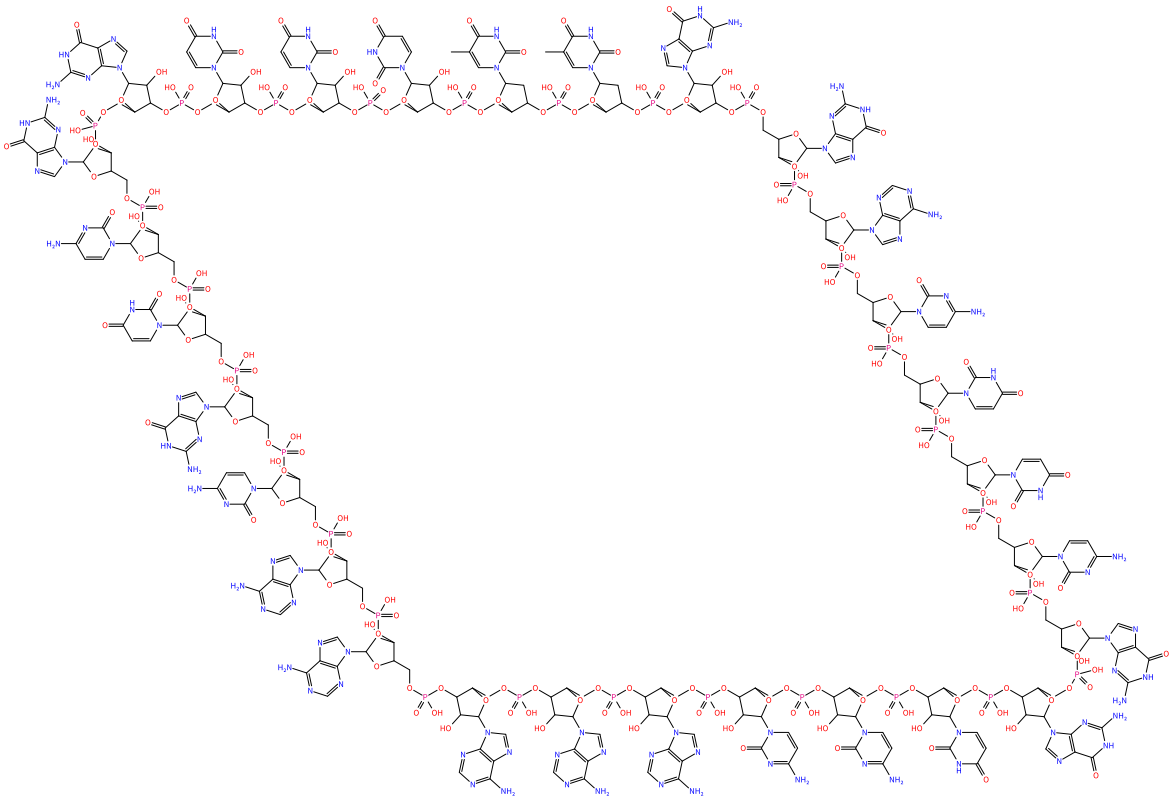
Pubchem SID: 138242971

Abb. A.1. Obwohl dieses Molekül starr aussieht, gibt es durch das Spiegeln der Ringe die Möglichkeit ein Layout ohne Kollisionen zu erzeugen. Einzelne Bindungen müssen dafür nicht verzerrt werden.



Pubchem SID: 40705740

Abb. A.2. Der mittlere Ring zwingt die ausgehenden Äste dieses Moleküls in vier Richtungen. Nicht alle Kollisionen konnten durch kombinatorische Operationen behoben werden und mussten mit verzerrten Bindungen korrigiert werden.



PubChem SID: 3720778

Abb. A.3. Ein sehr großer Makrozyklus sorgt bei diesem Molekül für eine rautenartige Form.

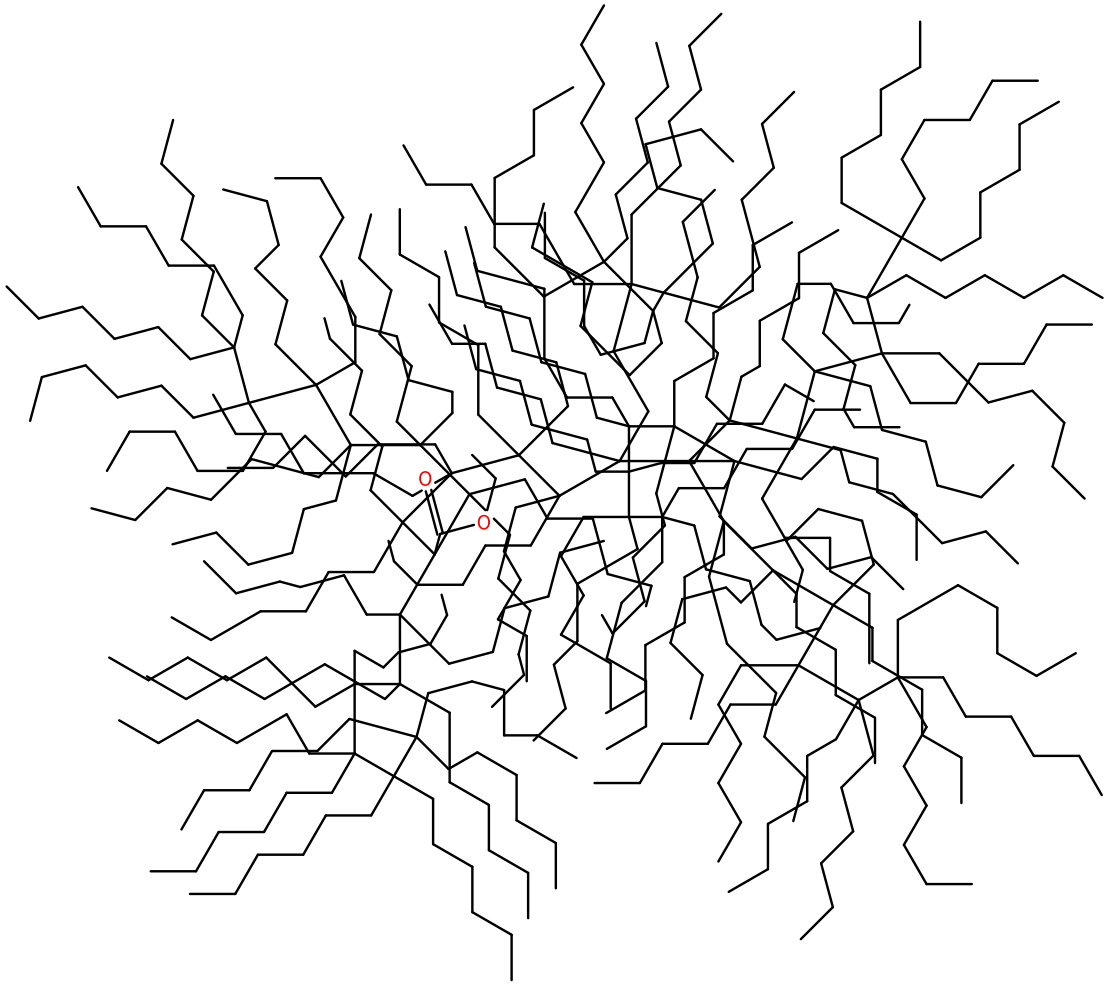


Abb. A.4. In der Abbildung ist das Molekül mit der längsten Berechnungszeit in der PubChem Datenbank abgebildet. Die Layoutberechnung benötigte 2744 s, wobei 2723 s davon auf die nachträgliche Kollisionsbehebung entfielen.

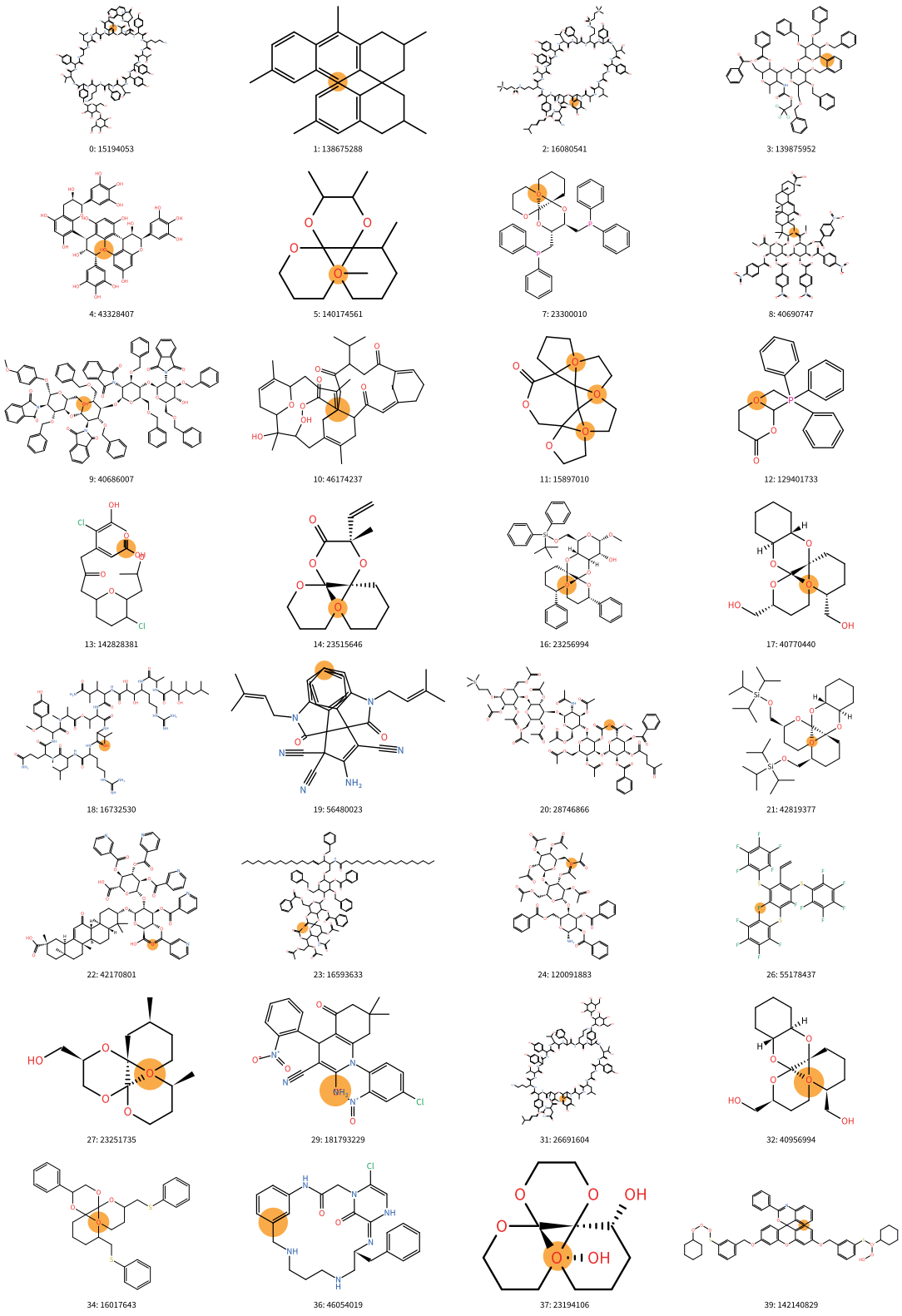


Abb. A.5. Die Abbildung zeigt 32 zufällig ausgewählte Beispiele aus allen 1505 Molekülen der PubChem, für die Naomi$_{2D}$ Layouts mit übereinander liegenden Atomen generiert. Die Stelle, an der zwei Atome dieselben Koordinaten haben, ist mit einem orangenen Kreis gekennzeichnet. Unter jedem Molekül findet sich die entsprechende PubChem-Substance-ID.

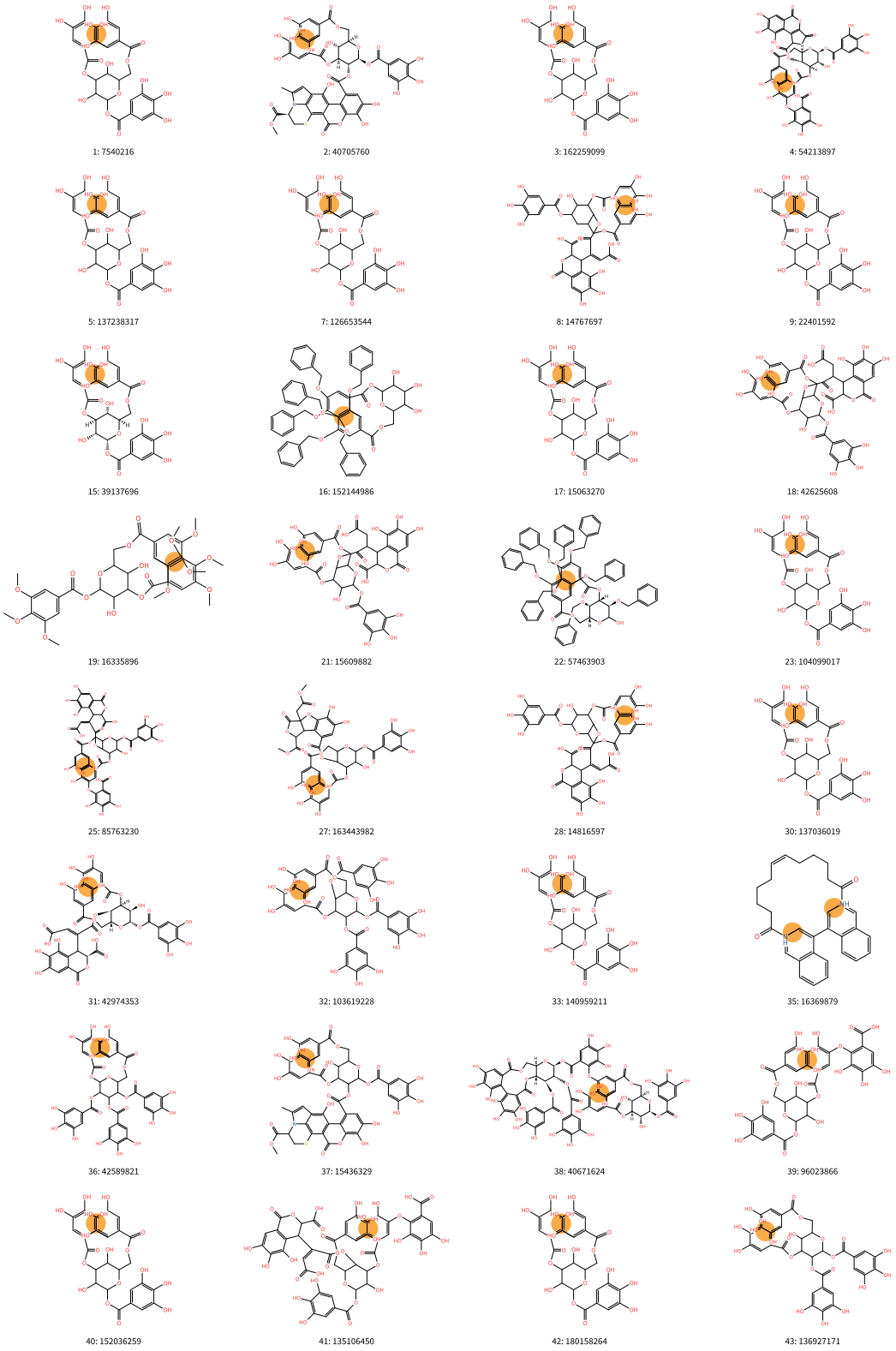


Abb. A.6. Die Abbildung zeigt 32 zufällig ausgewählte Beispiele aus allen 256 Molekülen der PubChem, für die Naomi_{2D} Layouts mit übereinander liegenden Bindungen generiert. Die Stelle, an der zwei Bindungen dieselben Koordinaten haben, ist mit einem orangenen Kreis gekennzeichnet. Unter jedem Molekül findet sich die entsprechende PubChem-Substance-ID.

Literatur

[1] Accelrys Software. *BIOVIA Pipeline Pilot*. Version 18.1. 2019. URL: http://accelrys.com/products/collaborative-science/biovia-pipeline-pilot/.

[2] Sanjeev Arora und Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 20. Apr. 2009. 519 S. ISBN: 978-1-139-47736-9.

[3] August Kekulé. „Ueber einige Condensationsproducte des Aldehyds“. In: *Justus Liebigs Annalen der Chemie* 162 (14. Feb. 1872), S. 77–124.

[4] August Kekulé. „Untersuchungen über aromatische Verbindungen“. In: *Annalen der Chemie und Pharmacie* 137.2 (1866), S. 129–196.

[5] Jonathan B. Baell und Georgina A. Holloway. „New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays“. In: *Journal of Medicinal Chemistry* 53.7 (8. Apr. 2010), S. 2719–2740. ISSN: 0022-2623. DOI: 10.1021/jm901137j.

[6] Giuseppe Di Battista u. a. *Graph Drawing: Algorithms for the Visualization of Graphs*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN: 978-0-13-301615-4.

[7] Helen M. Berman u. a. „The Protein Data Bank“. In: *Nucleic Acids Research* 28.1 (1. Jan. 2000), S. 235–242. ISSN: 0305-1048.

[8] Michael R. Berthold u. a. „KNIME: The Konstanz Information Miner“. In: *Data Analysis, Machine Learning and Applications*. Hrsg. von Christine Preisach u. a. Studies in Classification, Data Analysis, and Knowledge Organization. Springer Berlin Heidelberg, 1. Jan. 2008, S. 319–326. ISBN: 978-3-540-78239-1.

[9] Regine S. Bohacek, Colin McMartin und Wayne C. Guida. „The art and practice of structure-based drug design: A molecular modeling perspective“. In: *Medicinal Research Reviews* 16.1 (Jan. 1996), S. 3–50. ISSN: 0198-6325, 1098-1128.

[10] J. D. Boissonnat, F. Cazals und J. Flötotto. „2D-Structure Drawings of Similar Molecules“. In: *Graph Drawing*. Lecture Notes in Computer Science (1. Jan. 2001). Hrsg. von Joe Marks, S. 115–126. DOI: 10.1007/3-540-44541-2_11.

[11] R. S. Cahn, Christopher Ingold und V. Prelog. „Specification of Molecular Chirality“. In: *Angewandte Chemie International Edition in English* 5.4 (1966), S. 385–415. ISSN: 1521-3773. DOI: 10.1002/anie.196603851.

[12] Raymond E. Carhart. „A Model-Based Approach to the Teletype Printing of Chemical Structures“. In: *Journal of Chemical Information and Computer Sciences* 16.2 (1. Mai 1976), S. 82–88. ISSN: 0095-2338. DOI: 10.1021/ci60006a011.

[13] Fay Chang u. a. „Bigtable: A Distributed Storage System for Structured Data“. In: *ACM Transactions on Computer Systems* 26.2 (1. Juni 2008), S. 1–26. ISSN: 07342071. DOI: 10.1145/1365815.1365816.

[14] Chemical Computing Group. *Molecular Operating Environment (MOE)*. Version 2013.08. 2017. URL: http://www.chemcomp.com/.

[15] A.M. Clark, P. Labute und M. Santavy. „2D structure depiction“. In: *Journal of Chemical Information and Modeling* 46.3 (2006), S. 1107–1123. ISSN: 1549-9596. DOI: 10.1021/ci050550m.

[16] Alex M. Clark. „2D Depiction of Fragment Hierarchies“. In: *Journal of Chemical Information and Modeling* 50.1 (Jan. 2010), S. 37–46. ISSN: 1549-9596. DOI: 10.1021/ci900350h.

[17] Thomas H. Cormen u. a. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. 1312 S. ISBN: 978-0-262-03384-8.

[18] C. J. Date. *An Introduction to Database Systems*. 8. Aufl. Boston: Pearson, 1. Aug. 2003. 1024 S. ISBN: 978-0-321-19784-9.

[19] Daylight. *Daylight Theory: SMARTS – A Language for Describing Molecular Patterns*. URL: http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html.

[20] L. Peter Deutsch. *DEFLATE Compressed Data Format Specification version 1.3*. RFC 1951. Published: Internet Requests for Comments. RFC Editor, Mai 1996. URL: http://www.rfc-editor.org/rfc/rfc1951.txt.

[21] *Developmental Therapeutics Program (DTP)*. URL: https://dtp.cancer.gov/ (besucht am 07. 08. 2019).

[22] Peter Ertl und Bernhard Rohde. „The Molecule Cloud – compact visualization of large collections of molecules“. In: *Journal of Cheminformatics* 4.1 (6. Juli 2012), S. 12. ISSN: 1758-2946. DOI: 10.1186/1758-2946-4-12.

[23] David A. Evans. „History of the Harvard ChemDraw Project“. In: *Angewandte Chemie International Edition* 53.42 (13. Okt. 2014), S. 11140–11145. ISSN: 1521-3773. DOI: 10.1002/anie.201405820.

[24] Zukang Feng u. a. „Ligand Depot: a data warehouse for ligands bound to macromolecules“. In: *Bioinformatics (Oxford, England)* 20.13 (1. Sep. 2004), S. 2153–2155. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bth214.

[25] Tomasz Frączek. „Simulation-Based Algorithm for Two-Dimensional Chemical Structure Diagram Generation of Complex Molecules and Ligand–Protein Interactions“. In: *Journal of Chemical Information and Modeling* 56.12 (27. Dez. 2016), S. 2320–2335. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.6b00391.

[26] Arne Frick, Andreas Ludwig und Heiko Mehldau. „A fast adaptive layout algorithm for undirected graphs“. In: Springer-Verlag, 1995, S. 388–403.

[27] Patrick C. Fricker, Marcus Gastreich und Matthias Rarey. „Automated Drawing of Structural Molecular Formulas under Constraints“. In: *Journal of Chemical Information and Computer Sciences* 44.3 (1. Mai 2004), S. 1065–1078. ISSN: 0095-2338. DOI: 10.1021/ci049958u.

[28] Rajarshi Guha. *PAINS Substructure Filters as SMARTS*. URL: http://blog.rguha.net/?p=850 (besucht am 07. 08. 2019).

[29] Vincent L. Guilloux u. a. „Mining collections of compounds with Screening Assistant 2“. In: *Journal of Cheminformatics* 4.1 (31. Aug. 2012), S. 20. ISSN: 1758-2946. DOI: 10.1186/1758-2946-4-20.

[30] Karl Harrison, Jonathan P. Bowen und Alice M. Bowen. „Electronic Visualisation in Chemistry: From Alchemy to Art“. In: *arXiv:1307.6360 [physics]* (24. Juli 2013). EVA London 2013 Conference Proceedings, Electronic Workshops in Computing (eWiC), British Computer Society, 29-31 July 2013. URL: http://arxiv.org/abs/1307.6360 (besucht am 07. 08. 2019).

[31] Stephen Heller u. a. „InChI – the worldwide chemical structure identifier standard“. In: *Journal of Cheminformatics* 5 (24. Jan. 2013), S. 7. ISSN: 1758-2946. DOI: 10.1186/1758-2946-5-7.

[32] Harold E. Helson. „Structure Diagram Generation“. In: *Reviews in Computational Chemistry*. Hrsg. von Kenny B. Lipkowitz und Donald B. Boyd. John Wiley & Sons, Inc., 1999, S. 313–398. ISBN: 978-0-470-12590-8. DOI: 10.1002/9780470125908.ch6.

[33] Matthias Hilbig und Matthias Rarey. „MONA 2: A Light Cheminformatics Platform for Interactive Compound Library Processing“. In: *Journal of Chemical Information and Modeling* 55.10 (26. Okt. 2015), S. 2071–2078. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.5b00292.

[34] Matthias Hilbig u. a. „MONA — Interactive manipulation of molecule collections“. In: *Journal of Cheminformatics* 5.1 (28. Aug. 2013), S. 38. ISSN: 1758-2946. DOI: 10.1186/1758-2946-5-38.

[35] D. Richard Hipp, Dan Kennedy und Joe Mistachkin. *SQLite*. Version 3.8.6. 15. Aug. 2014. URL: http://www.sqlite.org.

[36] Holger H. Hoos und Thomas Stützle. *Stochastic Local Search*. Elsevier, 2005. ISBN: 978-1-55860-872-6. DOI: 10.1016/B978-1-55860-872-6.X5016-1.

[37] Wolf Dietrich Ihlenfeldt. *Cactvs*. Version 3.423. 2015. URL: http://www.xemistry.com/.

[38] Wolf Dietrich Ihlenfeldt u. a. „Computation and management of chemical properties in CACTVS: An extensible networked approach toward modularity and compatibility“. In: *Journal of Chemical Information and Computer Sciences* 34.1 (1. Jan. 1994), S. 109–116. ISSN: 0095-2338. DOI: 10.1021/ci00017a013.

[39] Adobe Systems Incorporated, Hrsg. *PostScript language reference*. 3. Aufl. Addison-Wesley, 1999. 897 S.

[40] John J. Irwin und Brian K. Shoichet. „ZINC – A Free Database of Commercially Available Compounds for Virtual Screening“. In: *Journal of Chemical Information and Modeling* 45.1 (1. Jan. 2005), S. 177–182. ISSN: 1549-9596. DOI: 10.1021/ci049714+.

[41] Jonathan Brecher. „Graphical representation of stereochemical configuration (IUPAC Recommendations 2006)“. In: *Pure and Applied Chemistry* 78.10 (2006), S. 1897–1970. ISSN: 0033-4545. DOI: 10.1351/pac200678101897.

[42] Jonathan Brecher. „Graphical representation standards for chemical structure diagrams (IUPAC Recommendations 2008)“. In: *Pure and Applied Chemistry* 80.2 (2008), S. 277–410. ISSN: 0033-4545. DOI: 10.1351/pac200880020277.

[43] Joseph Loschmidt. *Chemische Studien – Constitutions-Formeln der organischen Chemie in graphischer Darstellung ; Das Mariott'sche Gesetz*. Wien: Carl Gerold's Sohn, 1861. 53 S.

[44] Leonard Kaufmann und Peter Rousseeuw. „Clustering by Means of Medoids“. In: *Data Analysis based on the L1-Norm and Related Methods* (1987), S. 405–416.

[45] Mark J. Kilgard und Jeff Bolz. „GPU-accelerated Path Rendering“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 172:1–172:10. ISSN: 0730-0301. DOI: 10.1145/2366145.2366191.

[46] Sunghwan Kim u. a. „PubChem 2019 update: improved access to chemical data“. In: *Nucleic Acids Research* 47 (D1 8. Jan. 2019), S. D1102–D1109. ISSN: 0305-1048. DOI: 10.1093/nar/gky1033.

[47] Adrian Kolodzik, Sascha Urbaczek und Matthias Rarey. „Unique Ring Families: A Chemically Meaningful Description of Molecular Ring Topologies“. In: *Journal of Chemical Information and Modeling* 52.8 (27. Aug. 2012), S. 2013–2021. ISSN: 1549-9596. DOI: 10.1021/ci200629w.

[48] Greg Landrum. *RDKit*. Version Release_2015.03.1. 2015. URL: http://www.rdkit.org.

[49] *LigandExpo*. URL: http://ligand-expo.rcsb.org/ (besucht am 07. 08. 2019).

[50] Charles Loop und Jim Blinn. „Resolution Independent Curve Rendering using Programmable Graphics Hardware“. In: *24/3*. Association for Computing Machinery, Inc., Jan. 2005, S. 10.

[51] Eugene M. Luks. „Isomorphism of graphs of bounded valence can be tested in polynomial time“. In: *Journal of Computer and System Sciences* 25.1 (Aug. 1982), S. 42–65. ISSN: 00220000. DOI: 10.1016/0022-0000(82)90009-5.

[52] Hubert Maehr. „Graphic Representation of Configuration in Two-Dimensional Space. Current Conventions, Clarifications, and Proposed Extensions“. In: *Journal of Chemical Information and Computer Sciences* 42.4 (1. Juli 2002), S. 894–902. ISSN: 0095-2338. DOI: 10.1021/ci025518w.

[53] Gerald Maggiora u. a. „Molecular Similarity in Medicinal Chemistry“. In: *Journal of Medicinal Chemistry* 57.8 (24. Apr. 2014), S. 3186–3204. ISSN: 0022-2623. DOI: 10.1021/jm401411z.

[54] Makoto Matsumoto und Takuji Nishimura. „Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator“. In: *ACM Transactions on Modeling and Computer Simulation* 8.1 (1. Jan. 1998), S. 3–30. ISSN: 10493301. DOI: 10.1145/272991.272995.

[55] H. L. Morgan. „The Generation of a Unique Machine Description for Chemical Structures – A Technique Developed at Chemical Abstracts Service.“ In: *Journal of Chemical Documentation* 5.2 (1. Mai 1965), S. 107–113. ISSN: 0021-9576. DOI: 10.1021/c160017a018.

[56] Michael M. Mysinger u. a. „Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking“. In: *Journal of Medicinal Chemistry* 55.14 (26. Juli 2012), S. 6582–6594. ISSN: 0022-2623. DOI: 10.1021/jm300687e.

[57] OpenEye Scientific Software. *OEChem*. Version v2015.June. 2015. URL: http://www.eyesopen.com/oechem-tk.

[58] Optibrium. *StarDrop*. 2015. URL: http://www.optibrium.com/stardrop/.

[59] Mark Ovenden. *London Underground By Design*. Penguin, 2013. 288 S. ISBN: 978-1-84614-417-2.

[60] Keith Packard, Carl Worth und Behdad Esfahbod. *cairo graphics*. Version 1.12.18. 2014. URL: https://www.cairographics.org/.

[61] PerkinElmer. *ChemDraw*. Version 18. 2018. URL: http://www.perkinelmer.com/category/chemdraw.

[62] Geoff Pike und Jyrki Alakuijala. *CityHash, a family of hash functions for strings.* Version 1.0.1. 2013. URL: https://github.com/google/cityhash.

[63] PostgreSQL Global Development Group. *PostgreSQL*. Version 11.3. 2019. URL: https://www.postgresql.org/.

[64] Vladlmir Prelog und Günter Helmchen. „Basic Principles of the CIP-System and Proposals for a Revision“. In: *Angewandte Chemie International Edition in English* 21.8 (1982), S. 567–583. ISSN: 1521-3773. DOI: 10.1002/anie.198205671.

[65] Monty Python. *Spanish Inquisition – Nobody expects the Spanish inquisition in the references.* 22. Sep. 1970.

[66] Matthias Rarey und J. Scott Dixon. „Feature trees: A new molecular similarity measure based on tree matching“. In: *Journal of Computer-Aided Molecular Design* 12.5 (1. Sep. 1998), S. 471–490. ISSN: 0920-654X, 1573-4951. DOI: 10.1023/A:1008068904628.

[67] David Rogers und Mathew Hahn. „Extended-Connectivity Fingerprints“. In: *Journal of Chemical Information and Modeling* 50.5 (24. Mai 2010), S. 742–754. ISSN: 1549-9596. DOI: 10.1021/ci100050t.

[68] Thomas Sander. *DataWarrior – A Free Cheminformatics Program for Data Visualization and Analysis*. Version 5.0.0. 2019. URL: http://www.openmolecules.org/datawarrior/.

[69] Thomas Sander u. a. „DataWarrior: An Open-Source Program For Chemistry Aware Data Visualization And Analysis“. In: *Journal of Chemical Information and Modeling* 55.2 (23. Feb. 2015), S. 460–473. ISSN: 1549-9596. DOI: 10.1021/ci500588j.

[70] Till Schäfer u. a. „Scaffold Hunter: a comprehensive visual analytics framework for drug discovery“. In: *Journal of Cheminformatics* 9.1 (11. Mai 2017), S. 28. ISSN: 1758-2946. DOI: 10.1186/s13321-017-0213-3.

[71] John Scherk. *Algebra: A Computational Introduction*. 1 edition. Boca Raton: Chapman und Hall/CRC, 23. Juni 2000. 336 S. ISBN: 978-1-58488-064-6.

[72] Karen T. Schomburg, Lars Wetzer und Matthias Rarey. „Interactive design of generic chemical patterns“. In: *Drug Discovery Today* 18.13 (1. Juli 2013), S. 651–658. ISSN: 1359-6446. DOI: 10.1016/j.drudis.2013.02.001.

[73] Craig A. Shelley. „Heuristic approach for displaying chemical structures“. In: *Journal of Chemical Information and Computer Sciences* 23.2 (1. Mai 1983), S. 61–65. ISSN: 0095-2338. DOI: 10.1021/ci00038a002.

[74] Jonna C. Stålring u. a. „AZOrange – High performance open source machine learning for QSAR modeling in a graphical programming environment“. In: *Journal of Cheminformatics* 3.1 (28. Juli 2011), S. 28. ISSN: 1758-2946. DOI: 10.1186/1758-2946-3-28.

[75] Katrin Stierand und Matthias Rarey. „Flat and Easy: 2D Depiction of Protein-Ligand Complexes“. In: *Molecular Informatics* 30.1 (17. Jan. 2011), S. 12–19. ISSN: 1868-1751. DOI: 10.1002/minf.201000167.

[76] Katrin Stierand und Matthias Rarey. „From Modeling to Medicinal Chemistry: Automatic Generation of Two-Dimensional Complex Diagrams“. In: *ChemMedChem* 2.6 (Juni 2007), S. 853–860. ISSN: 18607179. DOI: 10.1002/cmdc.200700010.

[77] Stephen J. Tauber und Kirk Rankin. „Valid Structure Diagrams and Chemical Gibberish“. In: *Journal of Chemical Documentation* 12.1 (Feb. 1972), S. 30–34. ISSN: 0021-9576. DOI: 10.1021/c160044a010.

[78] The Qt Company. *Qt*. Version 5.9.1. 2017. URL: https://www.qt.io/.

[79] Sascha Urbaczek, Adrian Kolodzik und Matthias Rarey. „The Valence State Combination Model: A Generic Framework for Handling Tautomers and Protonation States“. In: *Journal of Chemical Information and Modeling* 54.3 (24. März 2014), S. 756–766. ISSN: 1549-9596. DOI: 10.1021/ci400724v.

[80] Sascha Urbaczek u. a. „NAOMI: On the Almost Trivial Task of Reading Molecules from Different File formats“. In: *Journal of Chemical Information and Modeling* 51.12 (27. Dez. 2011), S. 3199–3207. ISSN: 1549-9596. DOI: 10.1021/ci200324e.

[81] Sascha Urbaczek u. a. „Reading PDB: Perception of Molecules from 3D Atomic Coordinates“. In: *Journal of Chemical Information and Modeling* 53.1 (28. Jan. 2013), S. 76–87. ISSN: 1549-9596. DOI: 10.1021/ci300358c.

[82] Philippe Vismara. „Union of all the Minimum Cycle Bases of a Graph.“ In: *Electr. J. Comb.* 4 (1. Jan. 1997).

[83] Wendy A. Warr. „Tautomerism in chemical information management systems“. In: *Journal of Computer-Aided Molecular Design* 24.6 (1. Juni 2010), S. 497–520. ISSN: 0920-654X, 1573-4951. DOI: 10.1007/s10822-010-9338-4.

[84] David Weininger. „SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules“. In: *Journal of Chemical Information and Modeling* 28.1 (1. Feb. 1988), S. 31–36. ISSN: 1549-9596. DOI: 10.1021/ci00057a005.

[85] David Weininger, Arthur Weininger und Joseph L. Weininger. „SMILES. 2. Algorithm for generation of unique SMILES notation“. In: *Journal of Chemical Information and Computer Sciences* 29.2 (1. Mai 1989), S. 97–101. ISSN: 0095-2338. DOI: 10.1021/ci00062a008.

[86] Scott A. Wildman und Gordon M. Crippen. „Prediction of Physicochemical Parameters by Atomic Contributions“. In: *Journal of Chemical Information and Computer Sciences* 39.5 (27. Sep. 1999), S. 868–873. ISSN: 0095-2338. DOI: 10.1021/ci990307l.

[87] David S. Wishart u. a. „DrugBank 5.0: a major update to the DrugBank database for 2018“. In: *Nucleic Acids Research* 46 (D1 4. Jan. 2018), S. D1074–D1082. ISSN: 1362-4962. DOI: 10.1093/nar/gkx1037.

[88] Peng Zhou und Zhicai Shang. „2D molecular graphics: a flattened world of chemistry and biology“. In: *Briefings in Bioinformatics* 10.3 (1. Mai 2009), S. 247–258. ISSN: 1467-5463. DOI: 10.1093/bib/bbp013.